MACHINE LEARNING-BASED INTERFERENCE MITIGATION AND GHOST TARGET REDUCTION FOR ONE-BIT QUANTIZED SIMO FMCW AUTOMOTIVE RADARS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MUSA BURAK BAYTOK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2024

Approval of the thesis:

MACHINE LEARNING-BASED INTERFERENCE MITIGATION AND GHOST TARGET REDUCTION FOR ONE-BIT QUANTIZED SIMO FMCW AUTOMOTIVE RADARS

submitted by **MUSA BURAK BAYTOK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Ali Özgür Yılmaz
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Sencer Koç
Electrical and Electronics Engineering, METU _____

Prof. Dr. Ali Özgür Yılmaz
Electrical and Electronics Engineering, METU _____

Prof. Dr. Emre Aktaş
Electrical and Electronics Engineering, Hacettepe University _____

Date: 04.09.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: MUSA BURAK BAYTOK

Signature :

ABSTRACT

MACHINE LEARNING-BASED INTERFERENCE MITIGATION AND GHOST TARGET REDUCTION FOR ONE-BIT QUANTIZED SIMO FMCW AUTOMOTIVE RADARS

BAYTOK, MUSA BURAK
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Ali Özgür Yılmaz

September 2024, 85 pages

The use of automotive radars equipped with one-bit analog-to-digital converters offers a promising alternative to their high-precision counterparts, due to their cost-effectiveness and low power consumption. However, one-bit sampling can lead to the appearance of ghost targets in range-Doppler maps, potentially causing false detections. Furthermore, in scenarios involving radar-to-radar interference, the target-like appearance and high power of coherent interference exacerbates the ghost target problem, while non-coherent interference reduces target detectability by raising the noise floor.

This thesis addresses the issue of ghost targets caused by coherent interference and the noise floor increase due to non-coherent interference in one-bit quantized frequency-modulated continuous-wave radars. It explores the use of machine learning methods to mitigate interference and reduce ghost targets in range-Doppler maps.

Performance analysis of a neural network based solution is conducted based on the detection performance, the number of ghost targets in the network's output, and the SINR metric. The results demonstrate that the proposed neural networks effectively

eliminate the ghost targets caused by coherent interference, thus preventing false detections, and mitigate interference caused by non-coherent interference, thereby increasing SINR.

Keywords: Automotive Radar, One-Bit Quantization, Ghost Target, interference, Machine Learning

ÖZ

MAKİNE ÖĞRENMESİ TEMELLİ BİR-BİT ÖRNEKLENMİŞ SIMO FMCW OTOMOTİV RADARLAR İÇİN GİRİŞİM TEMİZLEME VE HAYALET HEDEF AZALTMA

BAYTOK, MUSA BURAK
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Ali Özgür Yılmaz

Eylül 2024 , 85 sayfa

Bir bitlik analog-dijital çeviriciler ile donatılmış otomotiv radarlarının kullanımı, düşük üretim maliyeti ve düşük güç tüketimi nedeniyle yüksek hassasiyetli muadillerine kıyasla umut verici bir alternatif sunar. Ancak, bir bitlik örnekleme, menzil-Doppler haritalarında hayalet hedeflerin ortaya çıkmasına yol açabilir ve bu da yanlış tespitlere neden olabilir. Ayrıca, radarlar arası girişim içeren senaryolarda, eşevreli girişimin hedef benzeri görünümü ve yüksek gücü hayalet hedef sorununu daha kötü hale getirirken, eşevreli olmayan girişim gürültü tabanını yükselterek hedefin tespit edilebilirliğini azaltır.

Bu tez, bir bitlik örneklenmiş frekans modülasyonlu sürekli dalga radarlarında eşevreli girişimin neden olduğu hayalet hedefler ve eşevreli olmayan girişim nedeniyle gürültü tabanının artması sorunlarını ele alır. Girişimi hafifletmek ve menzil-Doppler haritalarında hayalet hedefleri azaltmak için makine öğrenimi yöntemlerinin kullanımını araştırır.

Sinir ağının performans analizi, tespit performansı, ağın çıktısındaki hayalet hedeflerin sayısı ve Sinyal-Girişim-artı-Gürültü Oranı metriği bazında yapılmıştır. Sonuçlar, önerilen sinir ağlarının hayalet hedefleri etkili bir şekilde ortadan kaldırdığını, böylece yanlış tespitleri engellediğini ve eşevreli olmayan girişimi hafifleterek Sinyal-Girişim-artı-Gürültü Oranını artırdığını göstermektedir.

Anahtar Kelimeler: Otomotiv Radarı, Bir Bit Örnekleme, Hayalet Hedef, Girişim, Makine Öğrenmesi

To my family...

ACKNOWLEDGMENTS

First and foremost, I would like to extend my sincerest gratitude to my advisor, Prof. Ali Özgür Yılmaz. I have been fortunate to learn countless crucial aspects of research and engineering under his guidance. It has truly been a privilege to receive his mentorship and support throughout this journey. This thesis would not have been completed without his invaluable mentorship.

I am deeply grateful to Türk Telekom for funding my graduate studies through the 5G and Beyond Joint Graduate Support Program. I would also like to thank the Information and Communication Technologies Authority of Turkey (BTK) for organizing this impactful program, which plays a vital role in advancing research on communication technologies in our country.

I would also like to extend my heartfelt thanks to my dear friends Ahmet Furkan Odunkıran and Safa Çelik. We shared many stressful course studies and countless sleepless nights together, and I am truly grateful for their unwavering friendship and support throughout my thesis journey. Additionally, I wish to thank Anıl Kurt for his assistance and for generously sharing his valuable knowledge.

I would like to express my deep gratitude to my dear mother Hacer Baytok, father Rahmi Baytok, and sister Büşra Teker, whose constant presence and unwavering support have been a source of strength for me throughout my life. I also want to extend my thanks to my childhood friend, whom I consider like a brother, Kemal Ata Aydın, for his support.

Finally, I would like to extend my deepest gratitude to my beloved wife, Gülcenur Turgut Baytok. Since our childhood, she has been my unwavering friend, always by my side through challenging times, offering her constant support, and understanding me during my most stressful moments.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ACC	Adaptive Cruise Control
ADC	Analog-To-Digital Converter
ADAM	Adaptive Moment Estimation
ADAS	Advanced Driver-Assistance Systems
ADS	Autonomous Driving Systems
AEB	Automatic Emergency Braking
AI	Artificial Intelligence
AoA	Angle Of Arrival
ANN	Artificial Neural Network
AR	Autoregressive
AWGN	Additive White Gaussian Noise
BSD	Blind Spot Detection
CA-CFAR	Cell-Averaging Constant False Alarm Rate
CNN	Convolutional Neural Network
CFAR	Constant False Alarm Rate
CPI	Coherent Processing Interval
CUT	Cell Under Test
DFT	Discrete Fourier Transform
DL	Deep Learning
FFT	Fast Fourier Transform
FMCW	Frequency-Modulated Continuous Wave
GTN	Ghost Target Number
IMAT	Iterative Method With Adaptive Thresholding
LCA	Lane Change Assist

LNA	Low-Noise Amplifier
LPF	Low Pass Filter
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multilayer Perceptron
MIMO	Multiple-Input Multiple-Output
mGTN	Mean Ghost Target Number
MSE	Mean Squared Error
OS-CFAR	Order Statistic Constant False Alarm Rate
RCS	Radar Cross-Section
ReLU	Rectified Linear Unit
SAE	The Society Of Automotive Engineers
SINR	Signal-To-Interference-Plus-Noise Ratio
SIR	Signal-To-Interference Ratio
SISO	Single-Input Single-Output
SIMO	Single-Input Multiple-Output
SNR	Signal-To-Noise Ratio
ULA	Uniform Linear Array

CHAPTER 1

INTRODUCTION

Radar has a storied history dating back to the early 20th century. The origins of radar technology can be traced to multiple nations simultaneously exploring radio waves and their potential applications. Pioneering work in this field began with Heinrich Hertz's discovery of radio waves in the late 19th century, which established the foundational principles of radar. Practical radar systems first emerged in the 1930s, driven by the need for improved detection and tracking capabilities. In 1935, Sir Robert Watson-Watt demonstrated the feasibility of using radio waves to detect aircraft, leading to the development of the Chain Home system in the United Kingdom. This network of radar stations played a crucial role in the Battle of Britain during World War II, marking one of the first significant advancements in radar technology. Meanwhile, countries like Germany, the United States, and Japan were also developing their radar systems, further advancing the technology.

After World War II, radar technology evolved rapidly, benefiting from advancements in electronics and the burgeoning field of microwave engineering. The introduction of the magnetron in the 1940s enabled the production of high-power microwave signals, which significantly enhanced radar performance. This period saw the diversification of radar applications beyond military use, including air traffic control, weather forecasting, and maritime navigation [31].

The integration of radar technology into automotive applications began in the late 20th century. Early automotive radar systems were primarily used for crash avoidance and distance warning [38]. With advances in technology, features such as adaptive cruise control (ACC), automatic emergency braking (AEB), blind spot detection (BSD), and lane change assist (LCA) were introduced [38]. These features, along

with many others, are collectively referred to as advanced driver-assistance systems (ADAS). ADAS systems typically rely on a combination of sensors, including cameras, radar, and lidar technologies, to gather data from the vehicle's surroundings and make real-time decisions. Radar is essential for ADAS systems as it is particularly effective in adverse weather conditions such as fog, rain, or snow, where other sensors may struggle. Moreover, radar provides low cost and high precision in range, velocity, and angle measurements.

The primary goal of ADAS is to enhance driver safety and comfort while still requiring human intervention and oversight. These systems provide assistance and warnings to the driver but ultimately rely on the driver to take control in complex or unexpected situations. In contrast, autonomous driving systems (ADS) aim to achieve full automation of the driving task without the need for human intervention. ADS operates on the principle of perception, decision-making, and control, similar to ADAS but with a higher degree of autonomy. The Society of Automotive Engineers (SAE) defines driving automation across six levels as follows [34].

Level 0: No Driving Automation: The driver is entirely responsible for controlling the vehicle.

Level 1: Driver Assistance: Basic driver assistance features, such as adaptive cruise control or lane-keeping aid, are available; however, the driver must remain engaged.

Level 2: Partial Driving Automation: The vehicle can control both steering and acceleration/deceleration simultaneously under certain conditions, but the driver must remain vigilant and prepared to take control.

Level 3: Conditional Driving Automation: The vehicle can perform all driving tasks in specific conditions without constant driver supervision, but the driver must be ready to intervene if needed.

Level 4: High Driving Automation: The vehicle can operate autonomously in most driving scenarios within specific geographic areas or predefined routes, with minimal need for human intervention.

Level 5: Full Driving Automation: The vehicle can perform all driving tasks under all conditions without any human intervention, providing complete autonomy and convenience to passengers.

In Levels 0 to 2, the human driver is responsible for either fully or partially performing the dynamic driving tasks. At Level 2, the ADS can manage both steering and acceleration under certain conditions, usually in simpler driving scenarios. Beginning at Level 3, the ADS takes full responsibility for driving tasks while engaged. As we move up the levels, the need for the driver to be ready to take over decreases, until we reach Level 5, at which point a steering wheel is no longer necessary.

As the trend in ADAS systems progresses toward full autonomy, a dramatic increase in the number of radar sensors on vehicles is expected [3], with the anticipation that these sensors will maintain reliability even in increasingly complex situations. However, several challenges arise with the increasing number of sensors. One such challenge is the increase in cost and power consumption associated with the additional sensors [37]. Another important issue is the potential for interference between radar sensors [3]. Next, the mentioned challenges will be explained further.

As stated earlier, ADAS systems rely on the data from the vehicle's surroundings. For control-critical functions such as BSD and AEB, the radar system requires angular information of the target along with the range and Doppler information. The range and Doppler parameters can be estimated using a single receive antenna. However, estimating the angle parameter requires more than one receive antenna element. For this reason, multiple-input multiple-output (MIMO) radar is a suitable choice. Nevertheless, MIMO radar faces some challenges in practical implementation. One such challenge is the digitization of the received signal. Analog-to-digital converters (ADCs) are used to convert an analog signal into a digital signal. High-precision ADCs consume high power and are costly to produce, especially when considering the use of separate ADCs for multiple antenna elements. [9].

With the expected increase in the number of radar sensors on vehicles, as they move toward full autonomy, potential interference between radar sensors that could compromise system accuracy and reliability becomes a significant concern [3]. Currently, most radar systems utilized in automobiles are frequency-modulated continuous wave

(FMCW) radars [7]. According to [7], two types of interference can occur between FMCW radars, depending on their system parameters. In an interference scenario, the radar that causes the interference is called the interferer radar, and the radar that is affected by the interference is called the victim radar.

- **Similar System Parameters:** This type of interference occurs when the interferer and the victim FMCW radars have different system parameters, such as sweep bandwidths or sweep times. This interference type is called **non-coherent interference**. Non-coherent interference appears as a time-limited impulse-like signal in the time domain with a broadband frequency spectrum. As a result, non-coherent interference raises the noise floor in the spectral domain and reduces the detectability of the targets.
- **Identical System Parameters:** This type of interference occurs when the interferer and the victim FMCW radars have the same system parameters. This interference type is called **coherent interference**. Coherent interference occurs as persistent, structured signals in the time domain and narrowband peaks in the frequency spectrum. Consequently, coherent interference can create false targets and degrade the performance of the radar system by introducing systematic errors in target detection and measurement.

Although the likelihood of encountering coherent interference may be small [7], its occurrence can lead to the appearance of ghost objects, which may create dangerous situations.

1.1 Literature Review

In the literature, non-coherent interference mitigation for automotive radars utilizing high-precision quantization through signal processing methods has been extensively researched. These methods generally rely on signal detection and reconstruction in the time, frequency and time-frequency domain. In [26], a raised cosine window or zeroing is applied to the disrupted portion of a received signal. While these two methods attenuate the interference power, they also result in the loss of important signals

in the interference's overlapping region. In [29], the autoregressive (AR) model for signal reconstruction is presented. This model can retrieve more target information than the zeroing and windowed methods and can extrapolate usable signals in the interfered area. To overcome the signal gap that results from zeroing, the iterative method with adaptive thresholding (IMAT) is another signal reconstruction technique suggested in [6]. The IMAT method is a sparse reconstruction technique. Each of the mentioned methods above suppresses the interference in the time domain. However, these methods rely on the amplitude information of the received signal. An adaptive noise canceller is proposed in [18], which utilizes interference information in negative frequencies to cancel out interference in positive frequencies. This is an example of how interference mitigation is typically performed in the frequency domain. In [39], the authors utilized a constant false alarm rate (CFAR) detector in the time-frequency domain to detect and suppress interference.

Deep learning (DL) has rapidly advanced in recent years, demonstrating its effectiveness in addressing the mutual interference problem in FMCW radars. The authors of [24] proposed a recurrent neural network model with Gated Recurrent Units [8] to mitigate interference in the time domain signals. In [32], the authors utilized a convolutional neural network (CNN) to denoise automotive radar signal in the frequency domain, where the proposed network provides a signal-to-noise ratio (SNR) increase. Convolutional encoders are used to mitigate the interference effects in [14]. The authors of [12] utilized the U-Net deep neural network model for interference mitigation. In [23], the authors utilized deep learning methods in the time-frequency domain to mitigate the interference.

To the best of our knowledge, [22] is the only work that considers simultaneous coherent and non-coherent interference with high-precision quantization. Additionally, no work in the literature addresses non-coherent or coherent interference mitigation for automotive radars utilizing one-bit quantization. One of the main problems with one-bit quantization is the appearance of ghost targets in the range-Doppler maps. The authors of [17],[16] utilize iterative methods to address the ghost target problem with one-bit quantized signals without considering the interference effects.

The work of [22] considers a single-input single-output (SISO) radar system and does

not address scenarios involving multiple antenna elements. However, automotive radars are generally equipped with multiple antenna elements. Additionally, most of the work in the existing literature on interference mitigation using deep learning does not focus on radar detection performance. To investigate radar detection performance, false alarm probability is kept small and constant, and detection probability is observed. To maintain a constant false alarm probability, constant false alarm rate (CFAR) detectors are used. We aim to fill this gap by considering the single-input multiple-output (SIMO) radar system with one-bit ADCs and providing more commonly used radar detection performance metrics.

1.2 Thesis Motivation

With simplified designs and reduced data resolution, one-bit ADCs offer a potential solution to previously mentioned challenges about high-precision ADCs such as high production costs and power consumption [30]. However, despite the benefits, the non-linearity of one-bit radars introduces new challenges for conventional linear signal processing methods. Since one-bit quantization is essentially a sign comparator, samples are represented as +1 or -1. In the context of an automotive radar in a multiple-target or coherent interference scenario, after linear processing, self-generated and cross-generated harmonics appear as ghost targets in the range-Doppler maps and can be detected as real targets [30], [17]. The number of harmonics depends on the number of return signals and their amplitudes. Additionally, when addressing the coherent interference problem in automotive radars with one-bit quantized ADCs, the high power and target-like image of the interfering signal exacerbate the issue of self-generated and cross-generated harmonics appearing as ghost targets. Similarly, when considering the non-coherent interference problem, the lost amplitude information of the received signal in one-bit quantized ADCs makes it difficult to apply traditional methods.

As stated earlier, some work in the literature addresses the ghost target problem in one-bit quantized ADCs and utilizes iterative methods to solve it. This thesis aims to demonstrate that neural networks, when trained on appropriate data, can effectively reduce ghost targets caused by one-bit quantization and coherent interference, as well

as mitigate the effects of non-coherent interference with one-bit quantization, using separate networks for each problem.

In [7] it is shown that the likelihood of encountering coherent interference is small. However, the ghost target problem is not specific to automotive radars with one-bit quantization. Multistatic radar systems [10] may also suffer from ghost target problems. In addition, [15] and [36] address different aspects of improving the quality of SAR images obtained through one-bit quantization, with one focusing on residual attention mechanisms to restore images and the other on harmonic suppression through deep learning, respectively. Therefore, we believe that addressing the ghost target problem in one-bit ADCs is of significant importance.

1.3 Contributions and Novelties

This thesis considers the coherent and non-coherent interference problems in automotive FMCW radars that utilize one-bit ADCs and proposes a deep learning-based solution. Our contributions to literature in this thesis are listed as follows:

- We propose separate 2D U-Net models to reduce ghost targets and mitigate the interference in one-bit quantized single-input multiple-output (SIMO) FMCW radars affected by coherent and non-coherent interference.

- We compare the effect of using multi-beam against single-beam on neural network performance in the presence of coherent interference.

- We demonstrate that the neural network can effectively reduce the mean ghost target number (GTN) while maintaining the probability of detection $(\mathbf{P_d})$. We also demonstrate that the neural network can effectively mitigate non-coherent interference.

- We examine the limitations of the proposed neural network in restoring the original image when coherent interference is present.

1.4 The Outline Of the Thesis

In this section, we describe the outline of this thesis.

In Chapter 2, we cover the principles of FMCW radar, starting with an overview of Frequency Modulated Continuous Wave Radar and its signal analysis, including both single and multiple chirp scenarios. We then explore the signal processing techniques used for target detection, focusing on range and velocity estimation. The chapter continues with discussions on target detection methods like CA-CFAR and OS-CFAR, angle of arrival estimation, and challenges posed by FMCW mutual interference, concluding with an introduction to one-bit sampling and its effects on radar performance.

In Chapter 3, we introduce the fundamentals of machine learning, with a focus on supervised learning. We then explore artificial neural networks, highlighting multilayer perceptrons. The chapter covers the training process, including key concepts like underfitting, overfitting, loss functions, and backpropagation. We also discuss convolutional neural networks, their activation functions, and pooling operations, concluding with an overview of the U-Net architecture and its relevance to the research.

In Chapter 4, we detail the methodology, beginning with the generation of simulation data. We then describe the architecture of the neural network, including the choice of the loss function, data preprocessing steps, and considerations for test samples and dataset size. The chapter continues with a discussion of the training process of the neural network and concludes with the evaluation metrics used to assess the model's performance.

In Chapter 5, we present the results of our study, beginning with a qualitative evaluation of the three-channel neural network. We assess its generalization capabilities and explore its performance under different conditions. The chapter then shifts to a quantitative evaluation of the three-channel neural network, where we compare its performance with a single-channel input and analyze the limits of its improvement. Finally, we evaluate the neural network trained for non-coherent interference, first qualitatively and then quantitatively, to fully understand its effectiveness and potential.

In Chapter 6, we conclude the thesis and talk about future work.

CHAPTER 2

FMCW RADAR

2.1 Frequency Modulated Continuous Wave Radar

Frequency Modulated Continuous Wave (FMCW) radar is an example of a Continuous Wave (CW) radar system that continuously transmits a signal with a varying frequency at a known rate. The basic operation of FMCW radar involves the transmission of a continuous wave with a linearly increasing or decreasing frequency over time. When this signal encounters a target, it reflects off off the target, experiencing a frequency shift proportional to the target's distance and velocity when observed at the receive end. The received signal is amplified by a low-noise amplifier (LNA) and mixed with the transmitted signal followed by a low pass filter to generate the baseband signal. Lastly, an analog-to-digital converter (ADC) converts the signal to digital form before further processing. Fig. 2.1 shows the basic FMCW radar block diagram.

