

Pulse-Doppler Quick Look Processor for NeXtRAD



Prepared by:

Ethan Lawrence Morris

Prepared for:

Dr. Stephen Paine

Department of Electrical Engineering

University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town
in partial fulfilment of the academic requirements for a
BSc in Electrical & Computer Engineering

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.



October 29, 2023

Ethan Lawrence Morris

Date

Word Count: 17896

Acknowledgements

The conclusion of this report marks the culmination of the past four years in achieving my degree and my academic journey, at least for now. I am immensely thankful to all those who have played a role in helping me reach this point.

I would like to extend my gratitude to Dr. Stephen Paine for the guidance he has provided throughout the last three months, from the project's inception to its completion. Thank you for demonstrating the proper approach to problem-solving and equipping me with the tools to identify and resolve issues. Your expertise and willingness to educate were invaluable in the project's successful completion.

I'd like to express my appreciation to Grant Norrie for dedicating his time to proofread my report, aiding me in crafting well-structured content. I am very grateful for your willingness to assist and for the knowledge you have shared with me.

To Jahar, I consider myself fortunate to have met you during this journey, and I value the time we have spent studying (and not studying) over the last four years.

To my friends, Zaccary and Jason, who have accompanied me from high school through the challenging journey of completing our degree, words cannot fully convey my gratitude for your camaraderie, the countless hours we have invested in working and learning together, and the personal growth we have experienced together.

To Ethan, my closest friend and brother from another mother, I cannot begin to express my thankfulness for meeting you and growing alongside you. I am fortunate to have you in my corner, and I genuinely appreciate the unwavering and ongoing support you provide. We have shared many late nights studying on campus, over the phone, and in the library, and I look forward to sharing many more.

To my father, Steven, and step-mother, Melissa, your love and support has been invaluable. I appreciate the opportunities and experiences you have provided, propelling me to become the best version of myself. Thank you for the sacrifices made to help me achieve this milestone.

To my mother, I want to express my deep gratitude for bringing me into this world and affording me the chance to embark on a journey that I have cherished and would not trade for any other. You have been my moral compass, always guiding me toward what is right. Although you are not here to witness this milestone and the end of this journey, I know you are with me, watching over me, and ensuring I continue to follow the right path.

Abstract

In this project, a **Quick-Look Processor (QLP)** was designed and developed specifically for the NeXtRAD radar system, a substantial upgrade from the NetRAD system, offering dual-band operation and complete polarimetric sensing for improved target detection in complex environments. The work began with a comprehensive literature review to establish the background context for the study, encompassing radar technologies, signal processing techniques, and data visualisation methods. The **QLP** was meticulously designed to provide versatility and real-time data processing, enabling it to handle data from both the NetRAD and NeXtRAD systems and generate graphical representations of critical parameters such as range, time, and Doppler frequency. Moreover, a user-friendly **Graphical User Interface (GUI)** was incorporated to facilitate seamless interaction with the **QLP**, empowering radar operators, researchers, and analysts to customise data visualisation, configure system parameters, and conduct analyses with ease. Performance evaluation revealed the **QLP**'s effectiveness in plotting and tracking targets in **Range-Time Intensity (RTI)** maps and achieving real-time data processing for **Range-Doppler (RD)** maps and spectrograms, though challenges emerged in the processing of **Constant False Alarm Rate (CFAR)** maps, notably related to the **CFAR** detection threshold algorithm's configuration. Pulse integration showed potential in reducing processing times, highlighting the need for further optimisation in the **CFAR** algorithm to achieve real-time performance. In summary, the project successfully designed and demonstrated a **QLP** system that promises to enhance radar technology's capabilities, even though some areas require further refinement to optimise efficiency and real-time performance. This report presents the research, design, evaluation of the **QLP** system.

Contents

List of Figures	viii
Abbreviations	ix
1 Introduction	1
1.1 Background	1
1.2 Objectives	1
1.3 System Requirements	2
1.4 Scope & Limitations	2
1.5 Report Outline	2
2 Literature Review	3
2.1 Pulse-Doppler Radar	3
2.1.1 Brief History	3
2.1.2 Advantages & Disadvantages of Pulse-Doppler Radar	4
2.2 Signal Processing Techniques	5
2.2.1 Pulse Compression	5
2.2.2 Doppler Processing	6
2.2.3 Micro-Doppler Processing	7
2.3 Visualisation Techniques	8
2.3.1 Range-Doppler Map	8
2.3.2 Range-Time Intensity Map	9
2.3.3 Spectrogram	9
2.3.4 Constant False Alarm Rate Map	9
2.4 NeXtRAD Radar System	11
2.4.1 NetRAD	11
2.4.2 NeXtRAD	12
2.5 Quick-Look Processor	13
2.5.1 NeXtRAD Implementations	14
2.5.2 Alternative Implementations	14
3 Quick-Look Processor Design	15
3.1 System Structure	15
3.1.1 System Overview	15
3.1.2 Radar Theory	16
3.2 Pre-processing	17
3.2.1 Overview	17

3.2.2	Pre-processing Theory	18
3.2.3	Reading data from file	19
3.2.4	Enhancing data quality	20
3.2.5	Converting from real to complex	20
3.2.6	Two-dimensional reshaping	20
3.2.7	Zero padding	21
3.3	Signal Processing	21
3.3.1	Overview	21
3.3.2	Processing Theory	22
3.3.3	Pulse Compression	23
3.3.4	Doppler Processing	24
3.3.5	Micro-Doppler Processing	26
3.3.6	Constant False Alarm Rate	26
3.3.7	Pulse Integration	28
3.4	Visualisation	28
3.4.1	Overview	28
3.4.2	Visualisation Theory	28
3.4.3	Range-Time Intensity map	29
3.4.4	Range-Doppler map	30
3.4.5	Spectrogram	32
3.4.6	Constant False Alarm Rate map	33
3.4.7	Constant False Alarm Rate Level plot	34
3.5	Graphical User Interface	35
4	Quick-Look Processor Testing	37
4.1	Range-Time Intensity Map Comparison	37
4.1.1	Testing Procedure	37
4.1.2	Achieved Performance	38
4.2	Range-Time Intensity Time Test	39
4.2.1	Testing Procedure	39
4.2.2	Achieved Performance	40
4.3	Range-Doppler Time Test	41
4.3.1	Testing Procedure	41
4.3.2	Achieved Performance	41
4.4	Spectrogram Time Test	42
4.4.1	Testing Procedure	42
4.4.2	Achieved Performance	42
4.5	Constant False Alarm Rate Time Test	43
4.5.1	Testing Procedure	43
4.5.2	Achieved Performance	44
4.6	System Performance Analysis	45
5	Conclusions	46

6 Recommendations	48
6.1 Enhanced Constant False Alarm Rate Algorithm	48
6.2 Adaptation of System for NeXtRAD data	48
6.3 User Interface Improvements	49
6.4 C++ & Parallel Processing	49
Bibliography	50

List of Figures

3.1	Block diagram showing data handling part of QLP.	16
3.2	Operational deployment configuration of NeXtRAD taken from [1].	16
3.3	System parameters for each of the radars.	17
3.4	Block diagram showing pre-processing section of QLP.	18
3.5	Before and after of reshape of data values into 2D matrix.	19
3.6	Block diagram showing the signal processing section of QLP.	22
3.7	1D CFAR window taken from [2].	23
3.8	Matched filter output for a single pulse.	24
3.9	Conversion to RD data through the application of DFTs.	25
3.10	RTI map for 130000 pulse dataset.	30
3.11	Matrix representation of data for RD map.	31
3.12	Two RD maps showing the change in the map due to movement of a target.	31
3.13	Spectrogram for 130000 pulse dataset.	32
3.14	Two Range CA-CFAR maps showing the change in the map due to movement of a target. .	33
3.15	Two Doppler CA-CFAR maps showing the change in the map due to movement of a target.	34
3.16	Two CFAR Level plots showing the change in the map due to movement of a target. .	35
3.17	File page of GUI application.	36
3.18	Parameter page of GUI application.	36
4.1	First File comparison of RTI plot produced by system and given plot.	38
4.2	Second File comparison of RTI plot produced by system and given plot.	39
4.3	Third File comparison of RTI plot produced by system and given plot.	39
4.4	Comparison of processing time for the RTI map.	40
4.5	Comparison of processing time for the RD map.	41
4.6	Comparison of processing time for the spectrogram.	43
4.7	Comparison of processing time for the CFAR map and CFAR Level plot	44
4.8	Comparison of processing time for all the maps in the system.	45

Abbreviations

1D One-Dimensional

2D Two-Dimensional

ADC Analogue-to-Digital Converter

AWS Amazon Web Services

CA-CFAR Cell Averaging Constant False Alarm Rate

CFAR Constant False Alarm Rate

CNN Convolutional Neural Network

CoSaPD Compressive Sampling Pulse-Doppler

CS Compressed Sensing

CTFT Continuous-Time Fourier Transform

CUT Cell Under Test

DFT Discrete Fourier Transform

F/B Front-to-Back Ratio

FFT Fast Fourier Transform

FIR Finite Impulse Response

FMCW Frequency Modulated Continuous Wave

GEO Geosynchronous Earth Orb

GO-CFAR Greatest of Constant False Alarm Rate

GOCA-CFAR Greatest of Cell Averaging Constant False Alarm Rate

GPSDO Global Positioning System Disciplined Oscillator

GUI Graphical User Interface

HDF5 Hierarchical Data File version 5

HERM Helicopter Rotor Modulation

IaaS Infrastructure-as-a-Service

IIR Infinite Impulse Response

IQ In-phase and Quadrature Components

kNN k-Nearest-Neighbour

LEO Low Earth Orbit

LMS Least Mean Square

LOS Line-of-Sight

MIT Massachusetts Institute of Technology

MMF Mismatched Filter

MTI Moving Target Indication

NASA National Aeronautics and Space Administration

OS-CFAR Ordered Statistic Constant False Alarm Rate

PDR Pulse-Doppler Radar

PRF Pulse Repetition Frequency

QLP Quick-Look Processor

RAM Random Access Memory

RCS Radar Cross Section

RD Range-Doppler

RMMSE Reiterative Minimum Mean-Square Error

RTI Range-Time Intensity

SAR Synthetic Aperture Radar

SCR Signal-to-Clutter Ratio

SIR Signal-to-Interference Ratio

SNR Signal-to-Noise Ratio

SOC-A-CFAR Smallest of Cell Averaging Constant False Alarm Rate

STFT Short-Time Fourier Transform

SVM Support Vector Machine

SWGA Slotted Waveguide Antenna

TESS Transiting Exoplanet Survey Satellite

UAV Unmanned Aerial Vehicle

UCL University College London

UCT University of Cape Town

WVD Wigner Ville Distribution

Chapter 1

Introduction

This project has been primarily centred around the design and development of a sophisticated [Quick-Look Processor \(QLP\)](#) system for the NeXtRAD radar system. The NeXtRAD radar system represents a significant advancement over its predecessor, the NetRAD system, particularly due to its dual-band operation and comprehensive polarimetric sensing capabilities. This report aims to provide a comprehensive account of the research, design, and evaluation of the [QLP](#) system in the context of these advanced radar technologies.

1.1 Background

Within the continuously evolving radar systems domain, the demand for effective data processing tools has grown significantly. NeXtRAD and NetRAD systems play a crucial role in target detection and tracking, especially in challenging environments with high levels of sea clutter and interference. Realising the full potential of these radar systems hinges on the creation of a sophisticated [QLP](#) system.

The project commenced with an extensive review of existing literature, providing a solid foundation for subsequent research and development. NeXtRAD and NetRAD systems are at the forefront of modern radar technology, engineered to address the complexities of contemporary radar applications. To achieve optimal performance, the integration of a high-performance [QLP](#) system is necessary.

The research and development efforts aim to bridge the gap between the current capabilities of NeXtRAD and NetRAD systems and the ever-growing demands of radar technology. By incorporating a [QLP](#) system, the project strives to enhance precision and adaptability, ensuring these radar systems remain at the forefront of technological innovation. The initial focus on literature review is pivotal, as it grounds the research in the extensive knowledge and experience accumulated in the radar technology field.

1.2 Objectives

The primary objective of this project encompassed two key aspects. First, it involved designing and implementing a robust [QLP](#) system tailored for processing NeXtRAD radar data. This system's primary purpose was to extract essential data, such as range, time, and Doppler frequency, to enhance target detection and tracking capabilities.

Secondly, the project included the development of an intuitive and user-friendly [Graphical User Interface \(GUI\)](#). The objective here was to simplify interactions with the processing tool, making it accessible

to radar operators, researchers, and analysts. By creating an efficient and user-friendly interface, the project aimed to accommodate a wide range of users, ensuring the system's utility across various professional and research contexts.

1.3 System Requirements

The [QLP](#) system was purposefully crafted to meet the demanding needs of real-time and near-real-time data processing. It has been meticulously designed to ensure that recorded data can be analysed promptly, without unnecessary delays. The overarching goal of the [QLP](#) system is to usher in a new era of radar technology by enhancing its adaptability and versatility, making it suitable for a wide range of applications.

1.4 Scope & Limitations

The project, constrained to a tight thirteen-week timeline, revolved around designing and implementing a [QLP](#) system for the NextRAD radar system. This [QLP](#) system had a focused purpose – processing radar data to generate four types of maps: [Range-Doppler \(RD\)](#), [Range-Time Intensity \(RTI\)](#), [Constant False Alarm Rate \(CFAR\)](#), and Spectrogram. The system, developed in MATLAB, was modularised into functions and shared via a GitHub repository alongside this report. The report also contains a comprehensive account of the system's design, development, and performance testing, ensuring it aligns with the requirements outlined in Section 1.3.

However, certain limitations were imposed on the project. Firstly, the design and development relied on NetRAD data files, which were screened to ensure they contained relevant data for processing, such as targets or interference. These adjustments primarily focused on the pre-processing stage of the [QLP](#) pipeline. Additionally, the accuracy of the generated plots was assessed based on the precision of signal data processing and [RTI](#) maps due to the absence of comparative maps for the other three visualisation techniques present in the data.

1.5 Report Outline

The report consists of six chapters, each dedicated to distinct project phases. Chapter 2 initiates the report by delving into relevant literature concerning the project's topic. In Chapter 3, the design and development of the [QLP](#) system are explored in depth, accompanied by essential theoretical background and comprehensive documentation. This chapter also provides a breakdown of the subsystems and includes block diagrams that illustrate the data handling aspects of the pipeline. Chapter 4 details the system's performance and its alignment with the specified requirements. Lastly, Chapter 5 offers conclusions drawn from the project's outcomes, while Chapter 6 provides pertinent recommendations for future considerations.

For access to the implementation code of the [QLP](#) system, the GitHub repository at the following link provides a valuable resource for exploring the technical details and source code behind the project: [Github Repository](#).

Chapter 2

Literature Review

This chapter aims to offer context for the project by delving into existing research within the field. Such exploration enhances our understanding of the developments made thus far and highlights potential avenues for further investigation. The discussion covers the history and advantages of Pulse-Doppler Radar (**PDR**) technology, the use of various signal processing techniques, and methods employed to visualise the results. Following this, an evaluation of existing research on NeXtRAD and NetRAD, as well as projects utilising these technologies, will be carried out before shifting the focus towards prior research and implementations of **QLPs**.

2.1 Pulse-Doppler Radar

This section conducts a critical examination of essential literature related to **PDR**, beginning with a brief exploration of the historical context of **PDR**, followed by an assessment of its advantages and disadvantages.

2.1.1 Brief History

The history of **PDR** dates back to the end of World War II. The earliest documented ground-based **PDR** system, as described by Hansen in 1947 [3], employed a high duty cycle at a low carrier frequency to ensure unambiguous range and Doppler measurements.

In 1949, Barlow's work [4] led to the development of one of the initial Continuous Wave Doppler Radar systems. However, its range was limited by transmitter leak-through and near-range clutter in the main beam. To address this issue, the transmitter was pulsed while gating off the receiver, resulting in an effective **PDR** system.

Perkins et al.'s work in 1984 [5] details the history of the first operational **PDR** implemented in the Bomarc IM-99A by Westinghouse under contract for Boeing in the late 1950s. **PDR**'s use enhanced capabilities in detecting and tracking moving targets, especially in cluttered environments.

In 1983, Ringel et al. [6] published an article based on the performance of **PDR** in the F-16, focusing on detection and false alarm performance of the digital fire-control radar, the primary sensor for the F-16 fighter jet.

Fried [7] documented in 1993 how the use of **PDR** in navigation and target tracking expanded from military aircraft and drones to helicopters worldwide, assisting with navigation and hovering.

The NetRAD system is extensively detailed in multiple works spanning the years 2003 to 2008 [8, 9, 10, 11]. These works delve into the design and development of an experimental system. NetRAD's versatile functionality covers target detection and surveillance, applicable both in the air and on the ground. The design philosophy of this budget-friendly netted radar system places a strong emphasis on utilising commercial low-cost components.

Deployed between 2009 and 2011 for the establishment of a measurement database, the system is discussed in a 2011 paper by Inggs et al. [12]. Its primary objective was to concentrate on sea clutter and seaborne target measurements in the United Kingdom and Simon's Town. The three-node NetRAD system was assembled and tested, integrating a **Global Positioning System Disciplined Oscillator (GPSDO)** system to enable a comparison between bistatic clutter data and monostatic data. The conclusion of the data gathering for the sea clutter database was presented in a 2012 paper by Inggs et al. [13], scrutinising the collected data over the recording period to comprehend the sea clutter and vessel properties of the radar configuration.

A proposal for NeXtRAD was introduced by Inggs et al. [14] in 2014 as the successor to the NetRAD system. The NeXtRAD system, designed as a dual-band (X-band and L-band) polarimetric multistatic radar, represented a significant advancement over the S-band low-cost system used to test the eventual successor (NeXtRAD). The paper published in 2017 by Alhuwaimel et al. [15] documented the initial measurements taken with the new radar system. Subsequent measurements were conducted in 2018 [16] and 2019 [17]. The 2018 measurements focused on detecting and tracking targets in sea clutter and micro-Doppler, while the 2019 measurements were the first interleaved L and X-band measurements.

The historical evolution of **PDR** sets the stage for a deeper exploration of its advantages and disadvantages. The early developments in **PDR**, ranging from Hansen's pioneering work to the deployment of the NetRAD system, lay the foundation for understanding its historical significance and capabilities. This context now allows us to delve into a comprehensive discussion of the specific advantages and limitations that **PDR** presents in comparison to other radar technologies, shedding light on its strengths and areas where it faces challenges.

2.1.2 Advantages & Disadvantages of Pulse-Doppler Radar

PDR and **Frequency Modulated Continuous Wave (FMCW)** Radar each have distinct advantages and disadvantages in terms of functionality and performance, catering to specific applications and deployment scenarios.

In a 2013 study by Pulutan et al. [18], the discussion focuses on **PDR**'s inherent clutter suppression capabilities resulting from the non-simultaneous transmission and reception of data. This characteristic enables an increase in both peak and average transmit power, extending the radar's reach for long-range applications. The publication explores potential synergies by combining **FMCW** Radar with **PDR** in a unified system to harness the strengths of each radar type. Despite **PDR**'s vulnerability to blind range issues during simultaneous data transmission and reception, the study underscores the relatively lower impact of leakage on **PDR** compared to **FMCW** Radar. This is attributed to the continuous nature of **FMCW** and the pulsed nature of Pulse-Doppler. Due to its pulsed nature, **PDR** encounters ambiguities at short ranges due to strong signal returns from nearby clutter, necessitating a larger dynamic range.

Further insights into the advantages of **PDR** are presented in a 2020 work by Anju et al. [19], with a focus on Pulse-Doppler Processing. **PDR** systems prove useful when the required information pertains to the range or position of the target. Utilising the Doppler frequency, **PDR** enables the determination of the target's velocity information with high sensitivity. The Doppler frequency can be reconstructed by aligning consecutive pulses, facilitating coherent pulse integration. Spectral sensitivity ensures easy distinction of target velocities, with the demodulated signal manifesting as a sharp feature in the frequency domain.

However, [19] notes that determining the Doppler frequency in **PDR** is not as straightforward as in **FMCW** Radar. This complexity arises from spectral leakage within the Doppler spectra, resulting in power smearing across frequencies. This phenomenon hampers filtering performance within the system, potentially leading to false velocity calculations. The short pulse length in **PDR** may contain only a fraction of a Doppler cycle, introducing added challenges in frequency analysis. Nevertheless, the use of coherent pulse integration allows for the reconstruction of cycles based on the analysis of multiple pulses simultaneously.

2.2 Signal Processing Techniques

This section delves into fundamental signal processing techniques employed on the data to facilitate a more comprehensive understanding of the raw data. The methods addressed in this section are designed to improve the data pertaining to targets within the dataset or mitigate the impact of sea clutter to achieve a similar effect on the detected targets. The discussed methods in this section include pulse compression, Doppler processing, and micro-Doppler processing.

2.2.1 Pulse Compression

Pulse compression is a technique employed to enhance range resolution when pulse length is restricted or to increase **Signal-to-Noise Ratio (SNR)** when the peak power and bandwidth of the transmitted signal are limited. The implementation of pulse compression relies on the application of a matched filter on the data.

In a study by Wojaczek et al. [20], the research explores the use of a matched filter for range compression in a demodulation/remodulation scheme. The study assumes perfect synchronisation to the transmitter and that the transmitted signal has been perfectly reconstructed. The signal is initially split into batches for more efficient processing, followed by applying the matched filter to each batch individually using a convolution operation. The matched filter involves complex mathematical operations, including the **Discrete Fourier Transform (DFT)** and convolution within each batch, constructed based on a range compression filter defined for each batch earlier in the research.

Alternatively, Burrascano et al. [21] implement a matched filtering method involving the time-convolution of the signal recorded at the receiver with the reference or transmitted signal to maximize **SNR**. The paper also addresses the incorporation of windowing within the filtering process, accounting for it with a window function in the calculations.

