TALLINN UNIVERSITY OF TECHNOLOGY
Department of Computer Systems

Karl Kivi 204708IASM

IMPROVING PEDESTRIAN DETECTION ACCURACY USING SINGLE AND MULTIPLE MMWAVE RADAR CONFIGURATIONS

Master's Thesis

Supervisor
Karl Janson
PhD
Co-supervisor
Mairo Leier
PhD

Tallinn 2022

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutisüsteemide instituut

Karl Kivi 204708IASM

ÜHE JA MITME MILLIMEETERLAINERADARIGA JALAKÄIJATE TUVASTUSTÄPSUSE PARANDAMINE

magistritöö

Juhendaja
Karl Janson
PhD
Kaasjuhendaja
Mairo Leier
PhD

Tallinn 2022

Author's declaration of originality

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

Author: Karl Kivi .....
(signature)

Date: May 9, 2022

Annotatsioon

Millimeeterlaineradaritel on mitmeid kasutusvaldkondi nii kommerts- kui tööstussüsteemides, kuivõrd millimeeterlaineradarid laiendavad mitme teise anduri funktsionaalsust. Magistritöö eesmärk on parandada sihtmärgi tuvastamise täpsust, kasutades kahte radarit ja viit radari konfiguratsiooni. Lisaks arendada tarkvararakendus, et analüüsida eksperimentide tulemusi. Käesolevas töös keskendutakse kahele probleemile. Esiteks, millimeeterlaineradariga on võimalik tuvastada ainult mõned radaripunktid väiksematelt objektidelt nagu inimesed ja väiksemad sõidukid. Teiseks probleemiks on puuduv tarkvara radari andmete analüüsiks ja erinevate seadistuste võrdlemiseks.

Eksperimentide läbiviimiseks vajalikud andmed koguti kahe radariga. Autori arendatud tarkvararakendus genereeris soojuskaardi graafikud, et visuaalselt analüüsida kahe radari tulemusi viie erineva radari konfiguratsiooniga. Tarkvara kasutusvaldkond ulatub tegelikkuses kaugemale kui antud magistritöö, sest arendatud rakendust saavad kasutada kõik, kes soovivad radari andmeid analüüsida.

Eksperimente viidi läbi kasutades kolme erinevat kõndimismustrit – sirge muster, T-muster ja kolmnurk-muster. Need mustrid näitavad, kuidas jalakäija kõnnib vertikaalselt, horisontaalselt ja diagonaalselt radari suhtes. Vertikaalne liikumine tähendab radarist otse eemale ja tagasi kõndimist. Iga kõndimismuster salvestatakse kõigi viie radari konfiguratsiooniga ja tulemused analüüsitakse hiljem radari andmete visualiseerimistarkavaras. Kahe radari punktipilved liidetakse hiljem tarkvaras kokku.

Lõplikul eksperimentide tulemuste hindamisel kasutatakse kolme parameetrit. Esmalt hinnatakse, mitu punkti radarikaadri kohta tuvastati. Teiseks hinnatakse keskmist signaalitugevust. Kolmandaks kasutatakse visuaalsel hindamisel soojuskaardi graafikuid, mis näitavad objekti tuvastustäpsust. Kahe radari punktipilve liitmine suurendas punktipilve suurust, tänu millele punktide tõelevastavus suurenes. Üks radari seadistus töötas väga hästi, kolm radari seadistust töötasid hästi ja viimane radari seadistus ei andnud nii häid tulemusi. Viimaseks, arendatud tarkvara kasutati edukalt soojuskaardi graafikute analüüsis ja parima radari seadistuse leidmisel.

Kokkuvõtteks, kõik radari konfiguratsioonid parandasid jalakäija kõndimise tuvastustäpsust. Eksperimendid salvestati kahe radariga. Arendatud tarkvara andis vajalikud andmed, et leida kolme hindamisparameetriga parim radari konfiguratsioon.

Lõputöö on kirjutatud eestikeeles keeles ning sisaldab teksti 91 leheküljel, 6 peatükki, 62 joonist, 11 tabelit.

Abstract

Millimeter wave radars have numerous use cases in commercial and industrial systems as they extend the functionality of many other sensors. There are two problems with the millimeter wave radars that are addressed in this thesis. The first problem is getting few to no radar data points from small targets that are far away such as people and small vehicles and the second problem is the lack of software tools for analysing radar point cloud data. To this end, different radar configurations were investigated and a custom application was developed for comparing the configurations.

The experiments are conducted using a data collection device with two radars. All the desired software requirements are finished and all experimental results are analysed in the custom radar data visualization software. The custom application generated heat maps for visually analysing five different radar configurations. The application utility extends beyond this thesis as as it can used with future projects that use ROS bag format to store radar point cloud data.

The experimental results are conducted using three different walking patterns: line pattern, T-pattern and triangle pattern. These patterns showed a pedestrian walking in vertically, horizontally and diagonally in front of the radar. Vertical movement means walking straight away from the radar and then walking back. Each walking pattern is recorded with five different radar configurations and these results are later analysed in custom developed radar data visualization software. Two radars are recording simultaneously in each experiment and both radar point clouds are later combined in software.

Results are evaluated using three parameters: the average number of radar points per frame, average signal strength and visual evaluation in the heat map graph showing target detection accuracy. Combining radar point clouds from two radars reveal more detected data points, thus more information is received from pedestrians and small vehicles. Out of five radar configurations that were tested and compared, one radar configuration achieved excellent result, three configurations performed good and the last configuration has poor target detection accuracy.

In conclusion, almost all radar configurations improve the pedestrian walking detection accuracy with two radars. The developed radar data visualization application provided results for all three evaluation methods for finding the best radar configuration.

The thesis is in English and contains 91 pages of text, 6 chapters, 62 figures, 11 tables.

List of abbreviations and terms

API	Application Programming Interface
CPU	Central Processing Unit
IDE	Integrated Development Environment
IOT	Internet Of Things
VM	Virtual Machine
RADAR	Radio Detection and Ranging
AoA	Angle of Arrival
LIDAR	Light detection and ranging
LRR	Long-range radar
SRR	Short-range radar
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
NB	Narrow band
UWB	Ultra-wide band
FMCW	Frequency modulated continuous wave
EIRP	Equivalent isotropic radiated power
RF	Radio frequency
LCA	Lane-change assist
F/RCTA	Front/rear cross-traffic alert
AEB	Autonomous emergency braking
ACC	Adaptive cruise control
mmWave	Millimeter wave
MCU	Microcontroller
DSP	Digital signal processors
TI	Texas Instruments
LED	Light Emitting Diode
SSH	Secure Shell
USB	Universal Serial Bus
ROS	Robot Operating System
BLOB	Binary Large Object
LVDS	Low Voltage Differential Signaling

TI	Texas Instruments
SDK	Software Development Kit
mmWave	Millimeter wave Technology
PC	Personal Computer
HTTP	Hypertext Transfer Protocol
REST	Representational State Transfer
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
RCS	Radar cross section

Table of Contents

List of Figures

List of Tables

1. Introduction

Sensors and their various applications have changed the way we interact with the world around us. Sensing systems have numerous use cases in various industries. For instance, driverless cars, trucks and buses driving in smart cities where information is sensed, collected and used to reduce traffic congestions and promote safer roads. In addition to that, all types of robots are using sensors to sense the environment they are in. These systems are becoming more prevalent in commercial and industrial landscape as they are safer, more efficient and more autonomous [1]. The existing sensing technologies such as radars, LIDARs, thermal cameras, vision cameras and passive infrared are capable of detecting occupancy and the location of an object.

On the negative side, radars do not produce as many data points with detecting pedestrians as LIDAR systems do. More specifically, short-range and long-range radars working in the 77 GHz frequency band. However, radars have advantages over optical sensors. For instance, radar can easily work in foggy weather conditions [2]. Moreover, radars provide object distance and its velocity while optical sensors do not have this capability, except LIDAR which provides only object distance [3].

The millimeter wave radars are used in smart cities, smart transportation, smart home, security, rail transit, drones and in other commercial and industrial systems. For now, car industry is one the main drivers with millimeter wave radar technology and Internet of Things is closely following in the coming years.

AWR1843BOOST Evaluation boards [4] which have only one radar device on the module are used in this project. This systems costs around 300 euros. One other radar on the market is a cascade AWR2243 radar [5] that has 4 hardware synced radars situated on one board. This system costs around 3000 euros. The hardware synchronization ensures that all radars are transmitting and receiving signals using a master synchronization signal. Even though cascade radars have more accurate object detection, these systems are 10 times more expensive than one radar system.

The thesis elaborates on numerous topics regarding the millimeter wave technology radars. First, mmWave radars are used to improve smaller target detection accuracy such as people

and smaller vehicles. The target detection accuracy is improved using two radars and experimenting with several radar configurations. Secondly, radar data visualization tool is developed for analysing and comparing the results of several walking pattern experiments. Actually, the utility of the developed radar data visualization application goes beyond this thesis and can be later used for reading and analysing recorded ROS bag files produced by multiple radars.

1.1 Problem Definition

While radars have advantages over optical systems, they produce much less data points per object. In traffic monitoring, small moving targets such as children, adults, bicycles and other small vehicles are way harder to detect and classify than larger objects. For example, people crossing the road or walking along the sidewalk produces few to no data points depending on the distance to the measuring device.

As stated previously, the cascade AWR2243 radar system uses hardware synchronization between multiple radars for efficiently combining radar point cloud data into one stream of detected points. Without using a hardware synchronization signal, multiple radars might cause disruptive signal interference. Hardware synchronization has more advantages over combining multiple radar point cloud data in software. However, it can not always be used for various reasons.

Another problem working with millimeter radars is visualizing large recorded data files for giving a visual context through maps and graphs. The existing software solutions that are freely available are mainly used for displaying the recorded radar data. That is to say, these solutions lack features needed for radar data analysis and comparing various radar configurations with one and two radars.

1.2 Motivation

As stated in the previous section, the millimeter radars are great solution for detecting objects as they have some extended features over optical sensors. The radar system has a low system cost and works better in bad weather conditions. This is the reason what makes this technology attractive for various industries.

The motivation for using two AWR1843 radars is to find out if they provide better results after these radar point clouds are combined in software rather than in hardware as it is done in cascade radars. Another goal is to find out how much signal interference occurs when two

AWR1843 radars are operating at the same time without being synchronized in hardware. It is important to note that the intention is not getting better results with combining two radar point clouds than with a cascade radar that is using a hardware synchronized signals between multiple radars. The intention is understanding and revealing better ways to improve a walking pedestrian detection accuracy and getting more data points with more than one radar. In addition to improving target detection accuracy is to modify TI radar configuration file. This file contains information about how millimeter wave radars operate, thus changing various parameters will improve target detection in different environments where the device is used.

A better target detection accuracy is achieved when two radar point clouds are merged and radar data is analysed using radar data visualization software. The data visualization has a key role in creating a clear idea of what the information means. The translation of radar recorded data into a visual context such as a heat map or other graphs is giving insight how to proceed with the research and generates new ideas. In this thesis, several existing solutions are considered and discussed. The radar data visualization software should be able to play camera feed and multiple radar data and has tools for data analysis. None of the considered software tools have the desired features, thus custom Flutter web application is developed.

1.3 Task Definition

The main tasks in this thesis are defined below.

1. Finding the best radar configuration to achieve the best small object detection accuracy in the given limitation and environment
2. Developing a radar data visualization software for analysing and displaying recorded data from two radars
3. Analysing the walking pattern experiments with two radars to find the best pedestrian and small vehicle detection accuracy

Task 1. The radar configuration file defines the parameters, options, settings and preferences used in configuring the radar device. Two radar configurations are generated, modified and tested with two radars to achieve the best pedestrian detection. Two radars with the same radar configurations produces two streams of radar point cloud data and is later stored in a ROS bag file format. This is later synchronized and analysed in a custom developed radar data visualization application. The radar visualization application enables comparing and analysing multiple radar configuration file produced results.

Task 2. The main desired requirements for a radar data visualization application are the following.

- Parse ROS bag file containing information about two radars
- Display radar point cloud data from two radars
- Play camera feed with both radar cloud data
- Generate heat maps from the signal intensity of the detected targets
- Manipulate camera feed and radar data easily

The Flutter Web Application is going to be developed as there are not any other solutions available that met the desired requirements. The application can be used to generate a heat map for each radar and a combined results of two radars. The heat maps will show the object detection accuracy and the signal strength. These heat maps are later used to analyze and compare the experimental results.

Task 3. Three walking patterns are used to test the radar capability to detect a pedestrian movement in front of the two radars. The three walking patterns are **line pattern**, **T-pattern** and **triangle pattern**. All these patterns test the radar capability to detect objects with vertical, horizontal and diagonal movement. The vertical movement means walking straight away from the radar and then back. These walking patterns are tested with various radar configuration files and later compared and analysed visually with the Flutter radar data visualization application.

1.4 Thesis Outline

Chapter 2 provides information about the radar fundamentals and explains how radar range, velocity and angle are estimated with various examples and equations. In the end of the chapter, 24 and 77 GHz frequency bands are discussed as well as the benefits of using 77 GHz frequency band radars.

Chapter 3 gives an overview of the system used to conduct walking pattern experiments and explains the peripherals inside the system. In addition, this chapter elaborates on how the radar point cloud data is encapsulated and how two radar system stores the data. More, the process of generating radar configurations is discussed.

Chapter 4 describes radar data visualization software development. First, all desired requirements are listed. In the next section, three existing applications for radar data visualization are mentioned and then the developed Flutter web application for visualizing radar data is discussed with all the features and the guide on how to use the application. In

the following sections, the software design and implementation are described in depth.

Chapter 5 reveals the experimental results of three different walking patterns. First, three long-range radar configurations and two short-range configurations are described. Lastly, walking pattern experimental results of a two radar system is elaborated and compared with five different radar configurations.

2. Theory of radar operation

In this chapter millimeter wavelength radars are discussed. Millimeter wavelength (mmWave) is considered short wavelength in the electromagnetic spectrum. The short wavelength has advantages which makes this technology attractive for many applications for its high accuracy. In addition, sensor size can be reduced as antenna and other radio frequency components are smaller. A radar system operating at 76-81 GHz has the capability to detect object movements that are as small as a fraction of a millimeter.

2.1 Radar Fundamentals

A special class of radar technology uses short wavelength to achieve a millimeter electromagnetic waves. This class is called a millimeter wavelength radars (mmWave). Radar systems are using transmitter antennas to produce electromagnetic waves. These transmitted signals are sent as chirps. Chirps will be discussed later in this chapter. After chirp signal is transmitted it is reflected back from an object or from multiple objects situated in front of the radar. The reflected chirp signal is captured by a receiving antenna array and processed by the radar system to determine the range, velocity, and angle of the detected objects.

Millimeter wave radar system has multiple components. The radio frequency (RF) components include transmitting (TX) and receiving (RX) antennas with the matching network of resistors and capacitors. In addition, analogue components are used in clocking and in other timing features. Digital components are responsible for converting constant frequencies coming from the RF module. For instance, analogue-to-digital converters are used to digitize information and microcontrollers (MCU) process the sampled data. Digital signal processors (DSP) are constantly calculating object distances, velocities, and angles. All mentioned components are illustrated in Figure 1.

Traditionally, these modules were designed in separate packages which increased power consumption and overall system cost. Now, numerous semiconductor companies are designing everything in one package for better performance and efficiency in power domain. The complexity and high frequency components are making the system design challenging [6].

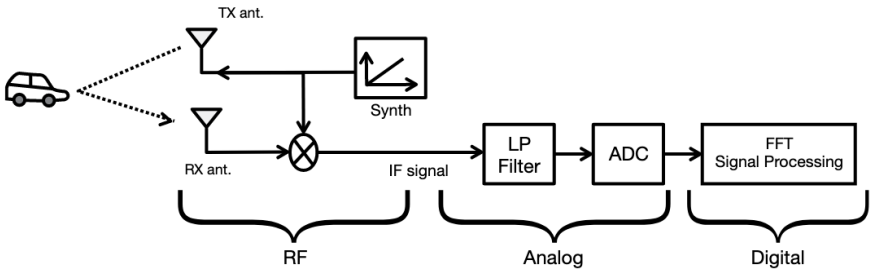

Figure 1. *RF, Analog and Digital Components of An FMCW Sensor* [6]

2.1.1 What Is A Chirp

There are two main types of radars available such as continuous wave (CW) and continuous wave frequency modulated (FMCW) radars. The continuous wave radars are transmitting and receiving a stable frequency and the continuous wave frequency modulated radars are using increasing frequency over certain amount of time. The FMCW signal is called a chirp.

Texas Intruments (TI) devices implement a special class of mmWave technology called frequency-modulated continuous wave (FMCW). The FMCW radars are transmitting chirps which are frequency-modulated signals sent continuously. This technique enables to measure object range, angle and velocity with high accuracy. This method results in better target resolution while using less peak input power. The nature of chirp signal has an advantage of being significantly harder to block or be disrupted as it uses a whole range of frequencies. In addition, chirp signal is more immune to certain forms of noise [7].

A chirp is a sinusoidal wave and its frequency increases linearly with time. There are two convenient ways to represent a chirp visually as shown in Figure 2. The first graph (a) is a frequency over time plot (or f-t plot) and the second graph (b), is a amplitude vs time (or A-t plot).

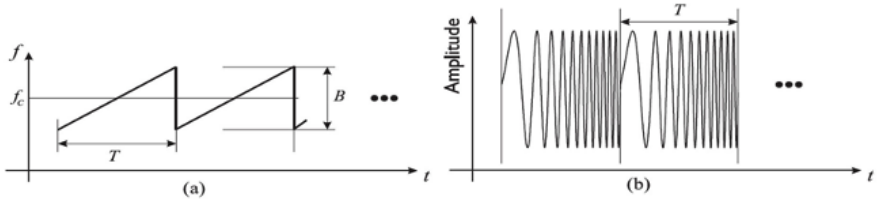

Figure 2. *Frequency over Time (f-t) Plot (a) and Amplitude over Time (A-t) Plot (b)* [8]

The main characteristics of a chirp are start frequency (f_c), bandwidth (B), duration (T_c)

and slope (S) as shown in Figure 3. As mentioned before, the short-range radars are operating in 77-81 GHz frequency range, thus the chirp start frequency is 77 GHz and ends at 81 GHz. This is why, the bandwidth in this case is 4 GHz. The slope (S) defines as how fast the frequency can ramp up from 77 to 81 GHz. This ramp up time is the duration of chirp.

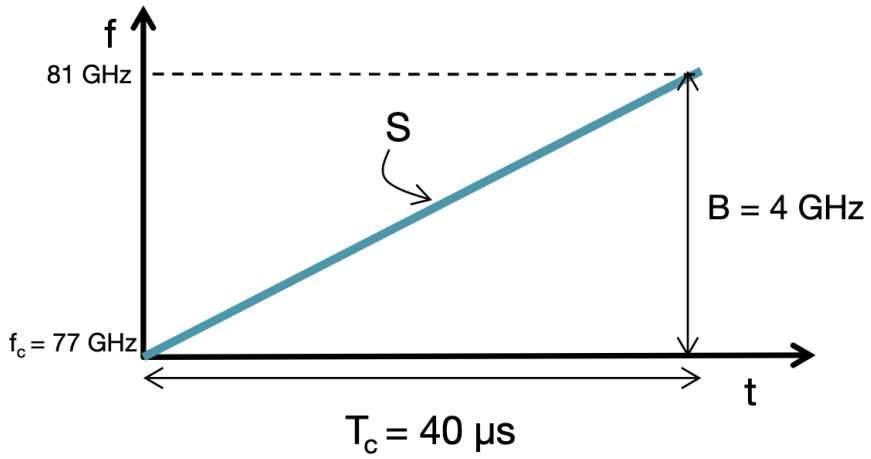

Figure 3. *Chirp Signal in Frequency over Time plot* [6]

For instance, if the parameters were taken from the Figure 3. The chirp has a sweeping bandwidth of $4GHz$ and the chirp duration is $40\mu s$. The slope of this chirp will result in $100MHz/\mu s$ [8].

2.1.2 Instantaneous Frequency

A FMCW radar system has numerous components before analogue signal from the RF components is sampled by the ADC. These components are illustrated in Figure 4 and labeled from one to four. The first component generates a chirp and is called a synthesizer. The second component is an antenna array transmitting the generated chirp. The transmitted chirp is reflected back from any objects situated in front of the radar and captured by the third component (receiving antenna array). Since the sampling rate of ADC is not capable to sample any frequency around 77 GHz the fourth component mixer is mixing the transmitting chirp with the receiving chirp signal and results in constant or instantaneous frequency (IF) signal. As the name implies, this frequency is constant and can be sampled by the ADC for further signal processing. Different frequencies of IF signal can be exploited to detect object distance and the frequency phase can be used to derive object velocity in certain cases. [6]

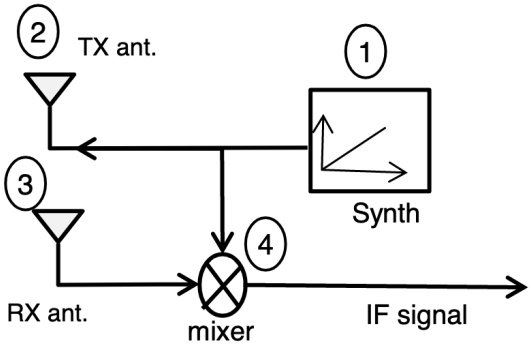

Figure 4. *A 1TX-1RX FMCW Radar System* [6]

Mixer Component in Radar Module

A mixer is a 3 port device with 2 inputs and 1 output as shown in the Figure 5. The two inputs are a transmitted chirp and a received chirp. The output is a sinusoidal wave equal to the difference of the two input chirps signals. In addition, the phase equals to the two input sinusoidal waves.