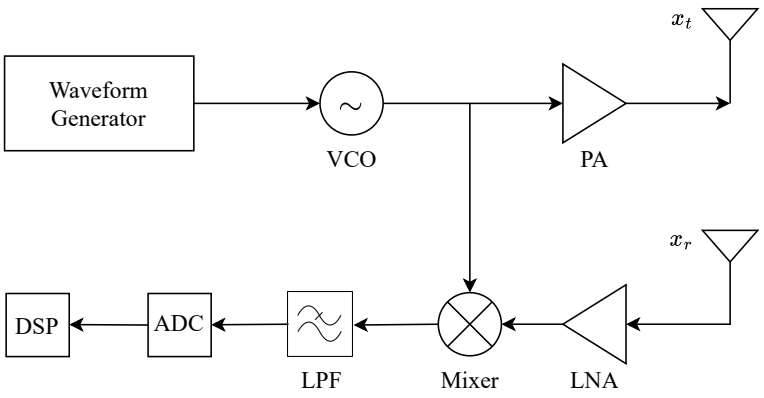


Figure 2.1: FMCW radar block diagram.

2.1.1 FMCW Radar Signal Analysis

As stated previously, the frequency of the transmitted signal increases or decreases linearly over time and this modulation type is called sawtooth modulation. Some other modulation types are triangular and sinusoidal modulations. The sawtooth modulated FMCW radar transmit and receive signal frequency can be seen in Fig. 2.2. The starting frequency of the signal is represented as f_c and the signal's bandwidth is denoted as B. The signal's frequency increases linearly as a function of sweep time T_s.

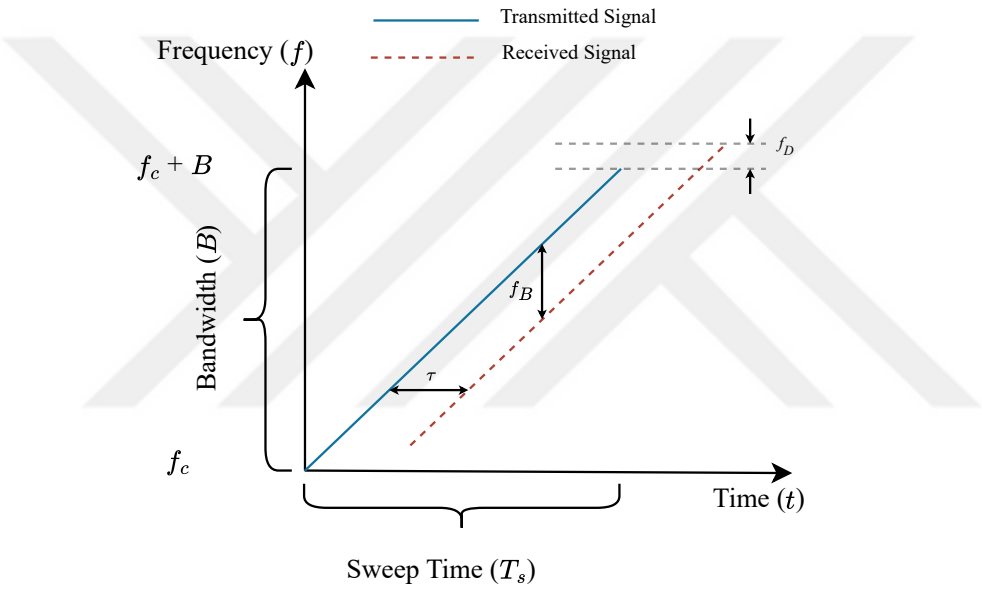


Figure 2.2: FMCW radar transmit and receive signal frequency.

The instantaneous frequency of the FMCW radar signal can be given as [20]

$$f(t) = f_c + \frac{B}{T_s}t = f_c + \alpha t, \tag{2.1}$$

where $\alpha = \frac{B}{T_s}$ represents the slope of the signal. The instantaneous phase of the signal can be found from the frequency change over time

$$\phi(t) = 2\pi \int_0^t f(t)dt = 2\pi \int_0^t (f_c + \alpha t)dt = 2\pi \left(f_c t + \frac{\alpha t^2}{2} \right), \tag{2.2}$$

and the transmitted signal can be obtained as

$$x_t(t) = \cos(\phi(t)) = \cos\left(2\pi (f_c t + \frac{\alpha t^2}{2}) \right). \tag{2.3}$$

Considering an object placed at an initial distance R_0 with a relative velocity v away from the radar, the reflected signal from the object will be a scaled and delayed version of the transmitted signal

$$x_r(t) = A_R \cos(\phi(t - \tau(t))) = A_R \cos\left[2\pi \left(f_c(t - \tau(t)) + \frac{\alpha(t - \tau(t))^2}{2} \right) \right], \tag{2.4}$$

where $\tau(t)$ is the round trip time delay between the radar and the object at time t and given as

$$\tau(t) = \frac{2(R_0 + vt)}{c} \tag{2.5}$$

with c being the speed of the light. The received signal is first amplified by an LNA and then mixed with the transmitted signal to obtain the IF signal as

$$x_b(t) = \text{LPF}\{x_t(t)x_r(t)\} = \text{LPF}\{A_R \cos\Big(\phi(t)\Big) \cos\Big(\phi\big(t - \tau(t)\big)\Big)\} \tag{2.6}$$

$$= \frac{A_R}{2} \cos\left[2\pi \left(f_c t + \frac{\alpha t^2}{2} \right) - 2\pi \left(f_c(t - \tau(t)) + \frac{\alpha(t - \tau(t))^2}{2} \right) \right] \tag{2.7}$$

$$= \frac{A_R}{2} \cos\left[2\pi \left(f_c \tau(t) + \alpha t \tau(t) - \frac{\alpha \tau(t)^2}{2} \right) \right]. \tag{2.8}$$

Substituting the delay term $\tau(t)$ given in Eqn. 2.5 into the Eqn. 2.8, the baseband signal can be obtained as [31]

$$x_b(t) = \frac{A_R}{2}\cos\Bigg[2\pi\Bigg((\frac{2f_cv}{c} - \frac{4\alpha R_0 v}{c^2} + \frac{2\alpha R_0}{c})t + (-\frac{2\alpha v^2}{c^2} + \frac{2\alpha v}{c})t^2 + (\frac{2f_cR_0}{c} - \frac{2\alpha R_0^2}{c^2})\Bigg)\Bigg]. \tag{2.9}$$

We can simplify Eqn. 2.9 by ignoring the small terms that contain c^2 in their denominator. The term $\frac{2\alpha v}{c}t^2$ is also small and thus, it can be neglected as well. As a result, after the mixer and the low-pass filter, the baseband signal can be obtained as

$$x_b(t) = \frac{A_R}{2}\cos\left(2\pi(\frac{2f_cv}{c}t + \frac{2\alpha R_0}{c}t) + \frac{4\pi f_cR_0}{c}\right). \tag{2.10}$$

In Eqn. 2.10, $\frac{4\pi f_cR_0}{c}$ is a constant phase term (will be represented as ϕ_0). $2\alpha\frac{R_0}{c}t$ is a range-dependent term which is the beat frequency and the $\frac{2f_cv}{c}t$ is a velocity-dependent term which is the Doppler frequency. The instantaneous frequency of $x_b(t)$ equals

$$F_B = \frac{2f_cv}{c} + \frac{2\alpha R_0}{c} \tag{2.11}$$

$$= f_D + f_{B,R_0}, \tag{2.12}$$

where

$$f_D = \frac{2f_cv}{c}, \quad f_{B,R_0} = \frac{2\alpha R_0}{c}. \tag{2.13}$$

Consequently, the baseband signal can also be represented as

$$x_b(t) = \frac{A_R}{2}\cos(2\pi F_B t + \phi_0). \tag{2.14}$$

In the preceding analysis, only a single chirp with the sawtooth modulation is considered. If the target reflecting the signal has a velocity relative to the radar, the Doppler frequency cannot be separated from the range-dependent term. This phenomenon,

known as range-Doppler coupling [31], introduces an error in the estimated target range and complicates the estimation of the target's velocity.

To overcome this problem, multiple chirps are transmitted and received although other approaches are possible as well. The extraction of range and velocity information will be discussed briefly, and the next section will present the signal analysis for the multiple chirp scenario.

2.1.2 Multiple Chirp FMCW Radar Signal Analysis

Considering the multiple chirp scenario for a single measurement frame, M chirps are transmitted and received. The time delay will become

$$\tau(t, m) = \frac{2\Big(R_0 + v(t + mT_s)\Big)}{c}, \tag{2.15}$$

where $m = 0, 1, ..., M-1$ represents the chirp indexes, and $t \in (0, T_s)$. Substituting $\tau(t, m)$ into Eqn. 2.8, and making the previous simplifications, the baseband signal will be represented as

$$x_b(t, m) = \frac{A_R}{2} \cos\left(2\pi(\frac{2f_c vt}{c} + \frac{2f_c mv}{c}T_s + \frac{2\alpha R_0}{c}t) + \frac{4\pi f_c R_0}{c}\right), \tag{2.16}$$

where A_R is the amplitude of the received signal. Consequently, the baseband signal for multiple chirps can also be represented as

$$x_b(t, m) = \frac{A_R}{2} \cos\Big(2\pi(F_B t + f_D m T_s + \phi_0)\Big). \tag{2.17}$$

Sampling the baseband signal using an ADC with a sampling frequency of f_s will yield the discrete baseband signal as

$$x_b(k, m) = \frac{A_R}{2} \cos\left(2\pi(F_B \frac{k}{f_s} + f_D m T_s + \phi_0)\right), \tag{2.18}$$

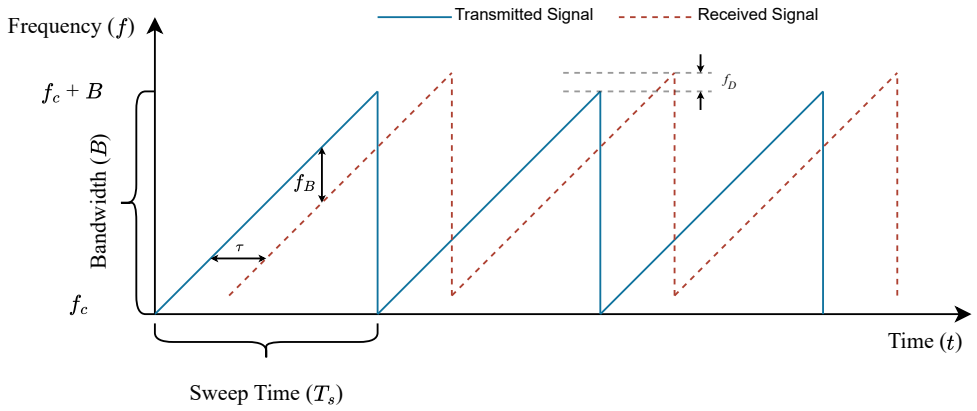


Figure 2.3: FMCW radar multiple transmit and receive signal frequency.

where $k = 0, 1, ..., K-1$ represents the sample indexes of one chirp. According to the formulation, we have obtained M chirps containing K samples each within the measurement frame. Thus, a $K \times M$ data matrix is formed in one measurement frame, also called the coherent processing interval (CPI). The columns of this matrix are referred to as the fast time dimension, while the rows are referred to as the slow time dimension. To extract the range and velocity information of the target, the discrete Fourier transform (DFT) operation is computed separately over slow and fast time dimensions.

2.1.3 Signal Processing for Target Detection

The following equation shows the DFT process across the slow-time and fast-time dimensions on the data matrix.

$$X_b(l,n) = \sum_{m=0}^{M-1} \sum_{k=0}^{K-1} x_b(k,m) \cdot \exp\left(-j2\pi \frac{kl}{K}\right) \cdot \exp\left(-j2\pi \frac{mn}{M}\right), \tag{2.19}$$

where $l = -K/2, ..., K/2-1$, and $n = -M/2, ..., M/2-1$ indexes the beat frequency and Doppler frequency bins, respectively. To compute the DFT, a fast Fourier transform (FFT) algorithm is often employed with proper K, M values. Fig. 2.4 illustrates the FFT processing of the collected signals. After mixing and digitizing the

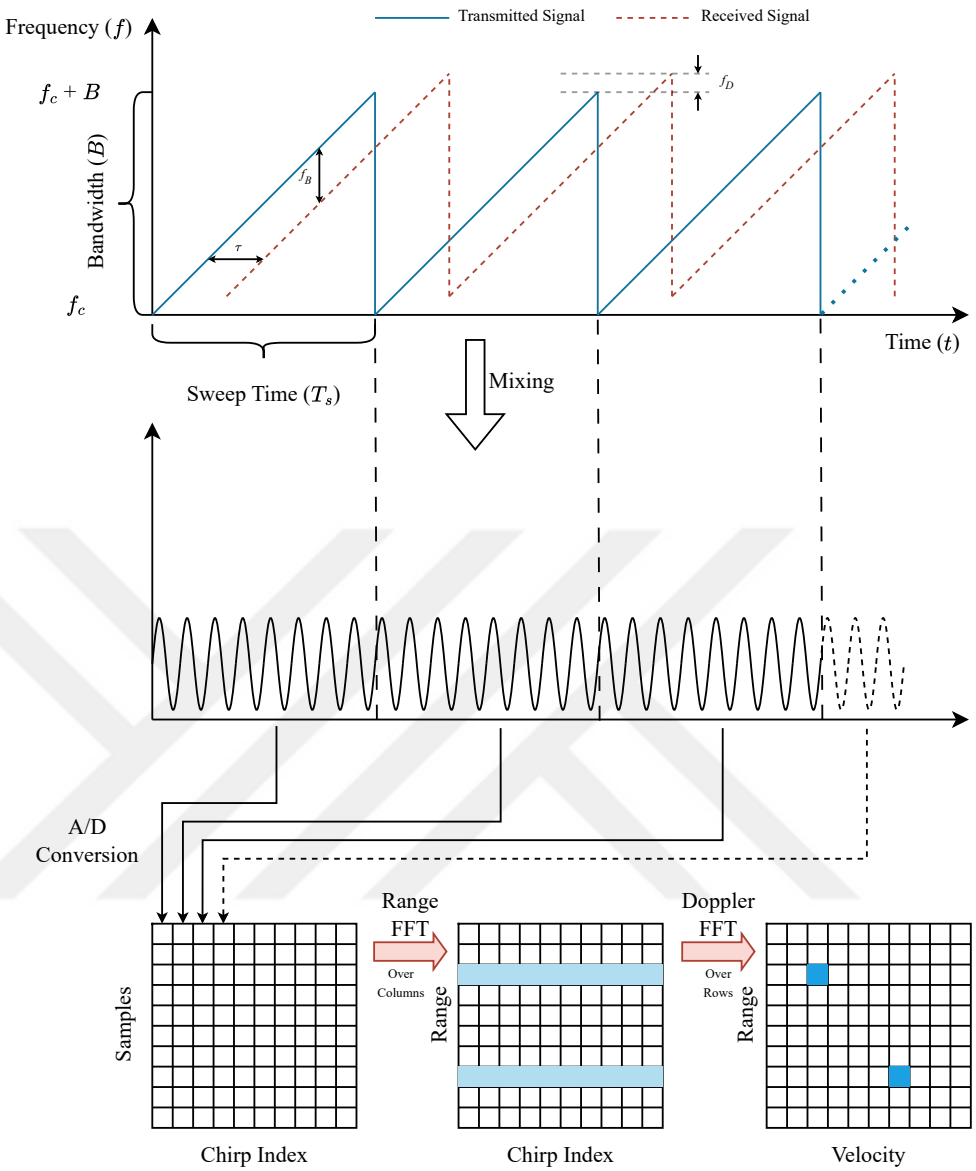


Figure 2.4: 2D FFT Process of the received signals.

received signals, the obtained samples are grouped into a two-dimensional matrix, with each column representing the samples of consecutive transmitted and received chirps. Utilizing the FFT across the fast time dimension (range FFT) provides the beat frequency, hence the range of the target, and utilizing the FFT across the slow time dimension (Doppler FFT) yields the Doppler frequency, which reflects the phase dif-

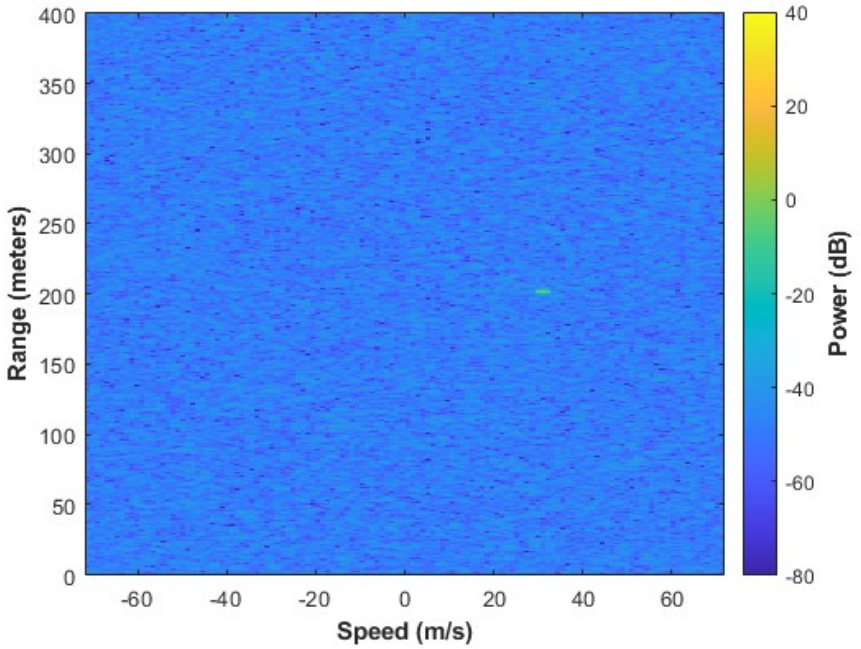


Figure 2.5: An example of a range-Doppler map with a target placed 200m away from the radar, moving at a velocity of 30m/s.

ference between consecutive chirps caused by the target's motion, thereby providing the target's velocity.

After the two-dimensional FFT process, the resulting output is referred to as a range-Doppler map. The range-Doppler map contains information about the range and velocity of the targets and can be further processed for detection purposes. Fig. 2.5 illustrates the range-Doppler map obtained after a two-dimensional FFT processing.

2.1.3.1 Range Estimation

Considering a target at a distance R with a zero relative velocity, the time delay will be expressed as

$$\tau = \frac{2R}{c}. \tag{2.20}$$

According to the previous equations, the beat frequency can be represented as

$$f_{B,R_0} = F_B = \alpha\tau = \frac{2\alpha R}{c}. \tag{2.21}$$

The IF signal contains a frequency component corresponding to the target. The distance can be calculated as

$$R = \frac{f_{B,R_0} c}{2\alpha}. \tag{2.22}$$

The first FFT across the fast-time samples will generate a peak at the range bin corresponding to the target distance. When there are multiple targets, the IF signal will contain several frequency components, each corresponding to a target. If two targets are very close to each other, the radar may put them into the same range bin causing them to appear as a single target. The radar's ability to distinguish between closely spaced targets depends on its range resolution. The range resolution of the radar is inversely proportional to the chirp duration $\Delta f_{B,R_0} = \frac{1}{T_s}$ and can be represented as

$$\Delta f_{B,R_0} = \frac{1}{T_s} = \frac{2\alpha\Delta R}{c} \Rightarrow \Delta R = \frac{c}{2B}. \tag{2.23}$$

As a result, the range resolution of the FMCW radar is directly related to the chirp bandwidth. To improve the range resolution, the bandwidth of the chirp must be increased.

Another important parameter of the FMCW radar is the maximum unambiguously detectable range denoted as R_{max}. According to the sampling theorem [35], if a continuous-time signal is sampled at a rate at least twice the maximum frequency present in the signal, it can be precisely reconstructed and fully represented in its samples. Therefore, the maximum range that radar can unambiguously detect is related to the sampling frequency f_s and can be expressed as [1]

$$f_{B,R_{max}} = \frac{f_s}{2} = \frac{2\alpha R_{max}}{c} \Rightarrow R_{max} = \frac{f_s c}{4\alpha}. \tag{2.24}$$

[1] This statement holds true when $f_s/2 < B$ so that f_s limits the maximum unambiguous range. When $f_s/2 > B$, B becomes the limiting factor and $R_{max} = \frac{Bc}{2\alpha}$.