A second alternative is discussed by Arya and Subha [22]. This paper details the design of a matched filter algorithm in MATLAB for detecting a known signal corrupted by noise. The algorithm involves

the time-convolution of the received echo signal with the complex conjugate of the transmitted signal. The filter described is a [Mismatched Filter \(MMF\)](#), allowing for potentially greater resolution at the expense of [SNR](#) value. This method is considered a potential solution to issues faced in pulse compression in weather radar implementations, as discussed by Argenti and Facheris [23].

In 2004, an alternative to the use of a matched filter was proposed in the work by Blunt and Gerlach [24]. The research introduces the [Reiterative Minimum Mean-Square Error \(RMMSE\)](#) algorithm for adaptive pulse compression. The objective of this algorithm is to apply the true matched filter for each individual range cell based on the actual received signal. This concept was revisited 13 years later by Kikuchi et al. [25], and similar results were achieved. The outcomes indicate that the [RMMSE](#) algorithm is effective in robustly estimating the impulse response representing the range profile. The algorithm exhibited better doppler tolerance than the matched filter implementation, and its response to dense target scenarios remained robust even when the matched filter's performance was sub-optimal.

2.2.2 Doppler Processing

Having explored the intricacies of pulse compression and its various implementations, the focus now shifts to Doppler processing. As per Richards [2], Doppler processing utilises Doppler shift information to achieve at least one of the following aims. The first goal is detecting and measuring target data in a clutter-dominated environment, while the second goal is measuring the Doppler shift and, consequently, the radial velocity of targets. Two processing methods will be examined: [Moving Target Indication \(MTI\)](#) pertains to the first goal, whereas pulse-Doppler processing addresses both objectives.

Moving Target Indication Processing

As per [2], [MTI](#) processing essentially involves processing the slow-time signal entirely in the time domain using a high-pass filter in the slow-time dimension.

In a study by Ma et al. [26], [MTI](#) is detailed in two stages. The first, known as basic [MTI](#) processing, is employed for high [Radar Cross Section \(RCS\)](#) target detection and is based on matched filtering. The second stage is long-time integration, which facilitates the detection of low and medium [RCS](#) targets. This involves compensating for range and Doppler migration in [RD](#) maps over extended integration times through target motion compensation and subsequent noncoherent integration.

In order to implement [MTI](#) utilising filtering, Ash et al.'s research [27] explores three techniques: Background Subtraction, [Finite Impulse Response \(FIR\)](#) filtering, and [Infinite Impulse Response \(IIR\)](#) filtering. Each method is implemented in the time domain. Background Subtraction estimates background clutter over time, subtracting it from future measurements before periodic updates. [FIR](#) filtering employs a single-delay filter and its generalisation to detect changes between consecutive pulses. [IIR](#) filters, suitable when linear phase is not essential, achieve a similar magnitude response as an [FIR](#) filter with fewer coefficients, resulting in reduced group delay. The research concludes that Background Subtraction is suitable for slow-moving target detection, while [FIR](#) and [IIR](#) filtering offer straightforward processes for moving target segmentation.

In 2018, Liu et al. [28] explored the application of a [Convolutional Neural Network \(CNN\)](#) for [MTI](#) processing. Considering [MTI](#) as a multi-class classification problem, the deep [CNN](#) performs

feature extraction and classification. Simulation results from synthetic and experimental data validate the effectiveness and robustness of the **CNN-MTI** approach, particularly in scenarios with limited **Signal-to-Clutter Ratio (SCR)** in non-homogeneous environments.

Pulse-Doppler Processing

Referencing [2], pulse-Doppler processing involves implementing a **Fast Fourier Transform (FFT)** on the signal in the time-domain, specifically in the slow-time dimension. This enables processing in the frequency domain through the application of spectral analysis.

In a paper authored by Liu et al. [29], they introduce a technique known as **Compressive Sampling Pulse-Doppler (CoSaPD)** processing. **CoSaPD** encompasses Doppler estimation, Doppler detection, and range estimation using sub-Nyquist data without the need to recover the Nyquist samples. Simulation results validate the effectiveness of **CoSaPD**, showing performance on par with classical processing with Nyquist samples, yet demanding only one-eighth of the Nyquist rate. This holds true as long as the input **SNR** remains above -25 dB.

In an alternative application, Beltrão et al. [30] introduce a distinctive subpulse-Doppler processing method, differing from conventional pulse-Doppler processing. This method incorporates an additional subpulse processing stage and employs a novel architecture based on the transmission of stochastic waveforms. The authors' suggested method uses subpulse processing to address the trade-off between unambiguous range and velocity estimation. The resulting subpulse-Doppler processing offers advantages, including enhanced Doppler tolerance, reduced ambiguity, and improved clutter mitigation compared to standard pulse-Doppler processing.

2.2.3 Micro-Doppler Processing

Moving from Doppler processing, which primarily deals with target velocities, the focus is now shifted to micro-Doppler processing. This transition enables the investigation into the intricate Doppler shifts caused by sub-components' micro-motions within a target. Micro-Doppler processing presents unique capabilities and applications for analysing complex target movements, further enriching this radar technology exploration.

In a 2015 study, Alemdaroğlu et al. [31] aimed to investigate the potential of distinguishing between different human motions by analyzing the micro-Doppler signatures of the target. The researchers employed a series of signal processing and micro-Doppler processing procedures to collect relevant data. These signal processing steps produced a matrix output through a matched filter, which was then subjected to a high-pass filter designed to reduce clutter in the data. Subsequently, a Hamming window was applied to further refine the processed information.

An alternative method of processing was addressed by Zeng et al. [32], where they investigated the integration of a micro-Doppler processing technique within a **CNN** developed for finger gesture classification. Micro-Doppler processing enabled the generation of feature maps corresponding to finger gestures. This approach involved coherent processing and **FFT** to capture intricate details of the finger movements. With an accuracy of 96.51%, these findings underline the effectiveness of micro-Doppler processing in enhancing the precision and reliability of finger gesture recognition within

the implemented [CNN](#) framework.

In research by Cammenga et al. [33], the integration of wideband radar systems and micro-Doppler analysis was explored to observe targets with micro-motions, overcoming limitations posed by conventional narrowband approaches. Simulations involving a walking person showcased that wideband systems offer intricate spatial details about scattering centres. Additionally, a customised analysis, employing the [Short-Time Fourier Transform \(STFT\)](#) on specific range bins, facilitated the extraction of micro-Doppler signatures over time.

2.3 Visualisation Techniques

In this section, exploration of fundamental methods for visualising meaningful data includes the [RD](#) map, [RTI](#) map, Spectrogram and [CFAR](#) map. These methods are crucial for extracting information from raw data across various domains. The discussion will delve into research on different techniques employed to plot these maps, offering insights into the diverse approaches in visualising such data.

2.3.1 Range-Doppler Map

[RD](#) maps serve as crucial tools in [PDR](#) applications, providing a comprehensive visualisation of targets and clutter in the range-velocity detection space.

In Ma et al.'s research [26], the calculation method for producing [RD](#) maps is demonstrated. The [RD](#) map is generated by applying a matched filter in the range direction, followed by organising the data into a matrix with fast-time (range) and slow-time (Doppler) components. Subsequently, a Doppler [FFT](#) is applied to this matrix, resulting in the [RD](#) map. This [FFT](#) operation facilitates the differentiation of targets and clutter with distinct Doppler characteristics.

In 2019, Feng et al. [34] conducted a study on [RD](#) map generation using [Compressed Sensing \(CS\)](#) theory applied to [PDR](#). The Batches Algorithm is described in the paper, where the received signal is divided into short batches. A [One-Dimensional \(1D\)](#) cross-correlation is then applied to obtain range profiles, followed by a slow-time [FFT](#) to estimate the Doppler frequencies of the targets. Fourier transform-based techniques, while increasing sidelobe levels, can be mitigated using alternative techniques, albeit at the cost of reduced [RD](#) resolution. The application of [CS](#) theory to the algorithm allows for better resolution and lower sidelobe levels in the final [RD](#) map. Aktar and Olsen [35] assessed this [RD](#) map creation method, incorporating sparse reconstruction. Their research concluded that a combination of [CS](#) and sparse reconstruction significantly improves upon alternative methods.

In their study, Alabaster and Hughes [36] tackled the need to determine clutter power within the range-velocity detection space in [PDR](#), a critical factor for evaluating target detectability in clutter-limited scenarios. Their work presents a method for mapping clutter in the range-velocity detection space, which is independent of radar resolution and [Pulse Repetition Frequency \(PRF\)](#). They derived the clutter backscatter coefficient, indicating how much electromagnetic energy is scattered back to a radar system by preexisting clutter, for different map locations. These coefficients were then used to derive clutter radar cross sections, and the clutter map was combined into one unambiguous range and Doppler interval, which is [PRF](#)-dependent. The clutter power was then allocated to the appropriate range and velocity cell, resulting in the unfolded clutter map, also known as the [RD](#) map.

2.3.2 Range-Time Intensity Map

Transitioning from the focus on **RD** maps, which vividly depict targets and clutter in the range-velocity detection space, we now delve into the realm of **RTI** maps. **RTI** maps provide a visual representation of the intensity of received data signals over both slow time and two-way range, offering valuable insights into radar system performance and target characteristics.

A 2021 paper authored by Palamà et al. [37] outlines the processing steps necessary for raw data before generating the **RTI** map. The paper explains that the **RTI** map illustrates the intensity of the received data signal in relation to both slow time and two-way range. To ensure a uniform colour scale for qualitative analysis, normalisation is applied before plotting the map.

2.3.3 Spectrogram

Moving from **RD** maps that illustrate range-velocity detection to **RTI** maps, which depict the intensity of received data signals over slow time and two-way range, we now explore spectrograms. Spectrograms offer a time-frequency representation, presenting micro-Doppler signatures as a **Two-Dimensional (2D)** plot with time on one axis and frequency on the other.

In their study on **Unmanned Aerial Vehicle (UAV)** Micro-Doppler Signature Analysis, Herr et al. [38] utilise spectrograms to illustrate **UAV** blade flash and **Helicopter Rotor Modulation (HERM)** line phenomenology, employing different **STFT** resolutions to differentiate between **UAVs** and similar-sized targets like birds, showcasing the unique micro-Doppler signatures.

Alemdaroğlu et al.'s work [31] compares two time-frequency transformations, **STFT** and **Wigner Ville Distribution (WVD)**, for plotting spectrograms in a human walking simulator model. The results indicate that, compared to **STFT**, **WVD** exhibits interference on the spectrogram plot.

In the study 'Learned Micro-Doppler Representations for Targets Classification Based on Spectrogram Images,' Alhadhrami et al. [39] discuss the methodology used to process raw data for spectrogram production, employing **STFT** in the time domain along with a window function.

2.3.4 Constant False Alarm Rate Map

CFAR is a radar signal processing technique used in target detection where the threshold for target detection is adaptively set based on the statistical characteristics of the background clutter. In the context of **CFAR**, **Cell Averaging Constant False Alarm Rate (CA-CFAR)**, **Ordered Statistic Constant False Alarm Rate (OS-CFAR)**, **Greatest of Cell Averaging Constant False Alarm Rate (GOCA-CFAR)**, and **Smallest of Cell Averaging Constant False Alarm Rate (SOCA-CFAR)** are different approaches to estimating the background clutter level to determine an appropriate detection threshold, with **CA-CFAR** being a basic form, **OS-CFAR** involving the ordered statistical values, **GOCA-CFAR** considering the maximum values, and **SOCA-CFAR** considering the minimum values of cell averages.

Cell Averaging

Kuang et al. [40] discuss the use of **CA-CFAR** in the range, Doppler and time domains for the detection of target vessels amid sea clutter. This method is computed through the use of parallel kernels, one in

the RD domain and the other in the time domain for each of the sequential RD spectra. In Abdullah's work [41], a 'modified CA-CFAR method' is discussed that performs better than alternatives in low clutter areas to detect targets with a low power loss.

Alternatively, a paper by Zhang et al. [42] outlines a Field Programmable Gate Array implementation of a 2D CA-CFAR algorithm – in the RD domain. This paper details the hardware implementation of the CA-CFAR algorithm in a real-time system. A reduction in computation time is achieved through the avoidance of repeated calculations when sliding the reference window, thus reducing the hardware cost.

A second alternative is proposed in work authored by Zhang et al. [43]. They propose a 2D Weibull CFAR detection algorithm for the detection of undersea targets. Within this study, a method of one dimensional CA-CFAR is also discussed in theory and practical implementation. The results show that the 2D Weibull CFAR algorithm performs with a higher rate of accuracy and lower false alarm rate than the comparative 1D CA-CFAR detection algorithm.

A more conventional system is discussed by Hameed [44]. They detail the applications of a 1D CA-CFAR for range detection. This paper details the use of the CA-CFAR algorithm to compare the target return with the surrounding noise floor. Jalil et al. [45] describe the method of calculation required for the threshold and detail how the threshold should be set in order to best manage the number of real and false detections of targets.

Ordered Statistic

Sor et al.'s paper [46] published in 2018 details the method used to implement both one and 2D CA-CFAR. The method of CA-CFAR utilises different CFAR algorithms depending on the environment of the data – the three options being homogeneous, non-homogeneous and reverberation. The performance of 2D CA-CFAR is better than CA-CFAR. Both forms of CA-CFAR have better false alarm rates during clutter edges and increases probability of detection for multiple targets.

Another application is detailed in work by Abdou et al. [47] details a distributed CA-CFAR detection algorithm which is made of a number of cells divided into two reference windows – either side of the Cell Under Test (CUT). The algorithm utilises the two windows to model the noise power of the system.

Alternatively, Tien et al. [48] propose a method of calculating the threshold for CFAR processing using a 2D CA-CFAR technique. The method utilises a sliding window in the clutter area where the calculation of the threshold values is based on an independent distribution analysis of the disturbance power of each window.

Greatest of Cell Averaging

Jalil et al. [45] outlines the GOCA-CFAR algorithm, employing reference cells positioned on either side of the CUT. The noise power estimate is determined by selecting the greater of the two windows, preventing false alarms at clutter edges. Target masking, where a stronger target obscures a weaker one, is identified as a challenge in this CFAR method. Yavuz [49] reports similar issues with GOCA-CFAR,

noting additional **CFAR** loss due to homogeneous clutter. De Maio and Aubry [50] attribute this loss to fewer samples used in **GOCA-CFAR** computation compared to **CA-CFAR**.

Additional tests conducted within Wang's research [51] explores error rates in **CA-CFAR** and Greatest of Constant False Alarm Rate (**GO-CFAR**) for detecting idle channels in cognitive radio systems, both yielding error rates just under 4%. Wang proposes a **GOCA-CFAR** method to reduce the error rate to 3.5%.

Smallest of Cell Averaging

Jalil et al.'s work [45] describes the **SOC-A-CFAR** method, comparing noise power estimates from reference windows on either side of the **CUT**. The smaller average value among the windows determines the noise power for threshold calculation. De Maio and Aubry [50] note clutter suppression in either window but not if clutter exists in both. Yavuz [49] suggests **SOC-A-CFAR** is best for target masking but faces high **CFAR** loss in other scenarios.

In another study based on the improvement of clutter detection, Luna Alvarado et al. [52] derive probabilities of false alarm and detection for a **SOC-A-CFAR** detector in Weibull-distributed clutter. The study aims to analyse current detectors and improve them using realistic clutter data, concluding that the number of reference cells directly affects system performance.

Xu et al.'s 2018 paper [53] details an enhanced **SOC-A-CFAR** algorithm for target detection. The algorithm incorporates feedback to replace the **CUT** value with calculated noise if it exceeds the threshold. This change aims to reduce target power sensitivity, mitigating masking effects on detection algorithms.

2.4 NeXtRAD Radar System

The NeXtRAD Radar System is an advanced radar technology operating in L and X-Bands. It offers hardware and software enhancements, including polarimetric operation and improved control systems. Its measurements encompass a wide range of applications, including target and clutter characterisation, micro-Doppler signature capture for **UAV** discrimination, and improved pulse resolution. NeXtRAD is a significant advancement in radar technology, providing valuable data for various radar applications. This section conducts a thorough review of the existing literature on NeXtRAD and its predecessor, NetRAD, exploring the differences and similarities between the two systems. Additionally, it addresses relevant studies that delve into both avenues.

2.4.1 NetRAD

The multistatic radar system NetRAD, developed by University College London (**UCL**) and the University of Cape Town (**UCT**), comprises three fully coherent S-Band (2 to 4 GHz) nodes – one transceiver node and two receiver nodes – synchronised through the use of clock and trigger cables or **GPSDOs** with programmable triggers. Inggs et al.'s work [12] in 2011 details the implementation of **GPSDOs** on each node of the NetRAD system, replacing the existing trigger cables. The trigger cables restricted bistatic baselines to 100 metres, whereas the **GPSDO** implementation allows for full

multistatic operation. **GPSDOs** enable wireless communication between the control software and the nodes, facilitating remote triggering and dynamic adaptation of data capture parameters.

The NetRAD system was employed in 2012 to compile a multistatic sea clutter and vessel database from data collected in the United Kingdom and South Africa. Inggs et al. [13] assembled the database to support the development of processing techniques for this form of data and enhance the understanding of sea clutter and vessel properties in a multistatic radar system.

In 2014, the work by Palamà et al. [54, 55] concentrated on the statistical and spectral analysis of sea clutter spikes captured by the NetRAD system in both monostatic and bistatic configurations. The research underscores the superior statistical properties of bistatic data, demonstrating that bistatic data yields more precise results in terms of spike width and the interval between two consecutive spikes compared to monostatic data. Ritchie et al.'s paper [56] follows a similar trend for bistatic data and suggests the implementation of Doppler **CFAR** detection for improved sensitivity.

In 2016, Hoffmann et al. [57] published a paper presenting an approach for detecting and tracking a micro-**UAV** using the NetRAD system. The study emphasises the importance of micro-Doppler signatures in discriminating between targets and ground clutter.

Machine learning was employed alongside the NetRAD sea clutter and vessel database to assess the viability of discriminating between targets and sea clutter. Callaghan et al.'s paper [58] in 2017 explores the use of two machine learning techniques, revealing that the **k-Nearest-Neighbour (kNN)** algorithm outperforms the **Support Vector Machine (SVM)** counterpart. Fioranelli et al.'s paper [59] uses NetRAD and machine learning to discriminate between armed and unarmed individuals based on human micro-Doppler signature data.

In 2017, the research conducted by Coetze and Inggs [60] explores the use of narrow bandwidth NetRAD data to produce high spatial resolution imagery. They achieve this by applying tomographic principles to a multi-site radar system that illuminates a target across a full 360-degree angular range. This approach involves taking measurements from various geometric positions with the radar system and using tomographic processing to mitigate the challenges posed by poor ranging, low range resolution, and the consistency issues associated with an extremely narrow bandwidth.

Inggs et al.'s paper [61] in 2017 discusses the data archive system for the NeXtRAD system, based on lessons learned from the NetRAD system. The paper highlights the re-inclusion of functionality that allows raw data to be grouped by day within recording campaigns, with improvements including the ability to search raw data for results based on specific recording conditions. The raw data and supplementary data are merged into a single **Hierarchical Data File version 5 (HDF5)** archive file, with important data parameters extracted and stored in a searchable database.

2.4.2 NeXtRAD

Transitioning to NeXtRAD represents a leap in radar technology. This section delves into the hardware and software enhancements featured in the NeXtRAD system, operating at L and X-Bands. These advancements encompass innovations in radar design, antenna technology, signal processing, and calibration techniques, marking a notable progression from the earlier NetRAD system.

In a 2014 review conducted by Inggs et al. [62], the superiority of multistatic radar compared to its monostatic counterparts was underlined. It stressed the significance of comprehending target and clutter characteristics, particularly for detecting low-signature targets. The study scrutinized NeXtRAD's parameters, showcasing enhancements over NetRAD, such as polarimetric operation in the L and X-Bands, improved Command and Control systems, and provisions for environmental resilience. Additionally, this review anticipated forthcoming measurement campaigns, including evaluations of small maritime targets and comprehensive multistatic multi-band dual-polarised sea clutter measurements across diverse meteorological and sea conditions.

A design for a **Slotted Waveguide Antenna (SWGA)** was detailed by Alhuwaimel and Tong [63] in 2015, demonstrating improved gain, **Front-to-Back Ratio (F/B)**, and elevation beamwidth compared to traditional **SWGA** arrays. This design, applied to the NeXtRAD X-band antenna, reduced the number of stacked waveguides needed to achieve the desired elevation beamwidth.

Alhuwaimel et al.'s paper [64] in 2017 reported NeXtRAD's first measurements, covering sea clutter, targets, and attempts at radar system calibration using corner reflectors. The paper highlighted the successful achievement of close to a 3-metre pulse resolution in preliminary tests of the low power loop back, conducted with a linear chirp of 50 MHz.

Trials carried out in 2018 were reported by Inggs et al. [16], focusing on detecting and tracking targets in sea clutter and micro-Doppler. The trials included recording micro-Doppler measurements using a drone target and sea clutter data with multiple bistatic node angles and grazing angles, resulting in successful datasets captured over two weeks.

Palamà et al.'s 2019 paper [17] presented results from measurements of small **UAV** signatures at L and X-band using NeXtRAD. The subsequent analysis by Beasly et al. [65] in 2020 discussed the first interleaved L and X-band measurements of multiple **UAVs** in simultaneous monostatic and bistatic configurations, revealing the potential of **UAV** micro-Doppler signatures for distinguishing between devices and similar-sized radar cross-section targets like birds.

In 2019, Sandenbergh and Inggs [66] concluded from a paper that discussed the functionality of **GPSDOs** in NetRAD and NeXtRAD multistatic radars. The paper addressed synchronisation requirements for networked radar, including time, frequency, phase synchronisation, and close-in phase noise. It introduced the concept of **Line-of-Sight (LOS)** phase compression to account for bistatic phase drift and noise, demonstrating significant improvements in phase noise through this approach.