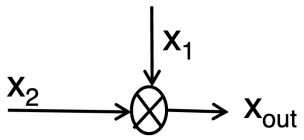

Figure 5. *A Mixer Diagram* [6]

The x_1 and x_2 sinusoidal waves in Figure 5 can be described in equation 2.1.

$$x_1 = sin[w_1 t + \phi_1], x_2 = sin[w_2 t + \phi_2] \tag{2.1}$$

This results in calculating the x_{out} as shown in equation 2.2. The constant frequency (IF) is further elaborated in range chapter.

$$x_{out} = sin[(w_1 - w_2)t + (\phi_1 - \phi_2)] \tag{2.2}$$

2.2 Range Estimation

The range estimation of an object is directly related to the constant frequency of an arrived signal. First, the IF signal can be understood easily using frequency over time plot as illustrated in Figure 6. Two plots are drawn on top of each other. The top plot shows how TX chirp is transmitted and the reflected RX chirp is received some time later. The time it takes the TX chirp to be transmitted and reflected back from an object to an receiver antenna is called τ. The slope (S) of the chirp is determined by the radar system and is limited by the ADC sampling rate. [8]

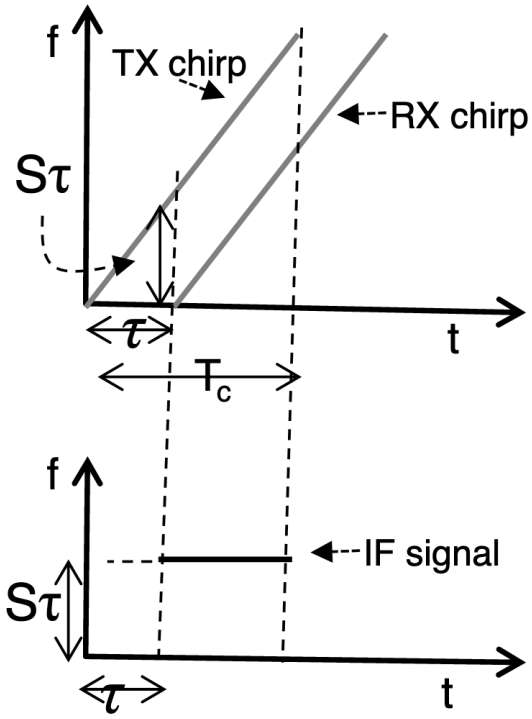

Figure 6. *The Instantaneous Frequency (IF) Signal* [8]

A single object in front of the radar produces a constant frequency in the mixer. This IF signal can be calculated as shown in equation 2.3.

$$S\tau = \frac{S2d}{c} \tag{2.3}$$

For example, the object is distanced at $d = 50m$, and the slope of the chirp is $S = 50MHz/\mu s$. The c is the speed of light. The constant frequency produced by the single object in front of the radar is $16.7MHz$ as shown in equation 2.4.

$$S\tau = \frac{S2d}{c} = \frac{50MHz/us * 2 * 50m}{3 * 10^8 m/s} = 16.7MHz \tag{2.4}$$

Multiple Object Detection

The concept of detecting multiple objects with one transmitted chirp is very similar to detecting only one chirp. First, a single chirp is transmitted by the radar antenna array and all the detected objects will produce a constant frequency tone in the frequency over time plot (graph a) as shown in Figure 7. The other plot (graph b) illustrates the constant frequency peaks produced by the three objects in front of the radar. If two objects are too closely spaced, then they might show up as one peak in the frequency spectrum. This issue can be mitigated by increasing the range resolution. The multiple tones are processed in range Fast Fourier Transform (range-FFT). These calculations are done in the radar DSP module.

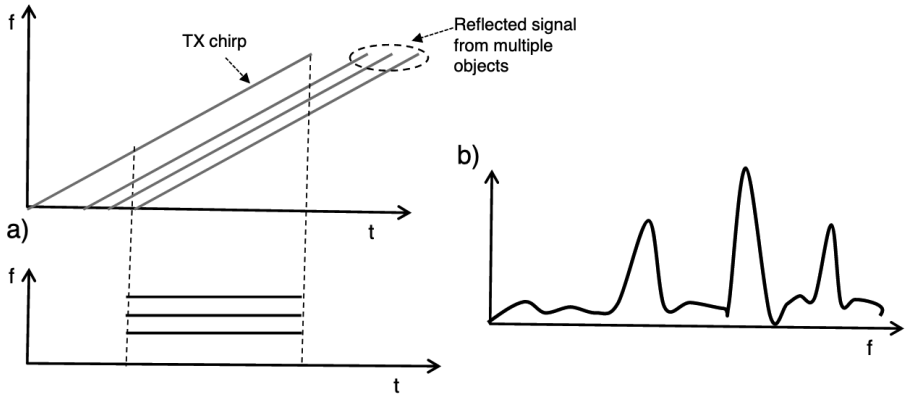

Figure 7. *A 1TX-1RX FMCW Radar System* [8]

It is important to note that if multiple objects are in the same distance to the radar, they will all produce the same constant frequency tone. These objects have all specific angle to the radar and can be separated by further signal processing. Finding the object angle will be discussed later in this chapter.

Maximum Range

The maximum distance of an object is directly limited by the ADC sampling rate, thus the IF signal must be commensurate with the desired maximum distance. The maximum constant frequency is defined as $f_c = \frac{S2d_{max}}{c}$.

For example, the maximum desired distance is $d_{res} = 300m$ with the chirp slope of $S = 40MHz/\mu s$. This will result to $80MHz$ as shown in equation 2.5. This means, the

ADC must be able to sample at least a frequency of 80MHz.

$$f_c = \frac{S2d}{c} = 40MHz/us * \frac{2 * 300m}{3 * 10^8 m/s} = 80MHz \tag{2.5}$$

When analysing the maximum distance formula, the slope of the chirp can be lowered to achieve greater range. Typically, shortwave radars tend to use lower slopes for lower ADC sampling rate.

2.2.1 Range Resolution

The range resolution shows the minimal distance between two objects where they are still detected as two objects. As mentioned in the previous sections, if two objects are closely spaced, then they might show up as one peak in frequency spectrum as shown in Figure 7. The range resolution (d_{res}) depends only on the bandwidth swept by the chirp. Thus, the range resolution can be increased by proportionally increasing the sweeping bandwidth [6]. The range resolution is defined in equation 2.6:

$$d_{res} = \frac{c}{2B} \tag{2.6}$$

For instance, a short-range radar in 77-81 GHz frequency band has a bandwidth of 4 GHz. This translates to a range resolution of 3.75cm [8].

2.3 Velocity Estimation

Further IF signal processing enables to measure the object's velocity. This processing is based on Doppler effect which determines if the object is moving or stationary. Moreover, further radar data processing determines the direction of an object and how fast is it moving. [8]

A single object velocity is measured using two chirps. The FMCW radar transmits two chirps separated by a certain amount of time T_c. Each reflected chirp is then processed through range-FFT to calculate the distance of all detected objects. Since those two chirps bounce off the same object, they both have the same distance. Hence, they have peaks in the same location in the frequency spectrum, but with different phases. This effect is illustrated in Figure 8 [6].

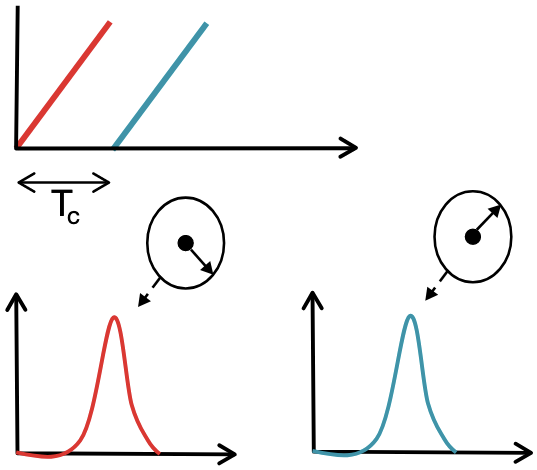

Figure 8. *Differing Frequency Phases of Two Received Chirps* [6]

Furthermore, the velocity is measured using two chirps as shown in Figure 8. Two chirps will be reflected off from the moving object. The displacement of an object is so small that range-FFT will recognize it as one object in the frequency spectrum. As stated before, the distance processed in range-FFT is the same, but both RX chirp IF signals have different phases. These phases are processed in the next FFT called Doppler-FFT. The phase difference (ω) must be measured between two IF signals. Velocity can be calculated using equation 2.7

$$V = \frac{\lambda\omega}{\pi 4 T_c} \tag{2.7}$$

IF Signal Phase

The phase of the instantaneous frequency (same as constant frequency) can be used to quickly and accurately measure the velocity of detected objects. The phase measurement is a foundation for some applications where high precision in object displacement is needed. For instance heartbeat monitoring and vibration detection.

The phase difference is illustrated in Figure 9. The displacement from an object is too small to be detected as two peaks in the frequency spectrum. Although, the phase of the signal has changes 90 degrees.

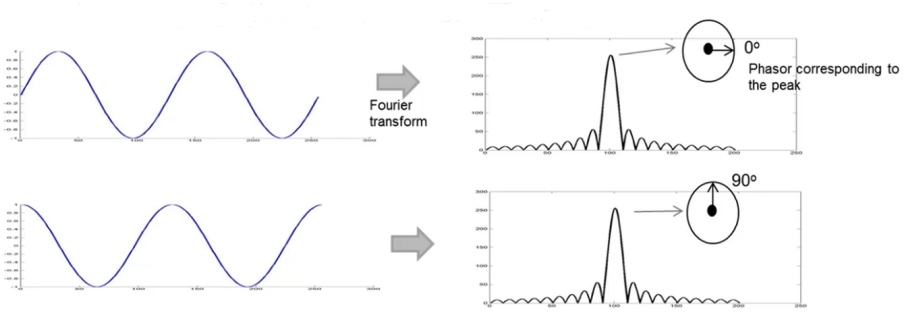

Figure 9. *The Phase Difference Between Two Chirps* [8]

The phase difference $\Delta\phi$ is calculated in equation 2.8 where Δd is the displacement from an object.

$$\Delta\phi = \frac{4\pi\Delta d}{\lambda} \tag{2.8}$$

For instance, a chirp has a slope of $S = 50MHz/\mu s$ and a chirp time of $T_c = 40\mu s$. The object in front of the radar changes its position by 1mm. (for 77 GHz radar $1mm = \lambda/4$). First, let's find out the phase change with equation 2.9.

$$\Delta\phi = \frac{4\pi\Delta d}{\lambda} = \pi = 180° \tag{2.9}$$

Next, let's find out how much IF signal changes with 1mm object displacement. The equation 2.10 shows that IF signal frequency changes by 333 Hz. The 333 Hz frequency change from 1mm object displacement sounds enough to be noticeable in the frequency spectrum. Although in the observation window, this displacement equals only $\Delta f T_c = 333 * 40 * 10^{-6} = 0.013$ cycles.

$$\Delta f = \frac{S2\Delta d}{c} = \frac{50 * 10^12 * 1 * 10^{-3}}{3 * 10^8} = 333Hz \tag{2.10}$$

Therefore, the range-FFT cannot be used for very small object displacement, but phase of the IF signal is very sensitive to small changes in object range.

Chirp Frame

If there are multiple objects moving at the same range, then two-chirp velocity measurement cannot be used. All moving objects equidistant to the radar are producing identical IF signals. Moreover, all objects will result in one peak in range-FFT, thus simple phase comparison technique will not work. This is why, more than two chirps have to be transmitted to get more accurate velocity measurements. The radar must send a set of N equally spaced chirps as shown in Figure 10. This set of chirps is called a chirp frame T_f. [6]

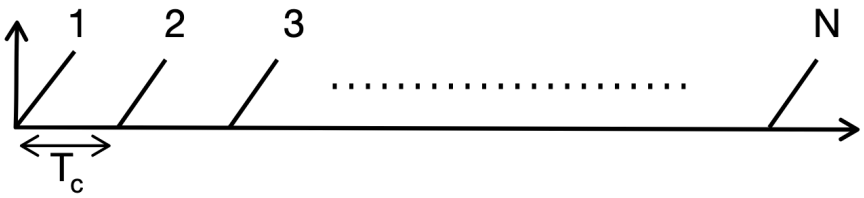

Figure 10. *Chirp Frame* [8]

Measuring velocity using chirp frames creates a lot of phasors as shown in Figure 11. The chirp frame will produce N amount of phasors and in this case, all these phasors will belong to the two moving objects.

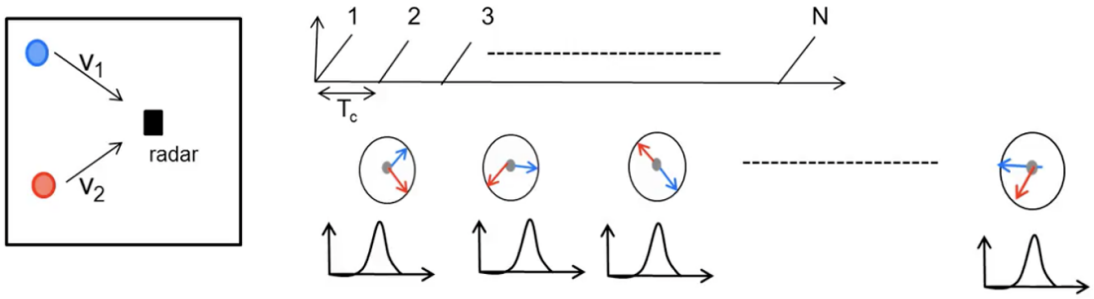

Figure 11. *Chirp Frame Helps Separating More Than One Object* [8]

This is why, further signal processing is needed after range-FFT. The second FFT is called Doppler-FFT. Signal processing is done on the N amount of phasors to resolve the velocity of two objects as shown in Figure 12. All N amount of phases resolved into two peaks in the phase plot. Therefore, this is how the velocity of two moving objects in front of the radar is measured. [6]

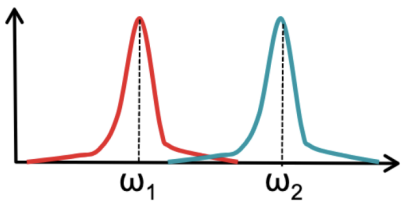

Figure 12. *Two IF Signal Phases Separated After Doppler-FFT Processing* [6]

The object velocities are calculated using the equation 2.11 where $\lambda = 1mm$ (for 77 GHz radar), T_c is the time between two chirps in the chirp frame and ω is the phase difference shown in Figure 12.

$$V_1 = \frac{\lambda\omega_1}{4\pi T_c}, V_2 = \frac{\lambda\omega_2}{4\pi T_c} \tag{2.11}$$

Maximum Velocity

The maximum relative speed (v_{max}) that can be measured by two chirps spaced T_c apart is shown in equation 2.12. Therefore, higher maximum velocity requires more closely spaced chirps.

$$V_{max} = \frac{\lambda}{4T_c} \tag{2.12}$$

Velocity resolution

The velocity resolution shows what is the minimum velocity increment that is detected when object is moving. More, it is inversely proportional to the frame time (T_f) and is given by the equation 2.13.

$$V_{res} = \frac{\lambda}{2T_f} \tag{2.13}$$

2.4 Angle Estimation

Angle estimation requires at least 2 RX antennas to successfully find out the location of an object. In simple terms, when chirp is transmitted, then each receiving antenna captures the signal at slightly different times. The distance from the object to the first antenna

is d and the distance to the second antenna is a little bit longer $d + \Delta d$ as shown in the Figure 13. This additional distance results in an additional phase of $\omega = \frac{2\pi\Delta d}{\lambda}$. This is the phase difference between the signal at the first RX antenna and the signal at the second RX antenna. [6]

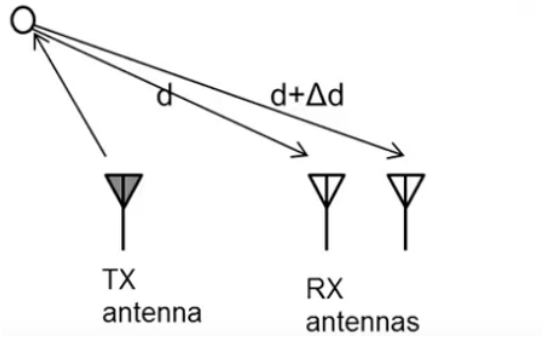

Figure 13. *Object Angle of Arrival (AoA)* [8]

Angle estimation is done in the next signal processing step called angle-FFT. This signal processing step takes place after both antenna range-FFT and Doppler-FFT processing is finished. The object position is calculated using the phase difference in Doppler-FFT. When two objects will result in the same distance in range-FFT and in the same velocity in Doppler-FFT, then this situation can be solved using the angle of both object relative to the radar. This can be done when comparing the results coming from two antennas. The two phases are different, thus the angle of the object can be derived from this information.

The angle estimation accuracy depends on the angle of arrival as shown in Figure 14. The most accurate reading is when θ is close to zero and the accuracy reduces when θ approaches to 90 degrees.

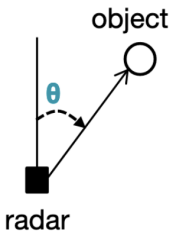

Figure 14. *Object Angle of Arrival* [6]

Maximum Radar Field of View

The maximum field of view depends on how closely (l) are two RX antennas placed on the sensor board. Therefore, the maximum field of view is shown in equation 2.14.

$$\theta_{max} = sin^{-1}(\frac{\lambda}{2l}) \tag{2.14}$$

Measuring Radar Angle of Arrival

The first step is TX antenna transmitting a chirp signal. The data is received at each of RX antennas. Each of these antennas process the data to create a 2D-FFT matrix (same as Doppler-FFT) with a peak corresponding to the range and the velocity of the object as shown in the Figure 15. The measured difference phase (ω) between the two RX antennas can be used to estimate the angle of arrival of the object.

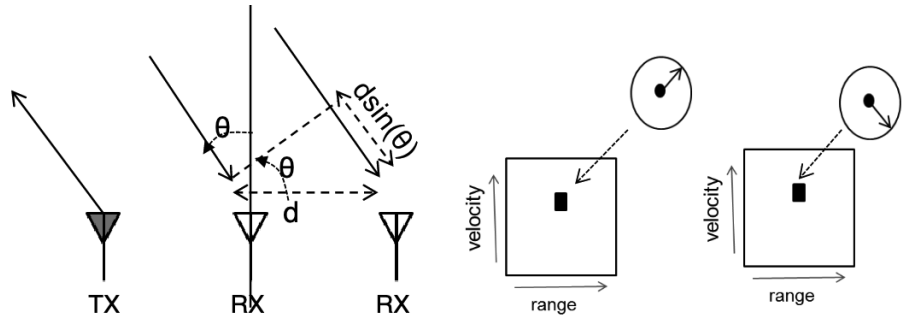

Figure 15. *The IF Signal Phase Difference for Two RX Antennas* [8]

The angle can be calculated using the following equation 2.15 where d is the distance between two RX antennas and ω is the phase difference.

$$\theta = sin^{-1}(\frac{\lambda\omega}{2\pi d}) \tag{2.15}$$

Angle Resolution

Angle resolution (θ_{res}) is the minimum angle separation for the two objects to appear as separate peaks in the angle-FFT. Therefore, angle resolution is often quoted assuming $d = \lambda/2$ and $\theta = 0 => \theta_{res} = \frac{2}{N}$, where N is the number of RX antennas.

2.5 24 GHz and 77 GHz Frequency Bands for Radars

Radar systems have numerous advantages over LIDAR systems and cameras as they can successfully operate in poor weather and in other extreme conditions where visibility is next to zero. In addition to that, radars are operating at longer distances. The main purpose of using radars is to determine object range, speed and their angle. This opens a new range

of applications for short-range and long-range radars using millimeter wavelength radars. This section discusses radars in the 24 GHz and 77 GHz frequency band. [9]

The majority of used radars today are pulse type radars. Radars are transmitting a very large burst of radio frequency signal energy for a very brief interval and then wait for the response. From these responses object position, size, type and movement may be determined. These pulse type radars are mainly used in weather observation, travel aids for the blind, airport flight control, and other detection systems. The frequency modulated continuous wave radars (FMCW) are popular within automotive and industrial space in 24 and 77 GHz frequency bands. They are more power efficient and show better results in numerous applications.[7]

2.5.1 24 GHz Frequency Band

The 24 GHz band includes an industrial, scientific and medical (ISM) frequency band from 24.05 GHz to 24.25 GHz, which is often called the narrow band (NB). The narrow band has a bandwidth of 200 MHz and is unlicensed. The 24 GHz band also includes an ultra-wide band (UWB) which is 5 GHz wide.

The narrow band and ultra-wide bands of 24 GHz are mainly used for legacy automotive sensors for short-radar purposes. For instance, blind spot detection is a great example for narrow frequency band, but UWB is used for high-range resolution application. Due to spectrum regulations and standards developed by the European Telecommunications Standards Institute (ETSI) and Federal Communications Commission (FCC), the UWB band is phased out since January 1, 2022 [10].

There are numerous applications working on ultra-wide frequency band, especially in automotive industry. The lack of 24 GHz wide bandwidth makes this band unattractive for new radar applications.

2.5.2 77 GHz Frequency Band

There are multiple bands for various applications available in 77 GHz frequency band for automotive and in industrial use shown in Figure 16. For instance, in 76-77 GHz frequency band has 1 GHz of bandwidth for automotive long-range radar (LRR); in 77-81 GHz band, there is 4 GHz of bandwidth for automotive short-range radar band (SRR) and in 75-85 GHz frequency band there is 10 GHz of bandwidth for industrial fluid level sensing.

The long-range radar applications have 76-77 GHz frequency band where higher equivalent isotropic radiated power (EIRP) is allowed. This means radars can transmit signals with higher power which results in detecting object further away. For instance, it can be used for adaptive cruise control where radar needs to see further away for keeping a safe distance between two cars. This frequency band is available both in Japan and in Europe. This makes it more attractive to use in numerous applications as in vehicle counting, traffic jam or accident detection or traffic light activation.

Short-range radar applications are using 77-81 GHz band. The ultra-wide bandwidth of 4 GHz is the reason why this band has gained a lot of industry adaption where high range resolution is needed. In future, automotive and industrial sensors are likely to use only 77 GHz frequency band sensors.

The last band in 77 GHz is for industrial fluid sensing. This industry needs very high resolution for industrial fluid and solid level sensing. Since wider bandwidth means higher resolution, this band enables applications to use 10 GHz of bandwidth in 75-85 GHz band.