2.1.3.2 Velocity Estimation

As previously discussed, multiple consecutive chirps may be utilized to calculate the target velocity. The target velocity induces a phase shift between consecutive chirps. By performing a second FFT across the slow-time samples, we obtain a peak that corresponds to both the velocity and range of the target. This second FFT is applied to the output of the first FFT. The target velocity can then be determined using the Doppler shift, which is given by

$$f_D = \frac{2f_c v}{c} \Rightarrow v = \frac{f_D c}{2f_c}. \tag{2.25}$$

The capability of the radar to distinguish between targets at the same distance but with different relative velocities depends on its Doppler frequency resolution. This resolution is influenced by the Doppler frequency changes over consecutive chirps and is determined by the total frame duration $M \cdot T_s$. The Doppler frequency resolution Δf_D is given by

$$\Delta f_D = \frac{1}{MT_s}. \tag{2.26}$$

Hence the velocity resolution is

$$\Delta v = \frac{c}{2f_c}\Delta f_D = \frac{c}{2f_c MT_s}. \tag{2.27}$$

Again, following the sampling theorem, the maximum unambiguously detectable velocity can be written as

$$f_{D,max} = \frac{1}{2T_s} = \frac{2f_c v_{max}}{c} \Rightarrow v_{max} = \frac{c}{4T_s f_c}. \tag{2.28}$$

2.2 Target Detection

One of the primary functions of the radar signal processor is detection ([31]). In radar, detection involves deciding if the measurement contains any echo signal from the target or if it is only the noise and clutter. The decision is very often made based on a threshold $\hat{T}$ to separate the target (if present) from the noise and clutter. Generally, the noise level is unknown and can change according to different environmental conditions, so the threshold should be estimated adaptively according to the changing noise level. The constant false alarm rate (CFAR) method is commonly used for adaptive threshold estimation. By continuously adjusting the threshold, CFAR keeps the false alarm rate constant. A generic CFAR detector utilizes a sliding data (reference) window to estimate the noise level around the cell under test (CUT). CUT is the current cell that the detector decides by comparing it with the threshold. The data window contains the noise level and its structure changes according to the CFAR type. Eventually, the calculated noise level is multiplied by a scaling factor and the threshold is obtained. Next, two CFAR algorithms, cell-averaging CFAR (CA-CFAR) and order statistic CFAR (OS-CFAR) will be examined.

2.2.1 CA-CFAR

The cell-averaging CFAR (CA-CFAR) detector calculates the threshold using the average noise power estimated from the reference window. The reference window in the case of CA-CFAR is constructed by guard cells and reference cells. Guard cells are placed around the CUT to ensure that any potential returns from targets do not influence the noise level estimate. Reference cells contain the noise levels and are used to estimate the background noise. Noise power estimation is performed as [31]

$$\hat{\sigma}_i^2 = \frac{1}{N} \sum_{i=1}^{N} x_i, \tag{2.29}$$

where N is the size of the reference cells and x_i holds the power value for each reference cell. The CA-CFAR threshold is estimated by multiplying the power estimate in the equation 2.29 with the CA-CFAR scale factor α_{CA} as

$$\hat{T}_{CA} = \alpha_{CA}\hat{\sigma}_i^2. \tag{2.30}$$

The scale factor α_{CA} may be calculated approximately according to the desired P_{fa} as

$$P_{\text{fa}} = \left[1 + \frac{\alpha_{CA}}{N}\right]^{-N}, \tag{2.31}$$

although more involved calculations are also possible. CA-CFAR is simple to implement and computationally efficient and works well in environments where noise and clutter are uniformly distributed. However, performance degrades in non-uniform scenarios. For example, targets are masked when multiple targets are present in the reference cell while another weak target is in the CUT increasing the estimated threshold and preventing the detection.

2.2.2 OS-CFAR

In this method, a sliding window, consisting of a CUT, guard cells (if desired), and reference cells are employed. The guard cells protect the CUT from contamination by strong signals, while the reference cells provide a local noise estimate. The signal strengths in the reference cells are ordered, and a specific order statistic, such as the k-th smallest value, is selected to represent the noise level. This chosen statistic is scaled by a factor to establish the detection threshold. Threshold is calculated according to [31] as

$$\hat{T}_{(\text{OS})} = \alpha_{\text{OS}} x_{(k)}, \tag{2.32}$$

where α_{OS} is the integer scaling factor, and $x_{(k)}$ is the selected k-th value. The scaling factor is numerically approximated according to the following equation [31]

$$P_{\text{fa}} = \frac{N!}{(N-k)!} \cdot \frac{(\alpha_{\text{OS}} + N - k)!}{(\alpha_{\text{OS}} + N)!}. \tag{2.33}$$

By comparing the CUT's signal strength against this threshold, targets are declared if the signal amplitude exceeds the threshold.

In comparison to CA-CFAR, OS-CFAR is more robust to outliers and performs better in heterogeneous environments, but comes with increased computational complexity and the need for careful selection of the order statistic.

2.3 Angle of Arrival Estimation

The Angle of Arrival (AoA) estimation process involves using an array of antennas to determine the direction of targets. Generally, the far-field assumption is made when a signal arrives from a distant target. The far-field region, also known as the Fraunhofer region [31], is the distance from the antenna array where the incoming wavefronts can be considered planar. This assumption is valid when the distance R_{far} from the target to the antenna array satisfies

$$R_{far} = \frac{2D^2}{\lambda}, \tag{2.34}$$

where D is the antenna array's largest dimension and λ is the wavelength of the signal[2]. One example of antenna array configuration is the uniform linear array (ULA), as shown in Fig. 2.6. In a ULA, multiple antenna elements are arranged in a straight line with equal spacing d between adjacent elements. The spacing d is usually a fraction of the signal wavelength λ, typically $d \leqslant \frac{\lambda}{2}$ to avoid spatial aliasing. When a plane wave arrives at an angle θ relative to the array axis, it reaches each antenna element at slightly different times, creating a phase difference between the signals received by adjacent elements. This phase difference $\Delta\phi$ is given by [31]

$$\Delta\phi = \frac{2\pi d \sin(\theta)}{\lambda}. \tag{2.35}$$

Considering the ULA configuration, rewriting Eqn. 2.18, we obtain discrete baseband signal as

[2] For the far-field assumption to be valid, the antenna array's largest dimension must be greater than the wavelength ($D > \lambda$) [5].

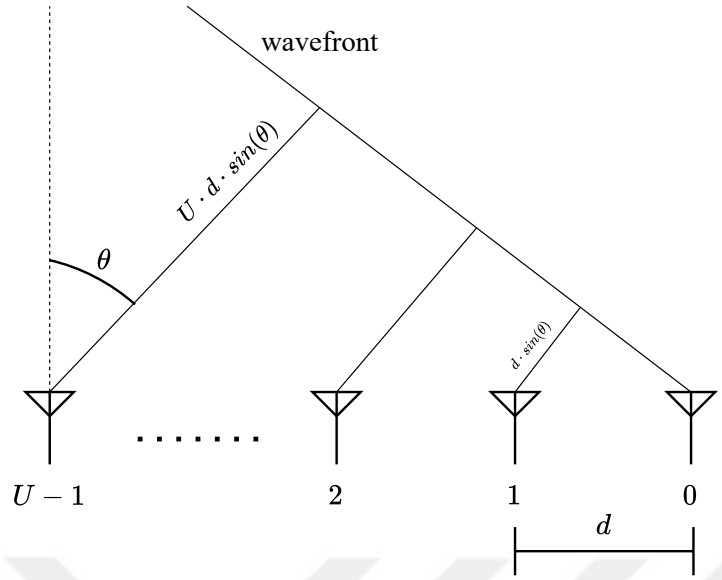


Figure 2.6: Representation of a wavefront arriving a uniform linear array with an angle θ.

$$x_b(k,m,u) = \frac{A_R}{2}\cos\left(2\pi(F_B\frac{k}{f_s} + f_D m T_s + f_\theta u + \phi_0)\right), \tag{2.36}$$

where $u = 0, 1, ..., U-1$ represents the index of antenna elements and $f_\theta = \frac{d\,sin(\theta)}{\lambda}$. As stated earlier, the DFT process can be applied to extract range and velocity information. Now, a third DFT process can extract the angle information as stated below

$$X_b(l,n,w) = \sum_{u=0}^{U-1}\sum_{m=0}^{M-1}\sum_{k=0}^{K-1} x_b(k,m,u)\cdot\exp\left(-j2\pi\frac{kl}{K}\right)\cdot\exp\left(-j2\pi\frac{mn}{M}\right) \tag{2.37}$$

$$\cdot\exp\left(-j2\pi\frac{uw}{U}\right), \tag{2.38}$$

where $w = -U/2, ..., U/2-1$ represents the normalized spatial frequency bins. Having multiple antenna elements will provide the AoA information of the target. However, when multiple targets are located at the same range with the same velocity, the radar's angular resolution determines whether those targets can be separated in space or not. If the angular resolution is high enough, the radar can distinguish between the

targets that are closely spaced in angle; otherwise, the targets will appear as a single object. The angular resolution of the antenna is determined by its 3-dB beamwidth, θ_3. The 3-dB beamwidth can be calculated according to [31] as

$$\theta_3 = 0.89 \cdot \frac{\lambda}{D}. \tag{2.39}$$

As a result, the angular resolution is related to the antenna size and operating frequency. Narrower beamwidth leads to better angular resolution, enabling the radar to resolve finer details and closely spaced targets.

2.4 FMCW Mutual Interference

In the area of autonomous driving, radar utilization stands as a cornerstone technology, having a significant role in shaping the future of transportation. As vehicles evolve towards greater autonomy, the need for robust and reliable sensing systems becomes increasingly important. FMCW radar, with its suitability for use in road scenarios, emerges as a crucial component in providing vehicles with a comprehensive awareness of their surroundings. However, the growing quantity of automobiles with radar systems raises concerns about the potential for interference between these devices, which can compromise the accuracy and reliability of each system [3].

To better understand the importance of the interference problem, consider a victim radar sensing an environment with the presence of an interfering radar in terms of received signal powers. The received signal powers can be written as

$$P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R_0^4}, P_i = \frac{P_{ti} G_{ti} G_r \lambda_i^2}{(4\pi)^2 R_i^2}, \tag{2.40}$$

where P_t is the transmit power, G_t is the transmitting antenna gain, G_r is the receiving antenna gain of the victim radar and $\lambda = c/f_c$ is the wavelength of the transmitted signal. In addition, P_{ti} is the transmit power, G_{ti} is the transmitting antenna gain of the interfering radar and $\lambda_i = c/f_{ci}$ is the wavelength of the transmitted interference signal. σ is the target's radar cross section (RCS). Lastly, R_0 is the distance between

the target and the victim radar and R_i is the distance between the interfering and the victim radars. The received signal power from a target decreases proportionally to the fourth power of the distance between the radar and the target because of the two-way propagation. However, in the presence of interference from another FMCW radar, the interfering signal power only decreases proportionally to the second power of the distance between the interferer radar and the victim radar because of the one-way propagation. Given the same transmit power for both radars, this significant power difference can lead to various issues including actual return signals from targets being buried under interference resulting in the prevention of target detection or the victim radar erroneously declaring so-called ghost target detections. Based on the above considerations, interference mitigation is an important problem. In the next subsection, FMCW interference signal analysis will be carried out to understand the issues mentioned previously.

2.4.1 FMCW Interference Signal Analysis

Considering the interferer radar transmitting an FMCW signal with a sawtooth modulation, the transmit frequency of the interfering signal is

$$f_i(t) = f_{ci} + \frac{B_i}{T_{si}}t = f_{ci} + \alpha_i t, \tag{2.41}$$

where $\alpha_i = \frac{B_i}{T_{si}}$ represents the slope and f_{ci} represents the starting frequency of the interfering signal. The instantaneous phase of the interfering signal can be written as

$$\phi_i(t) = 2\pi \int_0^t f_i(t)dt = 2\pi \int_0^t (f_{ci} + \alpha_i t)dt = 2\pi \left(f_{ci}t + \alpha_i \frac{t^2}{2} \right). \tag{2.42}$$

The received interference signal at the victim radar can be written as

$$x_{ri}(t) = A_{RI} \cos \left[2\pi \left(f_{ci}(t - \tau_i(t)) + \frac{\alpha_i (t - \tau_i(t))^2}{2} \right) \right], \tag{2.43}$$

where A_{RI} is the amplitude of the received interference signal and $\tau_i(t)$ is the time delay between the victim and the interferer radar. Eqn. 2.43 is a signal model that

can represent both coherent interference and non-coherent interference. Considering a single transmitted interference chirp and a single victim antenna receive element scenario, the mixed and low passed interferer signal at the victim radar will be

$$x_{bi}(t) = \text{LPF}\Bigg\{ \frac{A_{RI}}{2} \cos\Bigg[2\pi \Bigg((f_c - f_{ci})t + (\frac{\alpha - \alpha_i}{2})t^2 + f_{ci}\tau_i(t) + \alpha_i \tau_i(t) t - \frac{\alpha_i \tau_i(t)^2}{2} \Bigg) \Bigg] \Bigg\}. \tag{2.44}$$

It is important to denote the low pass filter notation because, in the presence of a non-coherent interference, some parts of the interference signal are filtered out. Now, considering that the interferer radar transmits M chirps in a single measurement frame, the time delay between the interferer radar and the victim radar will be

$$\tau_i(t, m) = \frac{2\Big(R_i + v_i(t + mT_{RIint})\Big)}{c}, \tag{2.45}$$

where v_i is the relative velocity of the interferer radar to the victim radar and T_{RIint} represents the chirp repetition interval of the interferer radar. Substituting Eqn. 2.45 into Eqn. 2.44, and making necessary simplifications, the baseband interference signal at the victim radar can be represented as

$$x_{bi}(t, m) = \text{LPF}\Bigg\{ \frac{A_{RI}}{2} \cos\Bigg[2\pi \Bigg((f_c - f_{ci})t + \frac{2f_{ci}v_i}{c}t + \frac{2f_{ci}mv_i}{c}T_{RIint} + \frac{2\alpha_i R_i}{c}t + (\frac{\alpha - \alpha_i}{2})t^2 \Bigg) + \frac{4\pi f_{ci} R_i}{c} \Bigg] \Bigg\}. \tag{2.46}$$

Considering the victim radar with a ULA configuration, the rearranged discrete baseband interference signal at the victim radar will be

$$x_{bi}(k,m,u) = \text{LPF}\Bigg\{\frac{A_{RI}}{2}\cos\Bigg[2\pi\Bigg(F_{B_i}\frac{k}{f_s} + f_{Di}mT_{RIint} + f_{\theta_i}u \\ +(f_c - f_{ci})t + (\frac{\alpha-\alpha_i}{2}t^2) + \phi_i\Bigg)\Bigg]\Bigg\}, \tag{2.47}$$

where

$$f_{Di} = \frac{2f_{ci}v_i}{c}, \quad f_{B_i,R_i} = \frac{2\alpha_i R_i}{c}, \quad f_{\theta_i} = \frac{dsin(\theta_i)}{\lambda_i}, \quad F_{B_i} = f_{Di} + f_{B_i,R_i}. \tag{2.48}$$

As a result, the combined discrete baseband target return and interference signal at the victim radar can be written as

$$\begin{aligned} x_s =& x_b + x_{bi} \\ x_s =& \frac{A_R}{2}\cos\left[2\pi\left(F_B\frac{k}{f_s} + f_D mT_s + f_\theta u + \phi_0\right)\right] \\ &+ \text{LPF}\Bigg\{\frac{A_{RI}}{2}\cos\Bigg[2\pi\Bigg(F_{B_i}\frac{k}{f_s} + f_{Di}mT_{RIint} + f_{\theta_i}u \\ &+(f_c - f_{ci})t + (\frac{\alpha-\alpha_i}{2}t^2) + \phi_i\Bigg)\Bigg]\Bigg\}. \end{aligned} \tag{2.49}$$

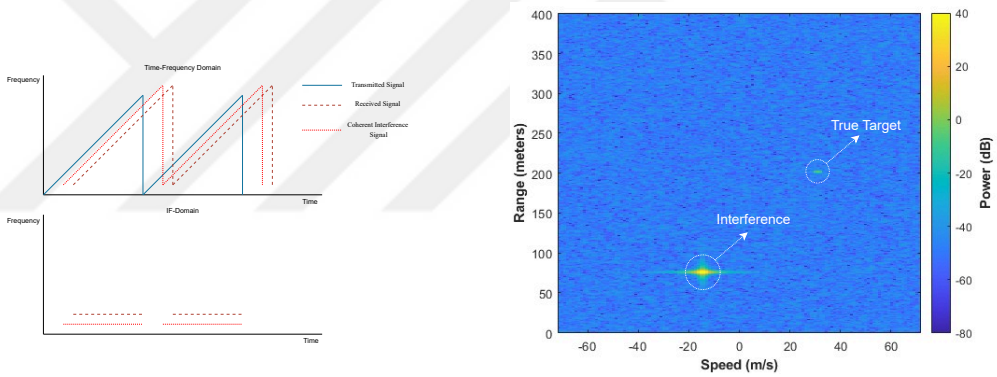
The effects of the interfering signal on range-Doppler maps can vary according to signal parameters such as bandwidth B_i, sweep time T_{si}, starting frequency f_{ci} etc.

Interference effects are analyzed in [7], where it is stated that there are two types of effects. These interference effects can occur due to either identical system parameters or similar system parameters. In the following subsections, the effects of different types of interference will be examined.

2.4.2 Identical System Parameters

Coherent interference happens when the victim and the interferer radars utilize the same system parameters, including operating frequency, sweep time, and bandwidth.

After mixing the target return and interfering signals with the transmitted signal and applying 2D FFT, the interferer signal appears in the range-Doppler map as if it is an object. However, this object-like image of the interfering signal appears at a distance and velocity that are half of the actual interferer radar's distance and velocity. This is because the victim radar processes the interfering signal as if it has been two-way propagated. Fig. 2.7 shows a representation of FMCW radar signals in the time-frequency domain and a range-Doppler map. Fig. 2.7a illustrates the frequency of transmit, receive and coherent interference signals. Since both the interferer and the victim radar utilize the same parameters, the interference signal acts as a target return signal in the IF domain. Fig. 2.7b shows an example range-Doppler map displaying the effect of coherent interference. The interferer signal appears as a ghost target in the range-Doppler map, although there is no such target at that position. The received target return and the interference signal are quantized with high bit precision in the figure.



(a) Representation of the transmit, receive and coherent interference signal in the time-frequency domain.

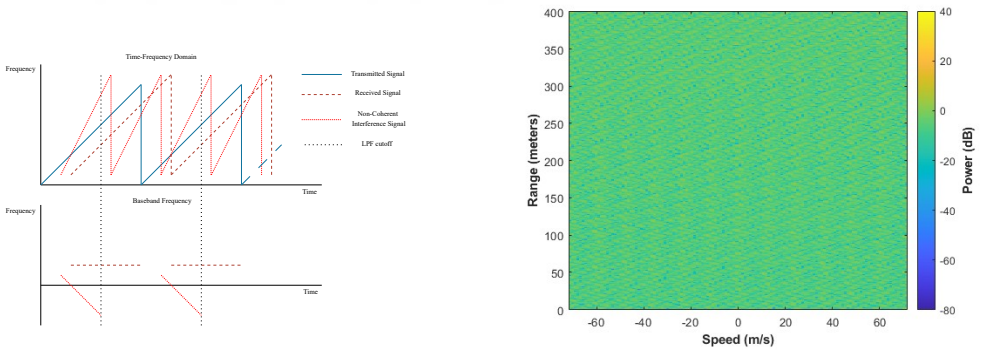
(b) Range-Doppler map example displaying the coherent interference effect.

Figure 2.7: An example representation of the coherent interference in time-frequency domain (a) and range-Doppler map (b).

2.4.3 Similar System Parameters

Non-coherent interference happens when the victim and interferer radars utilize similar system parameters. Unlike coherent interference, the interferer signal crosses

the IF bandwidth of the victim radar at different times on each chirp, resulting in the interference energy spreading across all range-Doppler bins. This type of interference causes a noise-like effect and increases the noise floor. Fig. 2.8 shows a representation of FMCW radar signals in the time-frequency domain and in the form of a range-Doppler map. Fig. 2.8a illustrates the frequency of transmit, receive and non-coherent interference signals. Since the interferer and the victim radar utilize different parameters, the interferer radar signal crosses the victim radar signal at different times. As a result, the interference signal energy spreads across both range and Doppler bins. The black dotted lines indicated the low pass filter cutoff instants. The non-coherent interference signal affects the victim radar only when its frequency falls below this cutoff frequency. Fig. 2.8b shows an example range-Doppler map displaying the effect of non-coherent interference. Since the energy is spread across the bins, the interference increases the noise floor, thereby reducing the visibility of the target which is located 200m away from the victim radar and moving at a velocity of 30m/s with a radar cross-section of 10 dBsm. The received target return and the interference signal are quantized with high bit precision in the figure.



(a) Representation of the transmit, receive and non-coherent interference signal in the time-frequency domain.

(b) Range-Doppler map example displaying the non-coherent interference effect.

Figure 2.8: An example representation of the non-coherent interference in time-frequency domain (a) and range-Doppler map (b).

2.5 One-Bit Quantization

Radar is crucial in autonomous driving, offering unparalleled perception in challenging conditions. Unlike optical sensors, radar's radio wave operation effectively penetrates fog, rain, and darkness. It excels in detecting objects at longer ranges, providing an early-warning system for proactive hazard response. The adoption of radar technology in autonomous driving brings advantages but also limitations. The use of high-precision ADCs in radar systems with multiple antennas increases production costs, posing economic challenges. Moreover, the increased precision of these radars often requires higher power consumption, raising concerns about energy efficiency compared to low-resolution radars [21, 2].