2.5 Quick-Look Processor

The Quick-Look analysis method is employed for real-time visualisation, offering a swift overview of data to extract initial insights without the need for a comprehensive or detailed examination. This approach serves as a rapid means to evaluate the fundamental characteristics of a signal or dataset.

This section explores the landscape of **QLP** systems, focusing on those designed for the NextRAD radar system and similar systems across the globe.

2.5.1 NeXtRAD Implementations

In Jonkers' thesis [67], a signal processing infrastructure was developed in the Julia Programming Language, focusing on methods such as pulse compression, pulse-Doppler processing, and a [Least Mean Square \(LMS\)](#) Filter. The study conducted extensive testing, revealing that the multi-core implementation of pulse compression was slower due to Julia's reliance on substantial [Random Access Memory \(RAM\)](#), while pulse-Doppler processing showed improved efficiency.

Another implementation is discussed by Jordan et al. [68] in a 2018 paper, where they introduced NeXtLook, a C++-based [QLP](#) supporting [PRFs](#) over 15 kHz. The system demonstrated efficient pulse compression and Doppler processing, with simulated results indicating processing times of less than 12 seconds. Furthermore, Inggs et al. [69] explored the feasibility of implementing computationally intensive algorithms on an [Infrastructure-as-a-Service \(IaaS\)](#) platform, specifically using [Amazon Web Services \(AWS\)](#). The study found comparable or improved performance compared to personal computers, emphasising the potential of [IaaS](#) for radar data processing. Alhuwaimel et al. [64] developed a multi-threaded implementation of NeXtLook, achieving a significant speed-up compared to the single-threaded version, with a focus on its application to the NeXtRAD [QLP](#).

2.5.2 Alternative Implementations

Below, various real-time radar processing methods are discussed. While NeXtLook and Jonkers' Julia-based infrastructure optimise pulse compression and Doppler processing, alternative implementations span applications like SAR image generation and planetary transit detection, showcasing diverse real-time processing in the radar field.

In the study by Zhang et al. [70], a [QLP](#) design was proposed for Synthetic Aperture Radar (SAR) image generation, specifically for [Geosynchronous Earth Orb \(GEO\)](#) and [Low Earth Orbit \(LEO\)](#) SAR. The software framework comprised data receiving and file reading software, image processing software, data display and interpretation software, and centralised control software. Implementation on a multi-node system allowed effective management of performance and resource requirements through the control software.

Setiadi et al. [71] presented the development of a [QLP](#) for a Circularly Polarised [SAR](#), utilising a modified version of the [SAR](#) spectral analysis (SPECAN) algorithm. The [QLP](#) subsystem incorporated a file processing loop for reading raw signal and motion data, converting the raw signal into images using the [QLP](#) algorithm for storage. The [QLP](#) demonstrated the ability to produce low-resolution images from extensive raw data, outperforming the original processing pipeline in terms of speed.

Massachusetts Institute of Technology ([MIT](#)) developed a Quick-Look Pipeline [72] for the [Transiting Exoplanet Survey Satellite \(TESS\)](#) mission by [National Aeronautics and Space Administration \(NASA\)](#). Huang et al. [73] presented light curves for a magnitude-limited sample of stars processed by the [MIT](#) Quick-Look Pipeline. The pipeline, designed to rapidly process [TESS](#) data as it is transmitted to Earth, combines data from each [TESS](#) sector to perform a multi-sector search for transiting exoplanets. The Quick-Look pipeline has successfully identified and alerted researchers to potential planetary candidates throughout the [TESS](#) Primary Mission [74, 75], observing nearly 15 million stars during its first two years of operation.

Chapter 3

Quick-Look Processor Design

This chapter details the design component of the [QLP](#). The core objective was to process data from a simulation conducted on the NetRAD and NeXtRAD systems before plotting the results in the range, time and Doppler frequency domains. This design is to create a tool which can be used alongside the radar system to allow for real-time, or near real-time, processing of the data. The implementation of this tool is to allow for a greater understanding of the data being recorded with the purpose of detecting targets and relevant information about said targets, such as their range and frequency of operation.

3.1 System Structure

3.1.1 System Overview

The entire system comprises a data handling pipeline, depicted in Figure 3.1, that channels processed data into plotting functions for generating maps. This system has been implemented in both a MATLAB-based code structure and a [GUI](#) application.

The [QLP](#) system comprises two core functional blocks for data handling: pre-processing and signal processing. The data handling component of [QLP](#) is responsible for taking in raw data files and converting them into a compatible format. Subsequently, it processes the data to generate maps, each displaying distinct results derived from the data.

In both the pre-processing and signal processing sections, each sub-block has been meticulously crafted using a modular MATLAB code construction approach. This modular design simplifies individual tasks and enhances the system's overall design, debugging, and testing.

The upcoming sections will provide in-depth information on these sub-blocks, with dedicated subsections for each.

The pipeline supplies functions to create various maps within the system. The visualisation section outlines the design of the plotting algorithm for different types of maps, including [RTI](#) maps, [RD](#) maps, spectrograms, [CFAR](#) maps, and [CFAR](#) Level plots.

The [GUI](#) application complements the code-only version of the project, offering more accessible parameter adjustments. It allows users to interact with the project without compromising processing efficiency or speed.

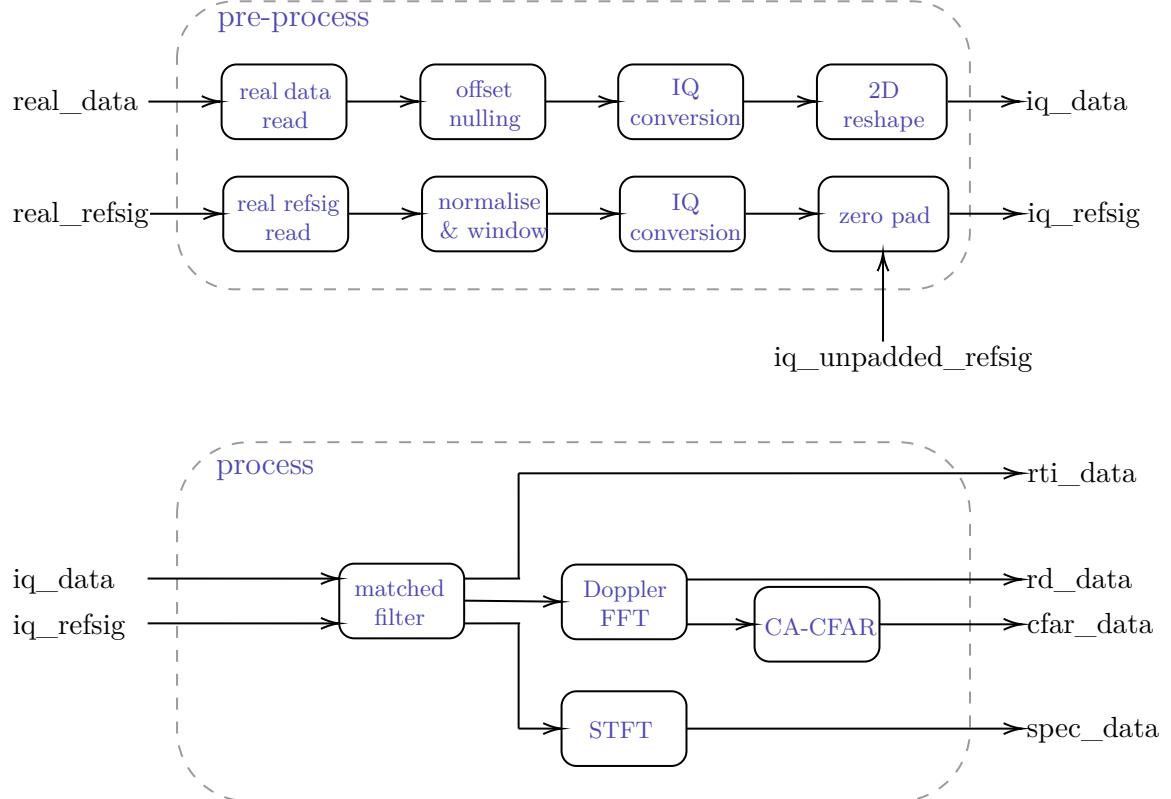


Figure 3.1: Block diagram showing data handling part of QLP.

3.1.2 Radar Theory

Illustrated in Figure 3.2, NeXtRAD is a multistatic PDR system comprising three nodes, offering dual-band functionality and complete polarimetric sensing.

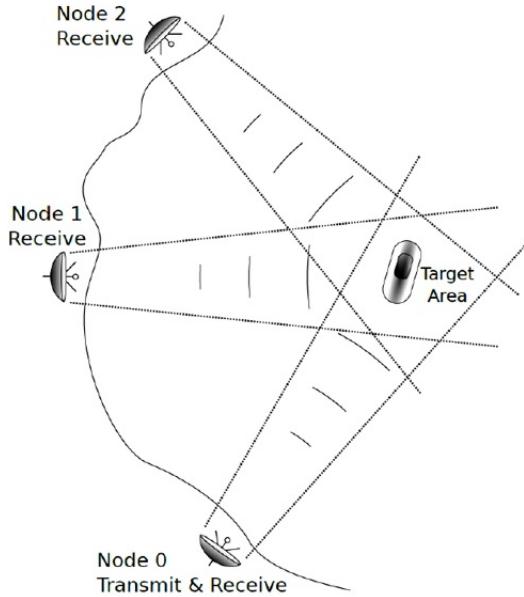


Figure 3.2: Operational deployment configuration of NeXtRAD taken from [1].

This radar system can function in a three-node configuration, consisting of one transmit-receive node

and two passive receiver-only nodes, as noted in [16]. In simpler terms, this setup involves one monostatic node and two bistatic nodes, as discussed in [67].

Recordings from each node are stored in separate files for processing. The monostatic node data includes simulation results and simulation parameters stored alongside the raw recorded results. NeXtRAD stores these measurements in [HDF5](#) archive files, combining raw radar data and simulation attributes in a single file.

These simulations use a Linear Frequency Modulated chirp pulse with a 50 MHz bandwidth and a 5 μ s duration.

NeXtRAD operates in the X-band at 8.5 GHz and the L-band at 1.3 GHz. Each node features three [Analogue-to-Digital Converter \(ADC\)](#) channels - one in the L-band and two in the X-band, as specified in [65].

Similarly, NetRAD is a three-node system, configured as depicted in Figure 3.2. NetRAD operates in the S-band at 2.4 GHz.

The operating conditions for both NeXtRAD (a) and NetRAD (b) are displayed in Figure 3.3. This figure illustrates that some parameters are shared between the two radar systems, while others, such as frequency of operation and antenna polarisation, differ between them.

Parameter	Value	Comments
X Band freq range	8.5 to 9.2 GHz	In practice, limited by power amplifier to 8500 MHz.
L Band	1.2 to 1.4 GHz	Full band possible.
Polarimetry X Band	co- and cross-polar	2 Receive channels
Polarimetry L Band	co- or cross-polar alt PRI	1 Receive channel
Instantaneous BW	50 MHz	Both bands
Peak power X Band	400 W	10% Duty cycle
Peak power L Band	1.6 kW	20% Duty cycle
X Band NF	3.5 dB	At IF output
L Band NF	6.3 dB	At IF output
Max PRF	Depends on blind range and amplifier duty cycle	
Max Pulse length		

(a) NeXtRAD [16]

Parameter	Value	units/comments
Frequency	2.4	GHz
Bandwidth	50	MHz
Nominal range resolution	3	m
PRF	50–3	Hz – kHz
Pulse length	0.1–10	μs
Waveforms	digitally-generated: linear FM chirp, Barker codes, polyphase codes, etc	
Transmit power	500	W (one node, others 200 mW)
Antenna gain	23	dBi
Antenna beamwidth	10 × 10	degrees
Antenna polarisation	linear	can be set V or H

(b) NetRAD [12]

Figure 3.3: System parameters for each of the radars.

3.2 Pre-processing

3.2.1 Overview

The pre-processing pipeline was initially designed for the NeXtRAD system and its data files. However, due to the unavailability of NeXtRAD [HDF5](#) data files, this subsystem was adapted for use with NetRAD data from previous trials in 2010 and 2011 [76]. Consequently, the pipeline's task was to read and prepare the file data, ensuring it matched the format expected for NeXtRAD data after pre-processing.

Graphically depicted in Figure 3.4, this pre-processing subsystem handled three distinct raw data files: recordings, reference signals, and simulation parameters. After reading these files, the pipeline

enhanced the data and reference signal values by applying offset nulling and signal normalisation. It then utilised a Hilbert transform to convert the data into its **In-phase and Quadrature Components (IQ)**, representing it in complex form. The data values were further processed by reshaping them into a **2D** matrix, modelling pulses in slow-time and samples per pulse in fast-time. The reference signal was zero-padded to match the signal length with the number of samples taken per pulse.

As an alternative, instead of retrieving the real reference signal from a file, the **IQ** reference signal stored in the parameter file could be used as input for the **QLP**. In this scenario, the steps prior to zero-padding the reference signal were omitted, and the final **IQ** reference signal was obtained from the pre-processing pipeline after applying the same zero-padding algorithm to the signal.

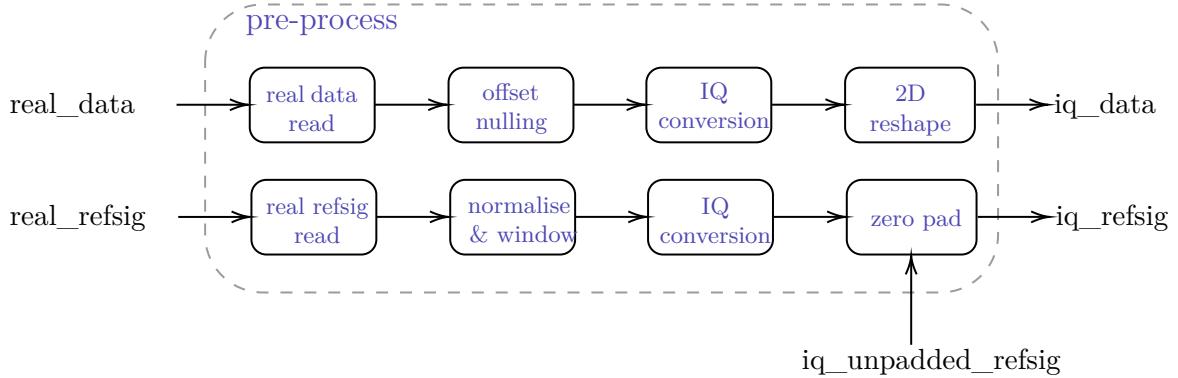


Figure 3.4: Block diagram showing pre-processing section of QLP.

3.2.2 Pre-processing Theory

Hanning Window

The Hann (or Hanning) window, represented by Equation 3.1, is a mathematical function employed in signal processing. This function exhibits a raised cosine shape, smoothly transitioning from zero at the signal's edges to a maximum value at its centre. Its purpose, as discussed in [77], is to reduce spectral leakage in frequency analysis applications.

$$w_{Hann}(n) = \begin{cases} 0.5 - 0.5\cos(\frac{2\pi n}{N-1}) & \text{if } x \in [0, N-1] \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

Hilbert Transform

Work by Dugundji [78] details the process of conversion of a real-valued function $u(t)$ where $t \in (-\infty, \infty)$. In this case the Hilbert Transform $\hat{u}(t)$ is defined as:

$$\hat{u}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{u(\tau)}{t - \tau} d\tau \quad (3.2)$$

Mathematically, this is expressed as the convolution of $u(t)$ with $\frac{1}{\pi t}$. This operation calculates the imaginary part of the complex-valued function, resulting in a complex function denoted as $u(t) + j\hat{u}(t)$.

Alternatively, for the Hilbert transform in the frequency domain of the original signal, after Fourier transformation, negative frequencies are set to zero, and positive frequencies are multiplied by $-j$, introducing a -90° phase shift in the time domain. This transformed spectrum provides both the amplitude and phase information required for envelope detection and phase analysis.

Data Reshaping

As shown in Figure 3.5, the process transforms a vector of $m \times n$ elements into a 2D matrix. In this matrix, there are n columns, each representing individual samples, and m rows representing successive pulses. This transformation arranges the data, where each row corresponds to a pulse, and each column corresponds to a specific sample moment. This organisation simplifies the analysis and visualisation of the radar signal's time and range characteristics.

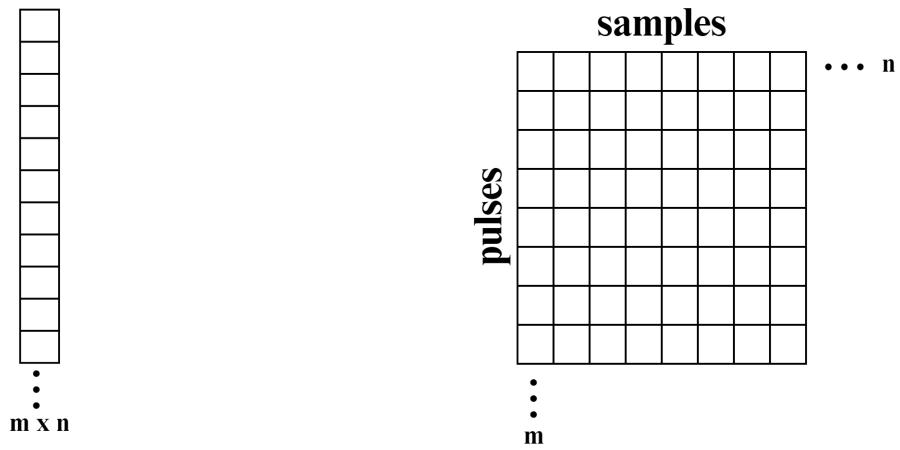


Figure 3.5: Before and after of reshape of data values into 2D matrix.

3.2.3 Reading data from file

Simulation recordings for the NetRAD system are stored as binary (.bin) files, while simulation parameters are saved as MATLAB (.m) files. Each node in the simulation saves its measurements in separate data and parameter files for later processing.

Another approach in the system involves using the reference signal stored in the parameter file. In this scenario, the reference signal, along with other necessary QLP parameters, is retrieved from the file. This bypasses the need to read the real reference signal, perform normalisation, and apply windowing.

The binary file format is chosen for NetRAD data storage to enhance data read/write efficiency and reduce file size, as noted in [79]. When raw data is read from these binary files, the data values are real floating point values stored as 16-bit unsigned integers.

The QLP system's database, collected in the Western Cape, South Africa, in 2010 and 2011, encompasses recordings with varying pulse counts, ranging from 1000 to 130,000 pulses.

The number of data values read from the .bin file depends on two factors: the number of samples per pulse obtained from the simulation parameters file and the desired number of pulses for processing. The pulse count in a single data file is determined by dividing the recording's time length in seconds

by the [PRF](#). The value to be read from the file is adjustable, with a maximum value retrieved from the parameter file.

It's important to note that when reading NeXtRAD data files, the operation of this block would likely differ due to the structure of [HDF5](#) files.

3.2.4 Enhancing data quality

Eliminating the offset from a data vector entails subtracting a constant from each element, ensuring the mean of the resulting vector is zero. The offset signifies a systematic bias in data measurement, making its removal crucial for accurate radar data analysis and interpretation.

Normalisation involves scaling data values to follow a standard distribution. This process maintains uniformity in data scale, simplifying analysis and preventing any dominant features due to larger magnitudes. It is achieved by dividing each data value by the maximum magnitude value in the real data.

Applying a windowing function to the reference signal reduces abrupt transitions, minimises frequency domain sidelobes, and offers more precise insight into the signal's spectral content. The Hanning window is chosen for the reference signal due to its desirable attributes, including smooth edge tapering, low sidelobes, and an optimal balance between mainlobe width and sidelobe suppression. This window effectively concentrates energy around the central frequency of interest, reducing spectral leakage, and improving accuracy in the matched filtering implemented in the pulse compression section.

3.2.5 Converting from real to complex

The conversion from real data to complex data utilises the Hilbert transform, as discussed in Section [3.2.2](#) for both the raw signal data and the reference signal. This transformation is achieved using the built-in `hilbert()` function, taking the real data vector as input and returning the corresponding [IQ](#) data.

The necessity for this transformation arises from the data recording method employed by the NetRAD system. Another approach, pairing every two real values into a single complex number to create a new vector with half as many elements, would lead to an incorrect reduction in samples per pulse. The NetRAD system records raw data as real data, requiring the Hilbert transform to generate the correct complex numbers and maintain the original number of samples per pulse.

To enhance system performance, a custom function was initially developed instead of using the MATLAB library. However, it was found that the library function provided a faster implementation of the transform. An alternative approach involving [2D](#) reshaping before the Hilbert transform and using a loop for column-wise conversion was also explored. However, the loop implementation introduced overhead, resulting in a slower overall conversion and reshaping process.

3.2.6 Two-dimensional reshaping

The reshaping process, explained in Subsection [3.2.2](#), was accomplished using a MATLAB code function provided in Listing [3.1](#). This function takes the [IQ](#) data vector as input, along with the pulse count

and the number of rows. To ensure the correct data format, the reshaped matrix was transposed due to the function's computation of columns from the vector data, and the data format's arrangement, with the number of samples per pulse across the columns.

The matrix generated by the code in Listing 3.1 is illustrated in Figure 3.5. In this representation, the columns correspond to individual range bins, while the rows correspond to individual pulses.

```
1 shaped_data = reshape(iq_data, num_samples, num_pulses).';
```

Listing 3.1: MATLAB code for reshaping complex data values.

3.2.7 Zero padding

Zero padding involves adding zeros to the reference signal to align its length with the number of samples in each pulse. This ensures that the correlation process covers the entire signal length, achieving the desired results in subsequent matched filtering.

Adding zeros to the front of the reference signal, rather than the back, helps align it with the actual beginning of the IQ data. This minimises the risk of losing critical information during the correlation, optimising the process.