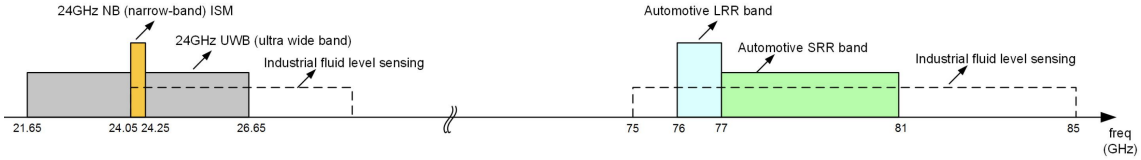

Figure 16. *Automotive and Industrial Sensor Frequency Bands In The 24 GHz and 77 GHz* [10]

77 GHz Frequency Band Benefits

One of the key advantages of 77 GHz frequency band is the wide bandwidth it provides: 1 GHz bandwidth for long-range radars; 4 GHz for short-range radars and 10 GHz for industrial fluid sensing. The 24 GHz frequency band only has narrow bandwidth of 200 MHz available. Therefore, 77 GHz is very attractive for new applications compared to the 24 GHz band. This is a key reason why many are leaving 24 GHz band for 77 GHz frequency band.

For instance, wide bandwidth significantly improves the range resolution to detect two closely spaced objects in front of the radar. In addition, wide bandwidth enables higher accuracy to measure distance of detected objects. Range resolution and accuracy are inversely proportional to the sweep bandwidth. The 77-81 GHz frequency band has 4 GHz of bandwidth compared to 200 MHz bandwidth in 24 GHz band. This comparison says that 77 GHz band radars can achieve 20x better performance in range resolution and accuracy compared to the 24 GHz frequency band. The achievable range resolution is

around 4cm versus 75cm for 24 GHz.

Likewise, velocity resolution and accuracy are significantly improved in 77 GHz frequency band compared to legacy radar systems. Velocity resolution and accuracy are inversely proportional to the radio frequency (RF), thus higher frequency band leads to better measurements. The 77 GHz band has three times better accuracy and resolution compared to 24 GHz band sensors (see Figure 17).

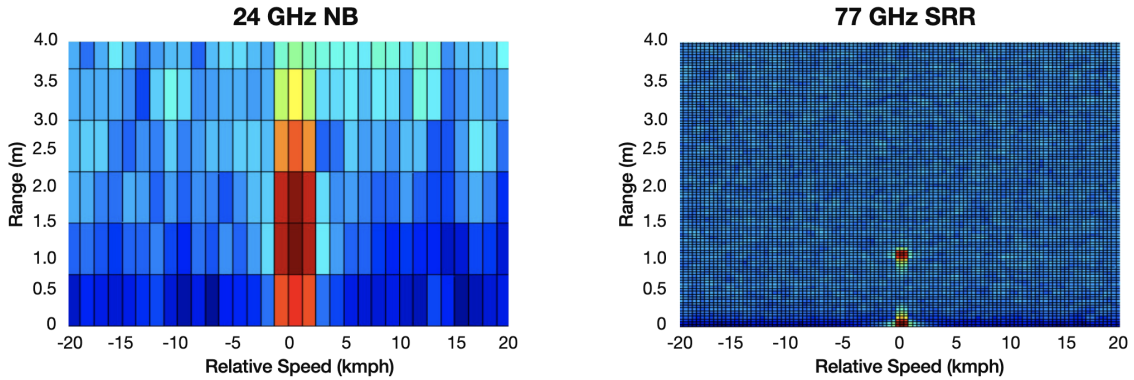

Figure 17. *The Speed and Range Resolution Comparison Between 24 and 77 GHz Frequency Bands* [10]

Higher radio frequencies enable to produce smaller sensors sizes as antenna array can be around 3 times smaller in both x and y dimension. In automotive applications size reduction is crucial since car doors and other places where sensor has to fit are tight. In industrial fluid sensing, higher frequencies enable using narrower beam for the same sensor and antenna size. This mitigates unwanted reflections from the sides of the tanks and also other interfering obstructions within the tank [11].

77 GHz Frequency Band Applications

In automotive space there are numerous applications for both short and long range radars. For instance, short-range car corner radars can be used as blind spot detection (BSD), lane change assist (LCA) or front/rear cross-traffic alert (F/RCTA). The long-range car front radars have other applications. For example, using mid-range and long-range radars for autonomous emergency braking (AEB) or adaptive cruise control (ACC)

In industrial space, radars can be used for fluid and solid level sensing, traffic monitoring or in robotics.

3. Data collection

In this thesis, all experimental data is collected using a data collection device, built by TalTech Embedded AI Research Lab for the Future Transport Ecosystem Management Solution [12] project. Data collection is important for testing the radar system and storing ROS bag files for later use. The radar system consists of multiple peripherals such as two radars, camera, embedded Linux platform and other smaller circuits. Moreover, radar configuration files are important part of how radar operates. These files are used to configure radar's parameters, options, preferences and settings.

3.1 Data Collection Device

The data collection device is composed of multiple radars and a raspberry PI camera. The purpose of camera is to give a better overview what the radars are detecting. The system is battery-powered and mobile.

3.1.1 Overview

The overview of the radar system is illustrated in Figure 18 where all the components are numbered. Components one and two are AWR1843BOOST Evaluation Modules [4]. They are connected to a NVIDIA Jetson Nano computer [13] (component 4). The radar evaluation modules and Nvidia Jetson Nano computer is connected via serial communication and powered using 5V power supply. The reset circuitry (component 5) is used to control the hardware reset for the evaluation modules. The third component is Raspberry PI camera module [14] and it is connected to the Jetson computer via MIPI CSI-2 connector. The whole radar system is powered using 12V battery. These components correspond to the same numbers in Figure 20.

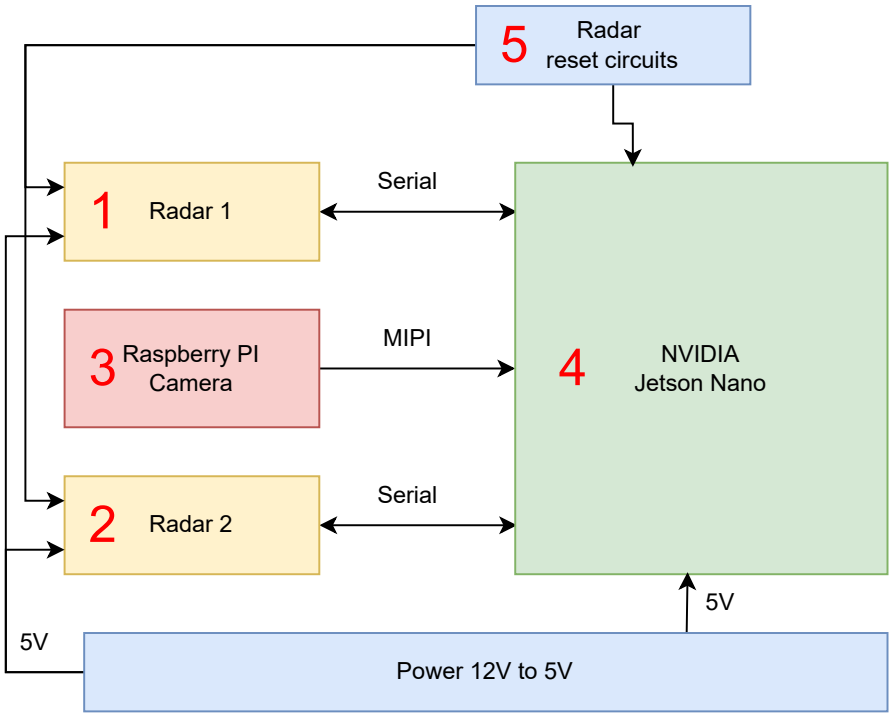

Figure 18. *The Data Collection Device Diagram*

This research is done with the Texas Instrument AWR1843BOOST Evaluation Module [4]. This module is easy-to-use 77 GHz millimeter wave sensor evaluation board for the single-chip AWR1843 device. This evaluation kit is supported by Texas Instruments mmWave tools and software. The AWR1843 is a single-chip 76-GHz to 81-GHz automotive radar sensor integrating antenna on package, DSP, and Arm Cortex-R4F based microcontroller.

All radar components are placed inside a plastic box. This a protection for weather. This plastic box is placed on a tripod as shown in Figure 19.

Figure 19. *The Data Collection Device*

3.1.2 Peripherals

The list of components used in this radar system is described below and illustrated in Figure 20.

1. Radar 1 (AWR1843Boost Evaluation Module)
2. Radar 2 (AWR1843Boost Evaluation Module)
3. Raspberry Pi camera module
4. Nvidia Jetson Nano Developer Kit
5. LED circuitry for indicating system status

All the components in the radar system are highlighted with red in Figure 20. The number one and two are the AWR1843BOOST Evaluation Modules. The component number three is a Raspberry PI camera module. This camera is used to film the scene while radars are recording data into a ROS bag file format. The number four component is the main computer Nvidia Jetson Nano Developer Kit. This computer is managing radars and camera data. In addition to that, ROS is running on the Jetson Nano. This enables recording ROS bag files. The component number five is controlling LEDs for indicating

system status.

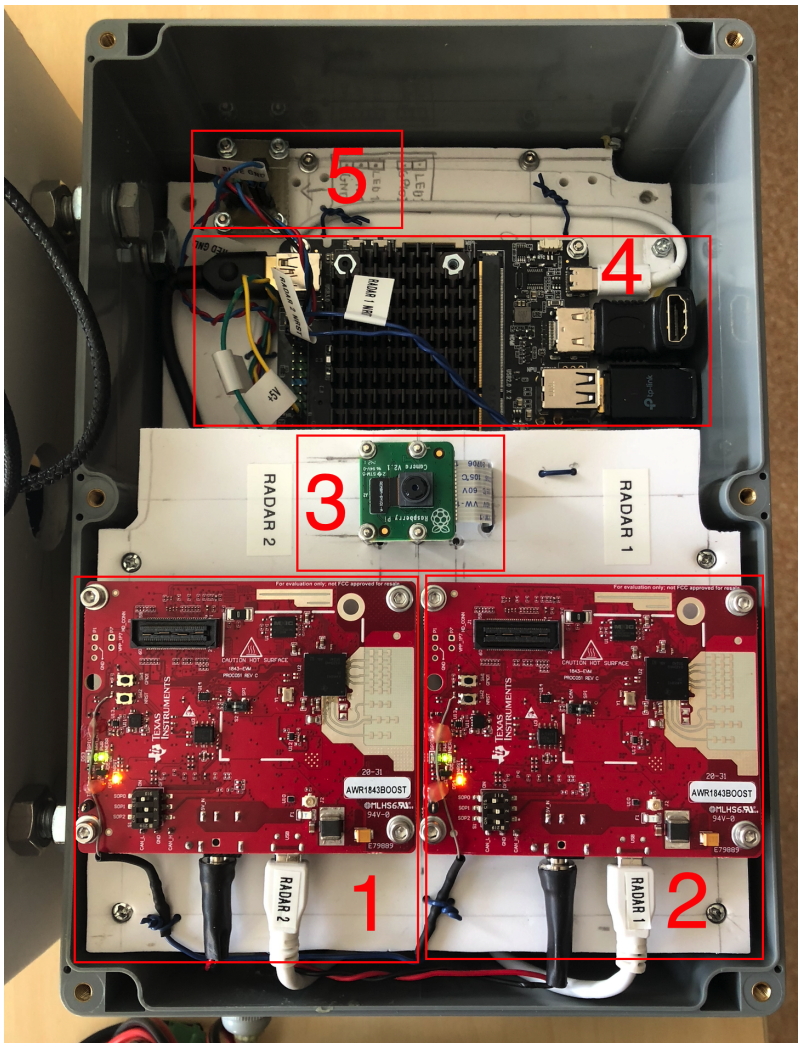

Figure 20. *The Data Collection Device Components*

The radar system is mobile and can be taken anywhere to start recording data. It is powered by 12V battery. First, the radar power is switched on and the LED power indicator shows when Linux has finished booting up the system. Then, SSH is used to remotely start the radar and start recording. The radar starts with the default configuration file, but this can be changed. Each time radar scripts are started, a radar configuration is sent to the radar.

The radar evaluation boards and Jetson Nano computer are connected via USB cable and communication is over serial at 921600 bit/s. The evaluation board also has a LVDS interface accessing sensor's raw data although this is not used in this system as it requires additional hardware. Texas instruments have provided numerous tools and software to start working with the radars. For instance, working MCU firmware, ROS support and other packages for data communication.

The ROS running on the Nvidia Jetson computer records different topics. For instance, both radars topics, camera topic, GPS topic and transform coordinate frame (tf) topic. These topics are recorded as soon as ROS recording is started with a script run remotely through SSH. ROS recordings are stored as a ROS bag format file.

3.1.3 Data Encapsulation

The two radars are storing ROS messages as pointcloud1 and pointcloud2. The point cloud message holds a collection of N-dimensional (x, y, z) points and additional information about these points. For instance, point distance, velocity and intensity. The point data is stored in the message as a binary blob and its layout is described by the header content in the message. Each message has at least one header and an array of points up to over 200.

The header in the point cloud message contains information about the layout in the binary data blob. For instance, fields are describing data endianness, total number of points in the binary data blob, length in bytes in each point and other fields describing the layout of the blob. In addition to that, message has a data array for the array of points and a timestamp for each detected point.

Each point in the message has four arguments describing the layout in the point structure. The name describes the name of the argument. For instance, these argument names are x, y, z, distance, velocity, or intensity. Next argument is an offset from the start of the point structure. Third arguments describes the datatype and finally fourth argument numbers the count of how many elements are in the field. Last argument is usually one, thus only one point is in one point structure.

Serial Communication

The point cloud information is encapsulated in the AWR1843 radar module and sent over serial connection to the Jetson Nano computer. The maximum baud rate of serial connections is used which is 921600 bits/s. The radar frame rate is directly limited by the radar system capability to ship out point data.

The AWR1843Boost evaluation board [4] has another data communication interface called low-voltage differential signaling (LVDS), which has multiple times higher bandwidth to send out raw radar data. This interface is not used as it requires additional hardware to communicate with the board. Therefore, the serial connections is preferred over LVDS interface.

ROS Bags

The radar system consists of camera, single radar or two radars and possible other sensors in the future. Therefore, robot operating system is helping to manage all these data packets to be included into one file, called ROS bag. Each node in ROS can create connections where messages can be interchanged between two or more nodes. In this radar system, there is only one radar node running on the Jetson Nano computer. Camera and radars are both connected to the Linux system.

The TI provides a driver for ROS. This means, the incoming messages over serial connection via USB can be translated to ROS messages. For instance, camera might have several connections - camera info and camera compressed image feed where messages can be interchanged. Each radar in the ROS system has its own connection where detected objects are represented as a point cloud (an array of points).

3.2 Generating Radar Configuration Files

The radar configuration file defines the parameters, options, settings and preferences used in configuring the radar device. A web application developed by Texas Instruments called mmWave Demo Visualizer [15] is used to generate config files. This application has a radar file configuring interface enabling users to select between different scene selections and setup parameters.

The first section in radar configuration is a "Setup Details" interface (Figure 21. This interface has various parameters such as radar platform, SDK version, antenna config, desirable configuration, frequency band and memory address for saving or restoring calibration data. The xWR18xx seria radars are used in this project with SDK version of 3.5.

The antenna config is selected as "4Rx,2Tx(15deg)". This selection implies that four RX antennas are used for receiving chirp signals and two antennas are for transmitting. Next parameter is a desirable configuration with three different options such as best range, best range resolution and best velocity resolution. These options are provided by Texas Instruments to guide users in creating a new configuration. It will be discussed more in depth in the scene selection interface. Frequency band parameter describes two available options including 76-77 GHz frequency band for long-range radars (LRR) and 77-81 GHz frequency band for short-range radars (SRR). LRR has the benefit of higher antenna output power and SRR has the benefit of higher bandwidth. Both frequency bands are used in creating new configuration files in the following chapters. The last parameter is not utilized

in this project since configuration files are loaded into device every power cycle, thus no need for saving or restoring the file from local radar storage.

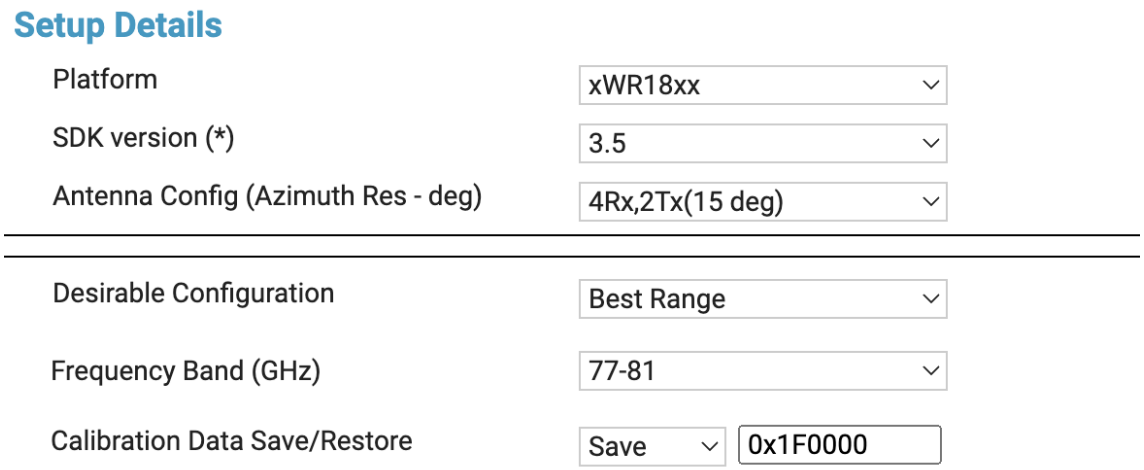

Figure 21. *mmWave Demo Visualizer Setup Details Interface*

Next section in the radar configuration interface is "Scene Selection" (Figure 22). As mentioned in the previous paragraph, the three options available are best range, best range resolution and best velocity resolution. Each scene is setting new lower and upper limits to available parameters in the scene selection interface. These limits guide users to select preferably the best parameters for the desired output. In addition to that, this ensures that user will not choose command parameters that can not run on the radar.

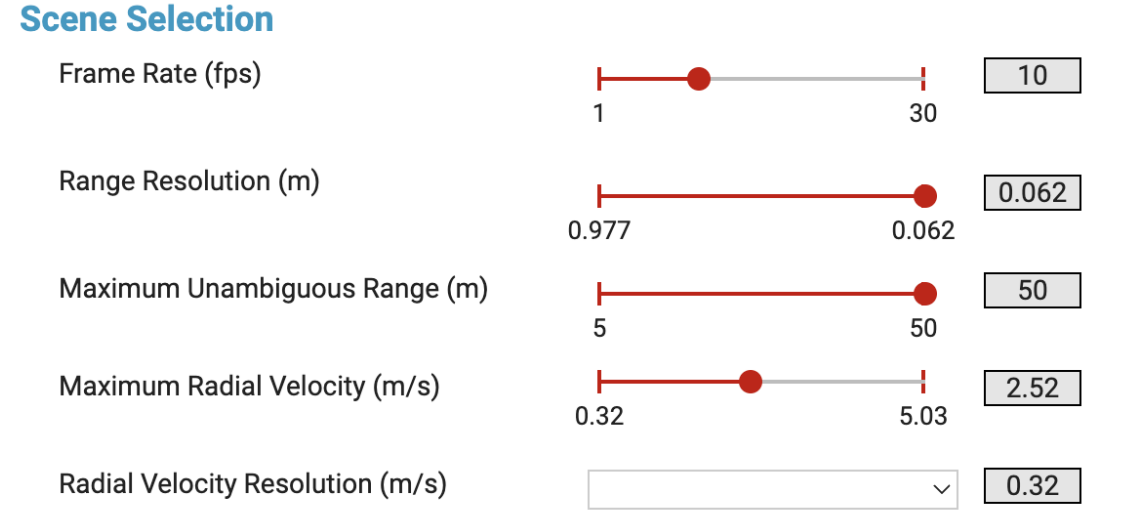

Figure 22. *mmWave Demo Visualizer Scene Selection Interface*

The parameters described in scene selection interface are frame rate, range resolution, maximum desired unambiguous range, maximum radial velocity and radial velocity resolution.

This applications makes it convenient to change one parameter and automatically other parameters available limits are modified. For instance, increasing maximum radial velocity will change the lower and upper limit for radial velocity resolution.

Frame rate is determining the data sampling speed and is limited by the serial communication bandwidth and detected cloud points. For example, long-range radar configuration has higher range resolution limits, thus fewer points are detected per targets. This should theoretically enable higher FPS. The maximum value is up to 30 FPS.

The range resolution shows how close two objects can locate and still be detected as two objects. This is directly related to the available bandwidth used in the frequency band. Therefore, short-range radars have higher range resolution than long-range radars. With this in mind, short-range radars have minimum range resolution around 4 cm with 4 GHz of bandwidth and long-range radars four times less (16 cm) since they have only 1 GHz. This value should be as small as possible for detecting smaller objects like people.

The maximum unambiguous range is determined by the desired radar cross section value of an object. For example, larger objects can be detected farther than smaller objects, thus maximum range is varying. This parameter is closely related to the selected range resolution.

The maximum radial velocity determines the possible maximum speed of an object that can be detected and its resolution shows the velocity increment of how much velocity has to change in order to be detected. These settings are desired values for detecting a certain type of target. In one scenario where traffic is monitored, then the desired maximum radial velocity should be matched to the speed of moving vehicles. In another case where moving people are recorded with radars, then the desired maximum radial velocity and resolution should be set accordingly to the speed of measuring targets.

There is one more interface called "RCS" (Figure 23) and it stands for radar cross section. While changing scene selection parameters described in the last paragraph, these numbers will automatically change in RCS interface. Desired radar cross section value field is an indication for selecting your target size. This will help to calculate the maximum distance where object is still likely to be detected.

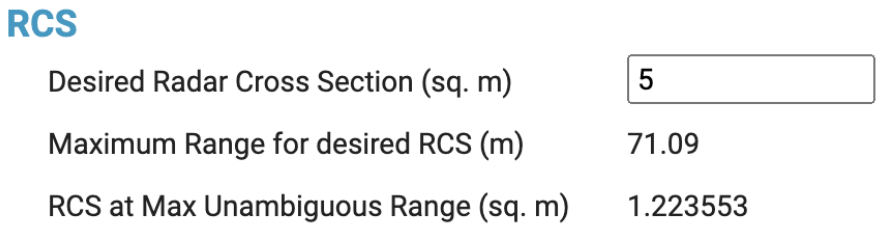

Figure 23. *mmWave Demo Visualizer RCS Interface*

The mmWave Demo Visualizer application provides some examples of object cross section values. For instance, approximate values for a truck is 100 m^2, car is 10 m^2, motorcycle is 3.2 m^2, an adult is 1 m^2 and a child is 0.5 m^2.