With simplified designs and reduced data resolution, radars that utilize one-bit quantization offer a potential solution to challenges such as high production costs and power consumption associated with traditional high-precision radars [30]. Despite the benefits, the non-linearity of one-bit sampling introduces new challenges for conventional linear signal processing methods. Since one-bit quantization is just a sign comparator, samples are represented as +1 or -1. In a multiple-target scenario, after linear processing, self-generated and cross-generated harmonics appear as ghost targets in range-Doppler maps and may be detected as real targets [30], [17]. The number of harmonics depends on the number of return signals and their amplitudes.

Considering the multiple chirp scenario in a ULA configuration, the received signals before ADCs can be represented as

$$x_b(t,m,u) = \sum_{p=1}^{P} \frac{A_R}{2} \cos\left(2\pi(f_{D,p} m T_s + f_{B,R_0,p} t + f_{\theta,p} u + \phi_0)\right) + w(t), \quad (2.50)$$

where $w(t)$ is the additive white Gaussian noise (AWGN), and P is the number of targets. Now, considering the one-bit quantizers being just a sign comparator, [13] expresses the sign function as

$$\text{sign}(u+n) = \frac{\text{j}}{\pi} \int_{-\infty}^{\infty} \frac{\exp(\text{j}(u+n)\xi)}{\xi} d\xi. \quad (2.51)$$

Let $\omega_p(t) = 2\pi(f_{D,p}mT_s + f_{B,R_0,p}t + f_{\theta,p}u + \phi_0)$. Substituting Eqn. 2.51 to the Eqn. 2.50, the one-bit signal $v(t)$ can be written as

$$\begin{aligned} v(t) &= \text{sign}\left(\sum_{p=1}^{p} \frac{A_R}{2}\cos(\omega_p(t)) + w(t)\right) \\ &= \frac{\text{j}}{\pi}\int_{-\infty}^{\infty} \frac{\exp(\text{j}w\xi)}{\xi}\left(\prod_{p=1}^{P}\exp\left[\text{j}\frac{A_R}{2}cos(\omega_p(t)\xi)\right]\right)d\xi \\ &\overset{(a)}{=} \frac{\text{j}}{\pi}\int_{-\infty}^{\infty} \frac{\exp(\text{j}w\xi)}{\xi}\prod_{p=1}^{P}\left(\sum_{m_p=0}^{\infty}\epsilon_{m_p}\text{j}^{m_p}J_{m_p}(\frac{A_R}{2}\xi)\times\cos(m_p w_p(t))\right)d\xi, \end{aligned} \tag{2.52}$$

where J_{m_p} is the Bessel function of the first kind, equality (a) follows from [27, p. 361] and $\epsilon_0 = 1$ and $\epsilon_m = 2$ for $m \geq 1$. Eqn. 2.52 shows that the one-bit signal $v(t)$ contains its fundamental harmonics as well as new high-order harmonics (infinite-order harmonics). As mentioned, the high-order harmonics include both self-generated terms from the original signal and cross-generated terms from different signals. As explained earlier, the FFT process is applied to the digitized signal. The FFT process can be considered as an averaging operation over the noise [17], and the amplitudes of both the fundamental harmonics and high-order harmonics can be approximated by their average values after the FFT [13]. Following the notation given by [17], the average value of the one-bit signal $v(t)$ can be represented as

$$\begin{aligned} \text{E}\left[v(t)\right] &= \frac{\text{j}}{\pi}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi\sigma_w^2}}e^{-\frac{w^2}{2\sigma_w^2}}\int_{-\infty}^{\infty}\frac{e^{\text{j}w\xi}}{\xi}\prod_{p=1}^{P}\left(\sum_{m_p=0}^{\infty}\epsilon_{m_p}\text{j}^{m_p}J_{m_p}(A_p\xi)\right. \\ &\quad \left.\times\cos(m_p\omega_p(t))\,dw\,d\xi\right. \\ &= \frac{\text{j}}{\pi}\int_{-\infty}^{\infty}\frac{e^{-\frac{\sigma_w^2\xi^2}{2}}}{\xi}\prod_{p=1}^{P}\left(\sum_{m_p=0}^{\infty}\epsilon_{m_p}\text{j}^{m_p}J_{m_p}(A_p\xi)\times\cos(m_p\omega_p(t))\right)d\xi, \end{aligned} \tag{2.53}$$

where the expectation $E[\cdot]$ is computed with respect to the noise w, and σ_w^2 represents

the noise power. Eqn. 2.53 shows that $\mathrm{E}[v(t)]$ is equal to zero for even m, indicating that the self-generated and cross-generated harmonics' average amplitudes are zero. Considering a single target case for odd m values, $\mathrm{E}[v(t)]$ only contains self-generated harmonics and given by [13] as

$$\mathrm{E}[v(t)] = \sum_{m=1,modd}^{\infty} c_m \cos(m\omega(t)), \tag{2.54}$$

where

$$c_m = \mathrm{j}^{m+1}\sqrt{\frac{2}{\pi}}\alpha_m \left(\frac{A_1}{\sigma_w}\right)^m F\left(\frac{m}{2}; m+1; -\frac{A_1^2}{2\sigma_w^2}\right), \tag{2.55}$$

where $F(\cdot,\cdot,\cdot)$ is the hypergeometric function [27, p. 504] and

$$\alpha_m = \frac{1}{\left(\frac{m-1}{2}\right)!m}2^{-\frac{3(m-1)}{2}}, \quad m \text{ is odd.} \tag{2.56}$$

Note that $\alpha_1 = 1$, $\alpha_3 = \frac{1}{24}$, and $\alpha_5 = \frac{1}{640}$, meaning that the amplitudes of the harmonics decrease rapidly as the order increases. If the received SNR $= \frac{A_1^2}{2\sigma_w^2} \gg 1$, the amplitude information is completely lost [13], which is generally the case in the presence of an interferer signal. Considering a two-target case for simplicity, $\mathrm{E}[v(t)]$ contains both the self-generated and the cross-generated harmonic terms. For m-th order self-generated harmonics the coefficient $c_{m,0}$ can be calculated as [17]

$$\begin{aligned} c_{m,0} &= \frac{2\mathrm{j}^{m+1}}{\pi}\int_{-\infty}^{\infty}\frac{\exp\left(-\frac{\sigma_w^2\xi^2}{2}\right)}{\xi}J_0(A_2\xi)J_m(A_1\xi)\,d\xi \\ &= \frac{4\mathrm{j}^{m+1}}{\pi}\frac{A_1^m\left(\frac{\sigma_w^2}{2}\right)^{-\frac{m}{2}}}{2^{m+1}\Gamma(m+1)}\sum_{i=0}^{\infty}\frac{\Gamma(i+\frac{m}{2})}{i!\Gamma(i+1)}\left(-\frac{A_2^2}{2\sigma_w^2}\right)^i \\ &\times F\left(-i,-i;m+1;\frac{A_1^2}{A_2^2}\right). \end{aligned} \tag{2.57}$$

For the m-th order cross-generated term $c_{m_1,m_2}\cos(m_2\omega_2(t))\cos(m_1\omega_1(t))$ where $m = m_1 + m_2$, the coefficient c_{m_1,m_2} can be calculated as

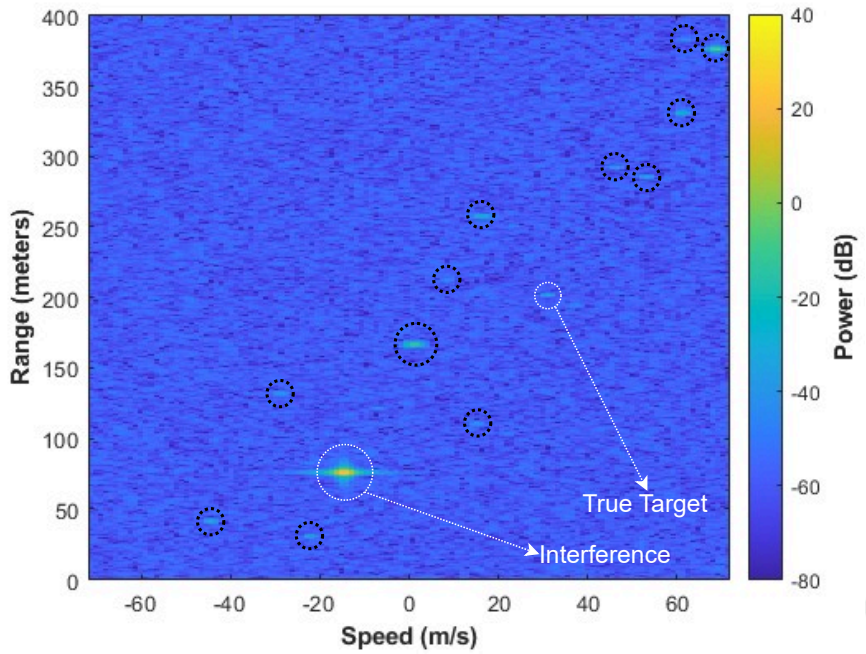


Figure 2.9: An example of a range-Doppler map with a target placed 200m away from the victim radar, moving at a velocity of 30m/s, and with an interferer radar 150m away from the victim radar, moving at a velocity of -30m/s.

$$
\begin{aligned}
c_{m_1,m_2} &= \frac{4\mathrm{j}^{m+1}}{\pi} \int_{-\infty}^{\infty} \frac{\exp\left(-\frac{\sigma_w^2 \xi^2}{2}\right)}{\xi} J_{m_2}(A_2\xi) J_{m_1}(A_1\xi)\, d\xi \\
&= \frac{8\mathrm{j}^{m+1}}{\pi} \frac{A_1^{m_1} A_2^{m_2} \left(\frac{\sigma_w^2}{2}\right)^{-\frac{m}{2}}}{2^{m+1}\Gamma(m_1+1)} \sum_{i=0}^{\infty} \frac{\Gamma(i+\frac{m_2}{2})}{i!\Gamma(i+\frac{m_2}{2}+1)} \left(-\frac{A_2^2}{2\sigma_w^2}\right)^i \\
&\times F\left(-i, -m_2 - i; m_1+1; \frac{A_1^2}{A_2^2}\right).
\end{aligned}
\tag{2.58}
$$

As a result, the provided calculations can determine the amplitudes and frequencies of the self-generated and cross-generated harmonics. However, it is important to note that these calculations will not be used further, as our solution relies on machine learning methods. Some work in the literature [17],[16] analyze the impact of harmonics on target detection and propose iterative approaches to address harmonic issues. In this thesis, we aim to demonstrate that a neural network, when trained on an appropriate dataset, can effectively solve the harmonic problem.

Fig. 2.9 shows a range-Doppler map containing a target and the effect of coherent

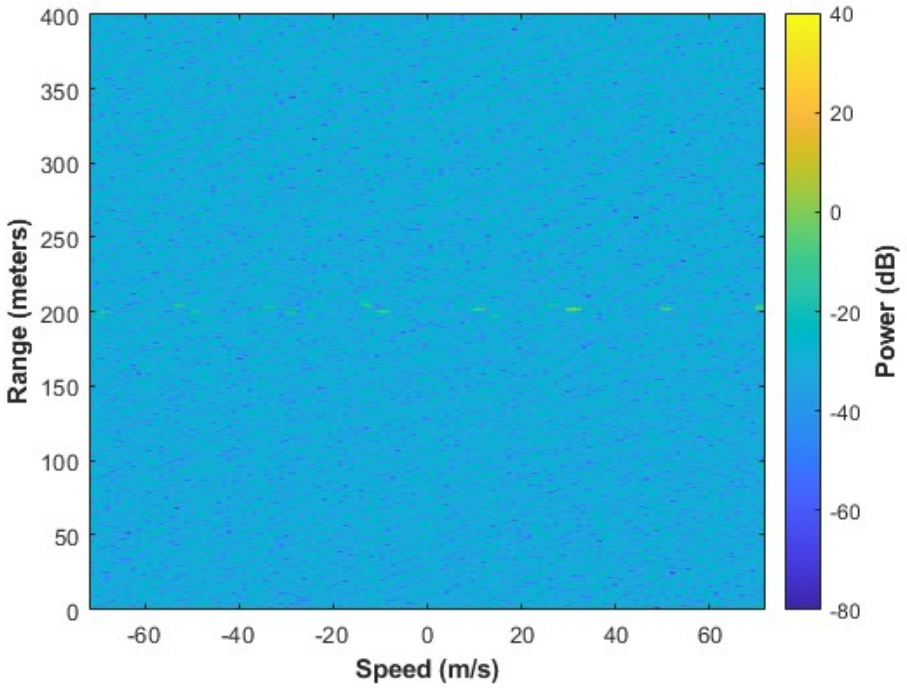


Figure 2.10: An example of a range-Doppler map with a target placed 200m away from the victim radar, moving at a velocity of 30m/s, and with an interferer radar 150m away from the victim radar, moving at a velocity of -30m/s.

interference. In this figure, both the received target return and the interference signal are one-bit quantized. As in Fig. 2.7b, the interference itself still appears in the map. Additionally, due to the one-bit quantization, self-generated and cross-generated harmonics also appear in the range-Doppler map as ghost targets. The signal power of the interference signal and the target return signal determines the amplitudes and numbers of these ghost targets. As the power of the received signals increases, the amplitude of the harmonics also increases, resulting in more harmonics appearing in the range-Doppler map. Ghost targets are represented with black circles in the figure.

Fig. 2.10 shows a range-Doppler map containing a target and the effect of non-coherent interference. In this figure, the received target return and the interference signal are one-bit quantized. Unlike Fig. 2.8b, since signals are one-bit quantized, the high power of the interference signal is limited and the increase in the noise floor is also constrained. However, before quantization, some parts of the weak target return signal are already affected by the high-power interference signal. After quantization,

the information about the target in the affected parts of the signal is completely lost. The extent of this loss depends on the radar system parameters. In the figure, due to the one-bit quantization, the self-generated harmonics of the target return signal are still visible in the range-Doppler map as ghost targets.

CHAPTER 3

MACHINE LEARNING

The purpose of the artificial intelligence (AI) field of machine learning is to create models and algorithms that allow computers to learn from data and forecast or make decisions based on future data without needing to be explicitly programmed. The key idea behind machine learning is to use statistical techniques to let a system gradually get better at a certain task, based on the patterns and information present in the data it is exposed to [25]. The machine learning field can be divided into several main approaches: supervised learning, unsupervised learning, and reinforcement learning.

- **Supervised Learning:** The algorithm is trained on a labeled dataset, where the input data is paired with the corresponding desired output. The model learns a mapping function from the input data to the output data.

- **Unsupervised Learning:** The algorithm is given unlabeled data and tasked with finding patterns or structures without explicit guidance on what to look for.

- **Reinforcement Learning:** Through interaction with the environment and feedback in the form of rewards or punishments, the algorithm acquires knowledge. The algorithm's goal is to eventually figure out a policy that maximizes accumulated rewards.

In this work, supervised learning is utilized for the problem and will be examined in the next subsection.

3.1 Supervised Learning

In the supervised learning case, the computer is fed with labeled inputs x_i along with their corresponding known output pairs y_i. The purpose of supervised learning is to learn a generalized mapping (a function) from inputs to outputs [4]. This mapping is then used to predict outputs from unseen data. The supervised learning problem is categorized into two parts.

- **Classification:** If the output y_i is a categorical variable within a set $(y_i \in \{1, ..., C\})$, where C is the number of classes, then the learning problem is called classification or pattern recognition [25]. An example can be given as classifying the handwritten digits.
- **Regression:** If the output y_i takes any real-valued number ($y_i \in \mathbb{R}$), the learning problem is called regression. An example can be given as predicting tomorrow's temperature at a certain location providing some previous information.

3.2 Artificial Neural Networks

The idea of artificial neural networks (ANNs) can be traced back to the work of Warren McCulloch and Walter Pitts in the 1940s. They introduced the first mathematical model of a neuron, laying the groundwork for the development of neural networks. After the foundational work by McCulloch and Pitts, Frank Rosenblatt made significant contributions to the field of neural networks, particularly with his development of the perceptron model in the late 1950s. Building upon the concepts of McCulloch and Pitts, Rosenblatt introduced the perceptron as a computational model of a single-layer neural network. Rosenblatt's perceptron consisted of an input layer connected directly to an output layer, with each connection assigned a weight representing the strength of the connection.

Fig. 3.1 shows the basic architecture of the perceptron. $\boldsymbol{x} = [x_1, x_2, ..., x_d]^{\mathrm{T}}$ is the input vector, where d represents the number of input values, $\boldsymbol{w} = [w_1, w_2, ..., w_d]^{\mathrm{T}}$ is the weight vector, b is the bias term, ϕ is the activation function and $\hat{y}$ is the output value. The perceptron operates by multiplying the input values with their corre-

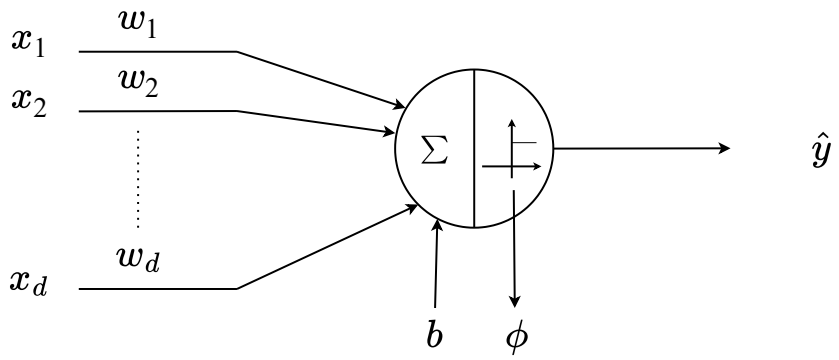


Figure 3.1: Architecture of the perceptron.

sponding weights, representing their importance in the decision-making process. The perceptron then sums up these weighted inputs, including an additional bias term, to compute a weighted sum. Next, the activation function adds non-linearity to the perceptron's output by passing the weighted sum through it. This activation function helps the perceptron make more complex decisions. The operation can be shown mathematically as follows [1]

$$\hat{y} = \phi\left(\sum_{i=1}^{d} w_i x_i + b\right). \tag{3.1}$$

We can also represent the operation in matrix form as

$$\hat{y} = \phi(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b). \tag{3.2}$$

The perceptron algorithm performs well on datasets that are linearly separable due to its simple structure [1]. However, a single artificial neuron performs poorly when the dataset is not linearly separable. To handle more complex datasets with multiple boundaries, more sophisticated networks, such as multilayer networks, are required.

3.2.1 Multilayer Perceptrons

Multilayer perceptrons (MLPs) extend the capabilities of single-layer perceptrons by incorporating multiple layers of neurons. While single-layer perceptrons can only learn linear decision boundaries, MLPs can learn and represent non-linear decision boundaries through the composition of multiple layers (input, hidden, and output) and the use of non-linear activation functions. This allows MLPs to capture more complex patterns and relationships in data, making them suitable for a wider range of tasks. Additionally, MLPs can learn hierarchical representations of features in data, with each layer extracting increasingly abstract and meaningful features from the input. In contrast, single-layer perceptrons are limited in their ability to learn hierarchical representations and can only capture simple linear relationships in data.

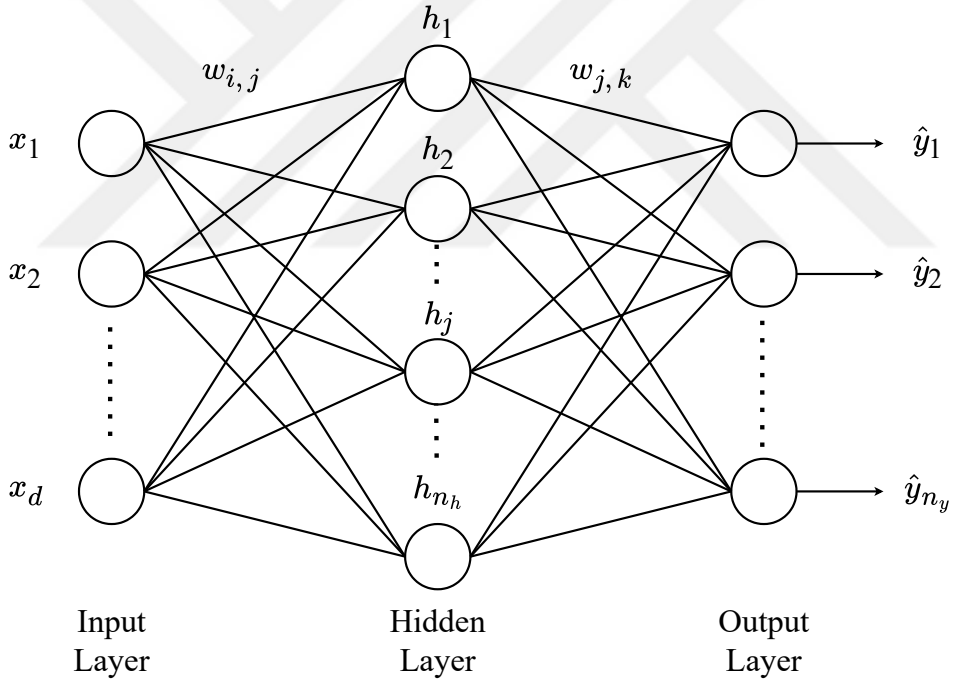


Figure 3.2: Architecture of the multilayer perceptron.