3.3 Signal Processing

3.3.1 Overview

The signal processing subsystem enhances radar and sensor technologies by incorporating several key components: Pulse Compression, Doppler Processing, Micro-Doppler Processing, and CFAR Processing. Pulse Compression, employing a matched filter, refines range resolution by correlating received signals with the transmitted pulse shape. Doppler Processing utilises a Doppler FFT to extract frequency information for target motion analysis. Micro-Doppler Processing enhances target classification by examining small-scale motions like rotations or vibrations through STFT. CFAR Processing, employing CFAR, reduces false alarms by adaptively setting detection thresholds. This comprehensive subsystem equips radar and sensor systems with the capability to detect, track, and classify targets accurately while minimising false alarms. Pulse integration combines multiple radar pulses to enhance SNR, a critical factor for improving map performance.

The pipeline generates matrices to store processed data, facilitating the creation of essential radar visualisations. These matrices serve as the basis for plotting various maps, including RTI maps, RD maps, Spectrograms, CFAR maps, and CFAR Level maps. RTI maps depict target intensity over time and distance, offering a comprehensive view of target behaviour. RD maps reveal the relationship between range and Doppler frequency, aiding in target detection and tracking. Spectrograms provide insights into both time and Doppler frequency domains, enabling the analysis of complex signals. CFAR maps illustrate detection outcomes, highlighting potential target locations, while CFAR Level maps visualise the adaptive thresholding process used to minimise false alarms. These matrices and associated maps are vital for understanding and interpreting radar data.

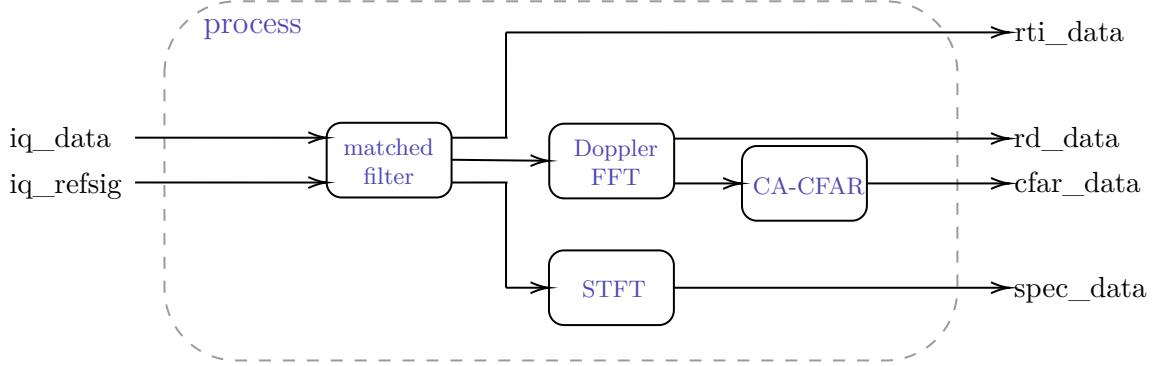


Figure 3.6: Block diagram showing the signal processing section of QLP.

3.3.2 Processing Theory

Matched Filter

The matched filter operation, as described in Equation 3.3, performs a cross-correlation between the known (broadcast signal) $s(t)$ and the measured signal $x(t)$, as discussed in [80]. The matched filter produces a peak in the measurement at the time when the known signal is present. This operation is analogous to convolving the measured signal with the time-reversed and complex-conjugated form of the known signal.

$$y(t) = x(t) * s(-t) = \int_{-\infty}^{\infty} x(\tau)s(\tau - t) d\tau \quad (3.3)$$

The application of this mathematical function in the continuous time domain is defined in Equation 3.3, which involves the **Continuous-Time Fourier Transform (CTFT)** within a convolution function. For discrete, digitised signals, the matched filter utilises the **DFT**, which is expressed as:

$$y[n] = x[n] * s[-n] = \sum_{k=-\infty}^{\infty} x[k]s[k-n] \quad (3.4)$$

Short-Time Fourier Transform

According to [81], the **STFT** focuses on a brief segment of a longer data sequence and calculates its Fourier transform. This is achieved by multiplying the longer time function by a rectangular window function, which isolates the desired short sequence without altering it.

The **STFT** is a time-frequency analysis technique used to study how the frequency content of a signal evolves over time. It involves segmenting a signal into shorter, overlapping sections and computing the **FFT** for each segment. The choice of window size and overlap parameters is critical. A larger window size provides better frequency resolution but reduced time resolution, while a smaller window size offers improved time resolution but less accurate frequency resolution. Overlapping segments help reduce information loss, although it increases computational load. Determining the optimal window size and overlap parameters depends on the specific application, finding a balance between time and frequency precision.

Cell-Averaging Constant False Alarm Rate Threshold

CA-CFAR, a target detection technique, flexibly adapts the detection threshold to the surrounding noise level, maintaining a consistent probability of false alarms P_{fa} even in the presence of changing background clutter or interference. This adaptive threshold setting ensures accurate target detection while mitigating the risk of false alarms. Figure 3.7 illustrates the structure of a **CFAR** window.

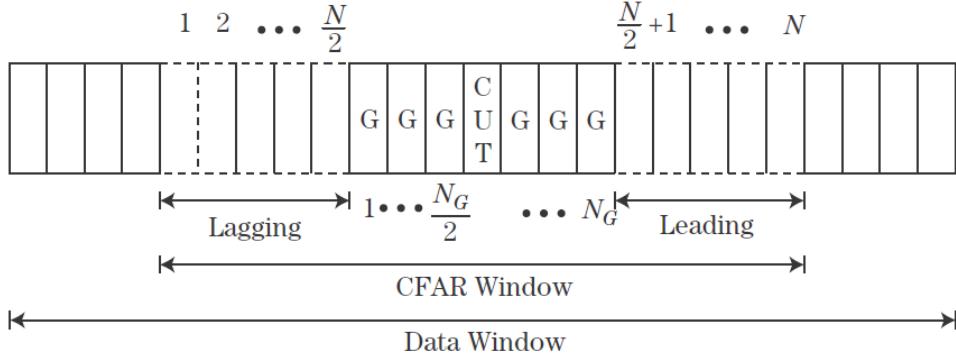


Figure 3.7: 1D CFAR window taken from [2].

The threshold calculation consists of two components: a constant determined by the implemented **CFAR** type α and the mean of the interference power σ_i^2 . For **CA-CFAR**, the threshold is calculated as follows:

$$T = \alpha_{CA} \hat{\sigma}_i^2 \quad (3.5)$$

where

$$\alpha_{CA} = N[P_{FA}^{-1/N} - 1], \quad (3.6)$$

$$\hat{\sigma}_i^2 = \frac{1}{N} \sum_{n=1}^N z_n \quad (3.7)$$

These equations enable the computation of the threshold for each **CUT** and its reference cells, denoted as $z = z_1, z_2, \dots, z_N$. It is evident from the equations that the determination of the detection threshold relies on both the average power level and the specified probability of false alarms.

3.3.3 Pulse Compression

Pulse compression enhances target detection and resolution in the range domain through the correlation of a transmitted pulse with the received echo, achieved using processing methods like a matched filter or an **RMMSE** algorithm.

In this pipeline's design, a matched filter was chosen for implementation due to the need for real-time or near real-time system operation. Using alternative methods, such as the **RMMSE** algorithm with more advanced and computationally intensive techniques, would result in longer processing times. The matched filter is a straightforward and computationally efficient pulse compression technique.

The matched filter implementation is most efficient when the transmitted signal is known and can be directly compared to the received echo, enabling the differentiation between targets and noise within the radar path.

For the matched filter, the samples measured for each pulse are directly compared with the zero-padded reference signal. The convolution of the known signal and the measured signal, as described in Equation 3.4, is computed for each pulse in the data and stored in a matrix containing the convolution results. The convolution operation involves calculating the [FFT](#) for each of the signals before performing element-wise multiplication of the two rows. The code for this functionality is displayed below:

```

1 for row = 1:num_pulses
2     matched_filter = ifft(fft(shaped_data(row, :)) .* fft(iq_refsig));
3     matched_filter_output(row, :) = matched_filter(1:length(iq_refsig));
4 end

```

Listing 3.2: MATLAB code for implementation of matched filter.

Before utilising `iq_refsig` in the convolution calculation described above, the known signal needed to be time-reversed and complex-conjugated, following the matched filter theory explained in 3.3.2.

The matched filter output for a single pulse is depicted in Figure 3.8. This plot was generated using the files ‘e10_10_16_1800_40_P1_1_130000_S0_1_2047_node3.bin’ and ‘e10_10_16_1800_40_P1_1_130000_S0_1_2047_node3.m’ as input to the model.

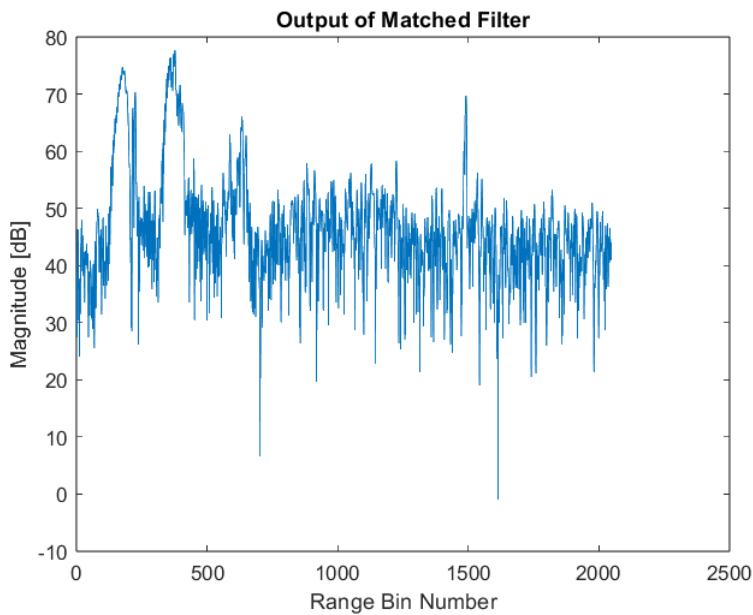


Figure 3.8: Matched filter output for a single pulse.

3.3.4 Doppler Processing

Doppler processing is a valuable signal analysis technique used for detecting and distinguishing moving targets from stationary clutter in measured signals. [MTI](#) filters out stationary objects by capitalising

on their absence of Doppler frequency shifts, preserving the Doppler signatures of moving targets. In contrast, Pulse-Doppler processing combines Doppler filtering with pulse compression, allowing the simultaneous measurement of target range and radial velocity. This enables the detection and tracking of moving objects, determination of their speed and direction, and filtering of unwanted stationary clutter.

The **MTI** system functions entirely within the temporal domain by employing a high-pass filter on the acquired data. This approach boasts the advantage of being computationally efficient; however, it offers only limited insights into the desired targets. The methodology entails the utilisation of a linear filter on the slow-time data sequence to suppress extraneous clutter components present in the data source, as indicated by the reference [2]. Subsequent to the high-pass filtering operation, a threshold-based detector assesses whether the remaining signal corresponds to a target for detection and tracking or mere noise embedded within the signal.

MTI processing is primarily concerned with detecting the mere presence or absence of moving targets and does not yield in-depth information regarding these targets. It is worth noting that potential challenges arise from fluctuations in transmitter amplitude and phase drift within the transmitted or received signals, as elaborated upon in [2]. Externally, the primary constraint on enhancing **MTI** performance is determined by the breadth of the clutter spectrum. A broader spectrum may permit a larger portion of clutter energy to remain outside the **MTI** filter, resulting in a reduced level of clutter energy being filtered out.

In contrast to **MTI**, Pulse-Doppler processing distinguishes itself by substituting slow-time domain filtering with an explicit spectral analysis of the slow-time data for each range bin. The outcome of Pulse-Doppler processing materialises as an **RD** matrix characterised by dimensions in both fast-time and Doppler frequency. Within the **RD** data, the energy stemming from moving targets is effectively isolated from clutter and competes solely with noise within the same Doppler bin. This spectral analysis is accomplished by employing a **DFT** on each of the slow-time data rows through the utilisation of a **FFT** algorithm, as visually represented in Figure 3.9. The **DFT** bins correspond to Doppler frequencies ranging between $-\text{PRF}/2$ and $+\text{PRF}/2$.

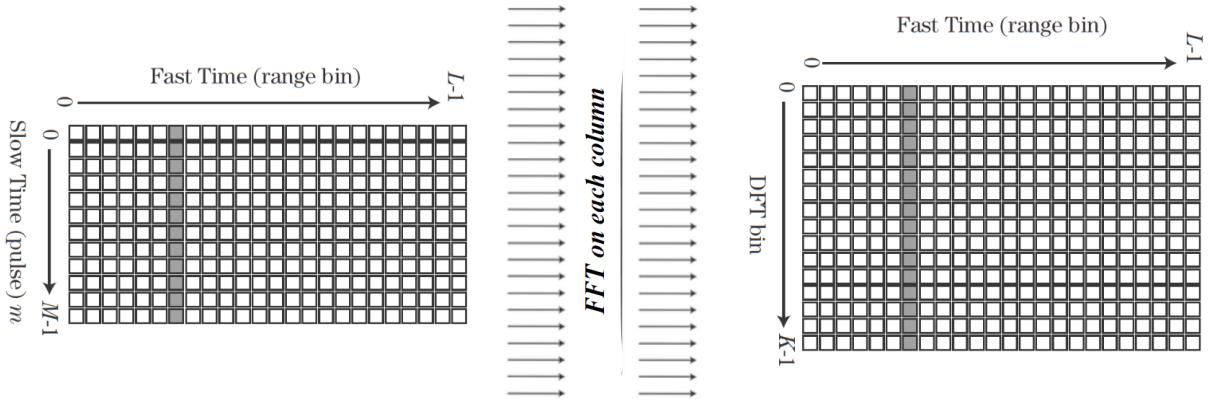


Figure 3.9: Conversion to RD data through the application of DFTs.

In this design, Doppler processing was implemented using pulse-Doppler processing. While pulse-Doppler processing demands more computational resources, it yields more information and provides a higher **Signal-to-Interference Ratio (SIR)** improvement. The implementation of this processing method was carried out to generate the **RD** map. The following MATLAB code line enables the application of pulse-Doppler processing as illustrated in Figure 3.9.

```
1 rdm = fftshift(fft(matched_filter_output, [], 1));
```

Listing 3.3: MATLAB code for implementation of pulse-Doppler processing.

The provided listing outlines the use of the ‘`fft()`’ function in a column-oriented fashion to compute the slow-time **FFT** on each of the provided columns. The ‘`fftshift()`’ operation is employed to centre the frequency components of the **FFT** output. This ensures that low-frequency components are positioned in the middle of the spectrum, while high-frequency components are placed at the edges.

3.3.5 Micro-Doppler Processing

Micro-Doppler processing facilitates the analysis of subtle Doppler frequency shifts caused by individual reflector motion within radar targets. This approach involves decomposing the measured data into various Doppler components, providing insights into the movement, rotation, and discrimination of different target features, enhancing target classification accuracy.

In this pipeline, micro-Doppler processing is realised through a series of **STFTs** applied to a specific range bin where a detected target is located. The **STFT** is applied to a user-selected range bin, and the processing details are explained in Section 3.3.2. The window size and percentage of overlap are customisable values that influence processing performance and results. This flexibility enables the identification and measurement of subtle Doppler shifts within radar targets, aiding in the comprehensive understanding of target dynamics and refining target feature discrimination. Customisation of processing parameters optimises performance, enhancing target classification accuracy and overall radar system effectiveness for a given set of data files.

3.3.6 Constant False Alarm Rate

CFAR, used for target detection amidst varying clutter and noise levels, involves calculating a detection threshold based on an estimate of local clutter or noise in radar data. This approach maintains a constant probability of false alarms, even as clutter conditions change across different radar positions.

CA-CFAR is a specific variant of **CFAR** that segments radar data into cells, computing the average power level within these cells to establish the detection threshold, as detailed in Section 3.3.2.

The choice of **CA-CFAR** in this design balances the computational intensity of threshold factor calculation and algorithm performance. In [2], **CA-CFAR** is described as ‘relatively simple’ based on the required detection threshold calculation. The algorithm is designed for operation in relatively benign interference or target environments.

There’s a trade-off between robustness, leading to increased **CA-CFAR** loss, greater complexity, higher computational load, and the need for iterative solutions for **CFAR** constants. Implementing **GOCA-**

CFAR instead of CA-CFAR introduces additional CFAR loss due to the fewer samples used to determine the GOCA-CFAR statistic for a specific dataset. While GOCA-CFAR does reduce false alarms at the clutter edge by biasing the threshold above that of CA-CFAR, the design considered the increased computational cost to be an unfavourable trade-off.

The SOCA-CFAR and OS-CFAR algorithms were designed to address the mutual target masking phenomenon. SOCA-CFAR's practical implementation is limited because it can suppress mutual target masking only in cases where interfering targets are limited to either leading or lagging windows. When interfering targets are present in both windows, its performance is significantly affected. Additionally, SOCA-CFAR exhibits relatively large CFAR loss compared to CA-CFAR and GOCA-CFAR, varying based on the number of reference cells and the probability of false alarm.

In a test by [2], adjusting the number of reference cells and the probability of false alarm showed that GOCA-CFAR had a CFAR loss peak 0.3dB larger than CA-CFAR, while SOCA-CFAR's loss ranged from 0.4dB greater than CA-CFAR to 4.6dB greater, as shown in Table 3.1.

The OS-CFAR algorithm exhibits a CFAR loss that falls between GOCA-CFAR and SOCA-CFAR. The implementation of OS-CFAR does not use guard cells, so all cells within the window form the reference cells for detection threshold calculation. Therefore, implementing OS-CFAR results in a larger number of windows per iteration loop or larger reference windows, introducing additional processing overhead and potentially impacting the system's overall processing time.

CFAR LOSS (dB)				
N	P_{FA}	CA-CFAR	GOCA-CFAR	SOCA-CFAR
8	10^{-4}	2.7	3.0	5.3
16	10^{-4}	1.3	1.5	2.3
24	10^{-4}	0.9	1.0	1.5
32	10^{-4}	0.6	0.8	1.0
8	10^{-6}	4.3	4.6	8.9
16	10^{-6}	2.0	2.2	3.8
24	10^{-6}	1.3	1.5	2.3
32	10^{-6}	1.0	1.1	1.7

Table 3.1: CFAR Loss associated with CA-CFAR, GOCA-CFAR and SOCA-CFAR, from [2].

In this system, two methods were chosen for implementing the reference window: 1D range CA-CFAR and Doppler CA-CFAR algorithms. These methods enabled CFAR implementation on the RD matrix, either in the range (fast-time) or Doppler (slow-time) orientation.

For the fast-time dimension, each row of the matrix was iterated individually. Within each row, a 1D CFAR window systematically slid through, processing each CUT.

In the slow-time dimension, columns were processed sequentially. A 1D CFAR window was applied to slide through each column and evaluate the CUTs.

Adjustments were made based on the CUT's location, with modifications to the allocation of guard cells and range cells.

3.3.7 Pulse Integration

When radar detects a target, it may reflect on multiple pulses in the measured signal, increasing **SNR** and detection probability. This process, known as pulse integration or coherent integration, combines these pulses, preserving their phase relationship and amplifying the signal amplitude.

In this pipeline, pulse integration is an optional user feature that compiles pulses, improving the differentiation and measurement of detected targets. This process enhances the radar's ability to discern and accurately assess target information over multiple pulses, ultimately leading to more precise results.

3.4 Visualisation

3.4.1 Overview

The visualisation section in this design report holds a pivotal role in the transformation of processed data into meaningful and interpretable maps. This section forms an integral part of the system, positioned at the end of the data handling pipeline (as shown in Figure 3.1). Here, it takes the processed data and utilizes plotting functions to create various types of maps. It's important to note that the system offers both a MATLAB-based code structure and a user-friendly **GUI** application for accessibility.

The primary function of the data handling blocks is to prepare raw data for analysis and then extract valuable insights, which materialise as different types of maps. These maps encompass **RTI** maps, **RD** maps, spectrograms, **CFAR** maps, and **CFAR** Level plots. The visual representation of these maps is achieved through the plotting functions, and this section will delve into the detailed design of these maps.

The structure of the visualisation section aligns with the modular design of the entire system, tailoring specific algorithms to each map type to cater to their unique requirements. Consequently, the section provides insights into how each map is plotted, offering clarity on the underlying processes that render the data visually comprehensible.

In the forthcoming sections, we will delve into the intricacies of each map type and its corresponding plotting methodology. This comprehensive approach ensures that the visual representation of the data is accurate, informative, and user-friendly, meeting the diverse needs of users across various applications.

3.4.2 Visualisation Theory

Range Resolution

Range resolution indicates a radar's capacity to differentiate and measure the separation between various targets within its **LOS**. This value represents the intervals between the range bins into which the continuous range monitored by the radar is divided. It reflects the radar's ability to provide precise range measurements and distinguish individual targets in close proximity. Higher range resolution means the radar can distinguish and represent objects positioned very closely in range, whereas lower

range resolution may result in multiple objects being perceived as a single target due to the radar's limited capacity to separate them. According to [2], the equation for calculating range resolution δR is as follows:

$$\Delta R = \frac{c\tau}{2} = \frac{c}{2B} \quad (3.8)$$

In the above equation, the pulse width in seconds τ is the reciprocal of the bandwidth B . The speed of light is represented by c .

Time Interval

The time interval between pulses, the **PRF**, and the number of pulses in an experiment are interrelated parameters in radar systems. **PRF** represents how often radar pulses are transmitted. The time interval between pulses is the reciprocal of the **PRF**, meaning that if the **PRF** is high, the time between pulses is short, and if the **PRF** is low, the time between pulses is long. The number of pulses in an experiment determines the total observation time. A longer observation time can increase the chances of detecting targets but may limit the ability to track rapidly moving objects due to a low **PRF**. Conversely, a higher **PRF** allows for faster target tracking but might reduce the observation time. The relationship between these parameters must be carefully chosen based on the specific radar system's requirements and the nature of the targets being observed.