In this thesis people and other smaller objects are targeted, thus the desired radar cross section value is around 1 m^2. Based on the parameter selection in the scene selection interface, "maximum range for desired RCS (m)" and "RCS at max unambiguous range" value will change accordingly. For example, the values illustrated in Figure 23 imply that the desired RCS is 5 m^2 and currently the selected parameters in the scene selection interface indicate that that 1.22 m^2 object is likely to be detected at the maximum distance of 71 meters. Since the calculated RCS is only 1.22 m^2 at 71 meters, the 5 m^2 RCS object is likely to be detected even farther.

4. Radar Software Development

The main motivation for developing a custom application is the lack of available software that meets the requirements of this project. The radar system is using a robot operating system (ROS) to record information from Raspberry PI camera module, Texas Instruments millimeter wave radars, and other peripherals connected to the radar system. The recorded data is stored in a ROS bag. The custom application will be used to parse ROS bag files and utilize the data in an interactive heat map and image player. One of the main aspect is the platform independence, thus web platform is chosen.

4.1 Requirements

The main objectives of this software development project are ROS bag file parsing and the data utilization in the web application for further analysis. The initial features are listed below.

- Display recorded Raspberry PI camera images
- Display two radar cloud data as a heat map
- Interactive controls for radar data analysis
- Setting frame delay when playing radar point cloud data
- Show statistics about the recorded radar data
- Generate a heat map with multiple radar point clouds

All camera images and radar cloud data points are displayed in the web application. These images and radar data points can be manipulated with various controls (buttons, slider and other widgets). One of the key functions is the ability to play radar point cloud data with a frame delay. This suggests that configurable frame delay can be chosen by the user. The frame delay is delaying the time when radar point cloud data disappears in the heat map. For instance, frame delay of 10 will leave the radar data in the heat map for the next 10 frames. Initially, there will be overall statistics displayed about the radar data in the ROS bag. For example, the total amount of points and average amount of radar points per frame. Lastly, all radar points' intensity will be used to draw a heat map. In case of multiple points in the same drawn box are detected, the average intensity is calculated.

4.2 Existing Solutions for Radar Data Visualization

The visualization and the interactivity with the data plays a key role in generating new ideas with millimeter wave radar technology and changes the approach for problem solving. This is why, there are certain requirements for a visualization software. Visualization is important for analysing the results and moving forward with the research.

4.2.1 mmWave Demo Visualizer

The visualization software provided by Texas Instruments is called mmWave Demo Visualizer [16]. This application is meant to be used in conjunction with the mmWave SDK demo running on the TI Evaluation module for mmWave devices.

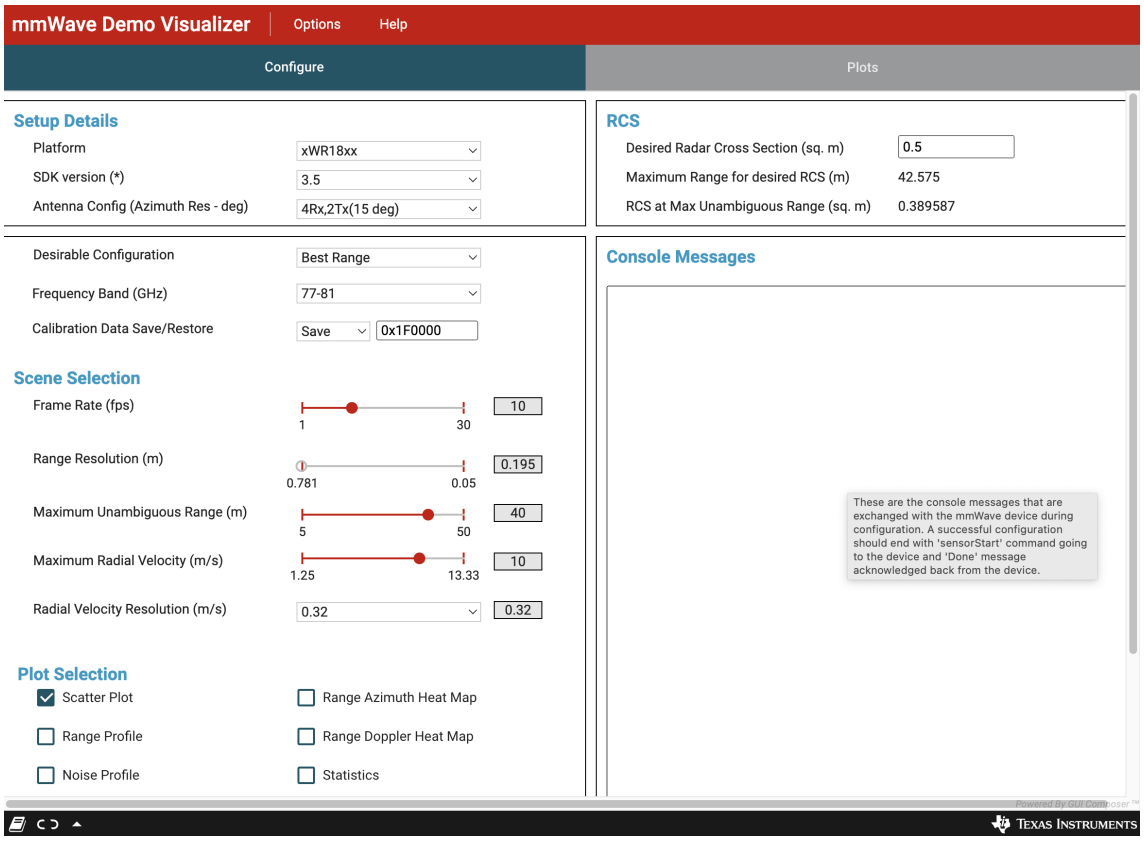

Figure 24. *mmWave Demo Visualizer User Interface*

This application has radar configuration settings panel and plots panel (Figure 24). The configuration settings panel can be used to modify the radar configuration and download the configuration to the device. The plot panel can be used to display live data from the device that is connected to the PC. There are numerous plots available for radar data visualization. These features are good to get the radar working and see results instantaneously on the

screen. However, this application has some major limitations working with our radar system as described below.

- Only one radar can be connected in this application
- It shows only real-time data
- Does not support reading ROS bag files

For instance, the major drawback is the lack of support for more than one radar. Our radar system consists of one or more radars, thus this software does not suffice. Another disadvantage of this system is the support for robot operating system file format. Our radar system is using ROS to connect all the sensors including one or more radars. Therefore, we cannot display any recorded data on the screen as a scatter or range plots.

4.2.2 ROS Visualization Tool RVIZ

The Rviz application is short for ROS visualization tool. It can visualize radar point cloud data in 3-dimensional space. It is meant to visualize what a robot is seeing and doing. In addition to that, the main purpose of this application is to visualize the state of a ROS based system.

Moreover, it allows the application users to view the simulated radar model with all the sensor information contained in the ROS bag file. For instance, Rviz displays 3D sensor data from cameras, lasers, and other 3D devices in the form of point clouds and depth images. More, 2D range finders can be viewed in the Rviz application as images data.

There are numerous absent features in this application for the radar system described in this thesis. First, there is no heat map generation without creating new plugins. Secondly, no support for delaying the radar data points and easily manipulating the recorded data. Lastly, it is inconvenient to use Rviz as it supported only on Linux platform.

4.2.3 Foxglove Studio

Foxglove Studio (Figure 25) is an open source visualization software . This application is downloadable for numerous platforms and the user interface is intuitive. The panel layouts can be arranged with ease and interactive visualizations help to understand the data in ROS bags.

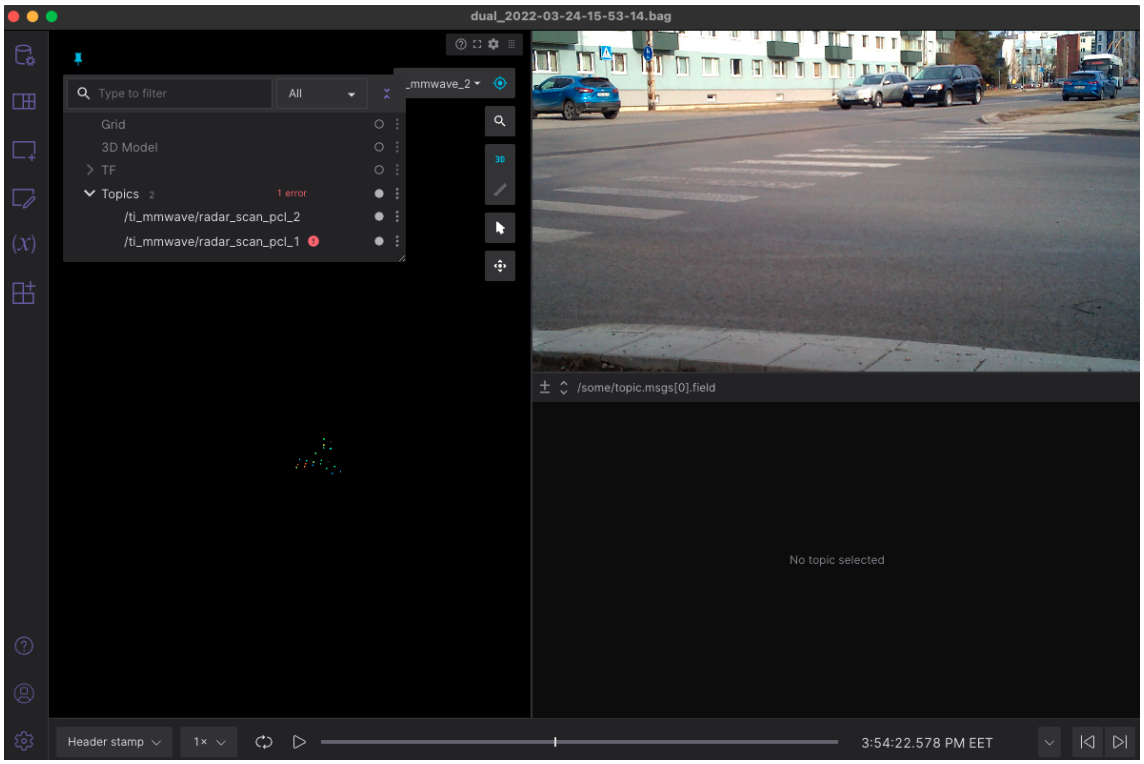

Figure 25. *Foxglove Application User Interface*

As with Rviz, this applications has two main lacking features. Firstly, there is no support for heat map generation. Lastly, the delayed radar data frames feature is missing.

4.3 Flutter Web Application

The custom data visualization software main purpose is to display instructive information from a ROS bag file format. The file contains interlaced and serialized ROS messages dumped directly to a single file as they come in from subscribed topics. This is one of the most performance and disk-friendly recording format possible to store messages. The ROS bag is subscribed to multiple topics. For instance, camera info, camera feed, radar point cloud data, and some other topics. These topics are used in the application to display a heat map, camera feed and other features.

4.3.1 Web Interface Overview

The Flutter web browser application user interface is illustrated in Figure 26. All interfaces, layouts, and functions are described in the next section.

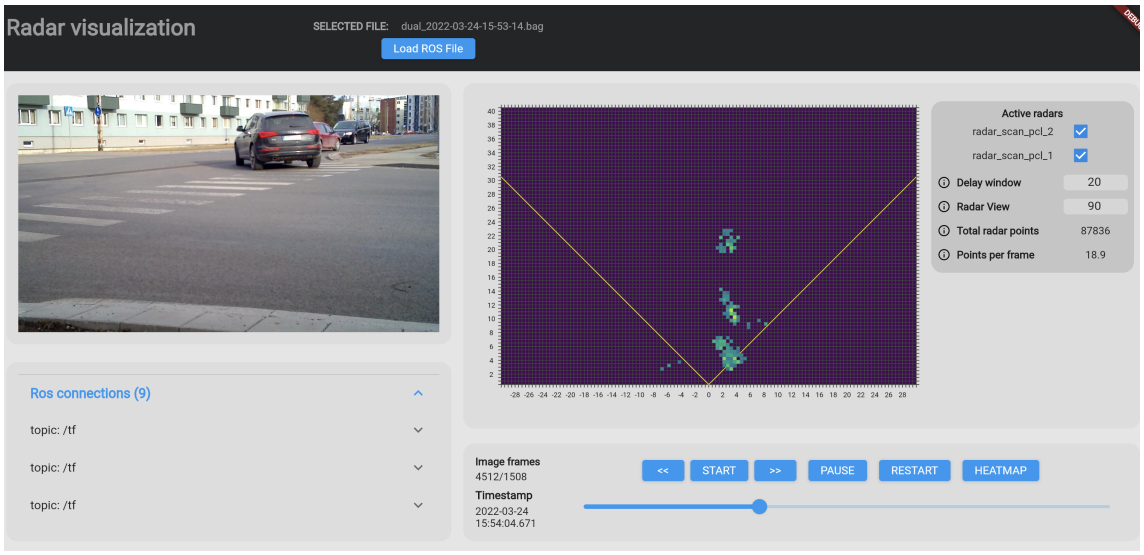

Figure 26. *Flutter Web Application User Interface*

The typical use of this application is illustrated in Figure 27. It begins by loading a ROS bag file from the local machine. Next, all the connections recorded in the bag file have topic names. Each connection has an identifier and this number can change, thus user is asked to select correct connection topic names for radar point cloud data and image as shown in Figure 35.

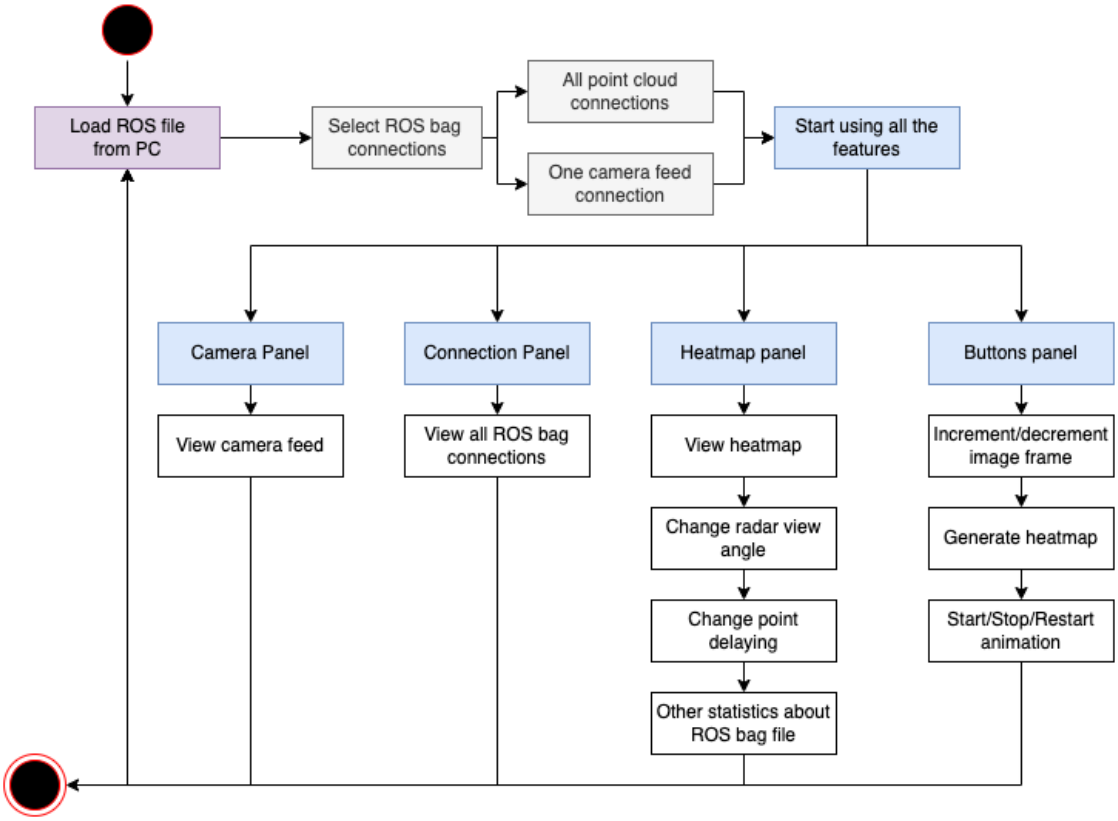

Figure 27. *Application Usage Flow Chart*

Moreover, all bag connections are displayed in ROS connection panel with the additional information as shown in Figure 33. The information about radar data cloud points and image files are used based on the connection identifiers user selects. The one or multiple radar connections are used to display visualization on a heat map graph. The image connection is used to fetch image byte data indexed by the ROS bag file structure. This data is displayed in the camera panel as shown in Figure 28 highlighted with red box number one.

4.3.2 Panel Layout and Functions

The Flutter web application composes of numerous panels as highlighted with red boxes in Figure 28. These panels are described more in depth below.

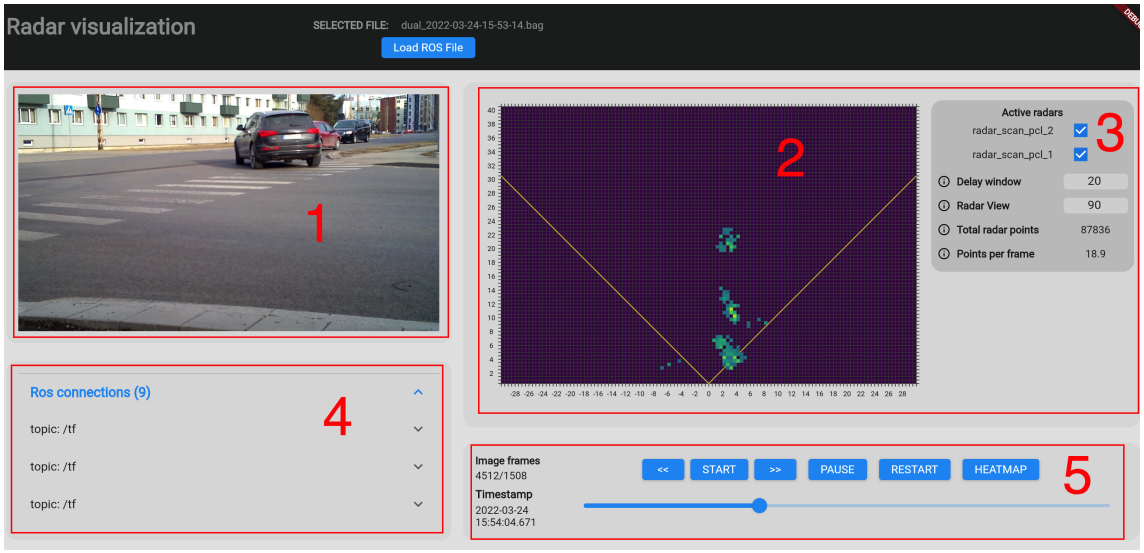

Figure 28. *Flutter Web Application Interface with Highlighted Panel Numbers*

Camera Panel

Panel 1. The camera panel interface shows the camera feed. This animation of images is not a video player. In order to make this interactive with the buttons panel, it uses sequence of images from the ROS bag. These sequence of images can be played forwards and backwards with any speed using the slider widget in buttons panel.

Heat Map Panel

Panel 2. The heat map panel is interfacing one or multiple radar point cloud data as a coloured heat map (Figure 29). This is a dynamic heat map changing data according to the timestamp value. The heat map graph x-axis and y-axis labels are both in meters using 0.5 meter increments. One data point is represented as 0.5x0.5m box on the graph. For now,

the graph size can only be resized programmatically.

The radar cloud points have several parameters. For instance, point position in x and y dimension (m), range (m), velocity (m/s) and signal intensity. The x and y dimension coordinates can be used to draw the map of objects and the signal intensity of the object can be used to color code that position on the graph. Thus, heat map can be drawn using x, y and intensity parameters to visualize the radar point cloud data. The velocity and range parameters are not utilized at the moment.

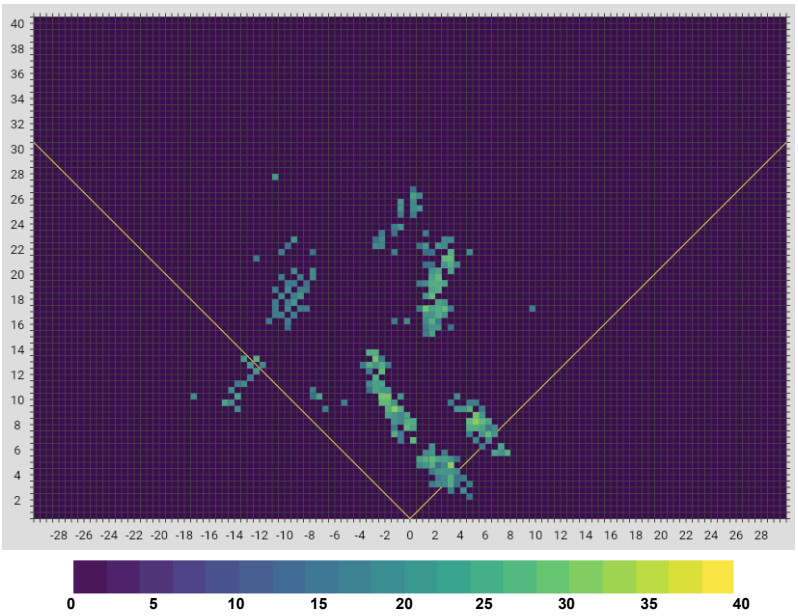

Figure 29. *Heat Map Displaying Radar Point Cloud Data*

Panel 3. The information interface in heat map panel as shown in Figure 30 is used to manipulate the heat map data and show statistics about the ROS file. Active radar section enables to select which radars are currently showing radar point data on the heat map graph. By default, all radars are selected. For instance, each radar can be enabled or disabled separately, in case of comparing results or analysing detected point movements. This will result in only showing selected radar connections.