Unlike a single-layer perceptron, an MLP, as shown in Fig. 3.2 contains hidden layers between the input layer and the output layer. Therefore, the weights of the hidden layers should also be included in the operation representation. Let the weights of

connections between the input layer and the hidden layer be represented as matrix $\boldsymbol{W}_{\text{xh}}$ with a size of $d \times n_h$. The output vector, $\boldsymbol{h} = [h_1, h_2, ..., h_{n_h}]^{\text{T}}$, of the hidden layer can be represented as [1]

$$\boldsymbol{h} = \Phi_h(\boldsymbol{W}_{\text{xh}}^{\text{T}}\boldsymbol{x}). \tag{3.3}$$

Considering the output layer with n_y nodes, the weights of connections between the hidden layer and the output layer can be represented by the matrix $\boldsymbol{W}_{\text{hy}}$ with a size of $n_h \times n_y$. The operation from the hidden layer to the output layer can be shown as

$$\hat{\boldsymbol{y}} = \Phi_o(\boldsymbol{W}_{\text{hy}}^{\text{T}}\boldsymbol{h}), \tag{3.4}$$

where $\hat{\boldsymbol{y}} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_{n_y}]$ is the output vector. The activation function Φ is applied element-wise to the values. Up to this point, the derivations provided are part of the forward propagation operation of neural networks. During forward propagation, the purpose is twofold: first, to obtain outputs that allow the network to be trained by calculating the error or loss, and second, after the network is trained, to use the forward propagation process to provide useful predictions based on new input data. Next, we will examine how networks are trained using the calculated loss value, with backpropagation being the backbone of the training process.

3.3 Training the Artificial Neural Networks

Training an ANN involves several sequential steps to enable the network to learn from data and improve its predictive capability. Initially, the network is initialized with random weights and biases, which are essential for breaking symmetry and starting the learning process. The network receives input data during forward propagation, and layer-by-layer calculations are carried out. To produce its output, each neuron adds a bias, calculates the weighted total of its inputs, and uses an activation function. The input for the following layer is the output from the previous layer, propagating through until the final output layer produces a prediction. Subsequently, a loss function (see Section 3.3.2) measures the difference between the predicted output and

the actual target value. Backpropagation follows, where gradients of the loss function with respect to each parameter are computed recursively using the chain rule of calculus. These gradients guide the update of weights and biases through an optimization algorithm like Gradient Descent [40], adjusting them to minimize the loss function. This iterative process is repeated over multiple epochs, with each epoch involving the forward and backward propagation steps on mini-batches of data to enhance efficiency. Periodically, the model's performance is evaluated on a validation set to monitor for overfitting and optimize hyperparameters accordingly. Finally, the trained model is tested on a separate test set to assess its generalization ability before deployment in real-world applications, where it makes predictions on new, unseen data.

3.3.1 Underfitting and Overfitting

Since the purpose of the learning algorithm is to learn a generalized function, meaning it should perform well on new, unseen data, it must extract meaningful patterns and make accurate predictions beyond its training set. In this context, generalization refers to a model's ability to apply what it has learned from the training data to new, unseen data, thereby making accurate predictions in real-world scenarios. To evaluate and improve generalization, data is typically split into three sets: training, validation, and test datasets. The training dataset is used to train the model, the validation dataset is used to tune hyperparameters, and the test dataset assesses the model's performance on entirely unseen data. However, two common pitfalls can prevent achieving this goal: underfitting and overfitting.

Large errors are produced on both the training and validation datasets when a model is underfitted, which happens when a model is too simplistic to identify the underlying patterns in the training data. This can be caused by insufficient model complexity, poor feature selection, or inadequate training time. On the other hand, overfitting occurs when a model grows too complex and picks up on both the noise and the real patterns in the training set. As a result, the model performs poorly on test and validation data but well on training data, indicating a failure to generalize.

Mitigation techniques for underfitting include increasing model complexity (e.g.,

adding more layers or neurons in a neural network), improving feature selection and engineering, and training the model for a longer period or adjusting the learning rate. Overfitting can be mitigated by reducing model complexity, using regularization techniques, collecting more data, and implementing early stopping. Achieving a balance between underfitting and overfitting is essential for creating models that generalize well and perform effectively on unseen data.

3.3.2 Loss Function

A loss function, or cost function, measures how well a neural network's predictions match the actual target values, guiding the learning process. It quantifies the difference between predicted and true outputs, and minimizing it during training improves the network's accuracy. By minimizing the loss function, the network learns to make accurate predictions on both training and unseen data. Different loss functions should be used according to the problem at hand. Some loss function examples are as follow [28],

- **Mean Squared Error (ℓ_2 loss)**: For continuous parameters (regression problems), the mean squared error (MSE) loss function is commonly used.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3.5}$$

  where y_i is the target value and $\hat{y}_i$ is the estimated output.

- **Mean Absolute Error (ℓ_1 loss)**: MSE is sensitive to outliers in the data. On the other hand, mean Absolute Error (MAE) is preferred over MSE when robustness to outliers and ease of interpretation are important. Unlike MSE, which squares errors and disproportionately penalizes larger errors, MAE treats all errors equally by using absolute differences. This makes MAE more balanced and less sensitive to outliers.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{3.6}$$

- **Hinge Loss**: If the task is a hard classification problem, Hinge loss is the most common loss function. Hinge loss can be defined for the case where $y_i \in$

$\{-1, 1\}$ as

$$\text{Hinge Loss} = \frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i\hat{y}_i). \tag{3.7}$$

- **Cross Entropy**: When the estimation of the probabilities of each class is preferred over hard classification, logistic loss functions are used. In this case, cross-entropy measures the difference between the true probability distribution and the estimated distribution.

$$\text{Cross-Entropy} = -\sum_{i=1}^{n}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}), \tag{3.8}$$

  where $y_{i,c}$ is the true label for sample i and class c, $\hat{y}_{i,c}$ is the predicted probability that sample i belongs to class c, C is the number of classes, and n is the number of samples.

3.3.3 Backpropagation

Backpropagation is a fundamental algorithm for training artificial neural networks by optimizing their weights. The goal is to minimize the difference between the predicted outputs and the actual targets, typically measured by a loss function. Backpropagation works by propagating the error backward through the network, and adjusting weights in order to reduce the overall error.

Following the derivations of [11], a model with only one hidden layer is considered. The process starts by calculating the loss value. Consider the squared error loss

$$J(\mathbf{w}) = \sum_{k=1}^{n}(y_k - \hat{y}_k)^2. \tag{3.9}$$

The weights of the network are initialized with random values and changed in a way that will reduce the error. The backpropagation algorithm that changes those weights is based on gradient descent. The gradient is

$$\Delta\mathbf{w} = -\frac{\partial J}{\partial \mathbf{w}}, \tag{3.10}$$

and the gradient for a single weight will be

$$\Delta w_{m,n} = -\frac{\partial J}{\partial w_{m,n}}, \tag{3.11}$$

Randomly initialized weights are updated iteratively as follows

$$\mathbf{w}(m+1) = \mathbf{w}(m) + \eta \Delta \mathbf{w}(m), \tag{3.12}$$

where the learning rate η determines the magnitude scale of the weights and m represents the training step.

Considering the three-layer MLP from Fig. 3.2 for simplicity, the hidden-to-output weights are first considered. The chain rule is required to calculate the partial derivative of the error J with respect to the weight $w_{j,k}$ since the error does not explicitly depend on the weight itself:

$$\frac{\partial J}{\partial w_{j,k}} = \frac{\partial J}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{j,k}} = \delta_k \frac{\partial net_k}{\partial w_{j,k}}, \tag{3.13}$$

where $net_k = \sum_{j=1}^{n_y} w_{j,k} h_j$ represents the weighted sum result before the activation function is applied and $\delta_k = -\frac{\partial J}{\partial net_k}$ represents how the overall error changes with respect to the unit's net activation and is called the sensitivity of the k-th neuron. δ_k is calculated using the chain rule

$$\delta_k = -\frac{\partial J}{\partial net_k} = -\frac{\partial J}{\partial \hat{y}_k} \cdot \frac{\partial \hat{y}_k}{\partial net_k} = 2(y_k - \hat{y}_k) \cdot \phi^{'}(net_k). \tag{3.14}$$

Since the net activation net_k of the k-th output neuron is

$$net_k = \sum_{j=1}^{n_y} w_{j,k} h_j, \tag{3.15}$$

the derivative in Eqn. 3.13 can be computed as

$$\frac{\partial net_k}{\partial w_{j,k}} = h_j. \tag{3.16}$$

Considering all the computed derivatives together, the update rule in Eqn. 3.11 for the hidden-to-output weights will be

$$\Delta w_{j,k} = \eta \cdot \delta_k \cdot h_j = \eta \cdot 2(y_k - \hat{y}_k) \cdot \phi^{'}(net_k) \cdot h_j. \tag{3.17}$$

After calculating the hidden-to-output weights, the input-to-hidden weights are now considered. Using Eqn. 3.11, and the chain rule, the following calculations are made

$$\frac{\partial J}{\partial w_{i,j}} = \frac{\partial J}{\partial h_j} \cdot \frac{\partial h_j}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{i,j}}. \tag{3.18}$$

Calculating the first term as

$$\frac{\partial J}{\partial h_j} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_j} \tag{3.19}$$

$$= -2 \sum_{k=1}^{n_y} (y - \hat{y}) \cdot \frac{\partial \hat{y}}{\partial h_j} \tag{3.20}$$

$$= -2 \sum_{k=1}^{n_y} (y - \hat{y}) \cdot \frac{\partial \hat{y}}{\partial net_k} \cdot \frac{\partial net_k}{\partial h_j} \tag{3.21}$$

$$= -2 \sum_{k=1}^{n_y} (y - \hat{y}) \cdot \phi'(net_k) \cdot w_{j,k}. \tag{3.22}$$

The sensitivity is defined for a hidden unit as:

$$\delta_j = \phi^{'}(net_j) \cdot 2 \sum_{k=1}^{n_y} (w_{j,k} \delta_k). \tag{3.23}$$

It is important to note that, δ_j is just the sum of all individual sensitivities of output units weighted by the respective hidden-to-output weights $w_{j,k}$, then all multiplied by the $\phi^{'}(net_j)$. So the input-to-hidden update rule can be expressed as

$$\Delta w_{i,j} = \eta \cdot x_i \cdot \delta_j = \eta \cdot x_i \cdot \left(2 \sum_{k=1}^{n_y} w_{j,k} \delta_k \right) \phi'(net_j). \tag{3.24}$$

For a three-layer MLP, Eqn. 3.17 and 3.24 give the backpropagation algorithm which is utilized to reduce the errors.

In summary, for each neuron, starting from the output layer and moving backward to the input layer, the algorithm calculates how much each weight and bias contributed to the total error. This involves determining the loss function's gradients with regard to each weight and bias. To lower the error rate, the weights and biases are adjusted in the gradient's opposite direction after the gradients have been computed. This step usually involves multiplying the gradient by a learning rate, which determines the size of the update steps. Forward pass and backward pass are repeated for many iterations (epochs), with each iteration using a new batch of training data, until the network's weights and biases are optimized to minimize the loss function.

3.3.4 Convolutional Neural Networks

Deep learning (DL) is a subset of machine learning (ML), a broader field where algorithms learn from data to make predictions or decisions. Deep learning utilizes neural networks with many layers, known as deep neural networks, to automatically learn and extract complex patterns and representations from large amounts of data. In comparison, traditional machine learning often relies on manual feature extraction and simpler models, which may require more domain expertise to select relevant features and typically perform well with smaller datasets. While both approaches aim to learn from data, deep learning's hierarchical approach to feature learning and its capacity to handle vast amounts of unstructured data give it an edge in solving more complex and large-scale problems. Fig. 3.3 shows a Venn diagram illustrating the relationship between AI, ML and DL.

One type of deep learning model called the convolutional neural network (CNN) is particularly effective for analyzing visual data. They have been widely used in image and video recognition, medical image analysis, and, more recently, in radar signal processing. CNNs are built with the ability to automatically and adaptably identify

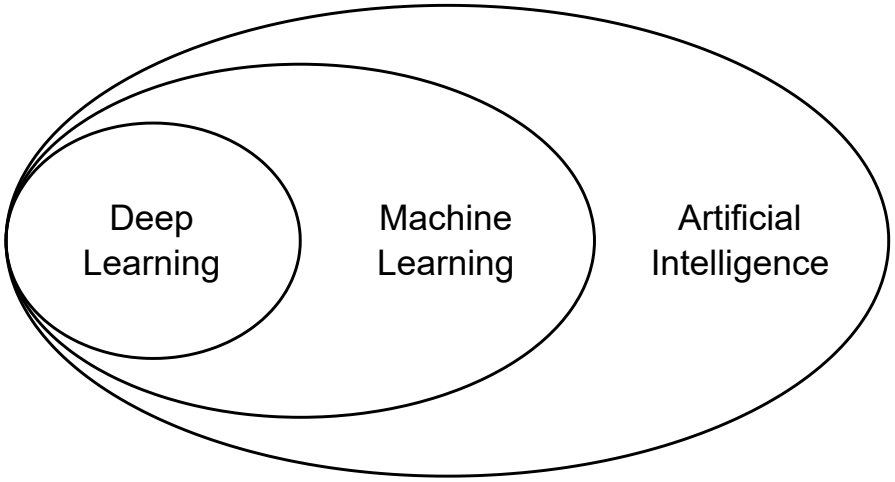


Figure 3.3: Relationship between artificial intelligence, machine learning and deep learning.

feature spatial hierarchies from given data. Considering the range-Doppler maps as images, CNNs can be useful for various radar-related interference problems through the useful features from input data.

CNNs consist of layers that perform convolutions and pooling operations. Convolutional layers apply a set of learnable filters to the input, capturing essential features such as edges and textures. These filters slide across the input data, producing feature maps that highlight various patterns. Pooling layers merge the extracted features and reduce the spatial dimensions of these feature maps, making the model more computationally efficient and less sensitive to positional changes. As data progresses through a CNN, the complexity of layers increases, enabling the network to learn from basic to abstract features. The initial convolutional layers detect low-level features such as edges, lines, and simple textures. As the network deepens, intermediate convolutional layers start to combine these basic features to recognize more complex patterns and structures, like corners and shapes. In the deeper layers, the network identifies high-level features, such as parts of objects or specific textures.

The convolution operation from the qth layer to the $(q+1)$th layer can be shown mathematically with the assumption that the filter in the qth layer has a dimension of

$F_q \times F_q \times d_q$ as follows [1]

$$h_{ijp}^{(q+1)} = \sum_{r=1}^{F_q} \sum_{s=1}^{F_q} \sum_{k=1}^{d_q} w_{rsk}^{(p,q)} \cdot h_{i+r-1,j+s-1,k}^{(q)} \quad \forall i \in \{1, \ldots, L_q - F_q + 1\} \tag{3.25}$$

$$\forall j \in \{1, \ldots, B_q - F_q + 1\} \tag{3.26}$$

$$\forall p \in \{1, \ldots, d_{q+1}\}, \tag{3.27}$$

where the three-dimensional tensor $W^{(p,q)} = [w_{ijk}^{(p,q)}]$ denotes the pth filter in the qth layer, the indices i, j, k denotes the positions of the filter through the height, width, and depth. The three-dimensional tensor $H^{(q)} = [h_{ijk}^{(q)}]$ denotes the feature maps in the qth layer.

The convolution operation is followed by the activation function and pooling operations.

3.3.4.1 Activation Function

Activation functions are integral to the functioning of neural networks because they introduce non-linearity, which is crucial for learning complex patterns and relationships in data. Without activation functions, a neural network would essentially be a series of linear transformations, regardless of its depth, and thus unable to model the intricate non-linear relationships inherent in most real-world data. Activation functions enable neural networks to capture and represent these non-linear relationships. For instance, phenomena such as image recognition, natural language processing, and radar signal analysis involve complex patterns that linear models cannot effectively handle. Rectified Linear Unit (ReLU) is a widely used activation function in neural networks, known for its simplicity and effectiveness. The function is defined as

$$f(x) = \max(0, x). \tag{3.28}$$

In simpler terms, ReLU returns the input x if it is positive, and zero otherwise. This results in a piecewise linear function where the output is zero for all negative values of x and increases linearly with x for positive values.

3.3.4.2 Pooling Operation

Pooling operation is a technique used in CNNs to reduce the spatial dimensions (width and height) of the input volume for each convolutional layer, while also reducing the number of parameters and computations in the network. There are two common types of pooling operations:

Max Pooling: By dividing the input image into a series of non-overlapping rectangles, max pooling produces the maximum value for each sub-region. It is often used to help make the representations smaller and more manageable.

Average Pooling: Average pooling computes the average of each sub-region of the input.

An elementary example of a CNN operation is shown in Fig. 3.4. The figure shows a convolution operation using a 3×3 edge detector filter. After the convolution operation, ReLu is applied as an activation function and lastly, a 2×2 max pooling operation is performed to obtain the result.

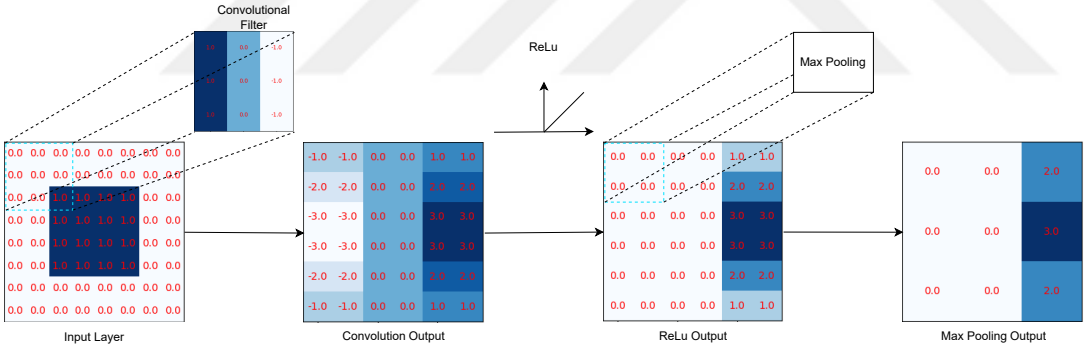


Figure 3.4: Basic convolutional operations: Convolution, activation, and pooling.

3.3.5 U-Net

U-Net [33] is a type of CNN primarily used for image segmentation tasks, where the goal is to classify each pixel in the image, enabling localization, unlike typical convolutional networks used for single-class classification tasks. Because of its success, U-Net has been widely used in many other disciplines besides biomedical image

segmentation, for which it was initially created. The architecture of U-Net is characterized by its "U" shape, which consists of a contracting path (encoder) and an expansive path (decoder).

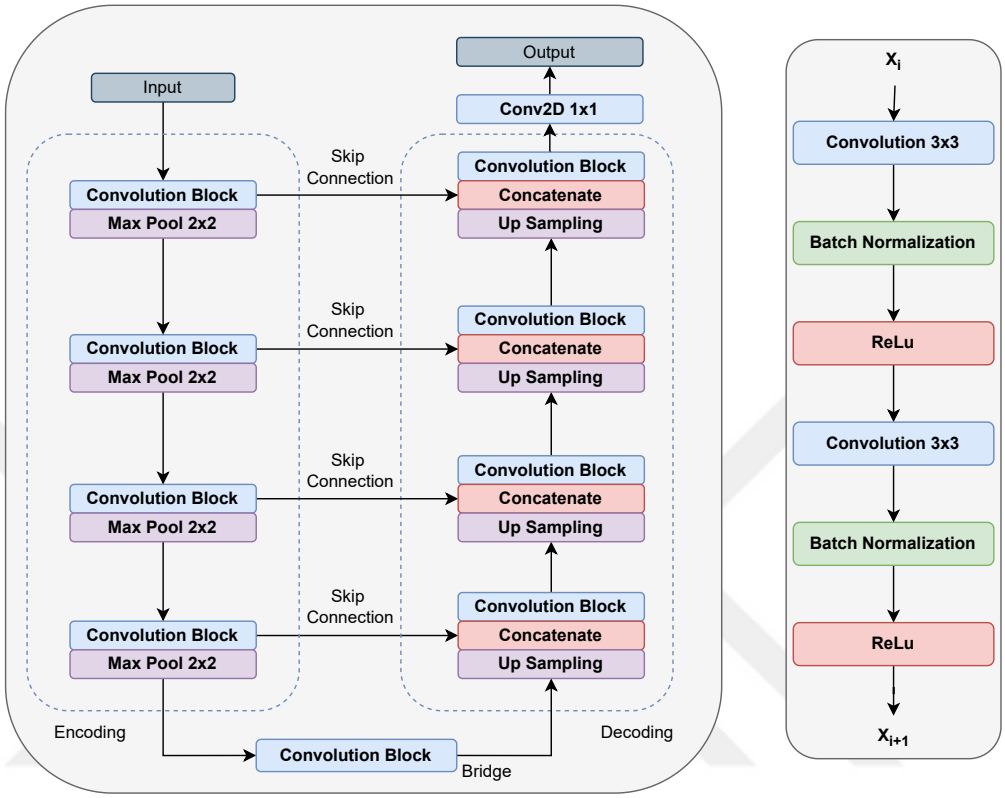


Figure 3.5: U-Net Architecture.