3.4.3 Range-Time Intensity map

The **RTI** map, a pivotal radar signal processing visualisation tool, provides a comprehensive representation of a radar signal in both the time and range domains. This map illustrates the temporal and spatial evolution of radar signals. In an **RTI** map, the horizontal axis represents the range, denoting the distance from the radar antenna to the target or reflector, as explained in Section 3.4.2. Meanwhile, the vertical axis signifies time, as detailed in Section 3.4.2. Each pixel in the map corresponds to a specific combination of range and time, with colour indicating the signal strength at that specific point. **RTI** maps are particularly valuable for detecting moving targets, analysing their velocity, and distinguishing them from clutter or noise.

To generate the **RTI** map, only pulse compression processing was required. After reading the data from the file and completing preprocessing, the sole data processing needed for **RTI** map generation was the application of a matched filter. The matched filter enhances target detection in the measured signal by accentuating discrepancies between the measured signal and the reference signal, where these differences represent the targets or interference.

Figure 3.10 displays an **RTI** map example, illustrating the detection and tracking of moving targets over time. In the figure, two stationary targets are marked by solid lines on the left side, while the wavy red line on the right side represents the detection and tracking of a target moving towards and away from the radar antenna as time progresses. This plot was generated using the files titled 'e10_10_16_1800_40_P1_1_130000_S0_1_2047_node3.bin' and 'e10_10_16_1800_40_P1_1_130000_S0_1_2047_node3.m'.

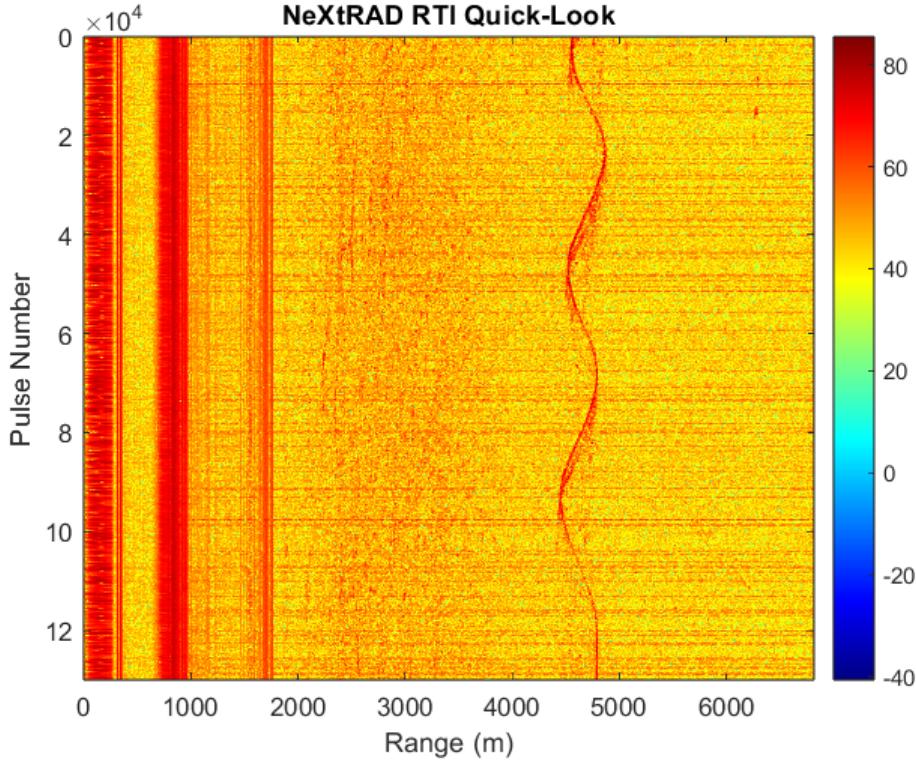


Figure 3.10: RTI map for 130000 pulse dataset.

The range axis for the **RTI** map displayed in Figure 3.10, was calculated by multiplying the number of samples recorded for each pulse by range resolution of the simulation. The time axis was defined as the number of pulses that elapsed from which the time can be determined based on the calculation detailed in 3.4.2 which utilises the **PRF** of the simulation.

3.4.4 Range-Doppler map

The **RD** map is a vital element in radar signal processing, offering a **2D** representation of the radar return signal. It displays the range of detected targets on one axis and their corresponding Doppler shifts on the other. This visualisation aids radar operators and analysts in understanding the location, speed, and direction of various targets within the radar's field of view. It is especially valuable for distinguishing moving targets from stationary clutter and provides a clear overview of the radar's surveillance area.

In this system, the **RD** map was generated through pulse compression using a matched filter and Doppler processing with a Doppler **FFT**. The resulting data is illustrated in Figure 3.11. The Doppler **FFT** was applied individually to each range bin, converting the data from pulse-based measurements to the frequency domain. This transformation unveils the distribution of Doppler frequencies within the radar data, facilitating the identification and tracking of moving targets.

Figure 3.12 displays two **RD** maps, each consisting of two sets of consecutive plots with 1000 pulses in each set. The variations between the plots on the left side of each image are attributed to a target's motion initially moving away from the radar antenna before coming to a halt. In this image, three targets are detected by the radar. However, it is evident that the two stationary tar-

gets are interference caused by signals from other systems operating in the same frequency band as the radar, the S-Band (2.4 GHz). The following plots were generated using the files titled ‘e10_10_16_1800_40_P1_1_130000_S0_1_2047_node3.bin’ and ‘e10_10_16_1800_40_P1_1_130000_S0_1_2047_node3.m.’

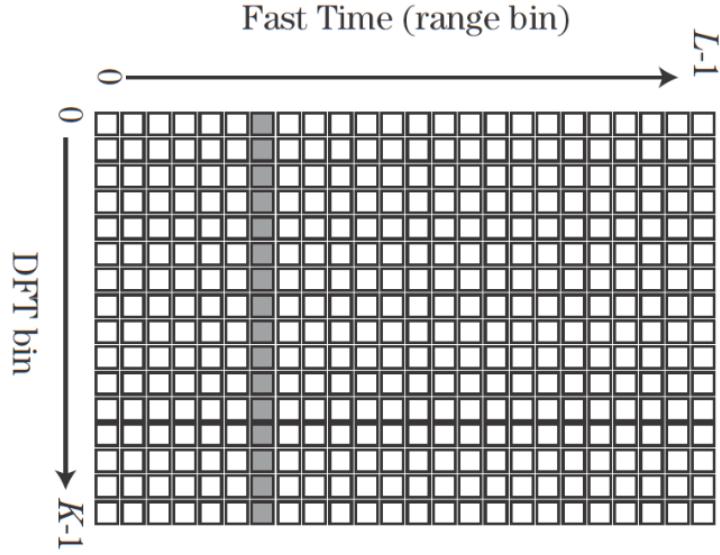


Figure 3.11: Matrix representation of data for RD map.

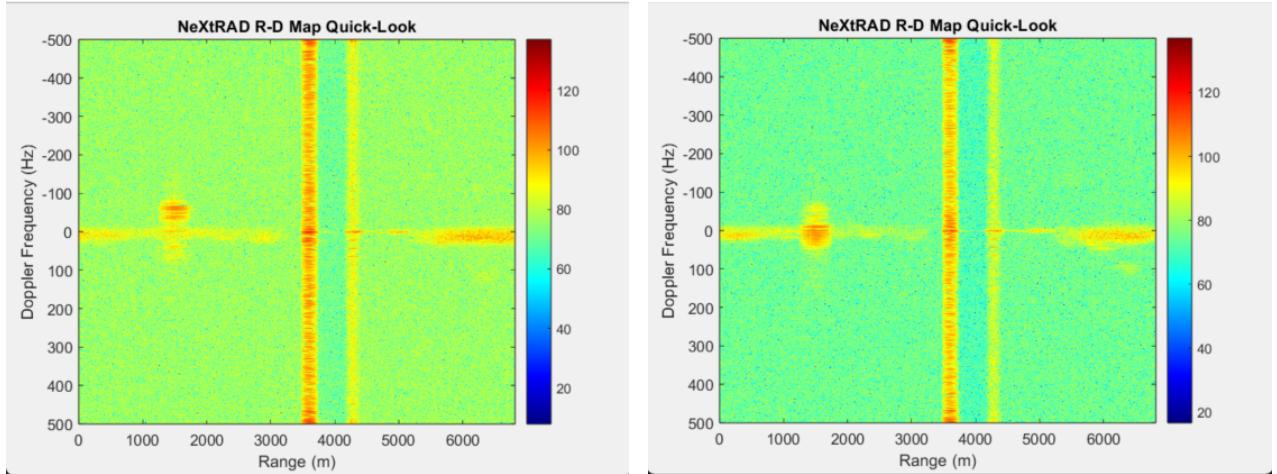


Figure 3.12: Two RD maps showing the change in the map due to movement of a target.

The axes of the RD map were established based on the methodologies described in Section 3.4.2 for the range bins and in Section 3.3.4 for the Doppler frequency bins. The range resolution was instrumental in converting the number of samples measured for each pulse into a range measurement metric. Likewise, the calculation of the Doppler frequency bins followed the principles outlined in the theory, which defined the minimum and maximum bounds of the axis as derived from the PRF of the simulation.

3.4.5 Spectrogram

A spectrogram is a visual representation of a signal's frequency content as it changes over time, serving as a valuable tool for analysing time-varying signals in various fields. It achieves this by dividing the signal into short, overlapping segments and applying [STFT](#) to each segment to capture its frequency components. In this system, the spectrogram is generated through pulse compression and micro-Doppler processing.

The outcome is presented as a [2D](#) image, with time on the horizontal axis and Doppler frequency on the vertical axis, and the colour intensity reflects the magnitude of each spectral component. Figure 3.13 illustrates the spectrogram generated for a single range bin. The parameters used for this spectrogram included a sliding [1D](#) window of 20 cells, a 95% overlap, and the range bin set at 1200 metres from the radar, covering a total of 130,000 cells. The data files employed for this spectrogram were ‘e10_10_16_1800_40_P1_1_130000_S0_1_2047_node3.bin’ and ‘e10_10_16_1800_40_P1_1_130000_S0_1_2047_node3.m’.

Spectrograms are highly valuable for identifying transient events, tracking frequency changes, and gaining insights into the dynamic characteristics of signals.

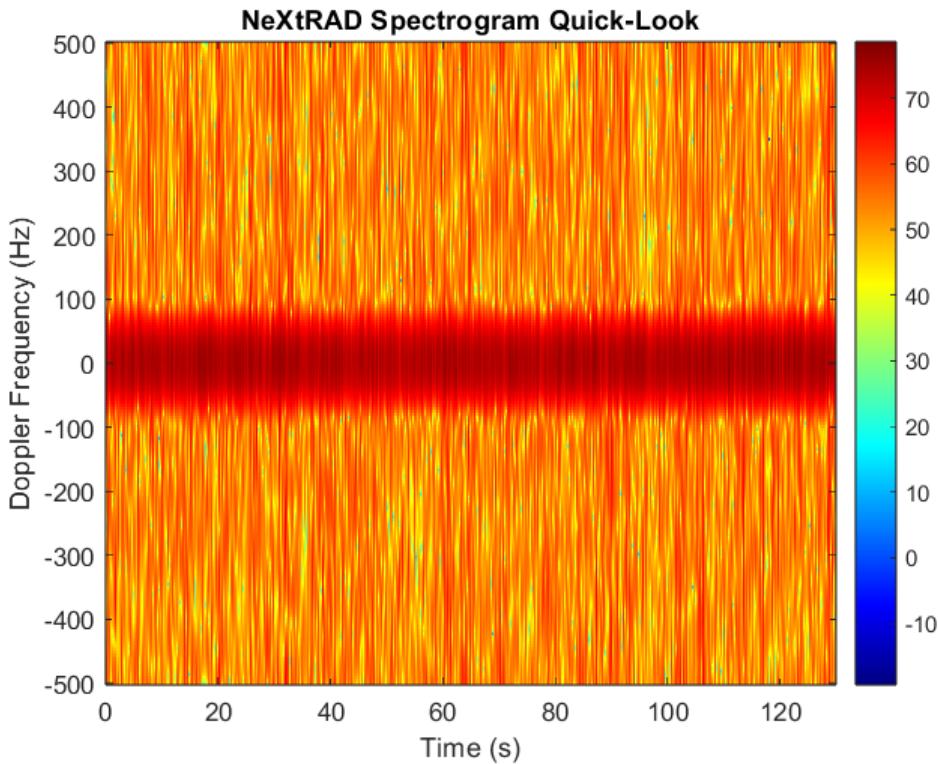


Figure 3.13: Spectrogram for 130000 pulse dataset.

In the plot depicted in Figure 3.13, interference is noticeable within the spectrum. This interference is primarily attributed to two stationary targets that fall within the radar's [LOS](#) and are caused by other systems operating in the same frequency band as the NetRAD radar at the time of data recording. The axes of this plot were calculated based on the principles discussed in Section 3.4.2. The Doppler frequency axis, for instance, was determined by specifying the minimum and maximum

frequency bounds as $-\text{PRF}/2$ and $\text{PRF}/2$, respectively. All the Doppler bins fall within this range. In the specific simulation illustrated in the Figure, a **PRF** of 1 kHz was employed, resulting in minimum and maximum values of -500 Hz and +500 Hz.

3.4.6 Constant False Alarm Rate map

A **CFAR** map is a valuable tool for detecting radar targets and anomalies in the presence of varying noise and clutter. This algorithm dynamically adjusts the detection threshold based on local data statistics, enabling robust target detection in different environmental conditions. The **CFAR** map highlights regions in the radar data where the signal power surpasses the threshold, thus revealing potential target locations. This consistency in **CFAR** performance is crucial for reliable target detection, even when noise and clutter are present.

Creating a **CFAR** map involves three processing methods. Firstly, pulse compression is applied to the **IQ** signal data using a matched filter. Secondly, Doppler processing is employed by applying a Doppler **FFT** to the data's slow-time domain along each range bin in the matrix. Lastly, each cell in the matrix undergoes a **1D CFAR** detection threshold algorithm, reducing noise and interference in the signal data. Users have the flexibility to implement **CFAR** thresholding in either the Doppler or range orientation.

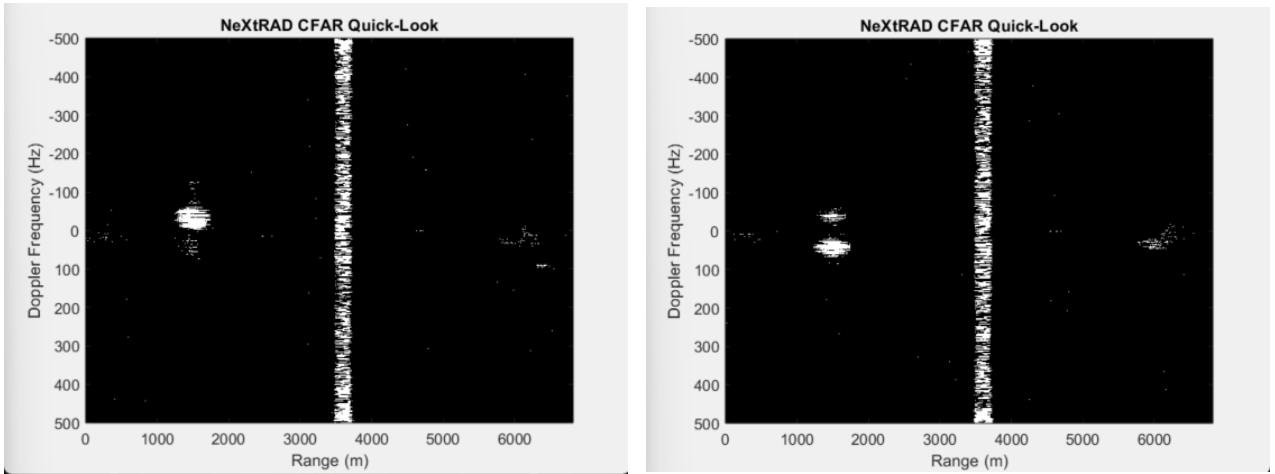


Figure 3.14: Two Range CA-CFAR maps showing the change in the map due to movement of a target.

Figure 3.14 presents maps containing two consecutive sets of 1000 pulses, where a Range **CA-CFAR** algorithm has been applied. The difference between the two maps is attributed to the motion of the target on the left side of the image. The **CFAR** detection values are binary, with values of 1 indicating detection and 0 indicating non-detection based on the detection observed in the **CUT** and the detection threshold calculated using the **CUT**'s sliding window.

In the **RD** maps featured in Figure 3.12, generated from the same pair of 1000 pulses, a pair of interferences is evident, represented as red vertical sections spanning all Doppler frequency bins. Additionally, noise is observed in and around the 0 Hz Doppler frequency bin, present across the entire spectrum of range values.

In an alternative implementation, Figure 3.15 displays the Doppler **CFAR** map derived from the application of a Doppler **CFAR** detection threshold algorithm. The primary difference between the two

maps within the Figure lies in the target's intensity around the 1200-meter range bin. The target's intensity diminishes from the first **CFAR** map to the second, indicating a shift in the radar environment. This change may result from variations in the target's **RCS** due to factors like orientation, shape, or material composition. Alternatively, the reduction in target intensity may be attributed to the introduction of interference around the target at that specific time. The clutter could obscure the target signal, subsequently reducing the **CFAR** map intensity.

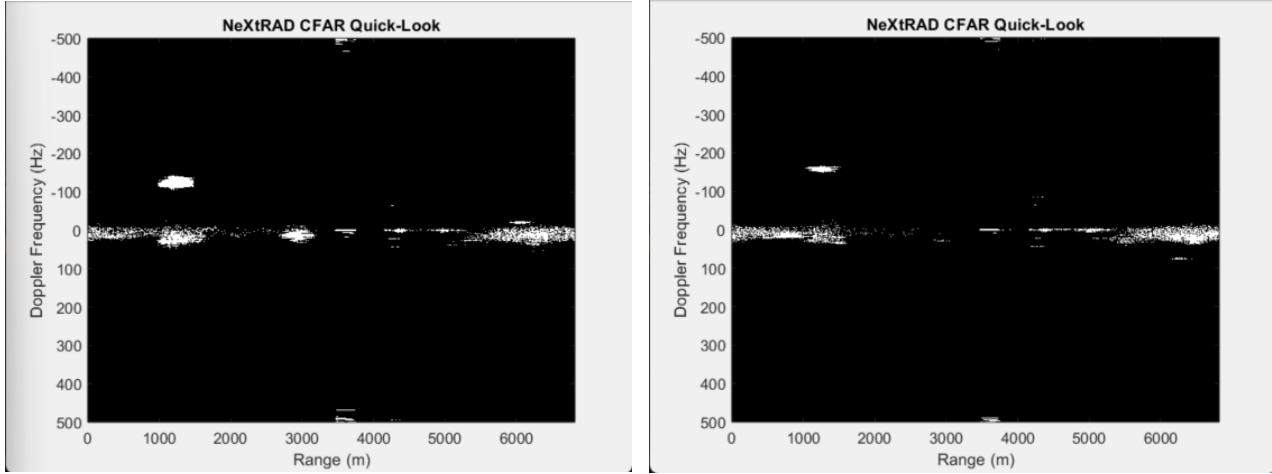


Figure 3.15: Two Doppler CA-CFAR maps showing the change in the map due to movement of a target.

3.4.7 Constant False Alarm Rate Level plot

The **CFAR** Level plot serves as a visual depiction of the threshold levels used in a **CFAR** detection algorithm across different range and Doppler cells in pulse-Doppler radar data. This plot effectively highlights how detection thresholds adapt dynamically to maintain a constant false alarm rate while considering variations in clutter and noise levels within the radar environment. The **CFAR** Level plot is a valuable tool for radar analysts, enabling them to assess the sensitivity and reliability of target detection throughout the entire range and Doppler domain. By observing variations in thresholds across different regions of the plot, analysts can gain insights into clutter conditions, radar system performance, and the effectiveness of target detection. This visualisation aids in optimising radar system settings and understanding the trade-offs between detection sensitivity and false alarms.

In Figure 3.16, it is evident that the threshold consistently remains higher than the **CUT** during a direct comparison between the two plotting functions. This occurs in every location where there is interference or noise spanning the entire window, as well as in situations when no target is present for detection. The locations where targets are detected and approved in the comparison between the **CUT** and threshold are those where interference is minimal. The horizontal axis of the plot denotes the number of cells processed in the **CFAR** detection threshold, while the vertical axis represents the power of each of the two data signals. The power on the vertical axis quantifies signal strength, providing a measure of the magnitude of the return signal within a specific range and Doppler cell.

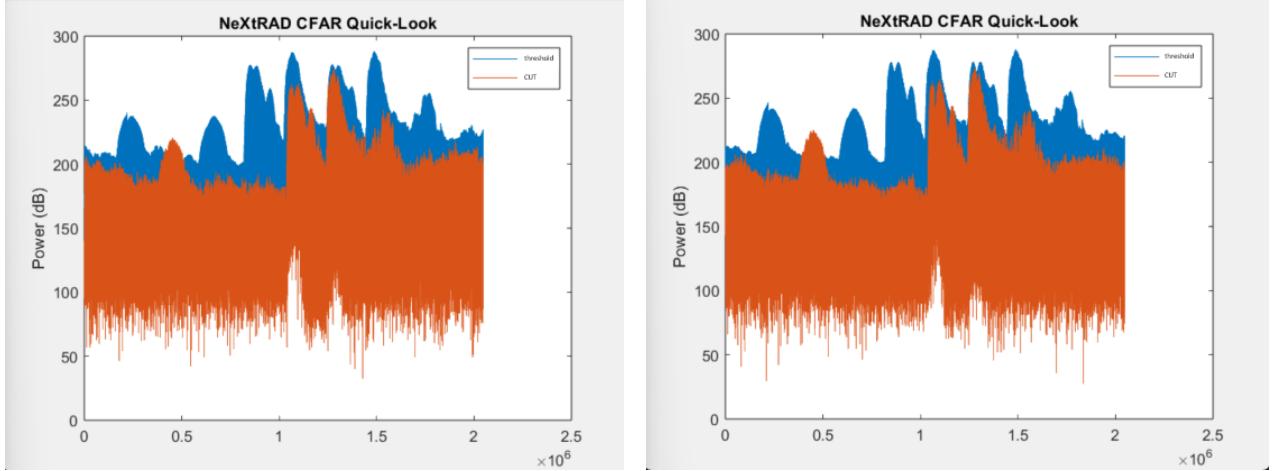


Figure 3.16: Two CFAR Level plots showing the change in the map due to movement of a target.