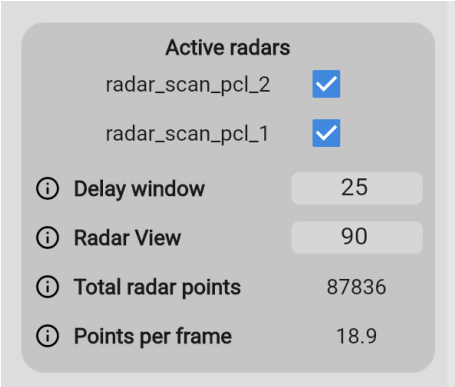

Figure 30. *Heat Map Information Interface*

Next parameter is a delay window. The frame delay feature enables to delay the point disappearance and the value indicates for how many radars frames. For instance, if delay window is set to 25, then all the radar data points are displayed for the next 25 frames. The frame delay for two objects is illustrated in Figure 31. Two cars are detected in the figure and highlighted for illustration purposes as red and green rectangles. Three different frame delays show how data points increases using higher frame delay. This shows where the object came from and helps visualizing and understanding the radar point cloud data. The frame delay value should depend on the size of an object you are analysing. Of course, other values can be used to experiment with this feature.

The radar view is for setting the radar maximum view. For instance, value set to 90 implies that radar view is 90 degrees wide. This value is only for illustrative purposes, thus it only visualizes the radar view in the heat map graph. The total radar points shows how many radar points were detected in the bag file for all the radars. The last parameter describes how many points are detected per each radar frame on average.

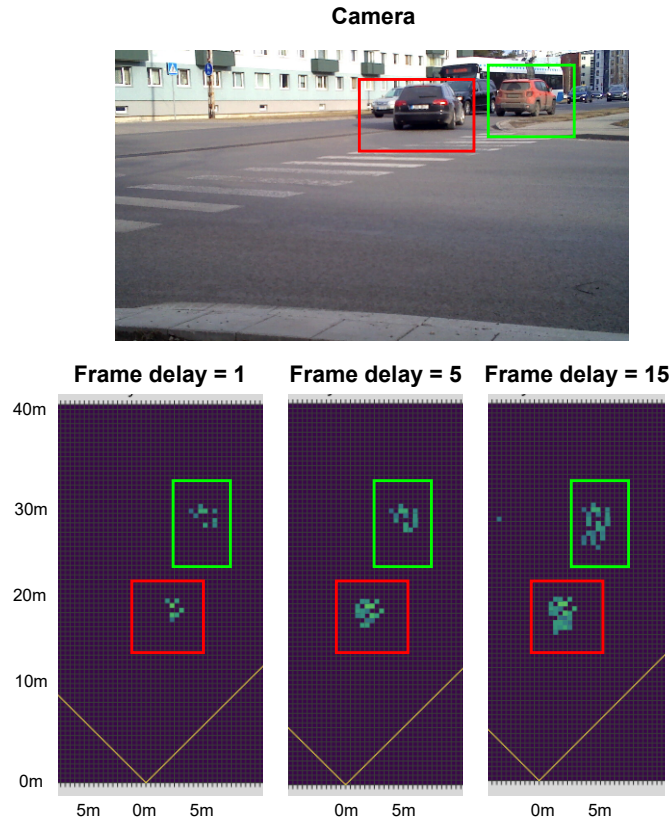

Figure 31. *Dual Radar Frame Delay for 1, 5 and 15 frames*

ROS Connection Panel

Panel 4. The connection panel is an interface for displaying ROS connections and the detailed data with every connection. The list of ROS bag file connections is created each time file is loaded in to the application. Each connection is in a collapsing and expanding widget. The collapsed widget is shown in Figure 32. The Expandable ROS connection widget is illustrated in Figure 33.

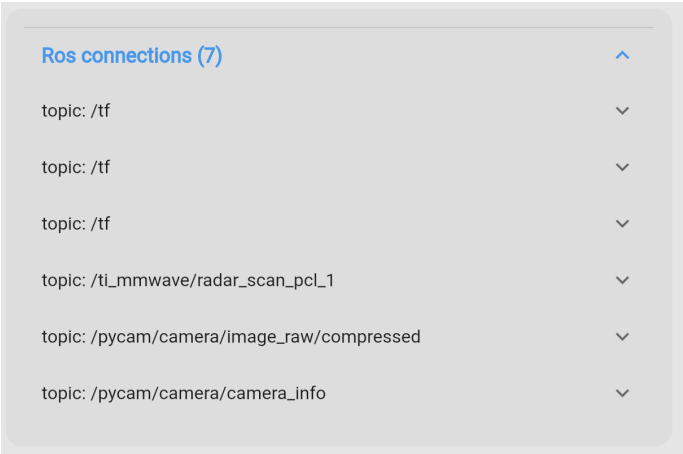

Figure 32. *ROS Collapsed Connection Panel*

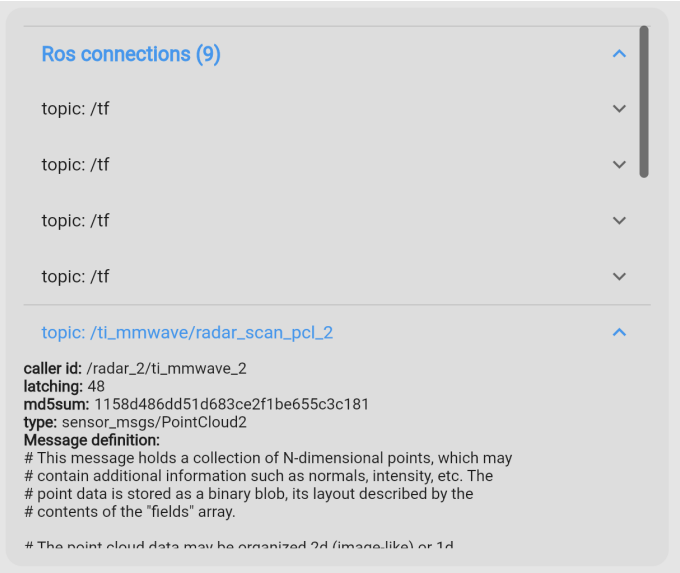

Figure 33. *ROS Expanded Connection Panel*

Buttons Panel

Panel 5. The buttons panel interface has multiple buttons and a slider widget as illustrated in Figure 34. The buttons are for manipulating the camera feed and point cloud data in the heat map graph. The heat map button generates heat map from all the currently selected active radar point cloud data. The slider widget can be used to skim through the ROS bag file by sliding it forwards or backwards. More, the slider widget is an convenient way to skip to a certain time in bag file. The animation of both camera feed and the heat map point data can be started using "START" button. More, "PAUSE" and "RESTART" button can be used to pause and restart the animation respectively. Double arrow buttons can be used to increment and decrement image frames in either direction.

Figure 34. *Buttons Panel*

Radar and Image Connection Panel

Panel 6. There is one more panel in this application called radar and image connection panel as shown in Figure 35. It is a simple interface as it consists of two identical lists of connection topics. One list is for selecting radar connections and the other list is for selecting an image connection.

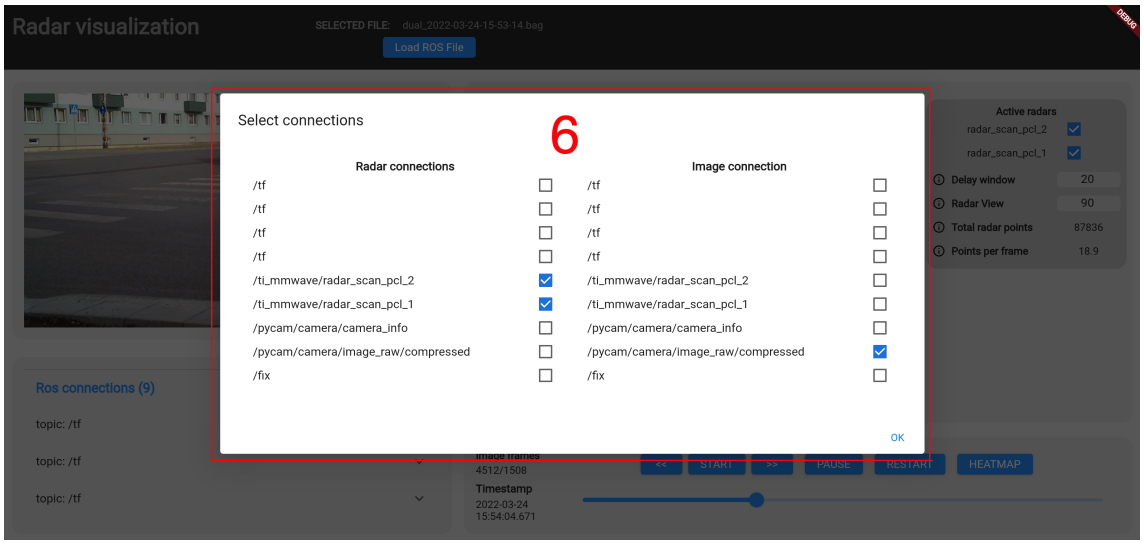

Figure 35. *Flutter Application Interface with Connection Panel Highlighted*

4.4 Flutter Software Design

The software development begins with the project system design. The web platform requires back-end and front-end development. There are plethora of programming languages and web frameworks available in the web. The selection of Python, Dart and Flutter was mainly based on the author's proficiency in these languages and in the web framework. In addition to that, Flutter and Dart are both developed by Google and have a thorough documentation with active community. This is helpful since code troubleshooting is inevitable.

Initially, the web platform application is divided into two projects. The front-end development includes using the web framework to build the layout and all the widgets on the screen and Dart is used to control them. More, Dart manages background services and data handling. The back-end project is running on Python. It was initially selected as it is easy to use for prototyping. In addition, Python has publicly listed libraries for ROS bag parsing (Python library called rospy [17]) and visualizing radar cloud data with matplotlib library [18].

Later in development, Python project was abandoned and Dart is used to parse ROS bag files and process the data visualization.

4.4.1 Initial Software Design with Flutter and Python

The project initial design featured two projects to fulfill the requirements stated in the requirements section. The first project, Flutter web application with Dart is used to display all the widgets and data on the screen. The second project, Python server is used mainly for ROS bag parsing and image and radar data processing.

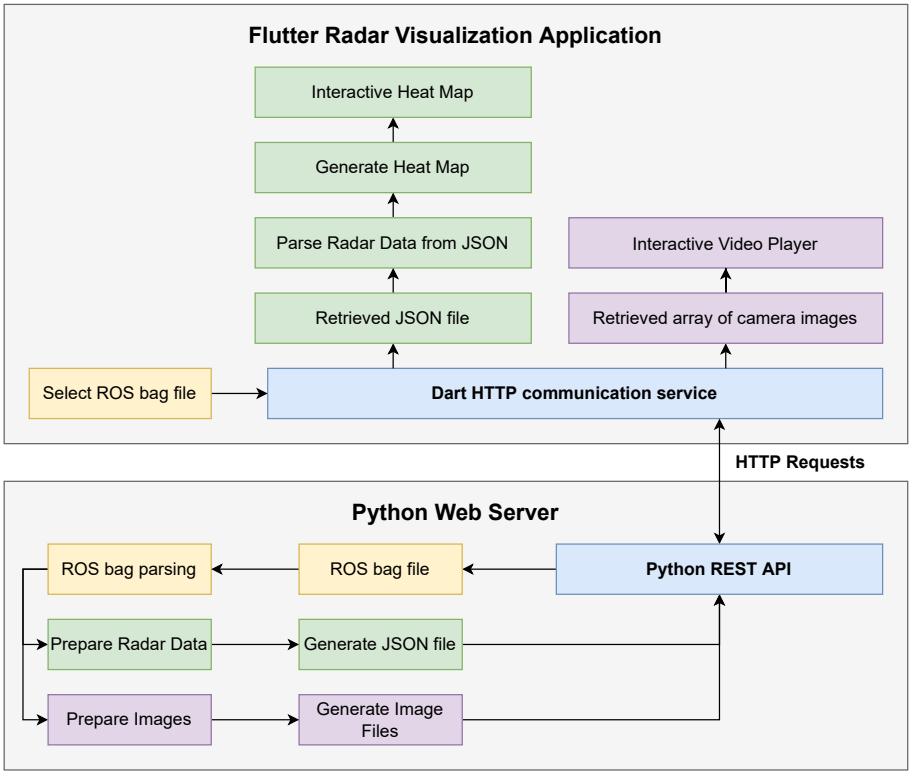

Figure 36. *Initial Software Application Design with Flutter, Dart and Python*

The Flutter web application is responsible for displaying the camera images and generated heat map images along with numerous button widgets to control the data flow. The Python Server application is tasked to run a server and parse ROS bag files into a JSON file and image files. The Flutter and Python applications are communicating via HTTP requests. The system overview of initial project is shown in Figure 36.

Flutter Web Application

The main aim of the web application is to display and interact with the ROS bag data. The web interface has a button for selecting ROS bag files from the computer. This bag file is then sent to the Python server for further data processing. Retrieved files from the server are then utilized. The JSON file has information about the image indexes and radar point cloud data as shown in the Figure 37. The image files are animated as a sequence of

images. The radar data from the JSON file is processed and heat map graphs are generated and animated along with the image files. The radar and image animation can be started, stopped, paused and restarted. In addition to that, a slider widget can be used to skim through the image frames. More features are added in the final system design described later in this chapter.

Python Server Overview

The Python Server application main task is parsing both radar point cloud data and image data from the ROS bag file. In addition to that, provide REST API functionality for Flutter to fetch and send information between two projects.

The ROS bag file is retrieved from the Flutter application. As previously mentioned, Python has a library called rospy [17]. This library is a client library for a ROS and enables Python programmers to quickly interface with ROS topics, services and parameters. The design of rospy library favors implementation speed over runtime performance. This has both advantages and disadvantages. The advantage is prototyping quickly algorithms and testing them with ROS bag files. At first getting algorithm and prototype running is crucial to proceed with the development of the custom web application. Therefore, Python was favored at first to get the understanding and concept working fast. It is also ideal for non-critical-path code, such as configuration and initialization code. The matplotlib [18] was initially used at the beginning to visualize the parsed radar point data in a graph. Later, visualization process was passed onto Flutter and Dart.

Python code was initially used to loop over all relevant connections and try to understand how to parse all various topics in the bag file structure. The ROS bag parsing is explained in depth later in this chapter. In the code, camera topic included messages about the camera settings and series of images as image byte arrays. The radar topic included messages about point cloud data. All this data was organized by timestamps included in the connection messages.

After processing radar and image data, the radar data with image file indexes were encoded into JSON file as shown in the Figure 37. This JSON file is used by the front-end project to generate heat maps and play image files as a video. The encoded file includes total camera and point cloud data frames in the ROS bag file. The camera frames per seconds were used to determine the correct video playing speed. The "frames" object in the JSON file is a list of all image frames. Each image frame has a frame count "frame", this is indicating the frame number. The frame number is used to determine which image file is playing. The start and end frame time is in Unix timestamp in seconds. The "pcl" object is an array of

radar data points. Each point has a timestamp, x and y coordinate, range, velocity and the intensity of the signal.

```JSON
{
  "bag_name": "2022-01-26-12-01-20.bag",
  "cam_frames": 35,
  "pcl_frames": 25,
  "cam_fps": 15,
  "pcl_fps": 10,
  "frames": [
  {
      "frame_start": 16431912898.2,
      "frame_end": 16431912898.3,
      "frame": 0,
      "pcl": [
        {
          "timestamp": 16431912898.23
          "x": 0.10,
          "y": 0.20,
          "range": 0.12,
          "velocity": 2.0,
          "intensity": 5.12
        },
        ...
      ]
  },
  ...
  ]
}
```

Figure 37. *Radar Data in JSON Format*

Abandoning the Python Project

The first task was to organize camera images and radar data. The radar data was constructed into JSON file format and array of image bytes was decoded into images. Initially, a folder of images and a JSON file containing radar cloud point data was manually copied to Flutter and Dart project, where files were read in and used. In order to automatize this process, Python code had to be running in a simple HTTP server or using some other service to fetch files. In addition to that, a representational state transfer (REST) application programming interface (API) had to be designed and programmed to download, upload and transfer files between two programs. The image files are large and transferring thousands of image files was not efficient. In addition to that, ROS bag file parsing and data restructuring was very slow in Python. This is why the ROS bag parsing task was switched to Dart programming language.

Python is considerably slower than Dart and not using communication protocols and API between multiple projects will increase the speed and performance of this web application. Dart was already used with the Flutter project and this meant all code base is now in one project. No more multiple projects and the communication between them. Dart was not considered at first as it does not have a support for ROS bag parsing. There was no publicly listed library for ROS bag parsing. Python was excellent at introducing the concept of parsing ROS bags, although Dart is much better choice in the end.

4.4.2 Final Software Design with Flutter

The main change was abandoning the Python project in the final design of the system. All required features are now implemented in Dart. This resulted having only one code base for the entire project, thus reducing complexity of this project. The Python tasks were now switched onto Dart. The final system design is illustrated in Figure 38.

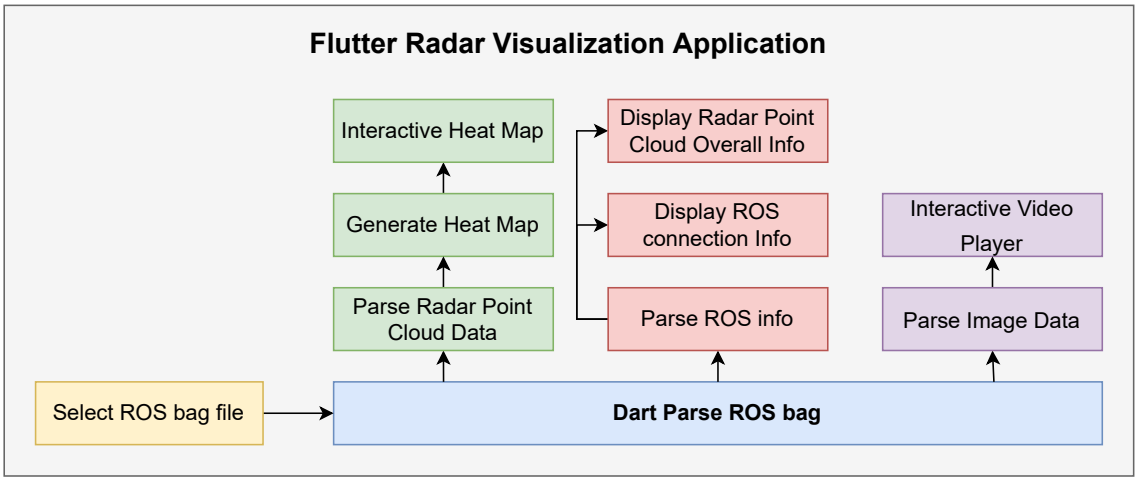

Figure 38. *The Final Flutter Software Design*

As mentioned previously, Dart does not have a publicly listed library for parsing ROS bag files. This created new software tasks to fulfill the requirements in the beginning of this chapter. The inner workings of ROS bag format had to examined, interpreted and comprehended. The mechanics of ROS bag format is explained later in this chapter. Dart has several advantages over Python. For instance, the capability to parse ROS bag files faster and the fact that Dart is already used in this project.

As shown in the Figure 38, the JSON file has been deleted. The communication between the methods and data structures are now all residing in one code base. All other ROS bag data processing is implemented using Dart. In depth implementation of all the features are described in the next section.

4.5 Implementation

The software tasks in this project are set in the beginning of this chapter. First, the ROS bag format section explains the inner workings of the bag file structure. Next, the implementation of parsing ROS bag using Dart is explained in depth. After that, all other main features of this projects are explained. For instance, heat map and image animation, heat map generation, and the ROS bag connection info.

4.5.1 ROS Bag Format

As mentioned previously, there is no publicly available ROS bag parsing library for Dart programming language. There are many benefits using Dart as parsing ROS bag files and one severe disadvantage in this custom visualization application. The numerous advantages of using Dart opposed to Python are mainly performance and development time. Firstly, all code is based in one Flutter and Dart project, thus no need for designing a Python web server with REST API. Secondly, the time for developing the data communication between multiple projects with HTTP requests can be now used for ROS bag parser in Dart. Finally, the speed and performance of the application is vastly improved. The one downside is the lack of support for ROS bag parsing.

Bag File Structure

The ROS bag parsing process begins by reading in the bag file format as a list of bytes in Dart. This project uses the latest ROS bag file format 2.0. The bag file is structured as a sequence of records, with a first line to indicate the format version number. The described sequence of ROS records are shown in the Figure 39. There are numerous types of records such as bag header, chunk, connection, message data, index data and chunk info. The bag header record is always the first record in the file. Next, there are many chunk records and each chunk is followed by multiple index records. These index records contain information about the preceding chunk. After chunk and index records, there are connection records. Each connection record describes one connection with certain topic. For instance, camera and radar data are on two different connections. Finally, the end of the bag file is composed of chunk info records.

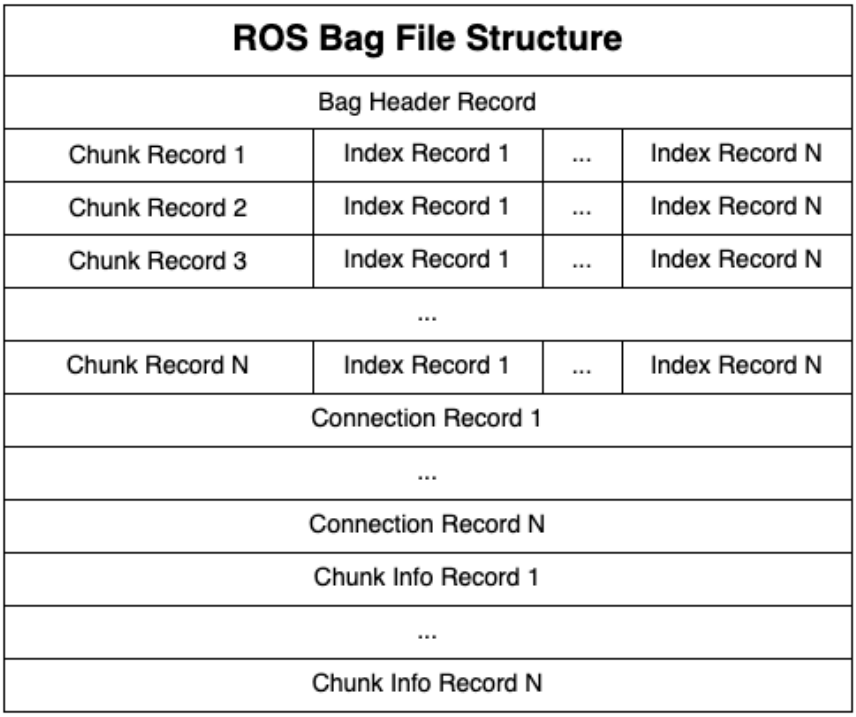

Figure 39. *ROS File Structure*

Bag records

All records have the same defined structure as shown in the Figure 40. The total length of the record is defined as the sum of the size of the header (four bytes), header data size, the size of the data (four bytes) and data section size.