The contracting path includes convolutional layers and downsampling layers. The input image passes through a series of convolutional layers. Each convolutional layer uses small 3x3 filters followed by a ReLU activation function. This extracts features from the image. After each convolutional layer, max-pooling layers with a 2×2 window are used to downsample the image, reducing its spatial dimensions while increasing the number of feature channels. This process captures the context of the image at various scales. At the end of the contracting path, the bottleneck layer connects the contracting path to the expansive path. The bottleneck is the deepest part of the network, where the image is represented in a highly abstract form. It consists of convolutional layers with ReLU activations, without any downsampling. The expan-

sive path includes upsampling, concatenation (skip), and convolutional layers. The upsampling layers increase the spatial dimensions of the image using transposed convolutions (also called deconvolution). For each upsampling step, the corresponding feature maps from the contracting path are concatenated with the upsampled output. This provides fine-grained information that was lost during downsampling. After concatenation, more convolutional layers (3×3 filters followed by ReLU) are used to refine the features. The last layer is a 1×1 convolution that associates each feature vector with the required number of classes (for pixel-wise classification). Fig. 3.5 shows the basic structure of the U-Net. On the left, the contracting and the expansive paths and the skip connections are shown, On the right, the structure of the convolution blocks is shown.

As a result, several defining features of the U-net make it powerful in the segmentation tasks. Firstly, its symmetrical encoder-decoder structure outputs images with the same dimensions as the inputs. In addition, the encoder captures context and extracts features from the input image at multiple levels of abstraction enabling the network to learn hierarchical features. However, during downsampling, information might be lost in the encoder. To prevent this information loss, another defining feature is introduced. Concatenation layers directly move feature mappings between the encoder and the decoder. Reintroducing the lost information by concatenation helps the learning process. In addition to that, by concatenating high-resolution features from the encoder with the upsampled features in the decoder, the network can better localize objects. This combination ensures that the segmentation map retains precise boundary details, which is crucial for accurate segmentation.

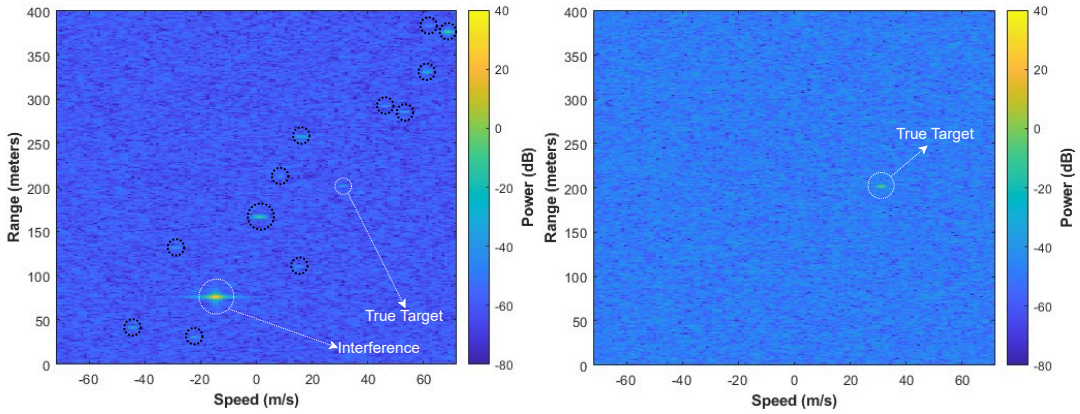
CHAPTER 4

METHODS

4.1 Simulation Data Generation

The experiments were conducted using datasets generated with MATLAB. In the case of supervised learning, a neural network requires a dataset that includes input data paired with the desired output. In this thesis, both the coherent interference and the non-coherent interference problem have been examined separately. For the coherent interference case, the input data is a one-bit quantized received signal, contaminated with interference, in the form of a range-Doppler map. Since the purpose is to teach the neural network to clear the provided input from the interference and one-bit quantization effects, the desired output is a received signal with high-bit precision, free from any interference effect, in the form of a range-Doppler map. Fig. 4.1 shows an example of the input data and corresponding label pair for the coherent interference case. Fig. 4.1a, represents the target, interference, and ghost targets appearing in the input data while Fig. 4.1b displays the desired output that only contains the target. For the non-coherent interference case, the input data is a one-bit quantized received signal, contaminated with interference, in the time domain. The desired output is a received signal with high-bit quantization, free from any interference effect, in the time domain. Fig. 4.2 shows an example of the input data and corresponding label pair for the non-coherent case. Fig. 4.2a, represents the received signal in the time domain, contaminated with interference, Fig. 4.2b displays the desired output in the time domain that only contains the target return signal.

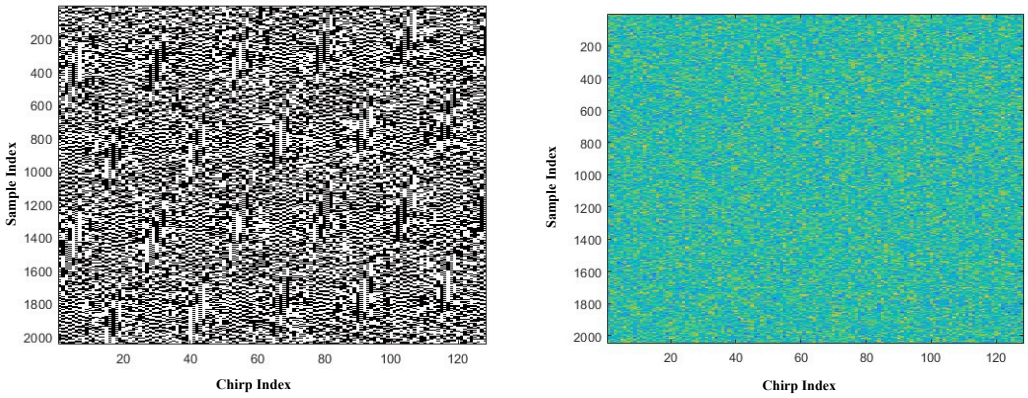
Fig. 4.3 illustrates the interference problem in a highway scenario, although it is not drawn to scale. The interferer is represented as an automobile with radar and



(a) Input data (b) Corresponding label

Figure 4.1: An example representation of the input data (a) and corresponding label (b) provided to the neural network for coherent interference case.

positioned in the opposite lane of the victim radar, while the target, also represented as an automobile, is placed in the same lane as the victim radar. For worst-case scenario analysis, the interferer radar and the target are positioned at the same angle. However, since the interferer radar has negative velocity and the target has positive velocity relative to the victim radar, they are resolvable in range-Doppler maps.



(a) Input data (b) Corresponding label

Figure 4.2: An example representation of the input data (a) and corresponding label (b) provided to the neural network for non-coherent interference case.

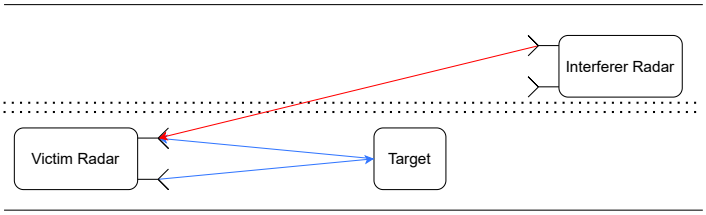


Figure 4.3: Highway interference scenario.

While creating the dataset for both interference scenarios, interferer radar placement for every input data is varied across a range of $[150, 200]$ m, and target placement for every input data is varied across a range of $[100, 390]$ m according to the uniform distribution. Interferer radar velocity for every input data is varied over a range of $[-60, 0]$ m/s and target velocity over $[0, 60]$ m/s according to the uniform distribution. Both the interferer radar and the target are placed at an azimuth angle of 2 degrees relative to the victim radar. The selected target is a point target with an RCS of 10 dBsm and it follows the Swerling 0 model. Note that the signal model here is realistic in that it is constructed towards the so-called "off-grid" case where target and interferer distances are not multiples of range resolution. When a target is placed at an off-grid distance, the returned signal energy spreads across the neighboring range bins after FFT processing. This results in SNR loss, making the target harder to detect. This may also adversely affect the training of the network and the performance it achieves. Table 4.1 provides the parameters of the victim radar, the radar causing coherent interference and the radar causing non-coherent interference. Note that the SNR is defined as $SNR = P_r/N_0B$, where P_r is received power, B is bandwidth and N_0 is noise in W/Hz and considered as an AWGN. Fig. 4.4 illustrates that, unlike conventional radars that utilize high-precision ADCs, one-bit ADCs are employed after each antenna.

To our knowledge, [22] is the only paper that considers the coherent interference problem. However, their work considers a SISO radar system and does not address scenarios involving multiple antenna elements. Automotive radars, however, are generally equipped with multiple antenna elements. In our work, both the victim and interferer radars have a single transmit element, and $U = 16$ receive elements spaced half a wavelength apart, indicating that a SIMO radar system is being considered. While

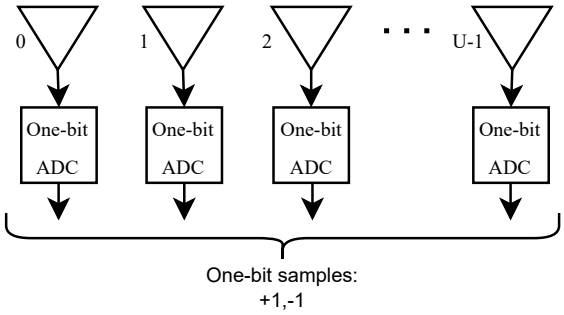


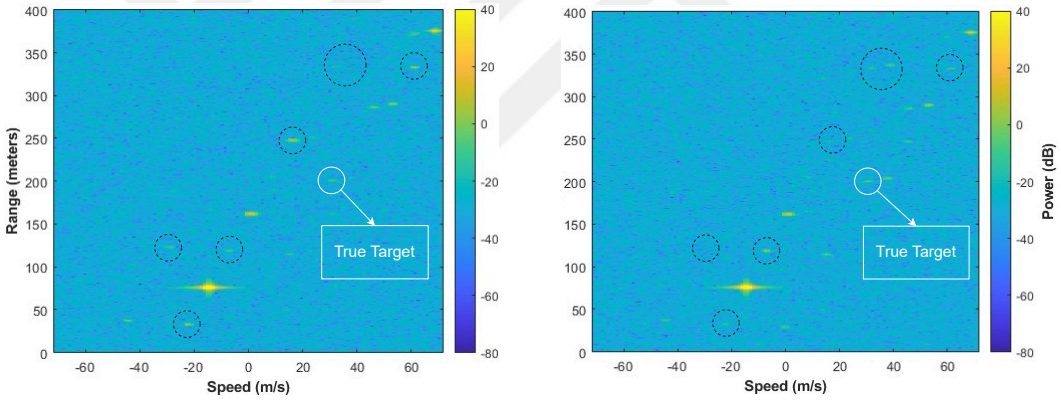
Figure 4.4: Basic representation of a receiver architecture.

preparing the data for the coherent interference scenario, the three-dimensional FFT is computed across time, chirp, and array domains after obtaining the one-bit quantized signal. The FFT across the array domain is computed with 64 points to increase the resolution. The range-Doppler image of the beam corresponding to the angle of the target and the interferer, along with two adjacent beams to its left and right, are fed as three-channel inputs to the network. Fig 4.5 shows the left and right beams. The figure demonstrates that the amplitudes of the harmonics change rapidly from one beam to another compared to the amplitude of the actual target. This attribute provides more information to the network, allowing it to learn the difference between the ghost targets and the actual target. It is exploited to train the network by providing a three-channel input.

To demonstrate the performance increase with the utilization of three-channel inputs for coherent interference, another network is trained using only a single-channel input. This single channel is the range-Doppler map of the beam corresponding to the angle of the target and the interferer. The performance of these two networks will be compared in the next chapter. For the non-coherent interference scenario, a single-channel network is trained because multiple channels will not provide any additional information to the network. Table 4.2 summarises the networks trained for interference scenarios. Overall three networks are trained.

Table 4.1: Victim and interferer radar parameters

Parameter	Values		
	Victim Radar	Coh. Int. Rad.	Non-Coh. Int. Rad.
Carrier Frequency f_c	77 GHz	77 GHz	76.8-77.2 GHz
Bandwidth B	150 MHz	150 MHz	90-210 MHz
Sampling Frequency f_s	150 MHz	-	-
Sweep Time Interval T_s	13.65 μs	13.65 μs	11-12.5 μs
Maximum Detectable Range R_{max}	400 m	-	-
# of Samples K	2048	2048	1650-1875
# of Chirps M	128	128	140-159
# of Receive Antenna Elements U	16	-	-
Transmit Power P_t, P_{ti}	30 dBm	30 dBm	30 dBm
Antenna Gain G_t, G_{ti}, G_r	15 dBi	15 dBi	15 dBi
Noise Figure	4 dB	-	-



(a) Range-Doppler map of the left beam.

(b) Range-Doppler map of the right beam.

Figure 4.5: Representation of the range-Doppler maps of the beams to the left (a) and right (b) of the beam corresponding to the angle of the target and the interferer for coherent interference case.

4.2 Neural Network Architecture

The network architecture of [33] is utilized in this paper. The U-Net architecture follows an encoder-decoder structure. The encoder path of the network consists of

Table 4.2: Summary of trained neural networks

Interference scenario	Trained network
Coherent interference	Three channel input
Coherent interference	Single channel input
Non-Coherent interference	Single channel input

convolutional blocks that contain two convolutional layers with a kernel of 3×3 and a padding size of 1, followed by a batch normalization layer and ReLu. After each convolution block, max pooling with a kernel of 2×2 is utilized for down-sampling. The number of feature maps is doubled at each stage of down-sampling starting from 32. The decoder path of the network consists of an up-sampling layer followed by skip connections where corresponding layers in the encoder and decoder path are concatenated and passed through a convolution block. The number of feature maps is halved at each stage. At the output, neither batch normalization nor ReLu is utilized and a 1×1 convolutional layer is used to generate outputs. So the network takes N channel inputs and produces one channel output. During the encoding path, essential features of the true target are captured and during the decoding path, images containing only true targets are reconstructed. Skip connections allow the utilization of both encoder and decoder features at the convolution layers of the decoder path. Fig. 3.5 shows the utilized neural network structure and Table 4.3 provides the parameters of the employed 2D U-Net. The same network parameters are utilized for the networks summarised in Table 4.2.

Table 4.3: Neural network parameters

Parameter	Values
Number of trainable parameters	8,635,809
Batch size	4
Learning rate	0.0005
Optimization algorithm	ADAM

4.2.1 Loss Function

A loss function is employed to measure the difference between predicted and expected values. It guides the optimization process by providing feedback for adjusting model parameters, ultimately minimizing prediction errors during training. In this work, MSE is utilized as a loss function with

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{4.1}$$

MSE measures the mean square error between the predicted value and the target output, where $\hat{y}_i$ is the predicted output and y_i is the actual output.

4.2.2 Preprocessing of the Data

To make sure that each feature contributes equally to the model, the data was standardized before being sent to the network. This was accomplished by scaling the data's mean to zero and its standard deviation to one using

$$\tilde{x} = \frac{x - \mu}{\sigma} \tag{4.2}$$

where μ is the mean of the data, σ is the standard deviation and $\tilde{x}$ represents the standardized data.

4.2.3 Test Samples and Dataset Size

For the three-channel network, a total of 2950 range-Doppler test samples were generated. The range-Doppler maps, used as three input channels, were constructed by 3D FFT processing of one-bit quantized receiver signals contaminated with interference. The interferer was placed randomly, as explained previously. However, the target was positioned 5 meters farther away for every 50 samples, starting from 100 meters and extending to 390 meters. The samples used for training had a size of $400 \times 128 \times 3$. 2700 three-channel range-Doppler samples were used for training, and 300 three-channel range-Doppler samples were used for validation.

For the single-channel network, the same number of test samples were generated in the same way, but only a single beam was used. The samples used for training had a size of $400 \times 128 \times 1$.

For the non-coherent interference case, a total of 200 time domain test samples were generated. The time domain samples were constructed by one-bit quantized receiver signals contaminated with interference. The target distance is fixed at 200 m and the interfering radar distance is fixed at 150 m. The samples used for training had a size of $2048 \times 128 \times 1$. 900 time domain samples were used for training, and 100 time domain samples were used for validation.

4.2.4 Neural Network Training

As shown in the Table 4.3, the network's learning rate η was set at 0.0005. The batch size was set to four, meaning four frames were fed into the network simultaneously. Adaptive Moment Estimation (ADAM) [19] is used as an optimization algorithm. ADAM adapts the learning rate for each parameter by estimating the first and second moments of the gradients. This results in faster convergence and better performance on large-scale data and sparse gradients.

The hyperparameters for ADAM are set to $\beta_1 = 0.9$, which smooths out gradient updates by controlling the decay rate for the first moment estimates, and $\beta_2 = 0.999$, which stabilizes the learning rate by normalizing updates through the second-moment estimates. The network was trained for 100 epochs. However, training was stopped early if the validation loss did not decrease for 15 consecutive epochs to prevent overfitting.

4.2.5 Evaluation Metrics

To investigate radar detection performance, a small and constant false alarm probability is maintained, while observing the detection probability. CFAR detectors are used to ensure a constant false alarm probability. In this thesis, a 2D CA-CFAR detector is utilized to evaluate the performance of the provided networks for coherent

interference, with the false alarm probability set to $P_{fa} = 10^{-5}$.

In addition to detection performance, another metric, the mean ghost target number (mGTN), is defined. This metric tracks the number of ghost targets that appear in the range-Doppler maps. Since the target distance changes every 50 samples in the test dataset, the "mean" in the metric represents the average number of ghost targets detected within each group of 50 samples.

To evaluate the performance of the neural network trained for non-coherent interference, the signal-to-interference-plus-noise ratio (SINR) is defined as the evaluation metric.

$$\text{SINR} = 10\log\left(\frac{\frac{1}{N_O}\sum_{\{n,m\}\in O}\left|\tilde{S}_{RD}[n,m]\right|^2}{\frac{1}{N_N}\sum_{\{n,m\}\in N}\left|\tilde{S}_{RD}[n,m]\right|^2}\right) \tag{4.3}$$

where O is the set of object peaks, N is the set of N_N noise cells, $\tilde{S}_{RD}$ is the range-Doppler maps and n and m are the row and column indexes of the range-Doppler matrix.

CHAPTER 5

RESULTS

5.1 Qualitative Evaluation of the Three-Channel Neural Network

In this section, the performance of the three-channel neural network will be evaluated qualitatively, while a quantitative evaluation will be conducted in the following section. Fig. 5.1 shows the output of the three-channel network when the scenario depicted in Fig. 2.9 is used as input. For better visual comparison, Fig. 2.9 is redrawn and presented here as Fig. 5.2. The proposed network successfully suppresses interference and ghost targets while preserving the actual target, preventing false target detections.

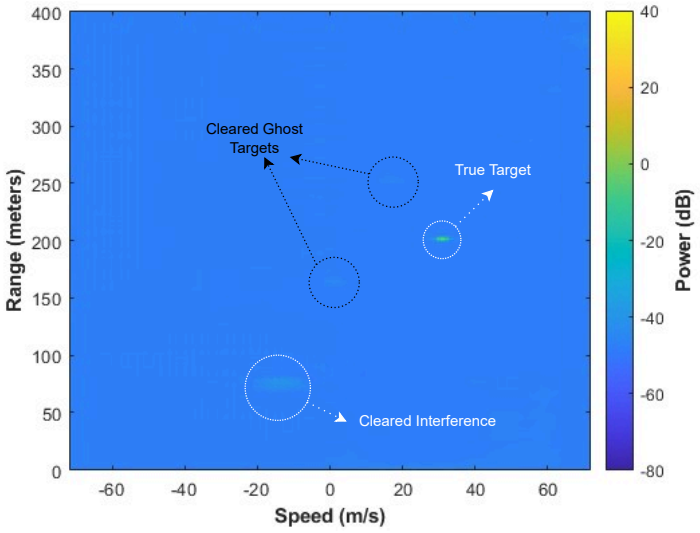


Figure 5.1: Output of the three-channel network.