3.5 Graphical User Interface

The system was developed in MATLAB to leverage its available libraries and offer easy implementation in both a code-based and a visual-based format. The GitHub repository containing both versions of the project can be accessed [here](#).

The [GUI](#) serves as the user-friendly front end of the [QLP](#), providing an interactive platform for users to interact with the radar data processing system effectively. It offers a visually intuitive way to configure and initiate data processing tasks, visualize radar maps, and interpret the results. The [GUI](#) includes features such as data file selection, parameter adjustments, and map selection, allowing users to customize data processing according to their specific requirements. It displays various radar maps, including [RTI](#) maps, [RD](#) maps, Spectrograms, [CFAR](#) maps, and [CFAR](#) Level plots in a user-friendly and comprehensible manner. This interface significantly enhances the usability of the [QLP](#), making radar data analysis accessible to a broader audience, from radar operators to researchers and analysts.

Figure 3.17 illustrates the [GUI](#) application's design, which enables users to input three files for processing and plotting. Each of the three files can be chosen from those stored on the user's computer. Additionally, an option is available to use the reference signal included in the parameter file. Upon pressing the 'Continue' button, the files are opened, and their contents are stored for use in the processing. Once the files are closed, the active tab switches to 'Parameters,' and the values read from the files are loaded into the display.

In Figure 3.18, the design of the 'Parameters' page is displayed. This page is loaded upon button press on the 'File' page, which also loads the simulation parameters defined in the measured data file supplied to the application. These values can be modified, and additional functionality can be enabled. Pulse Integration is available for implementation, and the number of pulses to sum is a parameter that can be adapted and set by the user. The [CFAR](#) parameters can be adjusted at will but will only take effect if the [CFAR](#) checkbox is selected. The same implementation applies to the spectrogram parameters. Changes and plot selections must be completed before pressing the 'Process & Plot' button. Upon pressing this button, the system will begin processing the data and plotting the selected maps, either in real-time or as a complete plot once the processing is finished. The plotting options are accessible on the 'Parameters' page. Within this page, you can select various types of maps, including the [RTI](#),

3.5. Graphical User Interface

RD, spectrogram, and the matched filter output for the first pulse in the dataset. The **CFAR** option is only enabled when the **RD** map is selected for plotting. The checkboxes related to **CFAR** remain disabled until all the prerequisites for activation are met. Additionally, the **CFAR** Level plot can only be selected if the **CFAR** functionality has been chosen.

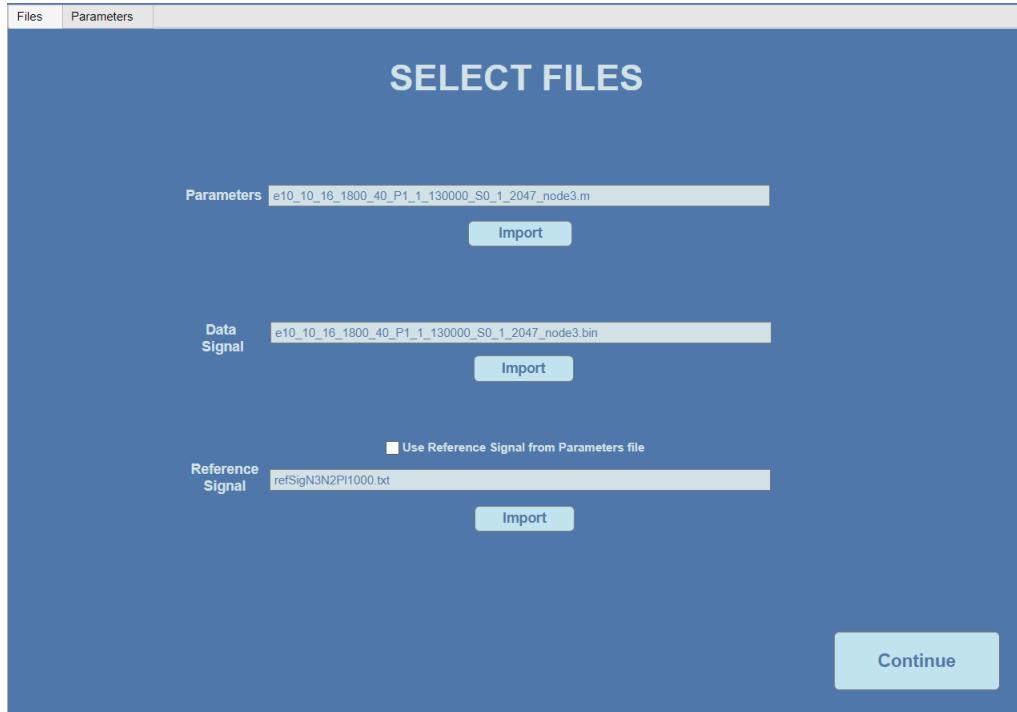


Figure 3.17: File page of GUI application.

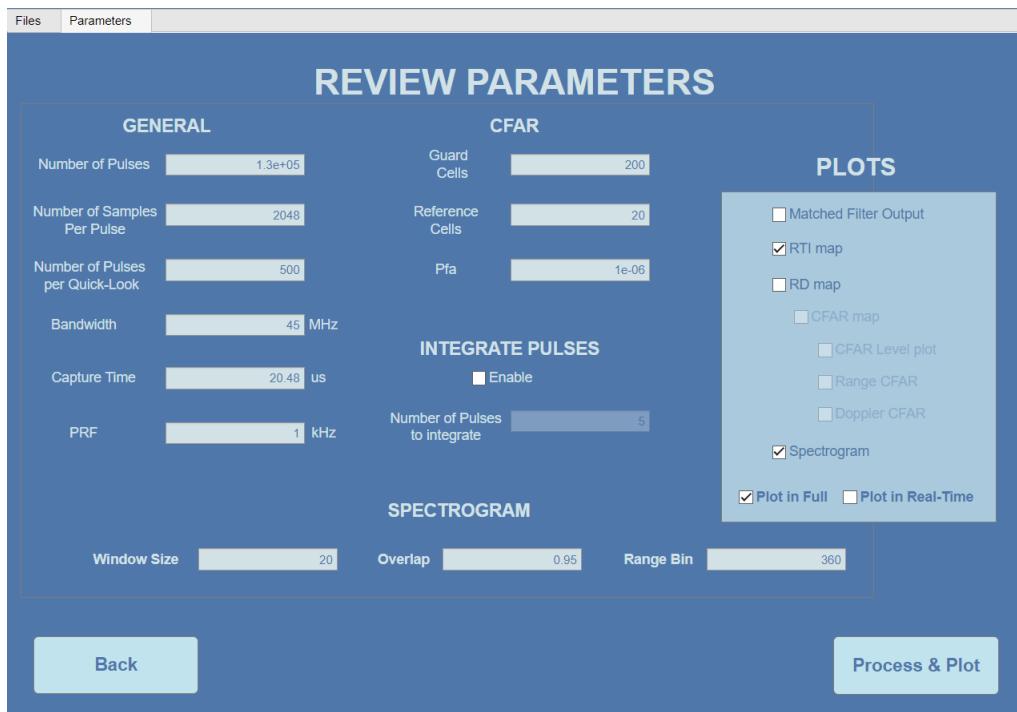


Figure 3.18: Parameter page of GUI application.

Chapter 4

Quick-Look Processor Testing

This chapter outlines the testing methodology for the six different tests performed on the [QLP](#) system. The initial test involves a visual comparison of the [RTI](#) map generated by the [QLP](#) with the map provided in the data files. Subsequent tests focus on evaluating the system's computational performance to ensure that the implemented visualisation techniques produce plots within a real-time time frame.

4.1 Range-Time Intensity Map Comparison

4.1.1 Testing Procedure

The [RTI](#) map provides insights into the motion of detected targets within a dataset. It displays the distance from the radar antenna to each target at the time each pulse is recorded, resulting in vertical markings on the map to indicate the presence of a target. To evaluate the performance of the [QLP](#) designed in this project, comparisons were made with sample [RTIs](#) provided alongside some of the data samples.

For some simulations, reference signals were computed using sources external to the provided data or the reference signals contained in the parameter file. The analysis of these results will focus on the comparison between the generated maps and the given maps when using the reference signal from the parameter file as the effective reference signal.

During this testing method, the maps will be compared and evaluated based on their capability to effectively plot and track the target's location, motion, and the colour intensity within the [RTIs](#). The system's ability to plot and track a target will be assessed by examining the target's location and its motion over time, represented as either a straight line for a stationary target or a wavy line for a moving target. The target's location is determined by the number of samples within the pulse that detects the target. Multiplying this number by the range resolution provides the distance from the radar for each pulse in the [RTI](#). The intensity of the plot reflects the signal strength of the detected target, enhancing the certainty of target presence and facilitating the interpretation of the [RTI](#) map.

The performance of the system was measured through the comparison of three pairs of .bin and .m files namely: 'e10_10_16_1800_40_P1_1_130000_S0_1_2047_node3', 'e10_10_21_1927_55_P1_1_130000_S0_1_2047_node3' and 'e11_04_20_1305_20_P1_1_130000_S0_1_2047_node3'. The full pulse dataset of 130000 pulses was used to produce the plots from the data files. The provided image files are the images of the [RTI](#) maps that are produced from the current [QLP](#) system used to process the NetRAD data. The parameters

from the provided files were used as default to plot the [RTI](#) maps from this system implementation.

4.1.2 Achieved Performance

Comparing the first pair of maps displayed in Figure 4.1, it is evident that the [RTI](#) map generated by the system developed in this project accurately depicts the location and relative distances between the targets. The thick vertical red strips on the left-hand side of both images indicate the detection of a series of closely spaced targets, which result from wireless interference within the data due to the operating frequency band of NetRAD. The general orange tint in the yellow areas of the data represents interference and noise, attributed to both interference and additional noise present within the dataset.

In both maps, a single moving target is detected and tracked on the right side. However, a slight discrepancy in the location of the target is observed between the [RTI](#) map produced by the system and the provided [RTI](#) map. The system's tracking of the target is not as precise as the reference plot due to the use of the reference signal from the parameter file instead of the exact signal used in the production of the provided plot. This discrepancy can be attributed to minor differences between the reference signal file and the reference signal from the parameter file.

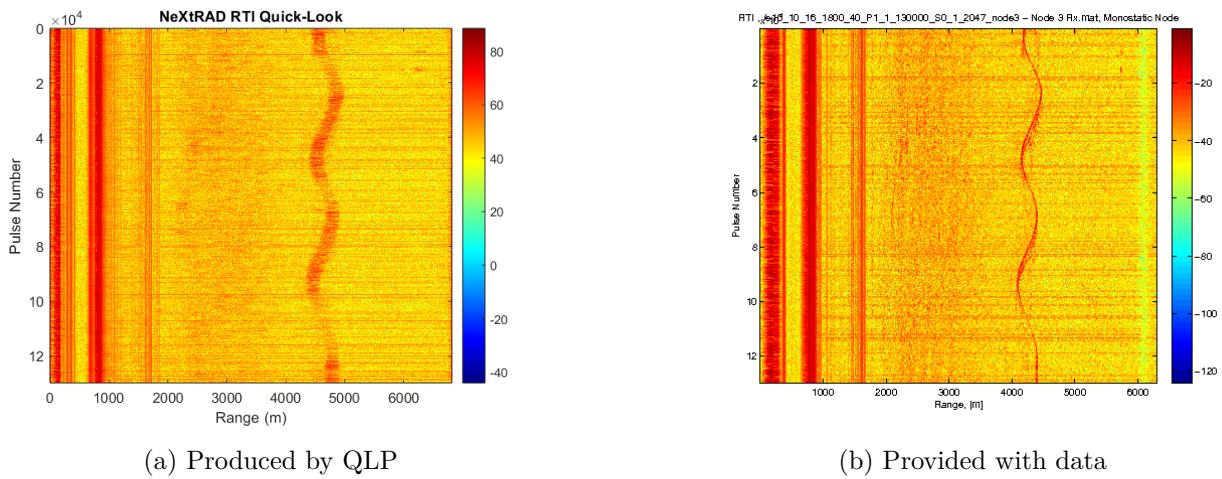


Figure 4.1: First File comparison of RTI plot produced by system and given plot.

A second comparison can be made based on Figure 4.2, where both maps display interference. In both maps, the interference and noise are the only signals present up to a range of 6000 meters from the radar antenna. Although they are modelled differently in the two maps, it is evident that interference and noise are clearly visible in both plots.

Furthermore, Figure 4.2b reveals the presence of a stationary target at the edge of the radar's [LOS](#). This target remains in the data from the beginning to the end of the observation and does not exhibit any movement. However, the map produced by the project's system does not show this stationary target. Instead, that location is masked by interference, which may be obscuring the presence of the target from view.

In the final comparison shown in Figure 4.3, the maps have interpreted the data differently. Figure 4.3b displays an [RTI](#) map that shows the presence of either numerous stationary targets or a significant

4.2. Range-Time Intensity Time Test

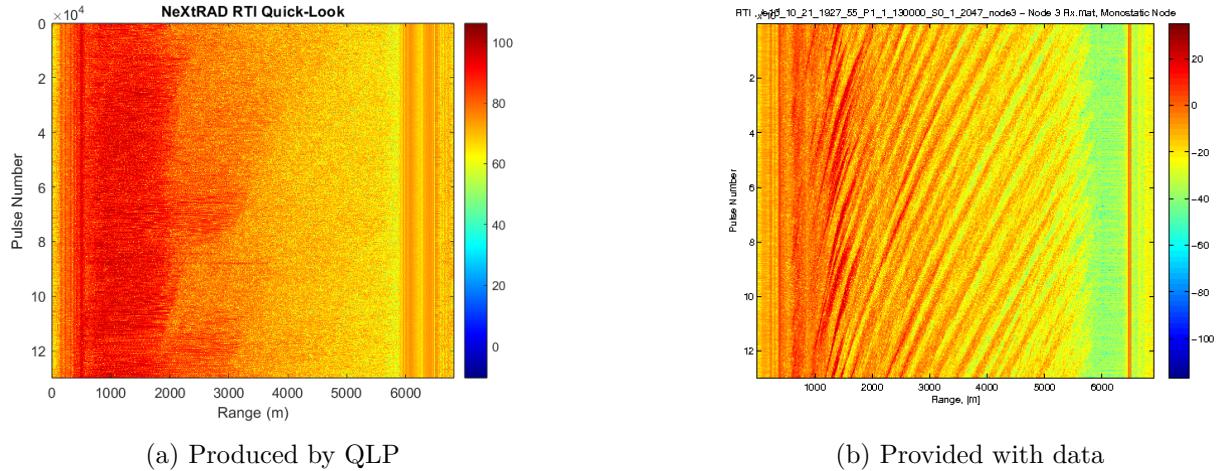


Figure 4.2: Second File comparison of RTI plot produced by system and given plot.

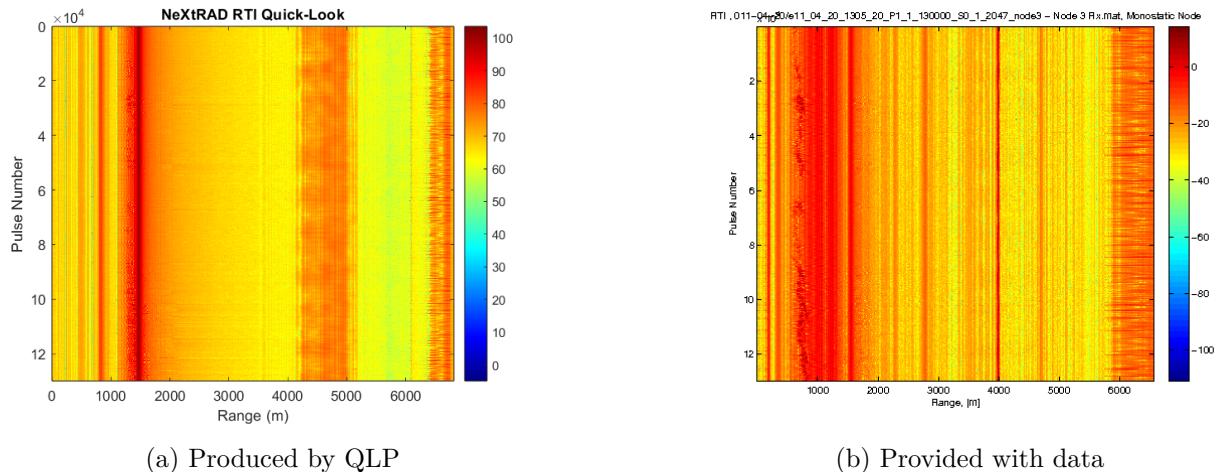


Figure 4.3: Third File comparison of RTI plot produced by system and given plot.

amount of interference within the measured data. It's uncertain whether these are stationary targets or interference. However, the plot produced by the project's system, as seen in Figure 4.3a, suggests that the QLP interpreted the vertical lines as interference rather than stationary targets. The colour intensity is similar between the two plots, but in Figure 4.3a, more of the map is depicted in yellow.

4.2 Range-Time Intensity Time Test

4.2.1 Testing Procedure

In this test, the processing time for generating the **RTI** map is evaluated. The time measurement encompasses data file reading, data processing involving matched filter pulse compression, and map plotting. The **RTI** map is generated incrementally in chunks, each representing a percentage of the total pulses to be processed.

The pipeline's performance in plotting the RTI map is assessed using both the original data signal and pulse integration. Pulse integration reduces the number of processed pulses and processing time while retaining data integrity.

The time measurement in MATLAB is achieved using the `tic()` function to record the start time and the `toc()` function to calculate the elapsed time.

The chunk-based approach for plotting groups of processed pulses accumulates a specific number of unplotted pulses before they are introduced to the map. A chunk size equal to 1% of the total processed pulses ensures a visually continuous plotting experience.

The pulse integration algorithm computes the number of pulses to sum together as 0.5% of the total pulses processed by the pipeline.

The system's time measurements were validated through five repetitions of each pulse iteration. Various pulse sizes were tested, ranging from 1000 to 130000 pulses, corresponding to time intervals from 1 to 130 seconds.

4.2.2 Achieved Performance

Figure 4.4 provides insights into the system's performance for real-time processing of RTI maps. It demonstrates the system's capability to process data in real-time for this specific map type. The figure also presents the system's performance for various pulse sizes, ranging from 1000 to 130,000 pulses, both with and without the use of pulse integration.

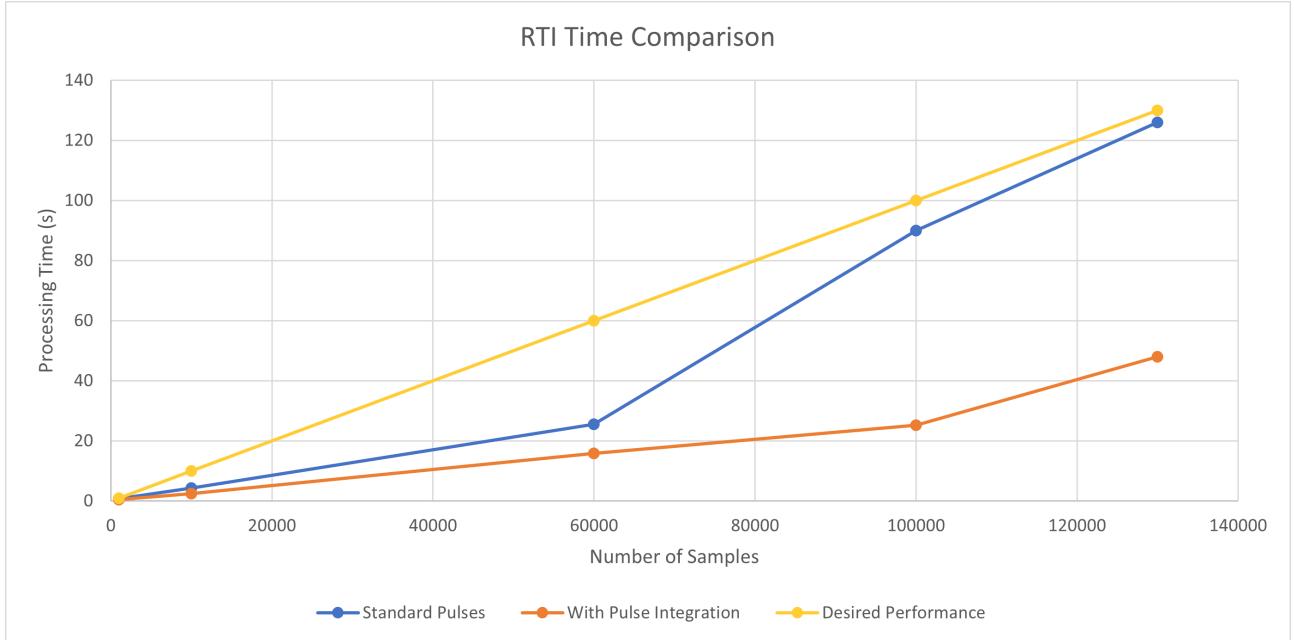


Figure 4.4: Comparison of processing time for the RTI map.