Record			
Header Size	**Header Data**	**Data Size**	**Data Section**
4-byte little-endian integer = X	X amount of bytes	4-byte little-endian integer = Y	Y amount of bytes

Figure 40. *ROS Bag Record*

Each record has a header data section. The header data section consists of header fields and each field consists of field size and data as shown in Figure 41. The header data is populated with header fields up to N amount. The header data size is determined in the four bytes of header size. Each field in header data consists of four bytes indicating the length of the field and then a name and value pair. The field name is always a string and field value can be either a string or an integer.

Header Data				
Header Field 1	Header Field 2	Header Field 3	...	Header Field N

Header Field			
Field Size	Field Data		
4-byte little-endian integer = X	Field Name (String)	"=" character	Field value (String, Integer)

Figure 41. *ROS Bag Record Header*

The data section of each record is dependent to the type of the record. For instance, the bag header record data section is padded out by filling data with ASCII space characters (0x20). This empty space can be used for additional information about the bag file. Another example of record data section is chunk record. In this case, the data section is filled with message data and connection records as shown in the Figure 42.

Chunk Data Section					
Message Data Record 1	...	Message Data Record N	Connection Record 1	...	Connection Record N

Figure 42. *ROS Chunk Record*

4.5.2 ROS Bag Parsing with Dart

The bag file is accessed and processed twice by Dart. The first time, all record headers are parsed through and connection headers are processed. This processing is not time consuming as header sections are relatively small binary blobs and each record has a defined length enabling to skim through the file fast.

The next step is using user interaction to determine which connection identifiers are radars using and which one is camera using. This process is achieved through the radar and image connection panel. Now connection identifiers can be used to parse image and radar messages.

The second time, all record headers in the bag file are parsed through again. The chunk and chunk info records are processed. Images and radar point cloud data are processed differently as image files are very large, as opposed to radar point cloud data. The connection identifiers determined in the radar and image connection panel are used to determine the message data records in the chunk records. The image message data is processed and the byte offset to the image data and its length in the bag file along with the

timestamp is stored in a class based model in Dart. This is why the image files are too large to be copied around in the memory efficiently, thus making the application unresponsive for a very long time. Therefore, when image is accessed in the application the byte offset and image length is used to display image in the bag file. The radar point cloud data is multiple times smaller binary blob than image, thus radar data is processed right away. After second processing of the ROS bag file, there will be one list of indexed images and one list of radar point cloud data ready to be used in the Flutter web application.

4.5.3 Image and Heat Map Animation

The image and heat map animation is driven by a timer in Dart code. The interval is calculated using the subtraction of end and start timestamp of image frames. Then, the subtracted amount is divided by all the image frames parsed in the ROS bag. This results an average interval time for playing the image files on the screen. This eliminates the need for user entering the correct frame rate.

The radar frames used for drawing heat maps are retrieved from the radar usually with a much smaller frame rate than the image frames. For instance, camera has a frame rate of 30 and radar has only 10 frames per second. Therefore, these radar data points must be rendered on the heat map with a certain frame delay. Otherwise, the data point is rendered only for one image frame. This is resulting in showing the data point for a single image frame time.

The delay window is using a mechanism to render radar points on the heat map for a longer period of time. For instance, there is an editable text field in the heat map panel for how many frames each radar point should be delayed on the heat map. This effect creates better visualization of radar data.

The performance of drawing heat map on the screen is increased by splitting the drawing process into two components. Initially, the heat map background is painted and only foreground is updated by new heat map radar points.

4.5.4 Heat Map Generation

The heat map generation begins by parsing ROS bag file for radar point cloud data. The heat map grid is represented as 0.5x0.5 meter boxes and each box has a color directly related to the signal intensity in radar point data. At first, all heat map grid boxes have signal intensity value of zero, thus they are purple. As the signal intensity increases, the

colors go from purple to green and then more yellow. In case of two data points residing in one 0.5x0.5 meter box, the intensities are averaged and the averaged value is used for coloring the box.

A list of heat map grid boxes are created when radar data is parsed. Each heat map box has information about the location in the heat map grid and the intensity value for the color. The Figure 43 shows heat map of 4 cars in the radar field of view.

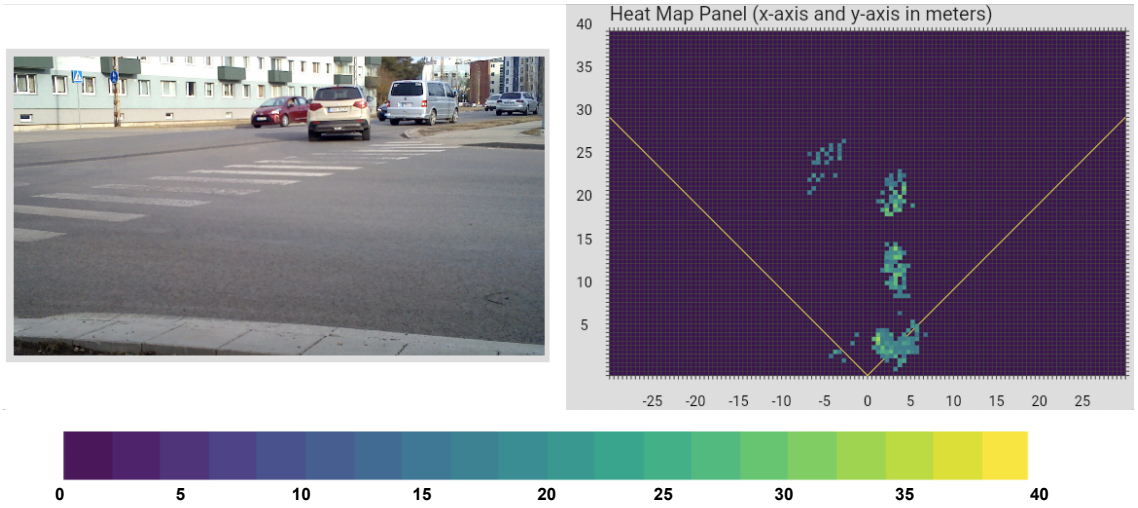

Figure 43. *Heat Map Generated with a Camera Feed*

5. Experimental Results

The data collection device is used to conduct experiments with three different walking patterns which are showing vertical, horizontal and diagonal movement in front of the radar. One person walks through all these movements with five different radar configurations. The results are achieved using three long-range and two short-range radar configurations and they are compared and analysed using the three different walking patterns. The radar configuration files are in project Git repository [19].

5.1 Radar Configuration Files

The radar configuration files are generated using mmWave Demo Visualizer software [15]. This application can be used to generate, modify and send the configuration files to the radar devices. Each line in the file describes a command with parameters. There are around 30 different commands including ADC, chirp profile, frame, power, data output mode, range, velocity, and many other configurable commands.

The radar system module (AWR1843BOOST Evaluation Module [4]) has support for two frequency bands such as long-range (76-77GHz) and short-range (77-81GHz). Both frequency bands are used to generate various configuration files.

5.1.1 Long-Range Configurations

Long-range radar configurations benefit from the higher antenna output power, thus objects are detected at longer distances. The downside is the frequency bandwidth of only 1GHz. This affects directly range resolution. Therefore, smaller objects are producing fewer data points.

People are considered small targets in comparison to vehicles in traffic and for this reason they produce fewer data points. There are some important factors when generating long-range configuration files. For instance, the range resolution should be set as close to the minimum as possible which is around 20cm for long-range radars. Maximum velocity is set around two to three meters per second (maximum person's walking or jogging speed). This enables setting lower velocity resolution as higher velocity increases velocity

resolution and vice versa. The data communication speed of the radar is configured from 10 to 15 FPS depending on the configuration file sent to the radar.

mmWave Demo Visualizer provides multiple scene selections such as best range, best range resolution and best velocity resolution. These scene selections set lower and upper limits to selecting different parameters. Numerous multiple long-range radar configuration files were generated and all did not work on the radars. Some configuration files were unable to be sent to the radar since it did respond to certain commands and the upload failed. These configuration files are not discussed in this thesis. Another configuration files were successfully sent to radar devices and produced false target reading. This was probably due to some reflections specifically affecting certain config files. Finally, a few files were successfully reading data points from detected targets and these files are discussed next.

Three long-range radar configuration files are used in result analysis and comparison. Two long-range radars are identical only differing on the FPS parameter (10 and 15 FPS). These configurations are described in Table 1.

Table 1. *Long-Range Radar Configurations*

Radar Parameter	LR_people_conf_1	LR_people_conf_4	LR_people_conf_4
Frequency Band	76-77GHz	76-77GHz	76-77GHz
FPS	10	10	15
Max Range	30m	45m	45m
Range Resolution	0.196m	0.2m	0.2m
Max Velocity	3.07m/s	3m/s	3m/s
Velocity Resolution	0.1m/s	0.2m/s	0.2m/s

5.1.2 Short-Range Configurations

As stated before, short-range radars are operating between 77 and 81 GHz frequency. The short-range radars benefit from the higher bandwidth of 4 GHz. This allows much smaller range resolution, thus more data points per target are detected.

Short-range radar configuration is focusing high range and low range resolution. Of course, velocity should be set to match a walking people behavior. This means the maximum velocity up to 3m/s or more and velocity resolution lower than 0.5m/s. As mentioned previously, lower range resolution allows more data points per target, thus short-range radars are great at detecting people.

Table 2. *Short-Range Radar Configurations*

Radar Parameter	**SR_people_conf_4**	**crit_infra_conf**
Frequency Band	77-81GHz	77-81GHz
FPS	10	15
Max Range	45m	30m
Range Resolution	0.056m	0.586m
Max Velocity	2.52m/s	23.05m/s
Velocity Resolution	0.32m/s	0.37m/s

As with long-range radar configurations, numerous short-range configurations were generated and not all did work. Only one working short-range radar configuration *SR_people_conf_4* (parameters shown in Table 2) is used for data comparison and analysis. Another short-range radar configuration was generated by TalTech Embedded AI Research Lab and is called critical infrastructure configuration. File parameters are shown in Table 2.

5.2 Experiments with Data Collection Device

The experiments are conducted using three walking patterns. These walking patterns are recorded by data collection device and analysed with the custom web application developed with Flutter. Three heat map graphs are generated for each radar and one with both radars combined. The graphs are showing all detected data points recorded and the signal intensity value is used to color code each 0.5x0.5m box on the graph. Brighter colors are used to indicate higher signal intensity value and dull colors are used to indicate lower signal intensity. If multiple data points are recorded in the same area on the graph, then average intensity is taken from all the points detected in that area. For instance, if one radar detects an object with high intensity, thus coloring box yellow and another radar detects the same object with very low signal intensity, thus coloring box blueish, then the average intensity is calculated and the final color in the combined radar graph is greenish.

Each pattern is recorded with all the previously mentioned radar configuration files. This allows comparing same walking pattern with other configuration files and determining each cons and pros. Both radars are always configured with the same configuration file. Files are recorded using ROS bag format where they are later analysed with Flutter web application. In application, heat map graphs are generated and all captured data is displayed on the three graphs for comparison. Two heat map graphs are illustrating both radars separately and third one is combining both data by averaging the signal intensities.

5.2.1 Walking Patterns

An experiment is conducted with data collection device. Each radar in the system will produce a point cloud and it is combined in the radar data visualization software. In order to measure or analyse how two radar system improves the person detection, three walking patterns are chosen for this experiment as shown in Figure 44. In addition to that, each walking pattern is tested with multiple short-range and long-range radar configuration files to determine the best results. These pattern results are later used for data analysis and comparison.

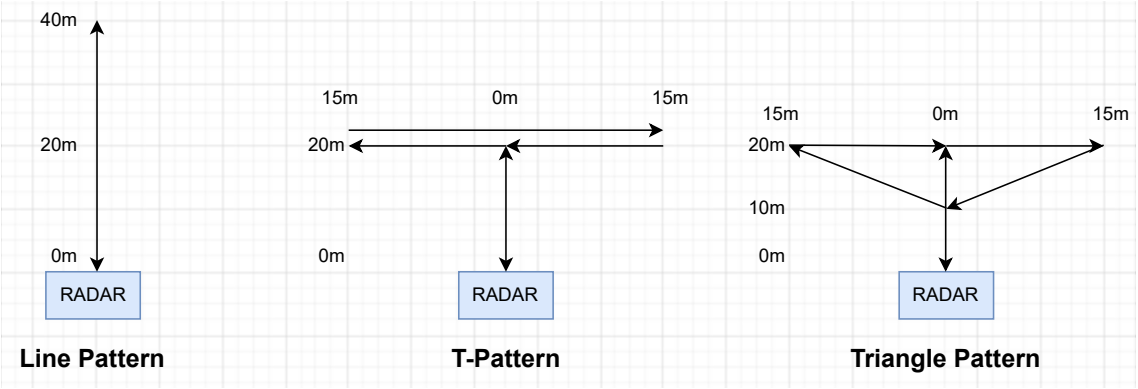

Figure 44. *Walking Patterns for Radar Experiments*

In Figure 44, there are three different types of patterns such as Line Pattern, T-Pattern, and Triangle Pattern. Each pattern is illustrating a person walking routine. The line pattern is a vertical movement and is used to test the maximum range for detecting a walking person. The vertical movement means walking away from the radar and then walking back to the radar. The signal in front of the radar is the strongest and moving away from the center weakens the reflected signals from the targets. Therefore, the T letter shaped pattern and triangle pattern are testing the radar ability to detect a horizontal and a diagonal movement.

5.2.2 Visual Evaluation for Experimental Results

There are total of three evaluation parameters that are used to analyze five different radar configurations as described below. This section describes the visual evaluation and other evaluation parameters are discussed later in this chapter.

1. Visual evaluation of target detection accuracy in the heat map graphs
2. Evaluating the average signal intensity of all walking patterns
3. Evaluating the number of detected radar points per frame

Parameter 1. Visual evaluation analyzes target detection accuracy in the heat map graph. The target detection accuracy shows how well a walking person is detected within 1.5 meter wide path since heat map uses half a meter boxes to indicate a detected target. Moreover, heat map displays colored boxes showing the detected target signal to noise ratio as a signal strength. The signal strength in the heap map 43 is shown from best to worse as bright yellow, green colors to duller green, blue and purple colors. The visual evaluation guide for analysing the radar configuration heat maps are described in Table 3.

Table 3. *Visual Evaluation Guide*

Accuracy	**Value**	**Definition**
Excellent	5	Evenly detected points in the graph with minimal gaps and reflected random points. High target detection accuracy of the given walking pattern. Heat map shows brighter yellow, greenish and blueish boxes indicating a stronger signal strength.
Good	4	Some gaps in the heat map graph and minimal detected reflections. Target detection accuracy shows a maximum of 1.5m wide walking path. (One colored box is 0.25m^2). Heat map shows mostly greenish and blueish colors indicating intermediate strength signal from the detected target.
Fair	3	Target is not detected in large areas (for a few meters). Target detection accuracy shows a maximum of 1.5m wide walking path. Heat map graph displays mostly greenish and blueish signal strength.
Poor	2	Target detection accuracy shows poor results of detecting a walking person. The captured data is inaccurate to the actual walking scenario.
Very Poor	1	Target detection accuracy is very low and walking pattern is hardly recognizable.

5.2.3 Line Pattern Experiments

In line pattern experiments long-range and short-range configuration files are used with data collection device to determine how far can radars detect a walking person. A person started walking in front of the radar and finished at 40 meters, then turned around and walked back to the radar.

Long-Range Radar Configuration Files

Figure 45 is illustrating three heat map graphs recorded with *LR_people_conf_1*. Both radars detected moving target up to 20 meters. After that, radar 1 signal weakened around 25 meters and radar 2 target detection was a little over 30 meters. The colored boxes are mostly up to three boxes wide, thus person was detected on 1.5m wide path. This path width is logical and accurate since the person width is little over 0.5m.

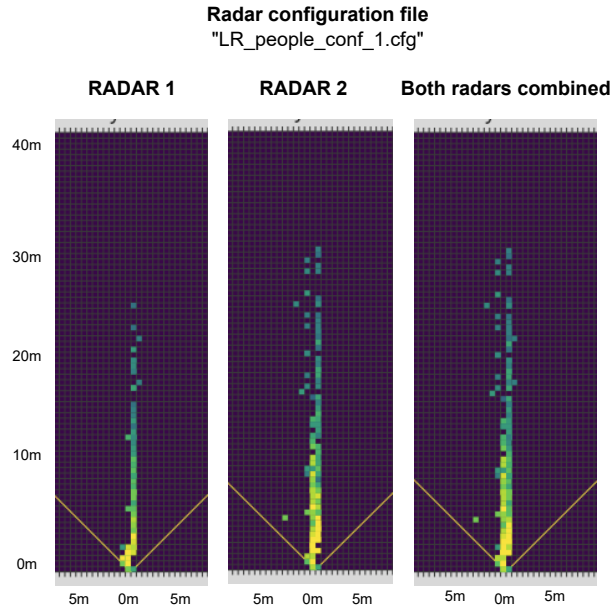

Figure 45. *Line Pattern with LR_people_conf_1 (10 FPS)*

Figure 46 shows radar 1 detecting a target at the maximum distance of 25 meters and radar 2 a little over 30 meters. One data point was captured around 35 meters. The detected walking path is mostly one color box wide. This means person was detected within 0.5m wide path.

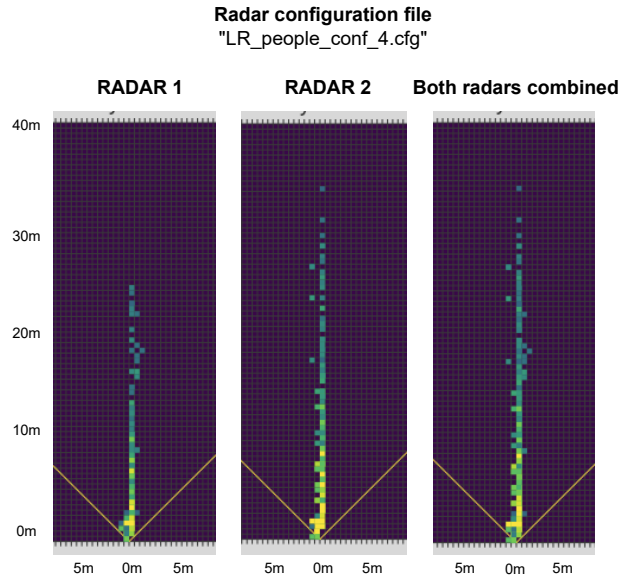

Figure 46. *Line Pattern with LR_people_conf_4 (10 FPS)*

Figure 47 displays radar 1 capturing a target moving around at 30 meters. Radar 2 is quite evenly detecting a target up to 35 meters. The detected path is mostly up to 3 boxes wide resulting in 1.5 meters. Higher frame rate definitely helped capturing more data points from the moving target.

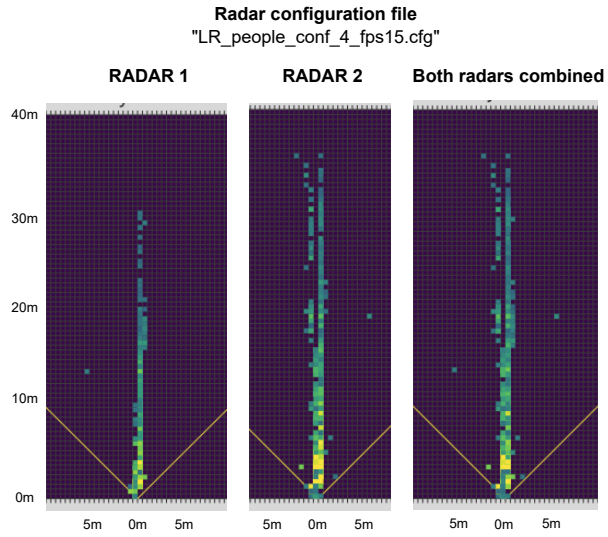

Figure 47. *Line Pattern with LR_people_conf_4 (15 FPS)*

Short-Range Radar Configuration Files

Figure 48 shows both radars having similar results. From radar to 25 meters data points are captured evenly. Then, from 25m to around 32m there are gaps in the heat maps. In the

heat map where data from two radars is combined, data points are more evenly distributed from 25m to 32m. The colored boxes are up to two boxes wide indicating one meter wide path where person was detected.

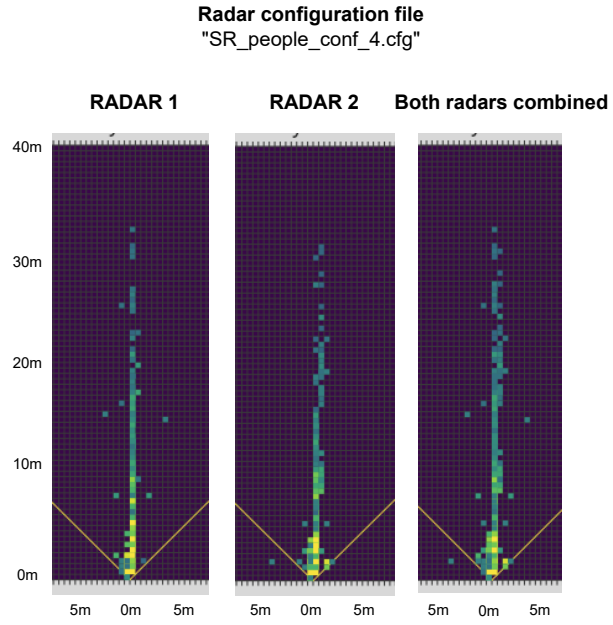

Figure 48. *Line Pattern with SR_people_conf_4 (10 FPS)*

Figure 49 displays radar 1 detecting a target farther than radar 2. The radar 1 is detecting a target up to around 32m and radar 2 almost up to 30m. The colored boxes are up to 5 or 6 in width, thus walking person is detected in 3m wide path. For some reason, this radar configuration detected a lot of reflections. The actual moving target width was around 0.50m.