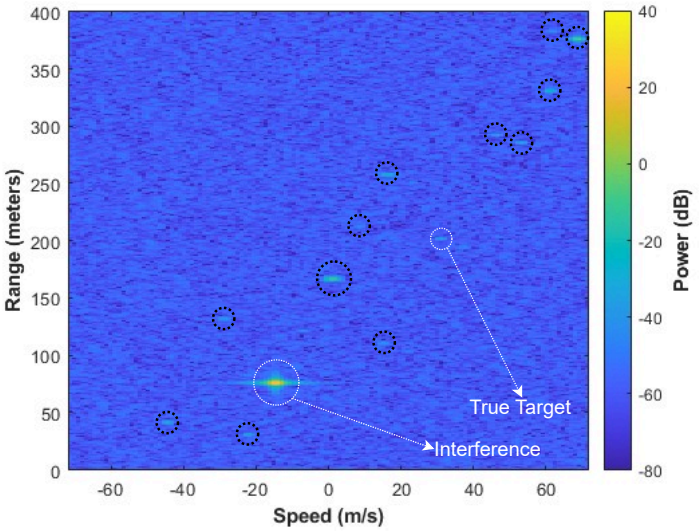


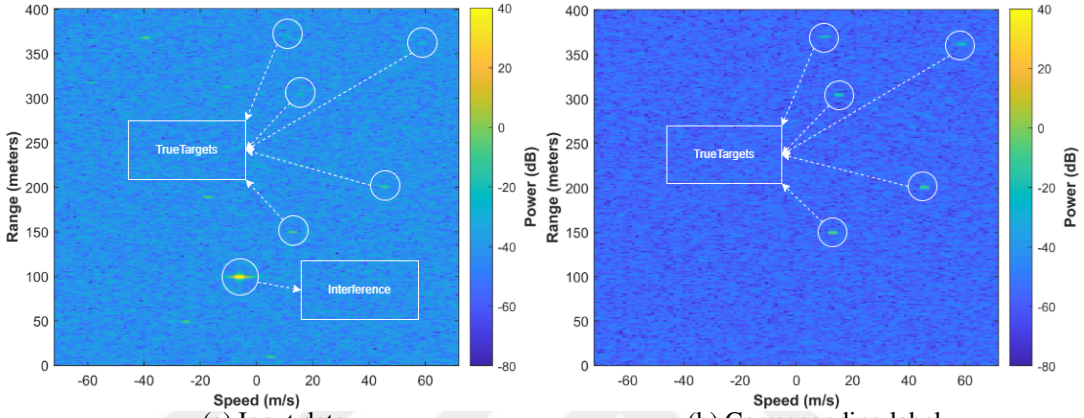
Figure 5.2: Redrawn Fig. 2.9.

5.1.1 Three-Channel Network Generalization Capability

Two different scenarios were employed to assess the generalization capability of the trained network. First, a multiple-target scenario was considered as a test sample instead of a single target. The velocity of each target varied within the range of $[0, 60]$ m/s for every sample. However, unlike the test samples described in 4.2.3, targets were randomly placed within a range of $[100, 390]$ m, and the number of targets was set to five. Second, to evaluate whether the network could detect the target and eliminate ghost targets independently of the provided angle information, the interferer radar and the target were placed at an azimuth angle of 10 degrees relative to the victim radar, rather than at the previously used angle of 2 degrees. In this case, a single target was considered. The qualitative evaluation of the second scenario will be omitted here, as it is very similar to what is shown in Fig. 5.1. For both scenarios, the range-Doppler image of the beam corresponding to the angle of the target and the interferer, along with two adjacent beams to its left and right, were used as three-channel inputs to the network.

Fig. 5.3 shows an input-output pair to test the performance of the three-channel network in a multiple-target scenario. Although the three-channel network is trained on range-Doppler maps containing only a single target, it demonstrates strong seg-

mentation capabilities by effectively learning the features of the targets. The rapid changes in the amplitudes of the harmonics from beam to beam further enhance its learning capacity. As a result, regardless of the number of targets provided in the test, the network is able to detect most of the targets. Another important point we need to mention is that, in a multiple-target scenario, targets are separated in the range-Doppler map and have different SNR values. Even though the three-channel network is trained using a single target, hence a single SNR value for every input sample, it can detect targets that have different SNR values.



(a) Input data (b) Corresponding label

Figure 5.3: An example representation of the multiple target input data (a) and corresponding multiple target label (b) provided to the neural network as a test for coherent interference case.

Fig. 5.4 shows the three-channel network output when Fig. 5.3a is used as a test input. The proposed network successfully suppresses interference and ghost targets while preserving the actual targets, thereby preventing false target detections.

5.2 Quantitative Evaluation of the Three-Channel Neural Network

To evaluate the performance, we present P_d vs. SNR graphs and mean GTNs with and without the network. Since the output of the network is a single-channel 400×128 matrix, we apply 2D CA-CFAR with a false alarm probability $P_{fa} = 10^{-5}$ to the network outputs. To compare how effectively the network mitigates coherent interference and reduces ghost targets, we generated 2950 samples with high-bit precision,

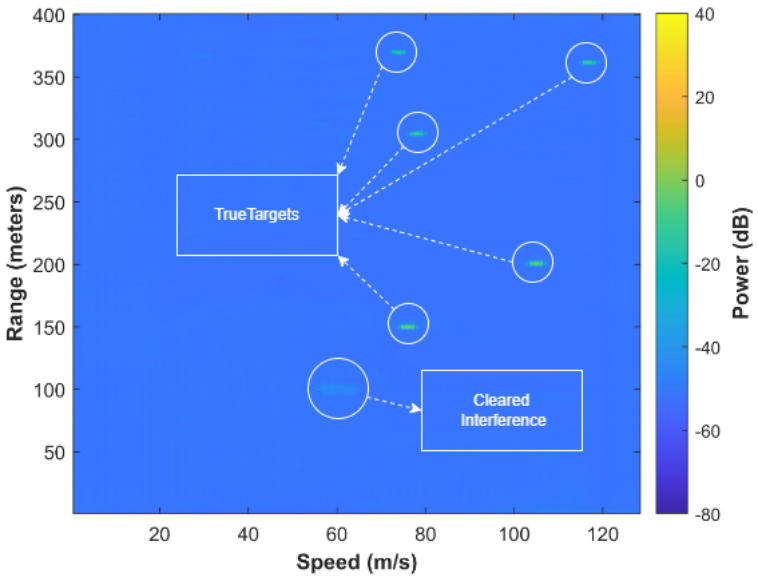
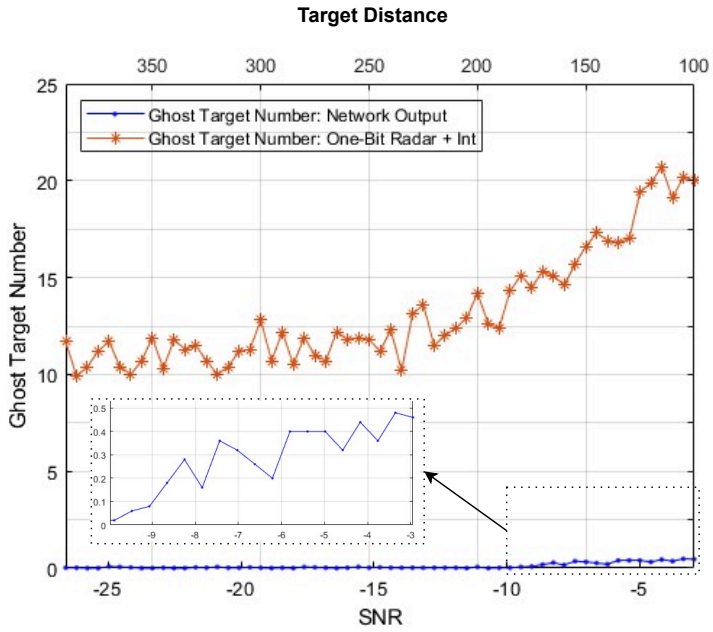


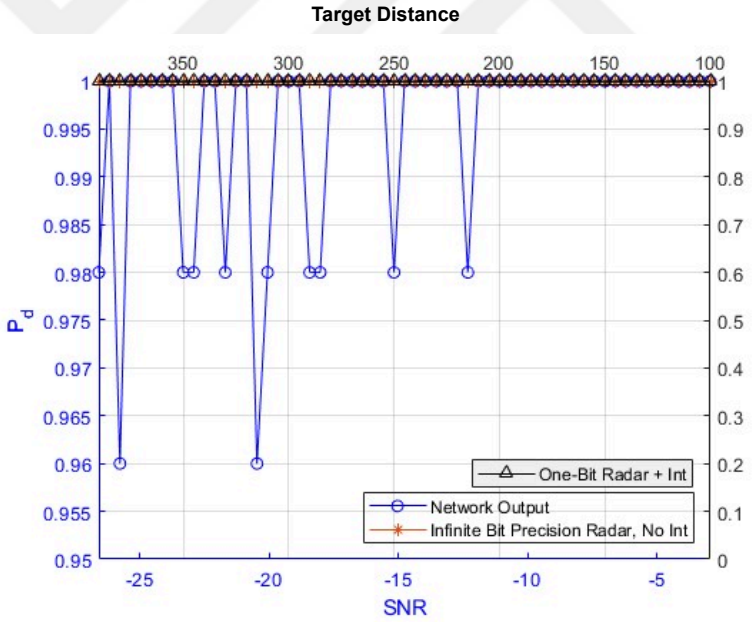
Figure 5.4: Output of the three-channel network of the multiple target test scenario.

which included only the target return signal and the noise. The placement of the target is the same as section 4.2.3. These samples represent normal radar operation with no interference and sufficient bit precision for quantization. Additionally, we generated 2950 samples with one-bit quantization, containing both the target return signal, coherent interference and noise. We apply 2D CA-CFAR with the same false alarm probability to the range-Doppler image of the beam corresponding to the angle of the target and interference for both sets of samples created for comparison purposes. The network that is trained for non-coherent interference will be discussed later.

Fig. 5.5 compares the mean GTN and detection performance with and without network utilization in the presence of coherent interference. Different SNR points were obtained by varying the target distance while maintaining a constant RCS during the acquisition of the results, as explained earlier. The mean GTN values of the network output and one-bit quantization with coherent interference, without network utilization, are compared in Fig. 5.5a. Without utilizing the network, up to 21 ghost targets are detected on average after CA-CFAR when one-bit quantization is used. The mean GTN values for the one-bit quantized scenario decrease and stabilize around $10 - 12$ for low SNR. This stabilization occurs because, as the SNR of the target return signal decreases, the amplitudes of cross-generated harmonics also decrease and cannot exceed the noise power. Consequently, only self-generated harmonics of the inter-



(a) Mean GTN vs SNR comparison with RCS=10 dBsm



(b) P_d vs SNR comparison with RCS=10 dBsm

Figure 5.5: Comparison of performance with and without network utilization.

ference are detected. The proposed network successfully removes ghost targets from range-Doppler maps, resulting in clear and accurate reconstructions, with fewer than 0.5 ghost targets remaining detectable on average. Fig. 5.5b compares the detection performances of the network output, high-bit precision case without interference, and one-bit quantization case with coherent interference. Considering the target as an automobile with an RCS of 10 dBsm in a highway scenario, up to its nearly maximum detectable range of 390 m, the network can detect the target without any significant loss in detection performance.

5.2.1 Three-Channel Input and Single-Channel Input Comparison

We now discuss the rationale behind selecting three channels as input data instead of a single channel and compare their performances. To test the single-channel scenario, we generated the same dataset as before, but only the range-Doppler image of the beam corresponding to the angles of the target and the interferer was used as input data, as explained earlier. The new network performed similarly in terms of detection performance compared to the previous three-channel network but was less effective

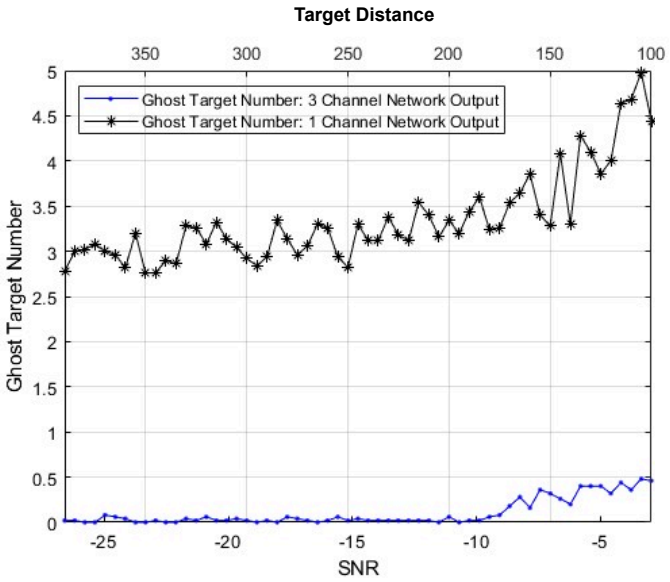


Figure 5.6: Mean GTN vs SNR comparison of the three-channel network and the single-channel network

in reducing ghost targets, as shown in Fig. 5.6. On average, up to 5 ghost targets were detected after CA-CFAR when the single-channel scenario was tested. The mean GTN values for the single-channel scenario decreased and stabilized around 2.5 to 3 for low SNR. This stabilization occurs because, as the SNR of the target return signal decreases, the amplitudes of cross-generated harmonics also decrease, as explained earlier.

5.2.2 Three-Channel Network Generalization Capability

In this subsection, the generalization capability of the three-channel network is quantitatively evaluated based on the scenarios described in 5.1.1. For the first scenario, P_d vs. SNR performance is shown in Fig. 5.7 [1]. Detection performance for low SNR targets is slightly decreased compared to the single target scenario. In addition, since the targets were randomly placed in each sample with equal probability, providing a mean GTN vs. SNR performance is not feasible. Instead, Table 5.1 presents the mean GTN over 1000 test samples. The results indicate that, on average, 0.41 ghost targets are detected across all test samples.

Table 5.1: Multiple target mean GTN performance

# of 5 target scenario	Targets SNR	# of mGTN (over 1000)
1000	-2.9 dB to -26.6 dB	0.41

For the second scenario, P_d vs. SNR performance was obtained similar to Fig. 5.5b and is therefore omitted here. Despite the different angle values, the target was successfully detected. Similar to Fig. 5.5a, an average of fewer than 0.75 ghost targets remained detectable on the range-Doppler maps. Fig. 5.8 compares the mGTN for test samples corresponding to the azimuth angle of 2 degrees and 10 degrees.

[1] A total of 1000 scenarios are generated, each containing 5 targets. In each scenario, target detections are evaluated within 5 m intervals and the results are plotted. For instance, one interval could range from 100 m to 105 m, while another ranges from 105 m to 110 m.

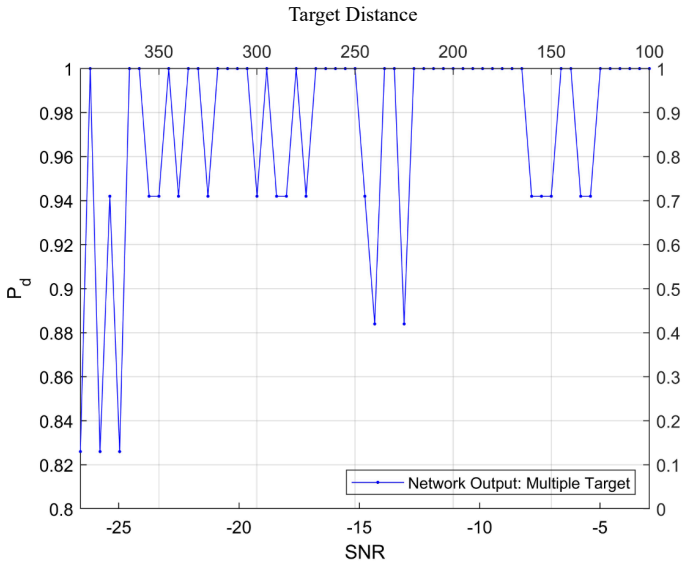


Figure 5.7: P_d vs SNR comparison for multiple target scenario with RCS= 10 dBsm.

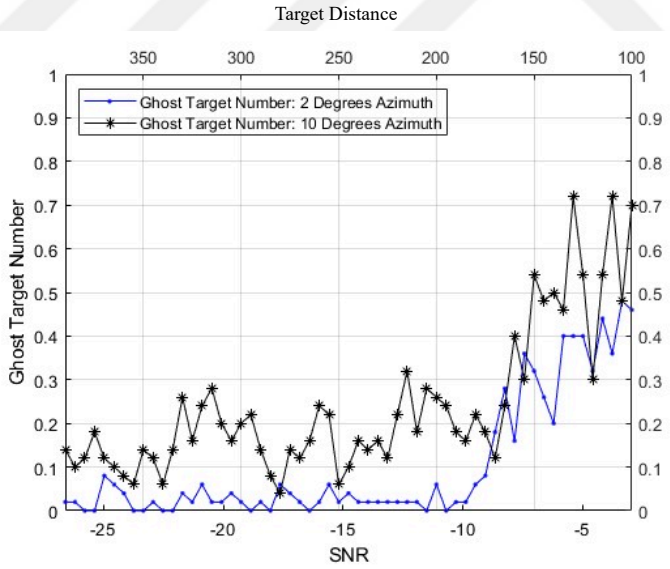


Figure 5.8: Mean GTN vs SNR comparison of the azimuth angle of 2 degrees and 10 degrees

5.2.3 Limit of Improvement of the Three-Channel Network

To explore the limits of the network, we compared the detection performance of the three-channel network, the high-bit precision case without interference, and the one-bit quantization case with interference, as shown in Fig. 5.9. With a target RCS of -14 dBsm, to maintain SNR continuity with 5.5b, detection performance in the presence of interference decreases rapidly. When a high-power signal, such as interference, is received, the performance of the quantizer deteriorates leading to this decline. However, it's important to note that such low RCS values are not realistic even for targets like pedestrians or bicycles. Note that, the signal-to-interference ratio (SIR) ranges between -36 dB and -59 dB according to the target distance.

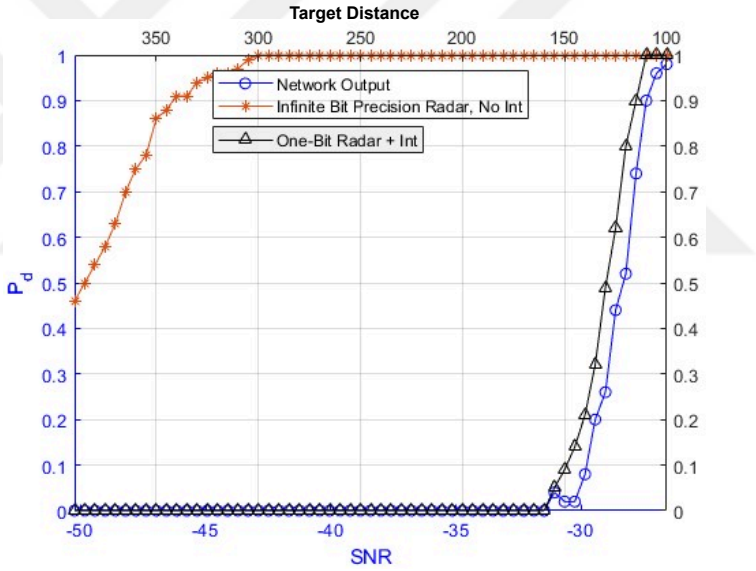
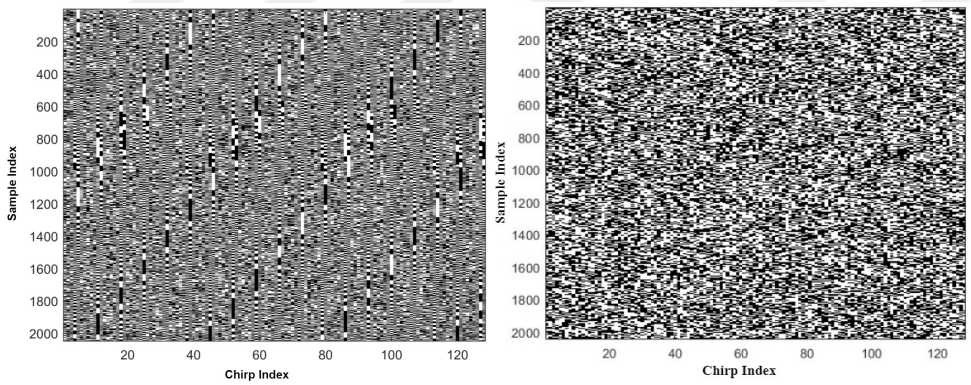


Figure 5.9: P_d vs SNR comparison with RCS=-14 dBsm.

In addition to this, in scenarios where the target is closer than 50 meters, outside the considered range (100-390) m, the network perceives the target as interference due to the high SNR, contrary to what it has learned, and is unable to distinguish the target. However, in these cases, since the target is very prominent, it can be differentiated from interference and ghost targets using a classification method.

5.3 Qualitative Evaluation of the Neural Network for Non-Coherent Interference

Fig. 5.10 presents an example of a one-bit quantized time-domain signal with and without non-coherent interference. Due to the selected radar parameters that cause non-coherent interference, Fig. 5.10a shows that nearly half of the samples in the data matrix are affected resulting in the loss of target information. Fig. 5.10b illustrates the one-bit quantized time-domain signal without the non-coherent interference to better demonstrate the non-coherent interference effect. Fig. 5.11 shows the output of the network trained for non-coherent interference. The Network effectively detects and removes the samples affected by non-coherent interference but also loses the information about the target at affected parts. However, in general, experimental results show that interference mitigation methods utilized in the frequency or time-frequency domains perform better than time-domain methods. Nonetheless, suppose it is necessary to perform interference mitigation in the time domain for one-bit quantized signals. In that case, this method might be useful, since this method operates without amplitude information.



(a) One-bit quantized time-domain signal with non-coherent interference

(b) One-bit quantized time-domain signal without non-coherent interference

Figure 5.10: An example representation of the one-bit quantized time-domain signal with non-coherent interference (a) and one-bit quantized time-domain signal without non-coherent interference (b).

Fig. 5.12 presents examples of range-Doppler maps under various scenarios. Fig.