The system's performance demonstrates that, with the inclusion of pulse integration, it efficiently processes data from raw input to RTI map plotting, ensuring satisfactory processing times for all tested pulse sizes. The benefits of pulse integration are particularly noticeable in cases with a higher number of pulses due to the aggregation of pulses during the process.

As the number of pulses approaches the maximum value, the computation time for the standard processing gradually nears the real-time threshold. This trend is primarily attributed to the additional overhead incurred by the plotting function when handling larger matrix sizes.

4.3 Range-Doppler Time Test

4.3.1 Testing Procedure

This test assesses the processing time for generating the RD map. Performance is evaluated based on the time it takes for tasks such as file reading, data processing, and map plotting. The processing involves pulse compression and Doppler processing through matched filtering and Doppler FFT. Unlike the RTI map, the RD map doesn't rely on a chunk-based approach, as a certain number of pulses are plotted in each quick-look.

The RD map is employed to track the direction of target movement efficiently. It provides valuable insights into how targets move and offers a clear representation of the radar's surveillance area.

Pulse integration's impact on the RD map was assessed using 0.5% of the total pulses for each calculation.

For consistency, the system generated five maps for each set of pulses tested, with each quick-look plot containing 20% of the total number of pulses.

The quantity and dimensions of the tests align with the simulations conducted on the RTI map.

4.3.2 Achieved Performance

The system's performance, depicted in the Figure below, was assessed against the real-time processing threshold. The recorded time measurements, with and without pulse integration, meet the real-time processing requirements.

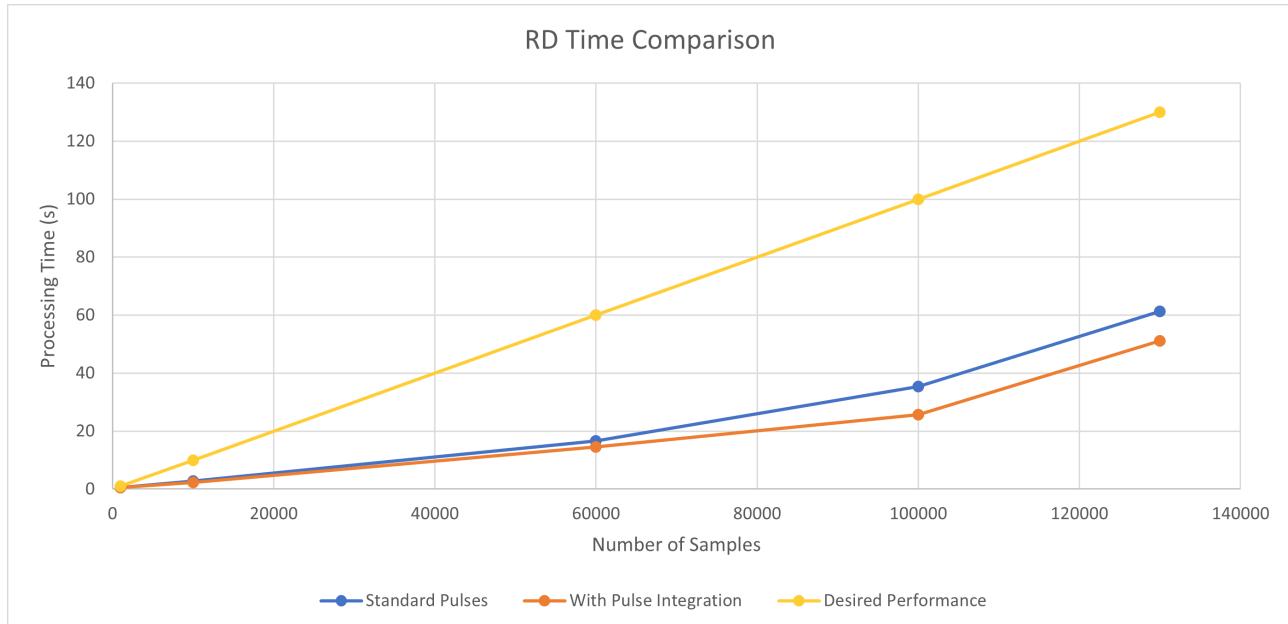


Figure 4.5: Comparison of processing time for the RD map.

In Figure 4.5, it is evident that the impact of pulse integration on performance is minimal in relation to the real-time threshold. The most significant difference is observed when processing the maximum number of pulses, where pulse integration reduces processing time by 10 seconds.

In the processing of the [RD](#) maps, a different trend emerges compared to what is observed in Figure [4.4](#), where the system without pulse integration approaches the real-time threshold as the number of processed pulses increases.

4.4 Spectrogram Time Test

4.4.1 Testing Procedure

Within this test, the performance of the system when only processing and plotting the data for the purpose of displaying the spectrogram is analysed. The spectrogram is a combination of pulse compression and micro-Doppler processing in order to draw meaningful results from the frequency shifts that occur within the system as time progresses.

The spectrogram was generated in chunks, with specific criteria set to determine the chunk size. The chunk size needed to be at least twice the size of the sliding window used in the spectrogram implementation. The sliding window itself had a size of 20 elements with a 95% overlap.

A particular range bin was selected within the region that experienced interference at the beginning of the [RTI](#) map. This specific range bin was located at column 360, which corresponds to a range of 1200 meters when considering the range resolution. The chunk size was determined to be 1% of the total number of processed pulses. However, if this calculated chunk size was smaller than twice the window size, it was set to be equal to double the window size. This adjustment ensured the effective operation of the spectrogram map.

Pulse integration was introduced in this test to compare the standard system with the integrated system and measure their respective performances. The integration was carried out using 0.5% of the total pulses for each calculation.

4.4.2 Achieved Performance

Figure [4.6](#) presents the system's performance when generating the spectrogram. The system meets the time-based criteria for real-time processing, completing the data reading, processing, and plotting in less time than it takes to record the dataset.

Two separate tests are displayed on the figure, illustrating the reduction in computation time when pulse integration is implemented. The impact of pulse integration is negligible for datasets with 60,000 pulses or fewer. However, in the case of the 130,000-pulse test, it results in a 17-second reduction in computation time.

The trend observed in this test aligns with the results shown in Figure [4.5](#), where the two sets of results start to differ when more pulses are processed. Overall, the system consistently performs data processing in a time frame shorter than real-time, categorising it as a real-time [QLP](#) when generating the spectrogram.

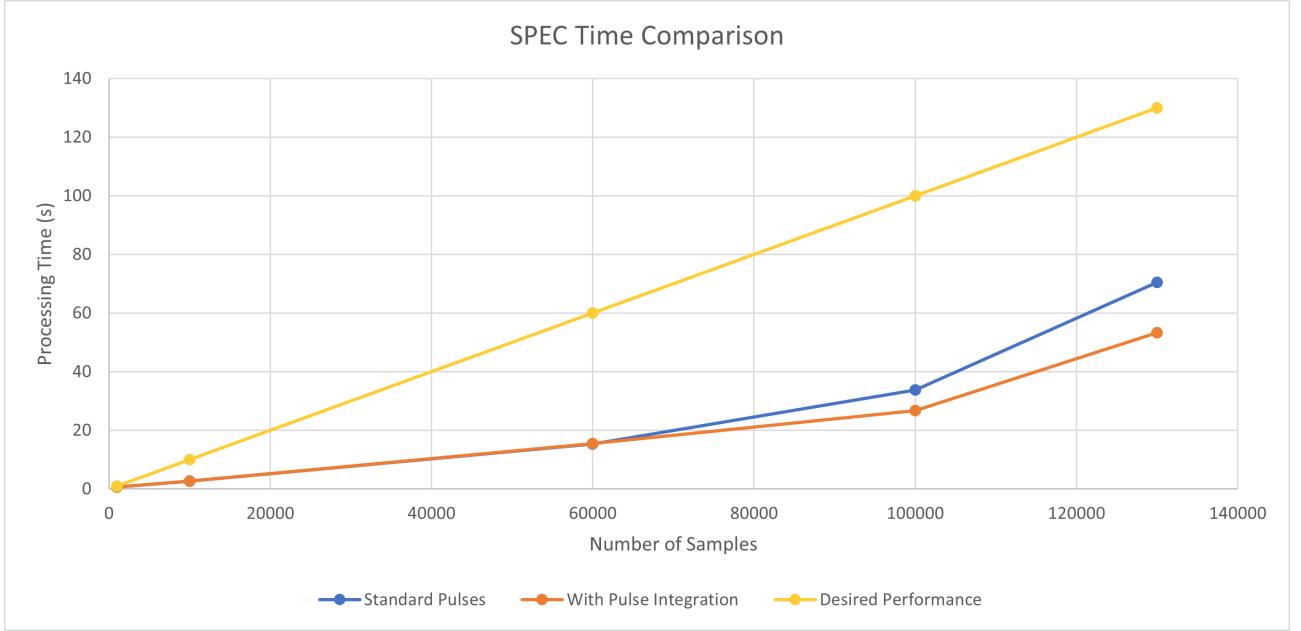


Figure 4.6: Comparison of processing time for the spectrogram.

4.5 Constant False Alarm Rate Time Test

4.5.1 Testing Procedure

Evaluating the performance of the implemented **CFAR** algorithm in this system involves reading raw data, applying pulse compression through a matched filter, Doppler processing using a Doppler **FFT**, and subsequently applying a **CFAR** detection threshold algorithm to the **RD** matrix. It's important to note that the **CFAR** functionality is exclusively available when the **RD** map is selected for plotting. Consequently, the performance of **CFAR** processing and plotting is expected to be relatively slower compared to the other three visualisation techniques available.

Pulse compression is achieved through a matched filter, followed by Doppler processing via a Doppler **FFT**, and lastly, a **CA-CFAR** detection threshold algorithm is employed to filter out interference and noise from the **RD** matrix.

The **CFAR** map shares some processing parameters with the **RD** map. Specifically, it involves dividing the number of pulses into five quick-look plots, ensuring that each map produced by the system during pulse processing contains 20% of the total number of pulses in the test. Another common parameter is the percentage of total pulses included in each pulse integration calculation, which is set to 0.5%.

Testing the **CFAR** visualisation technique is combined with the evaluation of the **CFAR** Level plot. Therefore, the testing encompasses the Range **CFAR** algorithm, the Doppler **CFAR** algorithm, the pulse-integrated Range **CFAR** algorithm, and the Range **CFAR** algorithm combined with the **CFAR** Level plot.

The parameters defined for the **CFAR** processing are as follows. Firstly, the number of reference cells is set to 20, ensuring sufficient data is used to generate a representative noise power value. Secondly, the number of guard cells is set to 200, with the larger guard cells size necessary due to the presence of large targets spanning numerous samples in each pulse. This size prevents interference in the calculation of

noise power due to the target's presence. Lastly, the probability of a false alarm is set to 1e-6.

4.5.2 Achieved Performance

Figure 4.7 reveals that the Range CFAR, Doppler CFAR, and CFAR map and Level plot tests took significantly more time to process and plot the data compared to the available time window. Each of these three simulations required over five times the defined real-time limit. Their processing speeds were quite similar, indicating that the bottleneck within the system is primarily attributed to the detection threshold algorithm used for comparing each CUT with the calculated threshold.

Upon closer examination, it was determined that the bottleneck is specifically caused by the allocation and configuration of the correct reference cells for threshold value calculations. The figure demonstrates that the Range CFAR algorithm is slightly faster than its Doppler CFAR counterpart. This is because the Doppler CFAR algorithm necessitates more computations per range cell, resulting in a higher processing load when compared to the Range CFAR algorithm.

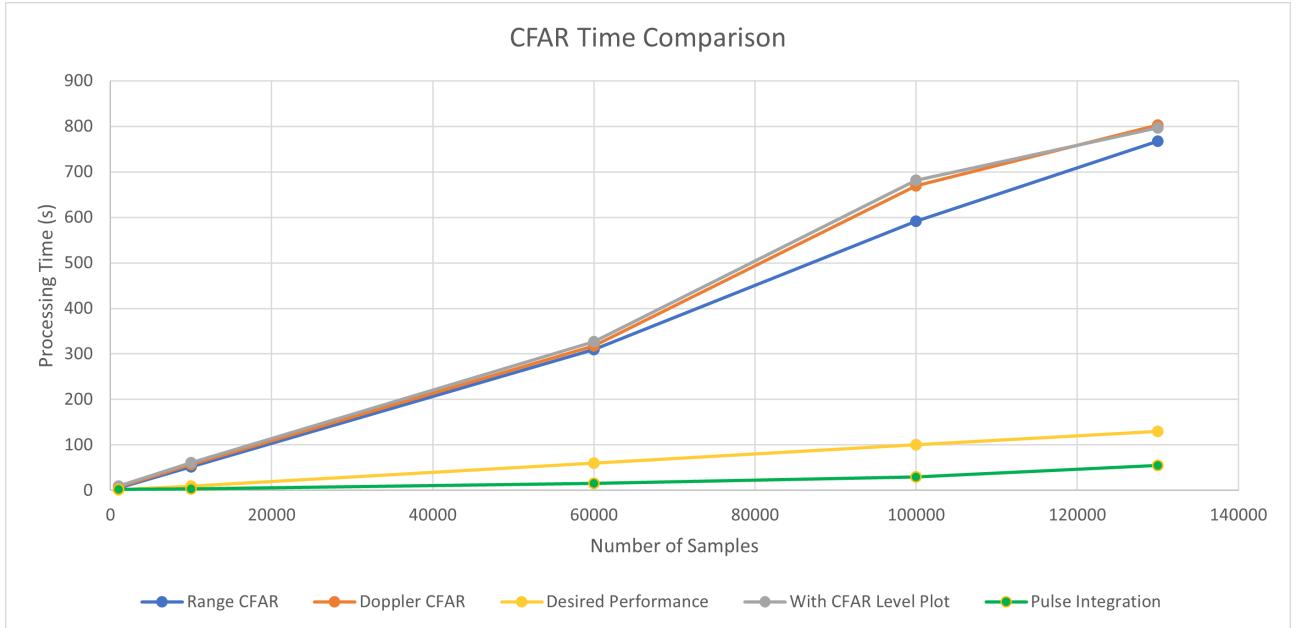


Figure 4.7: Comparison of processing time for the CFAR map and CFAR Level plot

Pulse integration has proven to be highly effective in reducing the processing time of the Range CFAR algorithm, especially evident in the case of the maximum pulse test, where it led to a 14-fold reduction in processing time. This remarkable performance enhancement is achieved by decreasing the number of pulses processed by the system, thus requiring fewer computations per quick-look implementation.

However, it's important to note that the system only meets the criteria for a real-time quick-look implementation when pulse integration is implemented to reduce the number of effective pulses processed. The identified bottleneck, as mentioned earlier, significantly impacts the processing time and overall system performance. Addressing this bottleneck is crucial for improving both map plotting and the system's efficiency as a whole.

4.6 System Performance Analysis

The testing conducted in this chapter was aimed at evaluating the system's performance, focusing on the accuracy of the generated maps and the time taken to produce each map. The results showed a wide range of accuracy, from maps closely resembling the reference plots to others that appeared significantly different, as exemplified in Figure 4.1 and Figure 4.3. It became apparent that the system's performance in producing accurate results is closely tied to the presence of targets in the radar data. While the inclusion of clutter and interference could lead to divergent plots, the presence of a target, even with a less than ideal reference signal, allowed the QLP to create representative RTI maps.

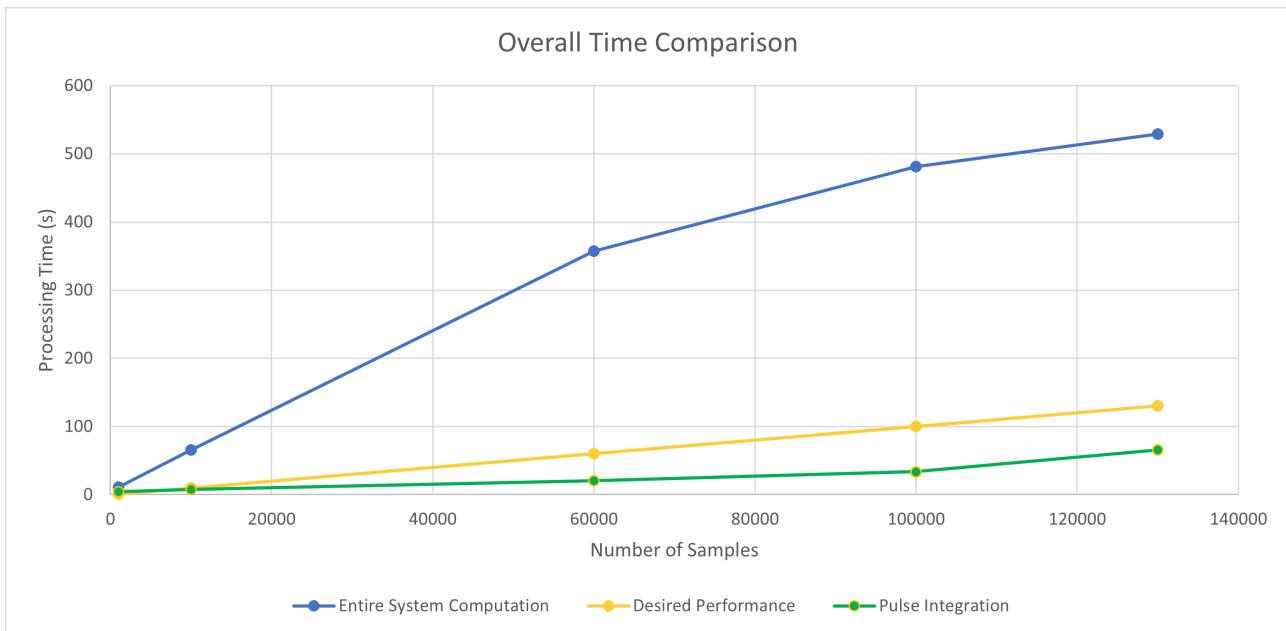


Figure 4.8: Comparison of processing time for all the maps in the system.

Figure 4.8 provides insight into the system's overall performance when plotting all available maps in a single processing cycle. The data is read from the file and undergoes pre-processing before pulse compression. After each line of data is processed, an identification tree determines whether it's ready for processing or should be combined with other pulses to form a chunk for plotting. The figure illustrates that the overall processing time for the system did not meet the desired performance threshold for any of the tests conducted with varying numbers of pulses. As a result, it failed to meet the criteria for a real-time QLP. However, when pulse integration was implemented, the system met the criteria for a real-time QLP for all tests, with computation times below the relevant threshold. The pulse integrated system operated eight times faster than the original system to meet the desired criteria.

The root cause of this inflated performance time metric was identified as the allocation of reference cells within the CFAR detection threshold calculation algorithm. This issue significantly impacted the overall processing time of the system and highlighted the need for optimisation to achieve real-time processing.

Chapter 5

Conclusions

The purpose of this project was to design and develop a [QLP](#) for the context of a NeXtRAD system. The NeXtRAD radar system is a new and improved version of the NetRAD system which was used for the purpose of target detection and tracking within a sea of sea clutter and interference. The NeXtRAD radar system offers significant steps forward from its NetRAD counterpart such as its dual-band operation and complete polarimetric sensing.

This report began with a thorough review of the existing literature, serving as the foundational chapter in Chapter 2. The literature review aimed to provide a contextual framework for the study, and successfully accomplished this objective by delving into various aspects of the field. The discussion initiated with a brief historical overview and performance comparison of alternative radar types before shifting its focus to potential signal processing techniques applied to radar data, highlighting their importance in extracting valuable information. Furthermore, techniques for effectively visualising processed radar data were explored, with a specific emphasis on map generation methods. Subsequently, the review narrowed its focus by examining the NeXtRAD and NetRAD radar systems before culminating with an evaluation of the current [QLP](#) applications on the NeXtRAD system and other radar technologies.

The bulk of the work for this project followed next, in Chapter 3. This chapter outlined the design of the [QLP](#) along with the underlying theory employed in creating a system dedicated to processing data generated from simulations carried out on the NetRAD and NeXtRAD systems. The principal aim of the design was to develop a versatile tool capable of handling data from these radar systems and generating graphical representations of the outcomes in the realms of range, time, and Doppler frequency. The implementation of this tool was tailored to enable real-time or near-real-time data processing, with the overarching goal of enriching comprehension of the recorded data. By detecting and tracking targets while extracting critical details such as their range and operational frequency, this design revolutionises the implementation of target detection and tracking systems. The comprehensive nature of the design underscores its importance in advancing the capabilities of radar systems and their applications across diverse domains.

Additionally, the [QLP](#)'s design incorporated the development of a user-friendly [GUI](#). The [GUI](#) was meticulously crafted to provide a seamless and intuitive platform for interacting with the processing tool. It empowers users to control and customise data visualisation, set system parameters, and analyse results with ease. This [GUI](#) design is instrumental in ensuring that radar operators, researchers, and analysts can fully harness the [QLP](#)'s capabilities, making it an invaluable asset for radar data analysis and target detection tasks.

Finally, Chapter 4 attempted to evaluate the performance of the QLP system through various methodologies. These tests encompassed the generation of RTI maps, RD maps, spectrograms, and CFAR maps. The testing methodology for the RTI map involved visual comparisons with reference maps, and the system demonstrated the capability to effectively plot and track targets. The tests for the RD map and spectrogram evaluated processing times and showed that the system efficiently processed data, meeting real-time requirements. However, the CFAR maps and CFAR Level plots exhibited significantly longer processing times, mainly due to the allocation and configuration of reference cells within the CFAR detection threshold algorithm. Pulse integration proved to be effective in reducing processing time, allowing the system to meet real-time criteria. The overall system performance analysis, as depicted in Figure 4.8, demonstrated that the system did not meet real-time requirements without pulse integration. The identified bottleneck in the CFAR algorithm significantly affected processing times and must be addressed for the system to achieve real-time performance. Further optimisation is necessary to enhance the overall efficiency and real-time capability of the QLP system.