The horizontal empty lines in the heat map are probably caused by the low range resolution. As shown in Table 2, the range resolution for *crit_infra_conf* is 0.586m and each colored box has a size of 0.5x0.5m. This is why, at certain distances one line of heat map boxes are skipped.

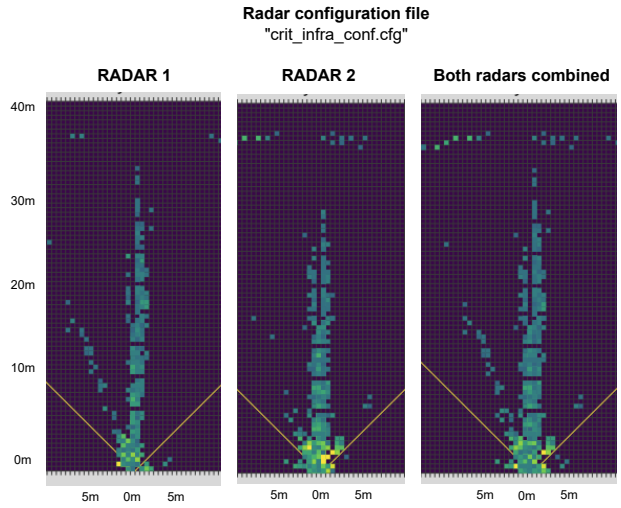

Figure 49. *Line Pattern with crit_infra_conf (15 FPS)*

Line Pattern Results

The results of line walking patterns with five different radar configuration files are illustrated in Table 4. This table analyses the results of three long-range and two short-range radar configuration files. The heat map visualization graphs are compared using a five point rating scale as shown and explained in Table 3.

Table 4. *Visual Evaluation of Line Pattern*

Radar Configuration	**Vertical Movement**
LR_people_conf_1 (10 FPS)	(4) Good
LR_people_conf_4 (10 FPS)	(4) Good
LR_people_conf_4 (15 FPS)	(5) Excellent
SR_people_conf_4 (10 FPS)	(4) Good
crit_infra_conf (15 FPS)	(2) Poor

The visual evaluation results for line pattern experiment shown in Table 4 conclude that one configuration (*LR_people_conf_4* with 15 FPS) has excellent target detection accuracy, three others have good and *crit_infra_conf* is performing poorly. The radar configuration performed poorly as the target detection did not match accurately to the actual target movement and there were many reflection. The reflections resulted in more falsely detected data points per frame as shown in column "Combined" in Table 5.

Table 5. *Line Pattern Evaluation of Points per Frame and Average Signal Intensity*

Radar Configuration	**Radar 1**	**Radar 2**	**Combined**	**Signal Avg**
LR_people_conf_1 (10 FPS)	1.49	1.65	3.14	28.98
LR_people_conf_4 (10 FPS)	1.64	1.5	3.14	26.82
LR_people_conf_4 (15 FPS)	1.46	1.47	2.93	26.85
SR_people_conf_4 (10 FPS)	1.86	2.23	4.10	27.05
crit_infra_conf (15 FPS)	9.13	9.85	18.98	26.92

Table 5 shows three columns for how many points were detected per one radar frame. Columns "Radar 1" and "Radar 2" show the points per frame for each radar and the "Combined" column displays the combined value from two radars. This combined value is later used in equation for finding the best radar configuration. The column "Signal Avg" shows the average signal strength for each radar configuration. All five configuration average signal strengths are similar where the best and worst values are 28.98 and 26.82.

The unusually high number of points per frame recorded with *crit_infra_conf* can be explained using the visual analysis. With this particular configuration, there were a lot of reflections (inaccurate data), thus increased number of points per frame.

5.2.4 T-Pattern Experiments

The T-pattern experiment shows how well radars are detecting a person walking 20 meters away from the radar and then side-to-side (horizontal) movement for 30 meters total. The reflected signal intensity is the strongest in front of the radar and as an object moves away from the center, then the signal weakens.

Long-Range Radar Configuration Files

Figure 50 displays radar 1 capturing less data points than from the radar 2 in vertical movement. The side-to-side movement data capturing is poor for both radars. Although, when two radar data clouds are combined, the two radars complement each other and final result is a bit more evenly and more accurately captured data points.

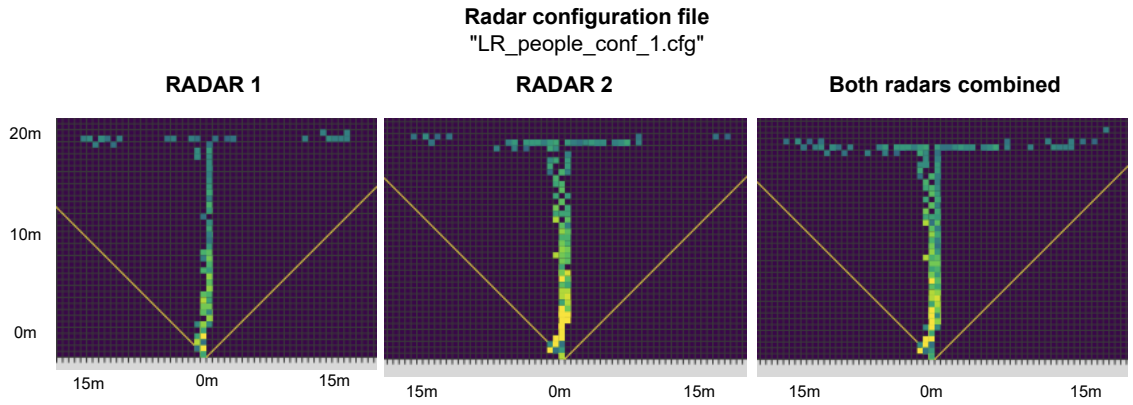

Figure 50. *T-Pattern with LR_people_conf_1 (10 FPS)*

Figure 51 shows radar 1 capturing a lot less data points than radar 2. The vertical movement of radar 2 is two boxes wide resulting in 1 meter wide walking path detection and up to 10 meters, the signal intensity is excellent. When combining two radar point clouds in a heat map graph, then the results are averaged and somewhat improved.

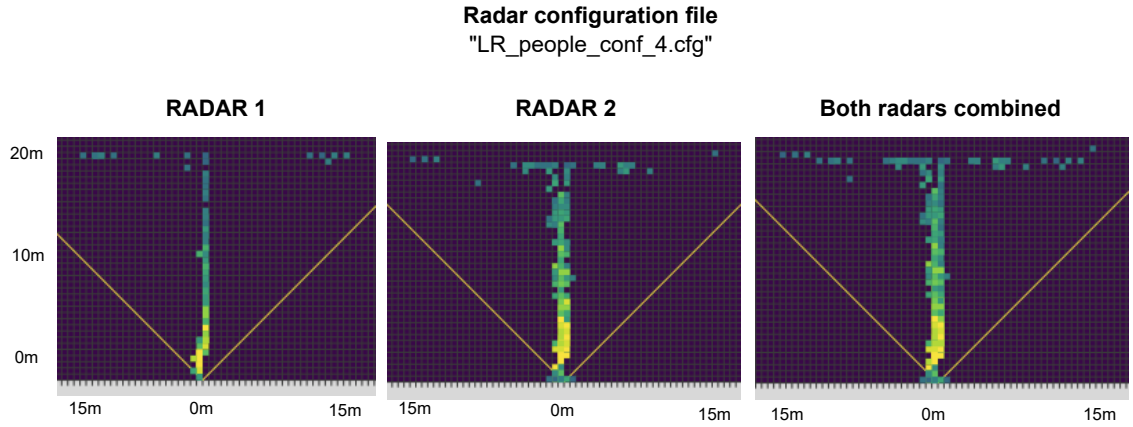

Figure 51. *T-Pattern with LR_people_conf_4 (10 FPS)*

Figure 52 shows radar 1 capturing less data points and signal quality is lower as well. On the other hand, radar 2 is excellent at capturing vertical and side-to-side target movement. The higher frame rate definitely helped capture more points resulting in more evenly and accurately detected points. The colored boxes in the combined radar heat map is up to 2 boxes wide indicating 1 meter wide path detecting a moving target.

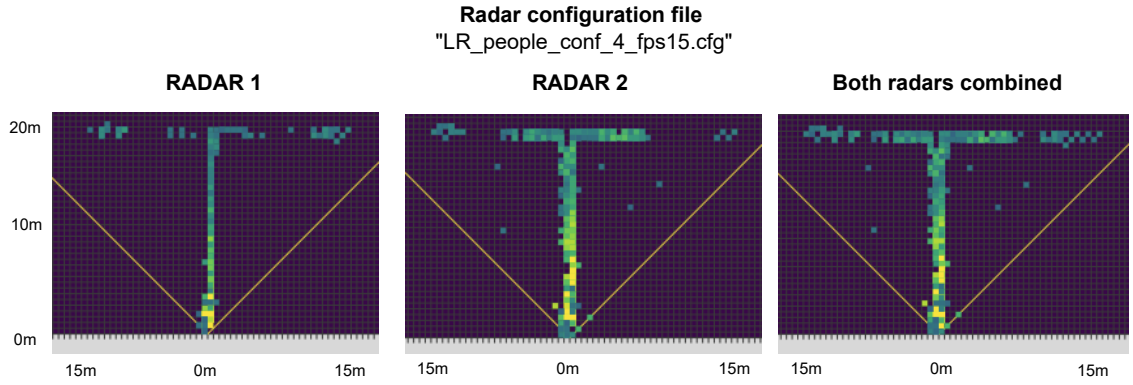

Figure 52. *T-Pattern with LR_people_conf_4 (15 FPS)*

Short-Range Radar Configuration Files

Figure 53 shows both radars having similar results. Vertical movement is evenly detected for both radars. Although, horizontal movement target detection is poor for both radars. The colored boxes are only one box wide resulting in 0.5 meter wide path where target was detected. Combining two radars improves the result.

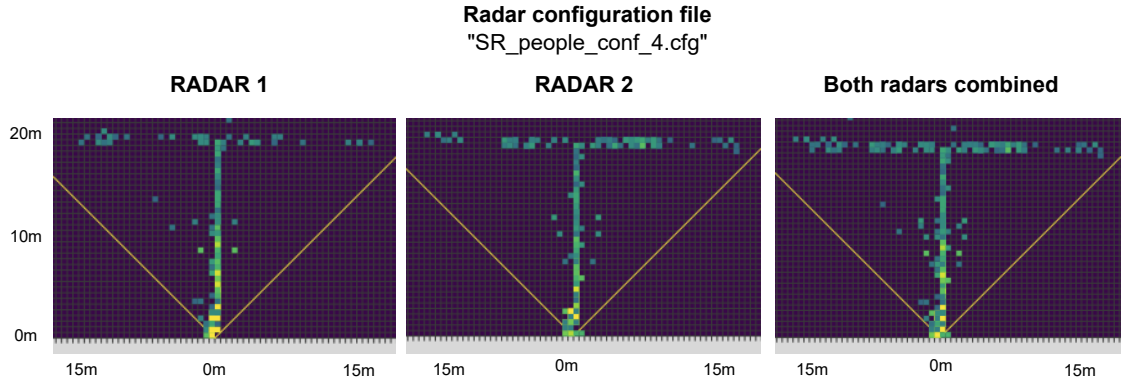

Figure 53. *T-Pattern with SR_people_conf_4 (10 FPS)*

Figure 54 displays both radars having similar results in vertical movement. As explained previously, the horizontal empty lines are coming from the low range resolution of 0.586m for this particular radar configuration. The horizontal movement is rated poor as there are large areas where target is not detected. Combining two radar results is showing improvements in target detection for both vertical and horizontal movement.

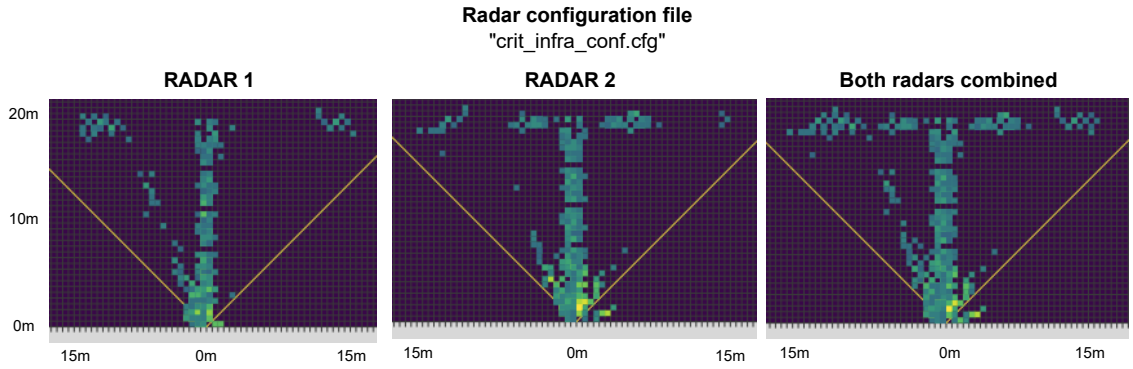

Figure 54. *T-Pattern with crit_infra_conf (15 FPS)*

T-Pattern Results

Table 6 illustrates the T-pattern walking experiment results with vertical and horizontal movements. Both movements are graded using five point scale described in Table 3.

Table 6. *Visual Evaluation of T-Pattern*

Radar Configuration Files	**Vertical Movement**	**Horizontal Movement**
LR_people_conf_1 (10 FPS)	(5) Excellent	(4) Good
LR_people_conf_4 (10 FPS)	(5) Excellent	(3) Fair
LR_people_conf_4 (15 FPS)	(5) Excellent	(4) Good
SR_people_conf_4 (10 FPS)	(4) Good	(4) Good
crit_infra_conf (15 FPS)	(2) Poor	(2) Poor

Table 7. *T-Pattern Evaluation of Points per Frame and Average Signal Intensity*

Radar Configuration	**Radar 1**	**Radar 2**	**Combined**	**Signal Avg**
LR_people_conf_1 (10 FPS)	1.56	1.56	3.12	27.30
LR_people_conf_4 (10 FPS)	1.44	1.64	3.07	27.39
LR_people_conf_4 (15 FPS)	1.49	1.51	3.00	26.27
SR_people_conf_4 (10 FPS)	1.86	1.72	3.58	25.14
crit_infra_conf (15 FPS)	9.45	9.56	19.01	26.84

The visual analysis of T-pattern walking experiments show that both *LR_people_conf_1* and *LR_people_conf_4* with 15 FPS radar configurations have excellent target detection accuracy in vertical movement and good detection accuracy in horizontal movement. Although, the average signal intensity and points per frame values in Table 7 show that

LR_people_conf_1 configuration has higher values. All five radar configurations have similar results regarding the average signal strength.

The worst performing radar configuration is *crit_infra_conf* showing poor results in vertical and horizontal movements as there are a lot of reflection. These reflections increase the combined points per frame value in column "Combined" in Table 7.

5.2.5 Triangle Pattern Experiments

The triangle walking pattern shows a vertical and a triangle movement. The triangle movement consists of horizontal and diagonal movement. At first, a person walks vertically up to 20 meters away from the radar. After that, turns 90 degrees left and walks 15 meters, then walks to the middle point of the vertical movement (10 meters from radar). Next, walks to the right side of 15 meters from the center and 20m away from the radar, turns around and returns to the center and finally walks 20 meters back to the radar. This walking routine is better explained in Figure 44.

Long-Range Radar Configuration Files

Figure 55 shows the vertical movement of radar 1 is worse than radar 2. The radar 2 has better signal strength and target detection accuracy up to 20 meters compared to radar 1. The triangle movement is very poor for radar 1 and poor for radar 2 as there are large areas where no points are detected. Combined result from two radars improves the triangle movement target detection accuracy.

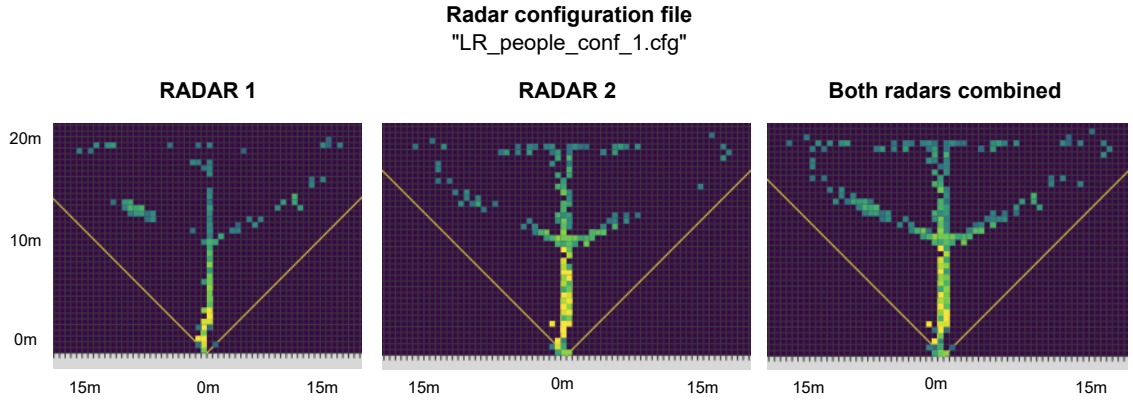

Figure 55. *Triangle Pattern with LR_people_conf_1 (10 FPS)*

Figure 56 displays radar 1 having poorer vertical movement target detection at 20 meters than radar 2. The signal quality is also better for radar 2 as there are more brighter yellow boxes. Triangle movement detection is poor for both radars although each radar detected a

target where other did not, thus combining two radars improved the final result.

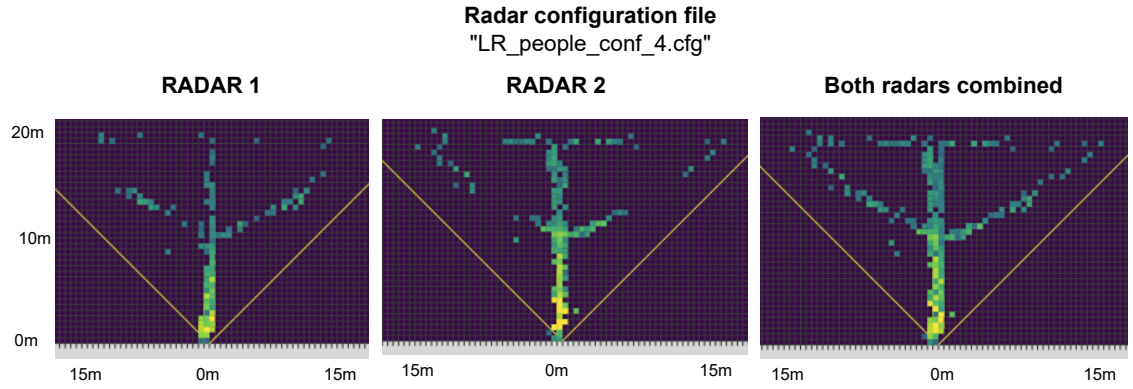

Figure 56. *Triangle Pattern with LR_people_conf_4 (10 FPS)*

Figure 57 shows better result for both radar 1 and radar 2. The radar 2 vertical movement is better as the signal quality is higher and target is detected more evenly. Combining two radar heat map graphs improves the overall target detection.

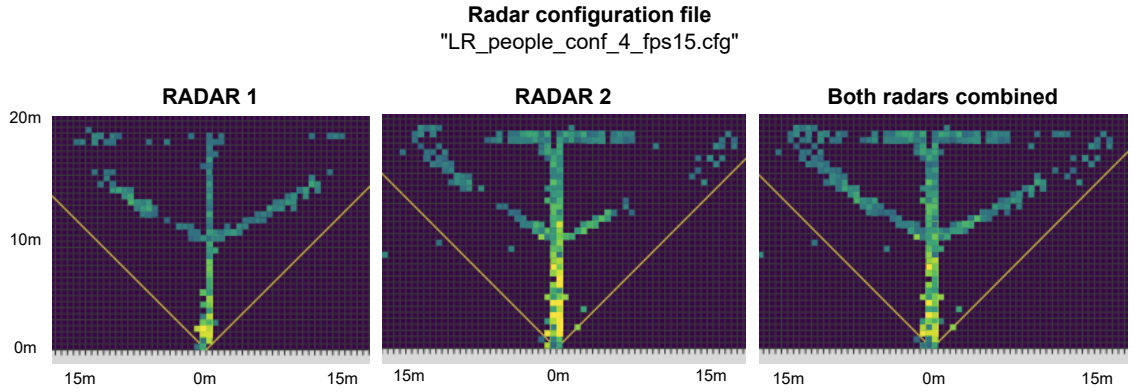

Figure 57. *Triangle Pattern with LR_people_conf_4 (15 FPS)*

Short-Range Radar Configuration Files

Figure 58 shows similar results for both radars. Vertical movement is similarly good for both. Triangle movement detection for both radars has empty areas where target was not detected. Combining the two radar results creates a bit finer result in target detection accuracy and in signal quality.

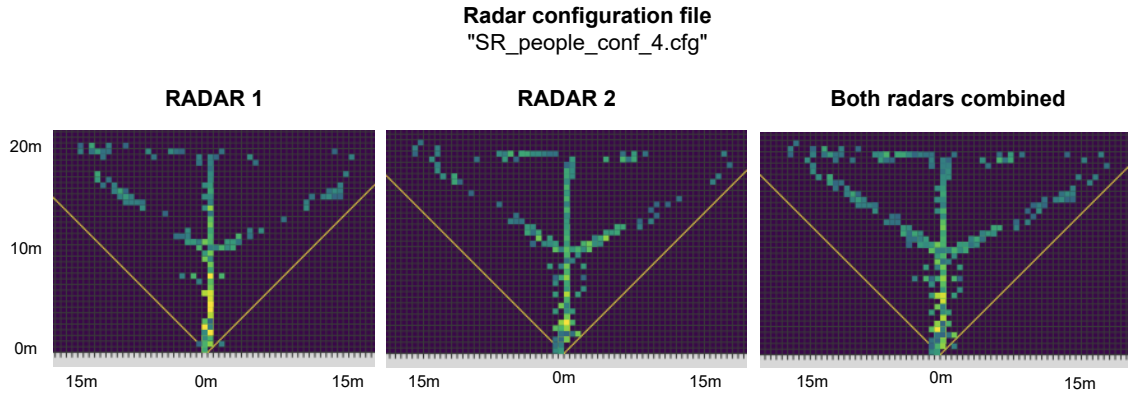

Figure 58. *Triangle Pattern with SR_people_conf_4 (10 FPS)*

Figure 59 shows good results for both radars. Radar 2 has one larger empty area where target is not detected. The target detection is excellent although the accuracy is bad as the target detection path is almost 3 meters wide. The horizontal empty lines are coming from the low range resolution setting.