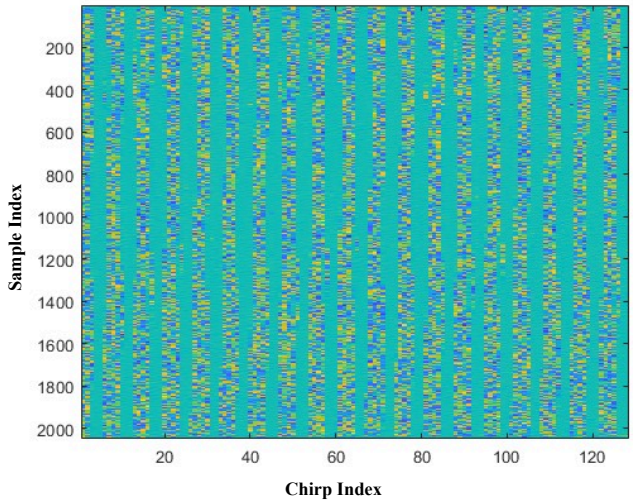
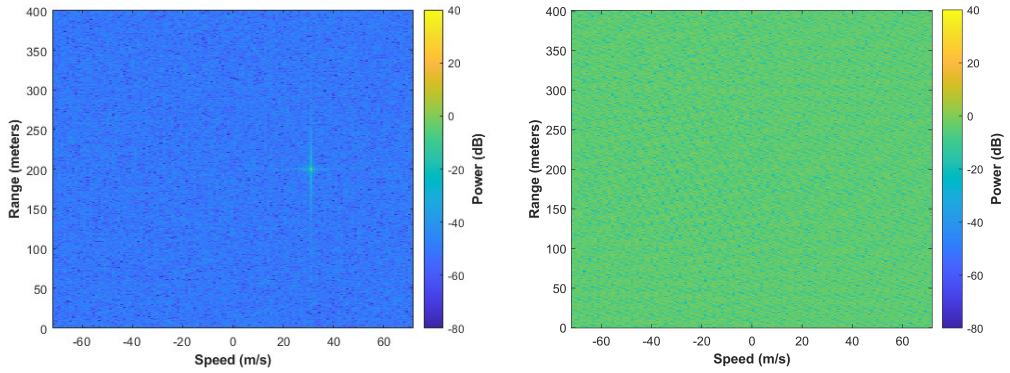


Figure 5.11: Output of the neural network trained for non-coherent interference.

5.12a shows the map with no interference and high-bit precision, where the target is clearly visible. In contrast, Fig. 5.12b illustrates the effect of non-coherent interference with high-bit precision, which raises the noise floor, making the target harder to detect. Fig. 5.12c depicts the scenario with non-coherent interference and one-bit quantization. Since the energy of the interference signal is limited the increase in the noise floor is also limited. However, one-bit quantization introduces ghost targets on the map. Finally, Fig. 5.12d illustrates the neural network output after 3D FFT processing. As seen in Fig. 5.11, the neural network effectively mitigates the samples impacted by non-coherent interference, thereby reducing the noise floor hence increasing the SINR. However, the network cannot clear the ghost targets.

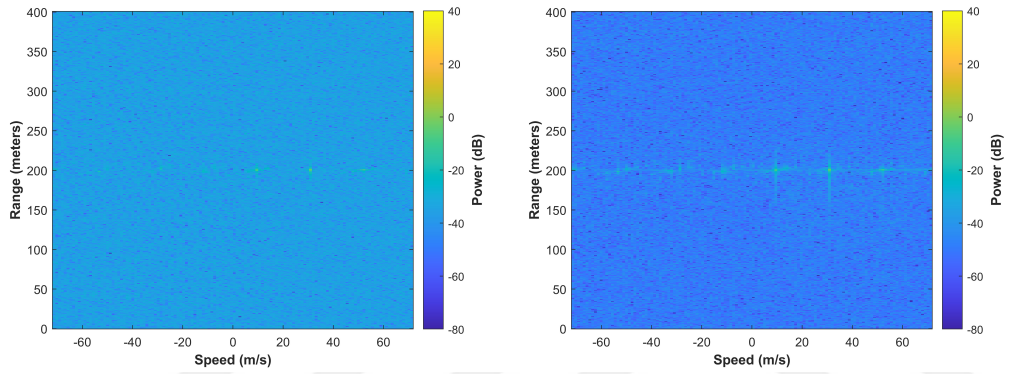
5.4 Quantitative Evaluation of the Neural Network for Non-Coherent Interference

Since interference mitigation for the non-coherent interference case is performed in the time domain, the neural network is applied to each one-bit ADC output. Following this, a 3D FFT is performed, and the SINR is calculated on the range-Doppler map of the beam corresponding to the angle of the target and interference. We will also calculate the SINR values for the following cases on the range-Doppler map of the



(a) Range-Doppler map example with no interference and high-bit precision.

(b) Range-Doppler map example with non-coherent interference and high-bit precision.



(c) Range-Doppler map example with non-coherent interference and one-bit quantization.

(d) Range-Doppler map example with non-coherent interference and one-bit quantization after utilizing the neural network.

Figure 5.12: Examples of Range-Doppler maps under various cases: (a) with no interference and high-bit precision; (b) with non-coherent interference and high-bit precision; (c) with non-coherent interference and one-bit quantization; and (d) with non-coherent interference, one-bit quantization, and neural network processing.

beam corresponding to the target's and interference's angle after applying the 3D FFT: the high-bit precision case with no interference, the high-bit precision case with non-coherent interference, and the one-bit quantized case with non-coherent interference. Table 5.2 compares the SINR values for provided cases obtained from only a single test sample.

The histogram in Fig. 5.13 illustrates the probability distribution of the SINR values

Table 5.2: SINR comparison of provided cases

Cases	SINR
High-bit precision, no interference + 3D FFT	50.7 dB
High-bit precision, with non-coherent interference + 3D FFT	6.7 dB
One-bit quantization, with non-coherent interference + 3D FFT	36.6 dB
One-bit quantization, with non-coherent int. + network + 3D FFT	47.2 dB

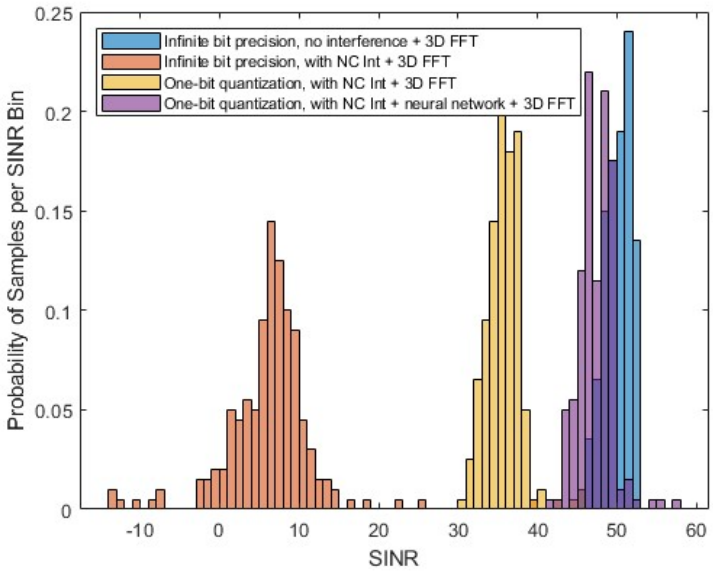


Figure 5.13: SINR comparison of provided cases with $U = 16$.

obtained from the provided cases when the number of receive antenna elements is set to 16. In this figure, the target distance is fixed at 200 m and the interfering radar distance is fixed at 150 m. For every case, 200 test samples are used. Before signal processing, the target return signal has an approximate SNR of -15 dB, while the interference has an approximate SNR of 34 dB. The high-bit precision case with no interference represents the normal radar operation. In the presence of non-coherent interference in the high-bit precision case, the interference increases the noise floor and significantly decreases the SINR. Although one-bit quantization limits the energy of the interference signal, the interference still benefits from the processing gain provided by the multiple-receive antenna configuration, leading to an increased noise floor, as shown in Table 5.3. Additionally, due to radar parameters that cause non-

coherent interference, nearly half of the samples are affected by interference, resulting in a loss of target information after one-bit quantization. This explains the SINR decrease of nearly 15 dB when one-bit quantization is used in the presence of non-coherent interference, compared to normal radar operation. Lastly, when the neural network is applied, it effectively clears most of the samples affected by interference but cannot retrieve or recreate the lost target information. As a result, the neural network improves the SINR, bringing it to approximately $3-4$ dB below the SINR value observed in normal radar operation.

Table 5.3: Processing Gain

Parameters	Processing Gain
# of Samples K , 2048	~ 33 dB
# of Chirps M , 128	~ 21 dB
# of Receive Antenna Elements U , 16	~ 12 dB

To justify the assumption regarding the processing gain provided by the multiple antenna configuration, another test set was created by setting the number of receive antenna elements to 2. The target distance was fixed at 200 m, and the interfering radar distance was fixed at 150 m. With two antennas, the processing gain is approximately 3 dB. The histogram in Fig. 5.14 illustrates the probability distribution of the SINR values obtained from the mentioned cases with two receive antenna elements. As previously mentioned, when one-bit quantization is utilized, nearly half of the samples are affected by interference due to radar parameters that cause non-coherent interference, resulting in a loss of target information after quantization. Additionally, the interference benefits from the processing gain provided by the multiple-receive antenna configuration. This explains the approximate 6 dB decrease in SINR when one-bit quantization is used in the presence of non-coherent interference compared to the normal radar operation.

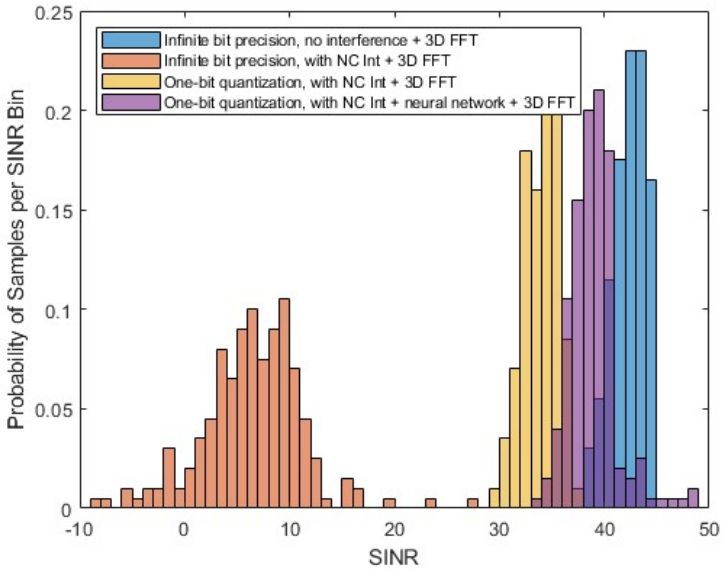


Figure 5.14: SINR comparison of provided cases with $U = 2$.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, we investigated machine learning methods for interference mitigation and ghost target reduction in one-bit quantized SIMO FMCW automotive radars. One-bit quantized ADCs have emerged as a viable solution to the high power consumption and production costs associated with high-precision ADCs. However, the non-linearity of one-bit radars introduces harmonic issues that complicate conventional linear signal processing methods. Additionally, coherent interference exacerbates the harmonic problem due to its high power and target-like image. Conversely, non-coherent interference raises the noise floor, thereby reducing the detectability of actual targets.

In this work, we explored machine learning methods to reduce ghost targets and mitigate interference when one-bit ADCs are utilized. We began by investigating the ghost target problem caused by one-bit quantization in the presence of coherent interference and proposed a three-channel 2D U-net to address this issue. We compared the proposed three-channel network with a single-channel network, evaluating its generalization capacity and identifying its limitations. Following this, we examined the problem of noise floor increase caused by non-coherent interference in one-bit quantization and discussed the corresponding simulation results.

Our findings demonstrate that utilizing a neural network for automotive radars with one-bit quantization in the presence of coherent interference can significantly reduce the number of ghost targets while maintaining minimal loss in detection performance, regardless of the number of targets or their angular positions. Additionally, in the presence of non-coherent interference, the proposed network can mitigate interference in the time domain, bringing the SINR close to that of normal radar operation.

In the future, this work can be enhanced by considering interference mitigation in the presence of non-coherent interference in the frequency or time-frequency domains for more comprehensive results. Additionally, developing a single neural network capable of handling both interference scenarios simultaneously, rather than using separate networks, is another promising direction for future research.

REFERENCES

[1] Charu C. Aggarwal. *Neural Networks and Deep Learning: A Textbook*. 1st. Springer Publishing Company, Incorporated, 2018. ISBN: 3319944622.

[2] Abdelmohsen Ali and Walaa Hamouda. "Low Power Wideband Sensing for One-Bit Quantized Cognitive Radio Systems". In: *IEEE Wireless Communications Letters* 5.1 (2016), pp. 16–19. DOI: 10.1109/LWC.2015.2487347.

[3] Stephen Alland et al. "Interference in Automotive Radar Systems: Characteristics, Mitigation Techniques, and Current and Future Research". In: *IEEE Signal Processing Magazine* 36.5 (2019), pp. 45–59. DOI: 10.1109/MSP.2019.2908214.

[4] Solveig Badillo et al. "An Introduction to Machine Learning". In: *Clinical Pharmacology & Therapeutics* 107.4 (2020), pp. 871–885. DOI: https://doi.org/10.1002/cpt.1796. eprint: https://ascpt.onlinelibrary.wiley.com/doi/pdf/10.1002/cpt.1796. URL: https://ascpt.onlinelibrary.wiley.com/doi/abs/10.1002/cpt.1796.

[5] Constantine A. Balanis. *Antenna Theory: Analysis and Design*. USA: Wiley-Interscience, 2005. ISBN: 0471714623.

[6] Jonathan Bechter et al. "Automotive radar interference mitigation using a sparse sampling approach". In: *2017 European Radar Conference (EURAD)*. 2017, pp. 90–93. DOI: 10.23919/EURAD.2017.8249154.

[7] Graham M. Brooker. "Mutual Interference of Millimeter-Wave Radar Systems". In: *IEEE Transactions on Electromagnetic Compatibility* 49.1 (2007), pp. 170–181. DOI: 10.1109/TEMC.2006.890223.

[8] Junyoung Chung et al. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. arXiv: 1412.3555 [cs.NE]. URL: https://arxiv.org/abs/1412.3555.

[9] Minglong Deng et al. “One-Bit ADCs/DACs Based MIMO Radar: Performance Analysis and Joint Design”. In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 2609–2624. DOI: 10.1109/TSP.2022.3176953.

[10] Stephen R. Doughty. “Development and Performance Evaluation of a Multistatic Radar System”. PhD Thesis. PhD thesis. London, United Kingdom: University of London, 2008. URL: https://discovery.ucl.ac.uk/id/eprint/1444153/.

[11] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. 2nd ed. New York: Wiley, 2001. ISBN: 978-0-471-05669-0.

[12] Ahmed A. ElSharkawy, Abdallah S. Abdallah, and Mohamed W. Fakhr. “A New Semantic Segmentation Technique for Interference Mitigation in Automotive Radar”. In: *2023 IEEE Wireless Communications and Networking Conference (WCNC)*. 2023, pp. 1–6. DOI: 10.1109/WCNC55385.2023.10118913.

[13] G. Franceschetti, V. Pascazio, and G. Schirinzi. “Processing of signum coded SAR signal - Theory and experiments”. In: *IEE Proceedings F: Radar and Signal Processing* 138 (June 1991), pp. 192–198.

[14] Jonas Fuchs et al. “Automotive Radar Interference Mitigation using a Convolutional Autoencoder”. In: *2020 IEEE International Radar Conference (RADAR)*. 2020, pp. 315–320. DOI: 10.1109/RADAR42522.2020.9114641.

[15] Li-Bo Guo, Yang-Yang Dong, and Chunxi Dong. “Residual Attention Augmented U-Shaped Network for One-Bit SAR Image Restoration”. In: *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024), pp. 1–22. DOI: 10.1109/TGRS.2024.3357812.

[16] Li-Bo Guo et al. “One-bit LFM signal recovery via random threshold strategy”. In: *Digital Signal Processing* 111 (2021), p. 102965. ISSN: 1051-2004. DOI: https://doi.org/10.1016/j.dsp.2021.102965. URL: https://www.sciencedirect.com/science/article/pii/S105120042100004X.

[17] Benzhou Jin et al. “One-Bit LFMCW Radar: Spectrum Analysis and Target Detection”. In: *IEEE Transactions on Aerospace and Electronic Systems* 56.4 (2020), pp. 2732–2750. DOI: 10.1109/TAES.2020.2978374.

[18] Feng Jin and Siyang Cao. “Automotive Radar Interference Mitigation Using Adaptive Noise Canceller”. In: *IEEE Transactions on Vehicular Technology* 68.4 (2019), pp. 3747–3754. DOI: 10.1109/TVT.2019.2901493.

[19] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

[20] J. R. Klauder et al. “The theory and design of chirp radars”. In: *The Bell System Technical Journal* 39.4 (1960), pp. 745–808. DOI: 10.1002/j.1538-7305.1960.tb03942.x.

[21] Bin Le et al. “Analog-to-digital converters”. In: *IEEE Signal Processing Magazine* 22.6 (2005), pp. 69–77. DOI: 10.1109/MSP.2005.1550190.

[22] Jiawang Li et al. “Interference Suppression for Radar Signal using 2D UNet based on Semantic Segmentation”. In: *2022 IEEE 5th International Conference on Electronic Information and Communication Technology (ICEICT)*. 2022, pp. 603–606. DOI: 10.1109/ICEICT55736.2022.9909297.

[23] Runlong Li et al. “Deep Learning for Interference Mitigation in Time-Frequency Maps of FMCW Radars”. In: *2021 CIE International Conference on Radar (Radar)*. 2021, pp. 1883–1886. DOI: 10.1109/Radar53847.2021.10028226.

[24] Jiwoo Mun, Heasung Kim, and Jungwoo Lee. “A Deep Learning Approach for Automotive Radar Interference Mitigation”. In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. 2018, pp. 1–5. DOI: 10.1109/VTCFall.2018.8690848.

[25] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN: 0262018020.

[26] Takuya Nozawa et al. “An anti-collision automotive FMCW radar using time-domain interference detection and suppression”. In: *International Conference*

on Radar Systems (Radar 2017). 2017, pp. 1–5. DOI: 10.1049/cp.2017.0366.

[27] Frank Olver et al. *The NIST Handbook of Mathematical Functions*. en. Cambridge University Press, New York, NY, 2010-05-12 00:05:00 2010.

[28] Josh Patterson and Adam Gibson. *Deep Learning: A Practitioner's Approach*. 1st. O'Reilly Media, Inc., 2017. ISBN: 1491914254.

[29] Muhammad Rameez, Mattias Dahl, and Mats I. Pettersson. "Autoregressive Model-Based Signal Reconstruction for Automotive Radar Interference Mitigation". In: *IEEE Sensors Journal* 21.5 (2021), pp. 6575–6586. DOI: 10.1109/JSEN.2020.3042061.

[30] Jiaying Ren and Jian Li. "One-bit digital radar". In: *2017 51st Asilomar Conference on Signals, Systems, and Computers*. 2017, pp. 1142–1146. DOI: 10.1109/ACSSC.2017.8335529.

[31] Mark A. Richards. *Fundamentals of Radar Signal Processing*. US: McGraw-Hill Professional, 2005, pp. -1. ISBN: 0071444742. DOI: 10.1036/0071444742. eprint: https://mhebooklibrary.com/doi/pdf/10.1036/0071444742. URL: https://mhebooklibrary.com/doi/book/10.1036/0071444742.

[32] Johanna Rock et al. "Complex Signal Denoising and Interference Mitigation for Automotive Radar Using Convolutional Neural Networks". In: *2019 22th International Conference on Information Fusion (FUSION)*. 2019, pp. 1–8. DOI: 10.23919/FUSION43075.2019.9011164.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[34] SAE International. *SAE International Recommended Practice, Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Tech. rep. SAE Standard J3016_202104. Revised April 2021, Issued January 2014. Apr. 2021.

[35] C.E. Shannon. “Communication in the Presence of Noise”. In: *Proceedings of the IRE* 37.1 (1949), pp. 10–21. DOI: 10.1109/JRPROC.1949.232969.

[36] Cuiqi Si et al. “A Convolutional De-Quantization Network for Harmonics Suppression in One-Bit SAR Imaging”. In: *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023), pp. 1–16. DOI: 10.1109/TGRS.2023.3330530.

[37] R.H. Walden. “Analog-to-digital converter survey and analysis”. In: *IEEE Journal on Selected Areas in Communications* 17.4 (1999), pp. 539–550. DOI: 10.1109/49.761034.

[38] Christian Waldschmidt, Juergen Hasch, and Wolfgang Menzel. “Automotive Radar — From First Efforts to Future Systems”. In: *IEEE Journal of Microwaves* 1.1 (2021), pp. 135–148. DOI: 10.1109/JMW.2020.3033616.

[39] Jianping Wang. “CFAR-Based Interference Mitigation for FMCW Automotive Radar Systems”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.8 (2022), pp. 12229–12238. DOI: 10.1109/TITS.2021.3111514.

[40] Jiawei Zhang. “Gradient Descent based Optimization Algorithms for Deep Learning Models Training”. In: *CoRR* abs/1903.03614 (2019). arXiv: 1903.03614. URL: http://arxiv.org/abs/1903.03614.