The project achieved the goals that were set out, by designing and demonstrating a QLP system specifically tailored for the NeXtRAD and NetRAD radar systems. All components of the pipeline were effectively implemented using NetRAD data files, and some were designed to be adaptable for processing data from either radar system. While the implementation of CFAR processing faced challenges, the project, as a whole, is deemed a success.

Chapter 6

Recommendations

The relatively short duration of this project has left certain areas within the design and development unaddressed. It is hoped that this work will serve as an entry point for future investigations in the field. The following recommendations outline potential directions for future work.

6.1 Enhanced Constant False Alarm Rate Algorithm

Dedicate additional resources to refining the [CFAR](#) detection algorithm, with a focus on enhancing its capability to filter out interference and noise. This refinement process may involve exploring alternative [CFAR](#) algorithms and fine-tuning parameters to enhance detection accuracy.

The existing [CFAR](#) implementation currently presents a bottleneck in data processing and result plotting. Streamlining the process of gathering reference cells for the relevant reference window could significantly boost the system's performance. Additionally, there's room for improvement in optimising key algorithm parameters, such as guard cell and reference cell sizes, probability of false alarm, and thresholding techniques.

Consider the possibility of substituting the current [CFAR](#) algorithm with alternative methods like [GOCA-CFAR](#) or [SOCA-CFAR](#). Such a change could potentially lead to improved [CFAR](#) processing accuracy without compromising the overall system's performance.

The overarching goal remains to fine-tune the [CFAR](#) algorithm for precise target differentiation and heightened accuracy in target detection. Extensive testing across diverse scenarios and radar conditions is imperative to validate the algorithm's robustness and adaptability.

6.2 Adaptation of System for NeXtRAD data

The pre-processing phase of the processing chain can be enhanced to efficiently handle the reading of the measured signal from the [HDF5](#) data files, along with loading the simulation parameters. These adjustments involve removing the conversion between real and complex data signals and eliminating the need for implementing the Hilbert Transform. In contrast, new components should be introduced into the pre-processing toolchain to accommodate various polarisation types and how each classification affects the data.

By streamlining the process, the system can read the simulation parameters and measured signal simultaneously from the same file, enhancing overall efficiency in data handling.

6.3 User Interface Improvements

The software tool's [GUI](#) can be further enhanced by introducing additional features and refining existing ones. One potential improvement is to create a 'Plots' tab where users can select and view all their chosen plots simultaneously, improving real-time monitoring. Another valuable addition could be the option to input previously plotted maps, enabling data and parameters to be reused for reprocessing and generating alternative plots. These are just a few suggestions, and the application's potential for improvement is extensive. The tool's adaptability allows for user-specific enhancements to meet their evolving needs.

6.4 C++ & Parallel Processing

The MATLAB code's design could be transferred to a language more suitable for real-time implementation, such as C++. C++ offers efficient memory management and supports parallel programming, making it an ideal choice for future development. Leveraging parallel programming in C++ could significantly accelerate some processing techniques, potentially addressing the bottleneck caused by the current [CFAR](#) detection threshold algorithm. This transition to C++ would enhance the system's overall performance and efficiency.

To enhance operational efficiency in the new programming language, a comprehensive reassessment of the system is necessary. After the transition to C++, it's crucial to benchmark the system's performance. This benchmarking process will help identify slower elements that need optimisation to meet the performance standards set by the rest of the system. By systematically addressing performance bottlenecks, the system can achieve a higher level of efficiency and functionality.

Bibliography

- [1] D. Du Plessis, “Pentek | NeXtRAD Multistatic Radar System.” [Online]. Available: https://www.pentek.com/applications/26_2/nextrad.cfm
- [2] M. A. Richards, “Doppler processing,” pp. 625–676, 2010. [Online]. Available: https://digital-library.theiet.org/content/books/10.1049/sbra021e_ch17
- [3] L. N. Ridenour, *Radar System Engineering*. McGraw-Hill, 1947.
- [4] E. Barlow, “Doppler radar,” *Proceedings of the IRE*, vol. 37, no. 4, pp. 340–355, 1949.
- [5] L. C. Perkins, H. B. Smith, and D. H. Mooney, “The development of airborne pulse doppler radar,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, no. 3, pp. 292–303, 1984.
- [6] M. B. Ringel, D. H. Mooney, and W. H. Long, “F-16 pulse doppler radar (an/apg-66) performance,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-19, no. 1, pp. 147–158, 1983.
- [7] W. R. Fried, “History of doppler radar navigation,” *Navigation : journal of the Institute of Navigation.*, vol. 40, no. 2, 1993-06.
- [8] T. Derham, K. Woodbridge, H. Griffiths, and C. Baker, “The design and development of an experimental netted radar system,” in *2003 Proceedings of the International Conference on Radar (IEEE Cat. No.03EX695)*, 2003, pp. 293–298.
- [9] T. E. Derham, “Design and evaluation of a coherent multistatic radar system,” Ph.D. dissertation, University of London, 2005.
- [10] T. Derham, S. Doughty, K. Woodbridge, and C. Baker, “Design and evaluation of a low-cost multistatic netted radar system,” *Radar, Sonar Navigation, IET*, vol. 1, pp. 362 – 368, 11 2007.
- [11] S. R. Doughty, *Development and performance evaluation of a multistatic radar system*. University of London, University College London (United Kingdom), 2008.
- [12] M. Inggs, G. Inggs, S. Sandenbergh, W. Al-Ashwal, K. Woodbridge, and H. Griffiths, “Multistatic networked radar for sea clutter measurements,” in *2011 IEEE International Geoscience and Remote Sensing Symposium*, 2011, pp. 4449–4452.
- [13] M. Inggs, A. Balleri, W. Al-Ashwal, K. Ward, K. Woodbridge, M. Ritchie, W. Miceli, R. Tough, C. Baker, S. Watts, R. Harmanny, A. Stove, J. Sandenbergh, and H. Griffiths, “Netrad multistatic sea clutter database,” in *2012 IEEE International Geoscience and Remote Sensing Symposium*, 2012, pp. 2937–2940.

- [14] M. Inggs, H. Griffiths, F. Fioranelli, M. Ritchie, and K. Woodbridge, “Multistatic radar: System requirements and experimental validation,” in *2014 International Radar Conference*, 2014, pp. 1–6.
- [15] S. Alhuwaimel, S. Coetzee, P. Cheng, D. Du Plessis, F. Fioranelli, H. Griffiths, M. Inggs, D. Jordan, W. Miceli, D. O’Hagan, R. Palama, S. Paine, M. Ritchie, J. Sandenbergh, and A. Stevens, “First measurements with nextrad, a polarimetric x/l band radar network,” in *2017 IEEE Radar Conference (RadarConf)*, 2017, pp. 1663–1668.
- [16] M. Inggs, S. Lewis, R. Palamà, M. Ritchie, and H. Griffiths, “Report on the 2018 trials of the multistatic nextrad dual band polarimetric radar,” in *2019 IEEE Radar Conference (RadarConf)*, 2019, pp. 1–6.
- [17] R. Palamà, F. Fioranelli, M. Ritchie, M. Inggs, S. Lewis, and H. Griffiths, “Measurements of multistatic xl band radar signatures of uavs,” in *2019 International Radar Conference (RADAR)*, 2019, pp. 1–5.
- [18] D. K. A. Pulutan and J. S. Marciano, “Design trade-offs in a combined fmcw and pulse doppler radar front-end,” in *IEEE 2013 Tencor - Spring*, 2013, pp. 567–571.
- [19] P. Anju, A. Bazil Raj, and C. Shekhar, “Pulse doppler processing - a novel digital technique,” in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2020, pp. 1089–1095.
- [20] P. Wojaczek, D. Cristallini, I. Walterscheid, and D. O’Hagan, “Range compression strategies for passive radar on airborne platforms,” in *2020 IEEE International Radar Conference (RADAR)*, 2020, pp. 25–30.
- [21] P. Burrascano, S. Laureti, L. Senni, and M. Ricci, “Pulse compression in nondestructive testing applications: Reduction of near sidelobes exploiting reactance transformation,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 5, pp. 1886–1896, 2019.
- [22] V. J. Arya and V. Subha, “Pulse compression using linear frequency modulation technique,” in *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 2017, pp. 921–926.
- [23] F. Argenti and L. Facheris, “Radar pulse compression methods based on nonlinear and quadratic optimization,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 3904–3916, 2021.
- [24] S. Blunt and K. Gerlach, “Adaptive pulse compression,” in *Proceedings of the 2004 IEEE Radar Conference (IEEE Cat. No.04CH37509)*, 2004, pp. 271–276.
- [25] H. Kikuchi, E. Yoshikawa, T. Ushio, F. Mizutani, and M. Wada, “Adaptive pulse compression technique for x-band phased array weather radar,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1810–1814, 2017.

- [26] H. Ma, M. Antoniou, D. Pastina, F. Santi, F. Pieralice, M. Bucciarelli, and M. Cherniakov, “Maritime moving target indication using passive gnss-based bistatic radar,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 1, pp. 115–130, 2018.
- [27] M. Ash, M. Ritchie, and K. Chetty, “On the application of digital moving target indication techniques to short-range fmcw radar data,” *IEEE Sensors Journal*, vol. 18, no. 10, pp. 4167–4175, 2018.
- [28] Z. Liu, D. K. C. Ho, X. Xu, and J. Yang, “Moving target indication using deep convolutional neural network,” *IEEE Access*, vol. 6, pp. 65 651–65 660, 2018.
- [29] C. Liu, F. Xi, S. Chen, Y. D. Zhang, and Z. Liu, “Pulse-doppler signal processing with quadrature compressive sampling,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 2, pp. 1217–1230, 2015.
- [30] G. Beltrão, L. Pralon, A. Barreto, M. Alae-Kerahroodi, and M. R. B. Shankar, “Subpulse processing for unambiguous doppler estimation in pulse-doppler noise radars,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 6, pp. 3813–3826, 2021.
- [31] T. Alemdaroğlu, Candan, and S. Koç, “The radar application of micro doppler features from human motions,” in *2015 IEEE Radar Conference (RadarCon)*, 2015, pp. 0374–0379.
- [32] Q. Zeng, Z. Kuang, S. Wu, and J. Yang, “A method of ultrasonic finger gesture recognition based on the micro-doppler effect,” *Applied Sciences*, vol. 9, no. 11, p. 2314, Jun 2019. [Online]. Available: <http://dx.doi.org/10.3390/app9112314>
- [33] Z. A. Cammenga, G. E. Smith, and C. J. Baker, “Combined high range resolution and micro-doppler analysis of human gait,” in *2015 IEEE Radar Conference (RadarCon)*, 2015, pp. 1038–1043.
- [34] W. Feng, J.-M. Friedt, G. Cherniak, and M. Sato, “Batch compressive sensing for passive radar range-doppler map generation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 3090–3102, 2019.
- [35] J. Akhtar and K. E. Olsen, “Formation of range-doppler maps based on sparse reconstruction,” *IEEE Sensors Journal*, vol. 16, no. 15, pp. 5921–5926, 2016.
- [36] C. M. Alabaster and E. J. Hughes, “Clutter mapping for airborne pulse doppler radar,” in *2008 IET Seminar on Radar Clutter Modelling*, 2008, pp. 49–55.
- [37] R. Palamà, F. Fioranelli, M. Ritchie, M. Inggs, S. Lewis, and H. Griffiths, “Measurements and discrimination of drones and birds with a multi-frequency multistatic radar system,” *IET Radar, Sonar & Navigation*, vol. 15, no. 8, pp. 841–852, 2021. [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/rsn2.12060>
- [38] D. B. Herr, T. J. Kramer, Z. Gannon, and D. Tahmoush, “Uav micro-doppler signature analysis,” in *2020 IEEE Radar Conference (RadarConf20)*, 2020, pp. 1–6.

- [39] E. Alhadhrami, M. Al-Mufti, B. Taha, and N. Werghi, “Learned micro-doppler representations for targets classification based on spectrogram images,” *IEEE Access*, vol. 7, pp. 139 377–139 387, 2019.
- [40] C. Kuang, C. Wang, B. Wen, Y. Hou, and Y. Lai, “An improved ca-cfar method for ship target detection in strong clutter using uhf radar,” *IEEE Signal Processing Letters*, vol. 27, pp. 1445–1449, 2020.
- [41] J. Abdullah and M. S. Kamal, “Multi-targets detection in a non-homogeneous radar environment using modified ca-cfar,” in *2019 IEEE Asia-Pacific Conference on Applied Electromagnetics (APACE)*, 2019, pp. 1–5.
- [42] M. Zhang and X. Li, “An efficient real-time two-dimensional ca-cfar hardware engine,” in *2019 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, 2019, pp. 1–3.
- [43] H. Zhang, M. Tian, Q. Ouyang, J. Liu, G. Shao, and J. Cheng, “Track detection of underwater moving targets based on cfar,” *Journal of Physics: Conference Series*, vol. 2486, no. 1, p. 012076, may 2023. [Online]. Available: <https://dx.doi.org/10.1088/1742-6596/2486/1/012076>
- [44] S. W. Hameed, “Peaks detector algorithm after cfar for multiple targets detection,” *EAI Endorsed Transactions on AI and Robotics*, vol. 1, no. 1, 7 2022.
- [45] A. Jalil, H. Yousaf, and M. I. Baig, “Analysis of cfar techniques,” in *2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, 2016, pp. 654–659.
- [46] R. Sor, J. S. Sathone, S. U. Deoghare, and M. S. Sutaone, “Os-cfar based on thresholding approaches for target detection,” in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1–6.
- [47] L. Abdou, O. Taibaoui, A. Moumen, and A. T. Ahmed, “Threshold optimization in distributed os-cfar system by using simulated annealing technique,” in *2015 4th International Conference on Systems and Control (ICSC)*, 2015, pp. 295–301.
- [48] V. V. Tien, T. Vu Hop, L. H. Nam, N. Van Loi, and T. T. Thanh, “An adaptive 2d-os-cfar thresholding in clutter environments: Test with real data,” in *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, 2018, pp. 116–119.
- [49] F. Yavuz, “Radar target detection with cnn,” in *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 1581–1585.
- [50] A. De Maio and A. Aubry, “Radar detection, performance analysis, and cfar techniques,” in *2019 IEEE Radar Conference (RadarConf)*, 2019, pp. 1–120.
- [51] S. Wang, “Research on multichannel spectrum detection algorithm in cognitive radio system,” in *Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering*, ser. ICTCE ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 290–294. [Online]. Available: <https://doi.org/10.1145/3291842.3291866>

- [52] M. C. Luna Alvarado, F. D. A. García, L. P. J. Jiménez, G. Fraidenraich, and Y. Iano, “Performance evaluation of soca-cfar detectors in weibull-distributed clutter environments,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
- [53] C. Xu, Y. Li, C. Ji, Y. Huang, H. Wang, and Y. Xia, “An improved cfar algorithm for target detection,” in *2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2017, pp. 883–888.
- [54] R. Palamà, M. Greco, P. Stinco, and F. Gini, “Analysis of sea spikes in netrad clutter,” in *2014 11th European Radar Conference*, 2014, pp. 109–112.
- [55] ———, “Statistical analysis of netrad high resolution sea clutter,” in *21st European Signal Processing Conference (EUSIPCO 2013)*, 2013, pp. 1–5.
- [56] M. Ritchie, A. Stove, K. Woodbridge, and H. Griffiths, “Netrad: Monostatic and bistatic sea clutter texture and doppler spectra characterization at s-band,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 9, pp. 5533–5543, 2016.
- [57] F. Hoffmann, M. Ritchie, F. Fioranelli, A. Charlish, and H. Griffiths, “Micro-doppler based detection and tracking of uavs with multistatic radar,” in *2016 IEEE Radar Conference (RadarConf)*, 2016, pp. 1–6.
- [58] D. Callaghan, J. Burger, and A. K. Mishra, “A machine learning approach to radar sea clutter suppression,” in *2017 IEEE Radar Conference (RadarConf)*, 2017, pp. 1222–1227.
- [59] F. Fioranelli, M. Ritchie, and H. Griffiths, “Classification of unarmed/armed personnel using the netrad multistatic radar for micro-doppler and singular value decomposition features,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 9, pp. 1933–1937, 2015.
- [60] S. Coetzee and M. Inggs, “Narrowband radar imaging using netrad and nextrad data,” in *International Conference on Radar Systems (Radar 2017)*, 2017, pp. 1–4.
- [61] M. Inggs, S. Coetzee, H. Griffiths, F. Fioranelli, M. Ritchie, and K. Woodbridge, “Database design for an experimental, dual band, polarimetric radar,” in *2015 IEEE Radar Conference*, 2015, pp. 417–421.
- [62] M. Inggs, H. Griffiths, F. Fioranelli, M. Ritchie, and K. Woodbridge, “Multistatic radar: System requirements and experimental validation,” in *2014 International Radar Conference*, 2014, pp. 1–6.
- [63] S. Alhuwaimel and K.-F. Tong, “Performance enhancement of a slotted waveguide antenna by utilizing parasitic elements,” in *2015 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC)*, 2015, pp. 1303–1306.
- [64] S. Alhuwaimel, S. Coetzee, P. Cheng, D. Du Plessis, F. Fioranelli, H. Griffiths, M. Inggs, D. Jordan, W. Miceli, D. O’Hagan, R. Palama, S. Paine, M. Ritchie, J. Sandenbergh, and A. Stevens, “First measurements with nextrad, a polarimetric x/l band radar network,” in *2017 IEEE Radar Conference (RadarConf)*, 2017, pp. 1663–1668.

- [65] P. Beasley, M. Ritchie, H. Griffiths, W. Miceli, M. Inggs, S. Lewis, and B. Kahn, "Multistatic radar measurements of uavs at x-band and l-band," in *2020 IEEE Radar Conference (RadarConf20)*, 2020, pp. 1–6.
- [66] J. Sandenbergh and M. Inggs, "A summary of the results achieved by the gps disciplined references of the netrad and nextrad multistatic radars," in *2019 IEEE Radar Conference (RadarConf)*, 2019, pp. 1–6.
- [67] S. C. Jonkers, "Software infrastructure for nextrad development in julia programming language," Master's thesis, University of Cape Town, 2016.
- [68] D. Jordan, M. Inggs, and D. O'Hagan, "Nextlook a lightweight, real-time quick-look processor for nextrad," in *International Conference on Radar Systems (Radar 2017)*, 2017, pp. 1–6.
- [69] M. Inggs, D. Jordan, and T. Learner, "Through the clouds: Exploring the potential of infrastructure-as-a-service for radar," in *International Conference on Radar Systems (Radar 2017)*, 2017, pp. 1–6.
- [70] Q. Zhang, T. Li, Y. Zhu, Z. Yu, Z. Ding, X. Guo, and Z. Lv, "Back projection algorithm-based geo sar real-time quick look imaging system," *The Journal of Engineering*, vol. 2019, no. 19, pp. 6341–6345, 2019. [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/joe.2019.0200>
- [71] B. Setiadi, Z. Baharuddin, G. Panggabean, H. Kuze, and J. Sumantyo, "Development of quicklook processor for circularly polarized synthetic aperture radar onboard gaia-ii microsatellite," *Progress and Communication in Sciences*, vol. 2, no. 2, 2016. [Online]. Available: <http://ojs.unsysdigital.com/index.php/pcs/article/view/876>
- [72] M. Fausnaugh, X. Huang, A. Glidden, N. Guerrero, and TESS Science Office, "TESS Data Processing and Quick-look Pipeline," in *American Astronomical Society Meeting Abstracts #231*, ser. American Astronomical Society Meeting Abstracts, vol. 231, Jan. 2018, p. 439.09.
- [73] C. X. Huang, "Tess lightcurves from the mit quick-look pipeline ('qlp')," 2020. [Online]. Available: <http://archive.stsci.edu/doi/resolve/resolve.html?doi=10.17909/t9-r086-e880>
- [74] M. Kunimoto, C. Huang, E. Tey, W. Fong, K. Hesse, A. Shporer, N. Guerrero, M. Fausnaugh, R. Vanderspek, and G. Ricker, "Quick-look pipeline lightcurves for 9.1 million stars observed over the first year of the tess extended mission," *Research Notes of the AAS*, vol. 5, no. 10, p. 234, oct 2021. [Online]. Available: <https://dx.doi.org/10.3847/2515-5172/ac2ef0>
- [75] M. Kunimoto, E. Tey, W. Fong, K. Hesse, A. Shporer, M. Fausnaugh, R. Vanderspek, and G. Ricker, "Quick-look pipeline light curves for 5.7 million stars observed over the second year of tess' first extended mission," *Research Notes of the AAS*, vol. 6, no. 11, p. 236, nov 2022. [Online]. Available: <https://dx.doi.org/10.3847/2515-5172/aca158>
- [76] M. Inggs, A. Balleri, W. Al-Ashwal, K. Ward, K. Woodbridge, M. Ritchie, W. Miceli, R. Tough, C. Baker, S. Watts, R. Harmanny, A. Stove, J. Sandenbergh, and H. Griffiths, "Netrad multistatic

- sea clutter database,” in *2012 IEEE International Geoscience and Remote Sensing Symposium*, 2012, pp. 2937–2940.
- [77] P. Podder, T. Z. Khan, M. H. Khan, and M. M. Rahman, “Comparative performance analysis of hamming, hanning and blackman window,” *International Journal of Computer Applications*, vol. 96, no. 18, pp. 1–7, 2014.
- [78] J. Dugundji, “Envelopes and pre-envelopes of real waveforms,” *IRE Transactions on Information Theory*, vol. 4, no. 1, pp. 53–57, 1958.
- [79] P. Murrel, “7.7 Binary files.” [Online]. Available: <https://statmath.wu.ac.at/courses/data-analysis/itdtHTML/node58.html>
- [80] N. R. Rypkema, “A straightforward derivation of the matched filter,” 2019.
- [81] L. Rabiner and R. Schafer, *Theory and Applications of Digital Speech Processing*, 1st ed. USA: Prentice Hall Press, 2010.