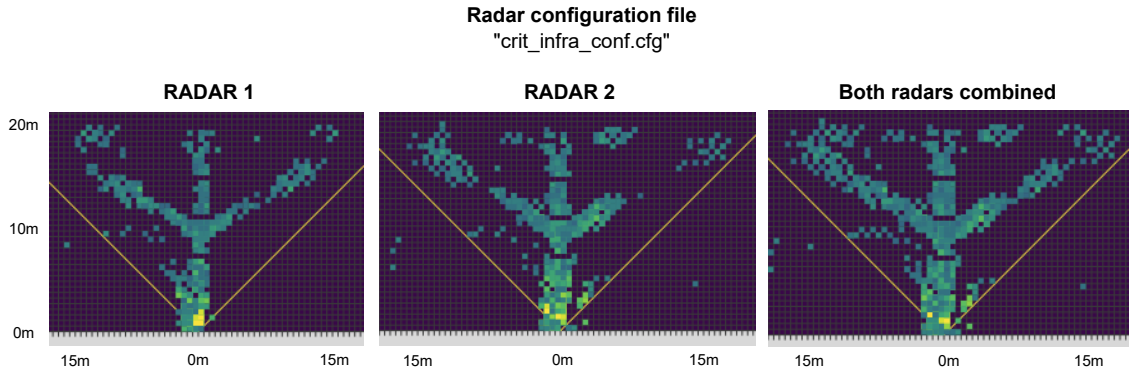

Figure 59. *Triangle Pattern with crit_infra_conf (15 FPS)*

Triangle Pattern Results

The triangle pattern results combine vertical, horizontal and diagonal target movements. These all three movements are visually analysed and the results are shown in Table 8. This analysis concludes that all long-range configurations performed well in vertical and diagonal movements. The "Horizontal Movement" column in the table shows relatively poor performance results with all radar configurations. The *crit_infra_conf* had the worst visual evaluation as it had many reflections and the target detection accuracy was low. This means that actual target movement and whereabouts slightly differ from the measured results. As mentioned in the previous pattern results, the high number of combined radar points per frame shown in Table 9 comes from the falsely reflected points.

Table 8. *Visual Evaluation of Triangle Pattern*

Radar Configuration	**Vertical Movement**	**Horizontal Movement**	**Diagonal Movement**
LR_people_conf_1 (10 FPS)	(5) Excellent	(3) Fair	(4) Good
LR_people_conf_4 (10 FPS)	(5) Excellent	(3) Fair	(4) Good
LR_people_conf_4 (15 FPS)	(5) Excellent	(3) Good	(5) Excellent
SR_people_conf_4 (10 FPS)	(4) Good	(3) Fair	(4) Good
crit_infra_conf (15 FPS)	(2) Poor	(2) Poor	(3) Fair

Table 9 shows detected points per frame for each radar and "Combined" column displays the summarized value from two radars. The *crit_infra_conf* high value comes from the reflected points as explained previously. Another short-range radar configuration having the highest of 3.58 points per frame can be explained with the higher range resolution they have. The radar configuration average signal strengths are shown in column "Signal Avg".

Table 9. *Triangle Pattern Evaluation of Points per Frame and Average Signal Intensity*

Radar Configuration	**Radar 1**	**Radar 2**	**Combined**	**Signal Avg**
LR_people_conf_1 (10 FPS)	1.48	1.62	3.10	27.52
LR_people_conf_4 (10 FPS)	1.37	1.49	2.86	25.88
LR_people_conf_4 (15 FPS)	1.41	1.49	2.91	25.90
SR_people_conf_4 (10 FPS)	1.66	1.92	3.58	24.86
crit_infra_conf (15 FPS)	8.27	8.70	16.97	26.18

5.2.6 Data Collection Device Recordings with Invalid Reflected Data Points

Some generated radar configuration files did not work as intended when recording different walking patterns. Numerous long-range and short-range radar configurations were not working on the radar device or produced invalid data points. These invalid data points are probably some reflections of actual data points. Some examples for each walking pattern are shown in Figures 60, 61 and 62.

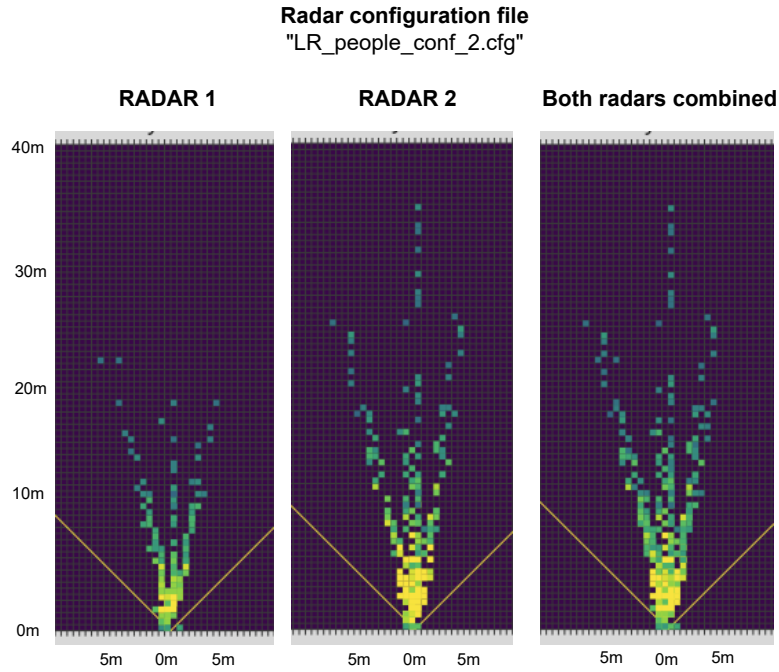

Figure 60. *Line Pattern with LR_people_conf_2 (10 FPS)*

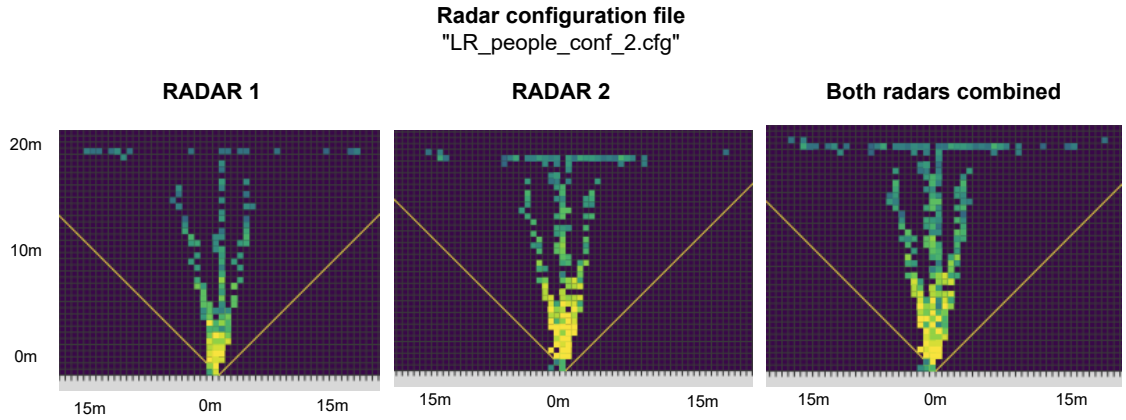

Figure 61. *T-Pattern with LR_people_conf_2 (10 FPS)*

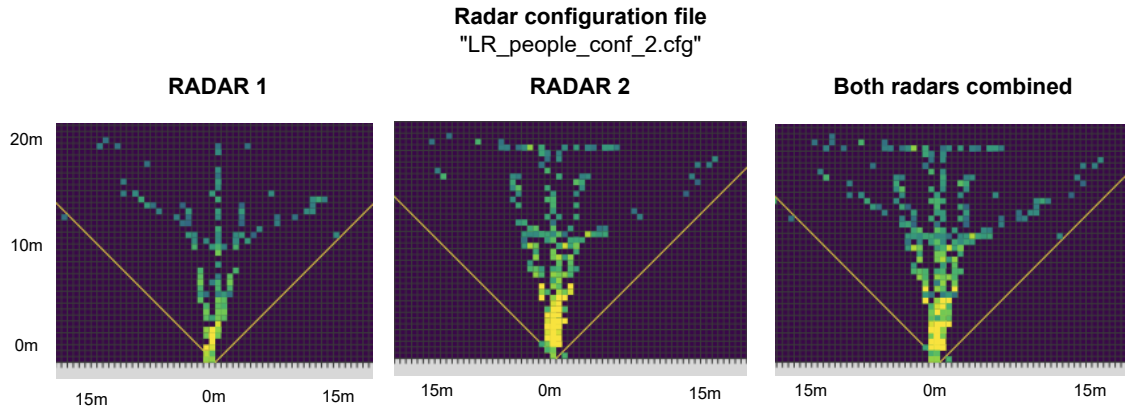

Figure 62. *Triangle Pattern with LR_people_conf_2 File (10 FPS)*

5.2.7 Summary of Conducted Experiments

Heat Map Visual Analysis Results

The three walking patterns consist of vertical, horizontal and diagonal movements. The "Vertical Movement" column from Table 10 averages all vertical movement evaluations from all three walking patterns. The column "Horizontal Movement" averages horizontal movement evaluations from T-pattern and triangle patterns. Finally, triangle pattern was used to get diagonal movement from a moving target and this value is shown in column "Diagonal Movement". The final column "Summarized concludes all previous three columns by finding the average. This summarized column shows the overall target detection accuracy of all five radar configurations and this value is used in final equation for finding the best radar configuration.

Table 10. *Overall Experimental Results for Target Detection Accuracy*

Radar Configuration	**Vertical Movement**	**Horizontal Movement**	**Diagonal Movement**	**Summarized**
LR_people_conf_1 (10 FPS)	Excellent (4.67)	Good (3.50)	Good (4.00)	Good (4.06)
LR_people_conf_4 (10 FPS)	Excellent (4.67)	Fair (3.00)	Good (4.00)	Good (3.89)
LR_people_conf_4 (15 FPS)	Excellent (5.00)	Good (4.00)	Excellent (5.00)	Excellent (4.67)
SR_people_conf_4 (10 FPS)	Good (4.00)	Good (3.50)	Good (4.00)	Good (3.83)
crit_infra_conf_4 (15 FPS)	Poor (2.00)	Poor (2.00)	Fair (3.00)	Poor (2.33)

Determining the Best Radar Configuration

As mentioned previously, there are three evaluation parameters that are used to find the best radar configuration. The first evaluation parameter is explained in Table 3. The other two evaluation parameters are the two and three in the following list.

1. Visual evaluation of target detection accuracy in the heat map graphs
2. Evaluating the average signal intensity of all walking patterns
3. Evaluating the number of detected radar points per frame

Parameter 2. This evaluation parameter shows the average signal strength of all the

detected points in the radar ROS bag file. Firstly, each radar point signal strength ($radar1\_signals$ and $radar2\_signals$) is added together for two radars and this is divided by the total number of radar points ($radar1\_points$ and $radar2\_points$) in the ROS bag file. This gives an average signal strength of all the detected radar points (Equation 5.1)

$$avg\_signal\_strength = \frac{radar1\_signals + radar1\_signals}{radar1\_points + radar2\_points} \tag{5.1}$$

Parameter 3. This parameter evaluates the number of detected points per one radar frame. This value is calculated using the equation 5.2. Both radar total number of points ($total\_radar1\_points$ and $total\_radar2\_points$) are divided with the total number of radar frames ($radar1\_frames$ and $radar2\_frames$) in the ROS bag. Then, adding these values together shows the result of combined radar points per one radar frame.

$$points\_per\_frame = \frac{total\_radar1\_points}{total\_radar1\_frames} + \frac{total\_radar2\_points}{total\_radar2\_frames} \tag{5.2}$$

Finding the Best Radar Configuration. Each radar configuration has a number between zero and one of how well it performed compared to other configurations. The final result is concluded using combined points per frame, average signal strength and the visual result from vertical, horizontal and diagonal movements. All these three evaluation parameters have different weights in the final equation as one parameter is more significant than other. Visual result (parameter 1) has 0.5, average signal strength (parameter 2) has 0.4 and points per frame result (parameter 3) has 0.1. The reason why the visual analysis has the highest weight is the importance of target detection accuracy. For instance, the points per frame and average signal strength does not matter if the detected target movement does not match to the actual target movement. The average signal strength results are more important than points per frame as a radar could easily detect a lot of reflections, thus increasing the points per one radar frame. Each parameter result is a number between one and zero as the best performing radar configuration value is used as the maximum value. This ensures that, with each parameter value is compared to the best performing radar configuration. Hypothetically, if one radar configuration has the highest values in all three parameters, then the final result would be 1.

The best performing radar configuration is determined using equation 5.3. As explained previously, the main idea is dividing the comparing value by the best result from each evaluation parameter. For instance, in Table 5.3 in column "Points per Frame" the highest

value is 18.32 and in column "Signal Strength" the highest value is 27.93. There is one exception with column "Visual Results" where the best value is five since five point scale is used. The parameter weights are multiplied with all the corresponding evaluation parameters and added together. The final result is between 0 and 1.

$$R_{conf} = 0.1 * \frac{PpF}{18.32} + 0.4 * \frac{AvgSS}{27.93} + 0.5 * \frac{V_{result}}{5} \tag{5.3}$$

In Table 5.3 columns "Points per Frame" (PpF), "Signal Strength" ($AvgSS$) and "Visual Result" (V_{result}) are used to calculate the values in "Final Result" column.

Table 11. *Best Radar Configuration*

Radar Configuration	**Points per Frame**	**Signal Strength**	**Visual Result**	**Final Result**
LR_people_conf_1 (10 FPS)	3.12	27.93	(4.06) Good	0.82
LR_people_conf_4 (10 FPS)	3.02	26.70	(3.89) Good	0.79
LR_people_conf_4 (15 FPS)	2.95	26.34	(4.67) Excellent	0.86
SR_people_conf_4 (10 FPS)	3.75	25.69	(3.83) Good	0.77
crit_infra_conf (15 FPS)	18.32	26.65	(2.33) Poor	0.72

The value in column "Final Result" shows how well radar configuration performed with all three evaluation parameters (points per frame, average signal strength and visual result). The results reveal that the best radar configuration is *LR_people_conf_4* with 15 FPS (0.86). The worst performing radar configuration is *crit_infra_conf* (0.72).

Conclusions

Overall, the data collection device with two radars worked well. The target detection accuracy was increased when two radar point clouds were combined and displayed in the heat map graph. The walking pattern experiments showed that two radars complemented each other as both radar point clouds overlapped resulting in better averaged signal strength, thus increasing the validity of detected objects. In general, two radar point clouds were overlapping with each other although there were areas where one radar detected target movements and other did not and vice verse. All things considered, data collection device with two radars improved smaller object detection accuracy.

6. Summary

Millimeter wave radars are used in numerous commercial and industrial systems as they are safer, more autonomous and extend the functionality of many other sensors. Sensing technologies such as vision cameras, thermal cameras and passive infrared sensors have their disadvantages over radars. Radars can operate in bad weather conditions and measure detected target's distance and velocity.

The motivation of this thesis was to find the best radar configuration to achieve the best walking pedestrian detection accuracy in given radar constraints. This was achieved using two millimeter wave AWR1843 modules to conduct experiments for finding the best radar configuration and testing how well two radars operate together. Both radars produced two separate radar point cloud data and it was later combined and analysed in custom developed radar data visualization software.

The experimental results of all defined tasks are described below.

1. Finding the best radar configuration to achieve the best small object detection accuracy in the given limitation and environment
2. Developing a radar data visualization software for analysing and displaying recorded data from two radars
3. Analysing the walking pattern experiments with two radars to find the best pedestrian and small vehicle detection accuracy

Task 1 result. The radar configurations define how radars operate. In this thesis, five different radar configurations are created, tested and utilized in walking pattern experiments. The walking pattern experiments showed vertical, horizontal and diagonal movement in front of the radar and the signal quality and target detection accuracy was measured using a five point scale. The vertical movement means walking away from the radar and then walking back. Only one radar configuration achieved excellent result, three other configurations were good and the last configuration performed not as well as the others.

Task 2 result. A custom radar data visualization software was developed to combine two radar point clouds together and use it to display and analyse the data. In the end, the

software project was successfully used to analyze and compare recorded radar configuration results. This application can be used in the future projects for analysing many other conducted experiments.

Task 3 result. The walking pattern experiment results showed that two radars were operating successfully and complemented each other. The two radars were not synchronized in hardware, thus data was later processed and combined in radar data visualization software. The radar data visualization software displayed the results of combining two radar point clouds and the visual analysis showed that combining two radars improved the target detection accuracy. Data was displayed as a heat map graph showing detected targets as half a meter square boxes in the graph. The box color indicated the signal strength of the detected target. Moreover, the target detection accuracy results are measuring by using two methods. First, two radars together produced larger point cloud, than only one radar. This resulted in more data points per one detected target. The second method showed more averaged signal strength. For instance, when two radars both detected an object at the same location, then these point signal intensities are averaged resulting in more reliable value.

In conclusion, two millimeter wave AWR1843 radars can be combined successfully in software and the target detection accuracy is improved compared to using only one radar. The Flutter radar data visualization application helped analysing the recorded experiments. Lastly, five radar configurations were compared and four of them out of five performed good or excellent.

Bibliography

[1] Fierce Electronics. “mmWave-based radar expands use cases for people, motion sensing”. In: (2020). URL: https://www.fierceelectronics.com/sensors/mmwave-based-radar-expands-use-cases-for-people-motion-sensing.

[2] M. Caris et al. “Detection of small UAS with W-band radar”. In: *2017 18th International Radar Symposium (IRS)*. 2017, pp. 1–6. DOI: 10.23919/IRS.2017.8008143.

[3] Utmel Electronics. “Millimeter Wave Radar: Advantages, Types, and Applications”. In: (2021). URL: https://www.utmel.com/blog/categories/sensors/millimeter-wave-radar-advantages-types-and-applications.

[4] *AWR1843Boost Evaluation Module*. [Accessed: 15-04-2022]. URL: https://www.ti.com/tool/AWR1843BOOST.

[5] *AWR2243 Cascade Radar*. [Accessed: 30-04-2022]. URL: https://www.ti.com/lit/an/swra574b/swra574b.pdf?ts=1651255611515&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FAWR2243#:~:text=Cascading%20of%20two%20and%20four,Semi%20and%20Fully%20Autonomous%20Driving..

[6] Cesar lovescu and Sandeep Rao. *The fundamentals of millimeter wave radar sensors*. [Accessed: 04-04-2022]. URL: https://www.ti.com/lit/wp/spyy005a/spyy005a.pdf?ts=1633879288914&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FAWR2243.

[7] Electronics World. “"Chirp" A New Radar Technique”. In: (2020). URL: https://www.rfcafe.com/references/electronics-world/chirp-new-radar-technique-january-1965-electronics-world.htm.

[8] Sandeep Rao. *Introduction to mmwave Sensing: FMCW Radars*. [Accessed: 04-04-2022]. URL: https://training.ti.com/sites/default/files/docs/mmwaveSensing-FMCW-offlineviewing_0.pdf.

[9] Patrick Hindle. “Comprehensive Survey of 77, 79 GHz Automotive Radar Companies – Sensors and ICs”. In: (2020). URL: https://www.microwavejournal.com/articles/33705-comprehensive-survey-of-77-79-ghz-automotive-radar-companies-sensors-and-ics.

[10] Karthik Ramasubramanian, Kishore Ramaiah, and Artem Aginskiy. *Moving from legacy 24 GHz to state-of-the-art 77 GHz radar*. [Accessed: 06-04-2022]. URL: https://www.ti.com/lit/wp/spry312/spry312.pdf?ts=1633541955309&ref_url=https%253A%252F%252Fwww.google.com%252F.

[11] Linga Reddy Cenkeramaddi et al. “A Survey on Sensors for Autonomous Systems”. In: *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. 2020, pp. 1182–1187. DOI: 10.1109/ICIEA48937.2020.9248282.

[12] Raivo Sell. *Future Transport Ecosystem Management Solution*. [Accessed: 09-05-2022]. URL: https://www.etis.ee/Portal/Projects/Display/ca68a8dd-ded9-43d0-815f-e044e7b1bec4.

[13] *Nvidia Jetson Nano Developer Kit*. [Accessed: 24-04-2022]. URL: https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

[14] *The Raspberry Pi Camera Module 2*. [Accessed: 24-04-2022]. URL: https://www.raspberrypi.com/products/camera-module-v2/.

[15] *mmWave Demo Visualizer application*. [Accessed: 24-04-2022]. URL: https://dev.ti.com/gallery/view/mmwave/mmWave_Demo_Visualizer/ver/3.5.0/.

[16] Texas Instruments. *mmWave SDK User Guide release 3.1.1*. [Accessed: 17-04-2022]. URL: https://dev.ti.com/gallery/view/mmwave/mmWave_Demo_Visualizer/ver/4.2.0/.

[17] Dirk Thomas Ken Conley. *Python rosypy library*. [Accessed: 21-04-2022]. URL: http://wiki.ros.org/rospy.

[18] The Matplotlib Development team. *Python matplotlib library*. [Accessed: 21-04-2022]. URL: https://matplotlib.org/.

[19] Karl Kivi. *Flutter Radar Data Visualization Application*. [Accessed: 05-05-2022]. URL: https://gitlab.cs.ttu.ee/karlki/radar_front_end/.

Appendices

Appendix 1 - Non-exclusive for reproduction and publication of a graduation thesis

Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Karl Kivi

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Improving pedestrian detection accuracy using single and multiple mmwave radar configurations", supervised by Karl Janson and Mairo Leier
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

09.05.2022

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.