



**UNIVERSITY OF
BIRMINGHAM**

**AUTOMATIC VEHICLE CLASSIFICATION IN A LOW
FREQUENCY FORWARD SCATTER MICRO-RADAR**

By

Nur Emileen Binti Abd Rashid

**A thesis submitted to School of Electronic, Electrical and
Computer Engineering of The University of Birmingham**

for the degree of

DOCTOR OF PHILOSOPHY

2011

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

Abstract

Forward Scattering Radar (FSR) is a special case of bistatic radar configuration where the desired radar signal is formed via the shadowing of the direct (transmitter-to-receiver) signal by the target body. FSR offers a number of interests including an inherent ability to detect stealth target, absence of signal fluctuations, reasonably simple hardware, enhanced target radar cross-section (RCS) compared to traditional radar and capability to use Inverse Synthetic Aperture algorithms for Automatic Target Classification (ATC). Of course as any system FSR has its own drawbacks and limitations.

This thesis presents the research results on development of ATC algorithm under a variety of external factors such as clutter and target's trajectories uncertainties. The peculiarity of this research are that the FSR operates at a low (VHF and UHF) frequency bands that in a strict sense does not correspond to an optical region for vehicles like targets and the system operate with omnidirectional antennas. There is no previous research considered this practically important case. The algorithm is developed based on Fourier transform, Principal Component Analysis (PCA) and K-Nearest Neighbour (KNN) classifier - for features extraction, transformation and classification, respectively. The ATC system is integrated with coherent signal processing algorithm in order to estimate target's motion parameters (i.e speed) prior to spectra normalisation process. The analytical and modelling results are experimentally confirmed. As ATC performance degraded when high level of clutter is present, clutter-compensated ATC model is introduced and its classification performance is analysed using measured signals with added simulated clutter.

Acknowledgements

First of all thanks to Allah the Al-Mighty who guided me to start this journey and gave me the strength to continue this learning process. There are a number of people who have helped me during my PhD through their support and encouragement.

First and foremost, I would like to express my gratitude to my supervisors Prof. Mike Cherniakov and Dr. Peter Jancovic for inspiration, guidance and providing the right balance of suggestions, criticism and academic independence. They have made this thesis possible.

Over the years, I have benefited greatly from interaction with other members of the Microwave Integrated System Laboratory, University of Birmingham especially Dr Marina Gashinova and Dr Vladimir Sizov. Thank you for creating such a pleasant and motivating atmosphere, for stimulating discussions, for the encouragement and support you have given me and also for all the fun we have had in the last four years.

I also want to thank my grant sponsors, Ministry of Education Malaysia and UiTM who have provided financial support throughout my study and made this opportunity possible.

Thank you to my parents and family– for always supporting me in whatever I wanted to do. And finally, to my daughters, Sofiya and Sabryna and my beloved husband, Azhar Ahmad – to whom this thesis is dedicated, thank you for your endless support and love during my endeavours.

Table of Contents

| | |
|---|----|
| CHAPTER 1: INTRODUCTION..... | 1 |
| 1.1 Background..... | 1 |
| 1.2 Motivation for the Present work..... | 3 |
| 1.3 Problem Statement and Original Work..... | 8 |
| 1.4 Thesis Organisation | 10 |
| CHAPTER 2: TECHNOLOGY OVERVIEW | 11 |
| 2.1 Introduction..... | 11 |
| 2.2 Radar Technology..... | 12 |
| 2.3 Forward Scatter Radar | 13 |
| 2.4 The Forward Scatter Radar Equation | 15 |
| 2.5 The Forward Scatter Radar Cross Section..... | 17 |
| 2.5.1 Optical approach to FS RCS estimation..... | 18 |
| 2.5.2 FS RCS prediction for electrically small targets | 21 |
| 2.6 Doppler Effect | 25 |
| 2.7 Target Signature in FSR | 28 |
| 2.8 Literature Review on Forward Scatter Ground Target Classification | 37 |
| 2.9 Pattern Classification | 42 |
| 2.10 Automatic Target Classification (ATC) procedure | 44 |
| 2.10.1 Pre-processing | 46 |
| 2.10.2 Classification System | 54 |
| 2.11 Limitation | 60 |
| 2.12 Summary..... | 61 |

| | |
|--|-----|
| CHAPTER 3: TARGET SIGNATURE EXPERIMENTAL COLLECTION AND ANALYSIS | 62 |
| 3.1 Introduction..... | 62 |
| 3.2 Experimental data collection | 63 |
| 3.2.1 Hardware description..... | 63 |
| 3.2.2 Recording data format | 69 |
| 3.2.3 Data collection method..... | 73 |
| 3.3 Experimental data description | 75 |
| 3.4 Preliminary Study | 77 |
| 3.4.1 Low frequency..... | 77 |
| 3.4.2 Effect of different positions of sensors..... | 80 |
| 3.4.3 Effect of different baseline lengths..... | 84 |
| 3.4.4 Effect of different crossing points | 89 |
| 3.4.5 Effect of different trajectory angles..... | 91 |
| 3.4.6 Effect of obstacles | 94 |
| 3.5 Conclusion | 98 |
| CHAPTER 4: CLUTTER SIMULATION | 99 |
| 4.1 Introduction..... | 99 |
| 4.2 Foliage clutter | 100 |
| 4.2.1 Non-stationary Clutter | 101 |
| 4.3 Non-stationary Clutter Simulation..... | 108 |
| 4.3.1 Generation of Coloured noise/stationary clutter..... | 110 |
| 4.3.2 Envelope creation | 111 |
| 4.3.3 Modulation Process | 114 |
| 4.4 Conclusion | 120 |

| | |
|---|-----|
| CHAPTER 5: TARGET MOTION PARAMETER ESTIMATION AND DOPPLER SIGNATURE NORMALISATION AS THE MAJOR PROCESSING ROUTINE FOR ATC | 121 |
| 5.1 Introduction..... | 121 |
| 5.2 Target Motion parameter estimation | 122 |
| 5.2.1 Coherent signal processing algorithm | 126 |
| 5.2.2 Algorithm verification in clutter free conditions | 131 |
| 5.3 Doppler spectra normalisation using estimated speed of target | 135 |
| 5.3.1 Analysis of speed estimation error in “clutter free” environment | 136 |
| 5.3.2 The influence of clutter on the speed estimation | 140 |
| 5.4 Doppler spectra normalisation for different baselines | 145 |
| 5.4.1 Simulated target | 147 |
| 5.4.2 Measured signal | 164 |
| 5.5 Conclusion | 169 |
| CHAPTER 6: ATC PERFORMANCE DEGRADATION IN CLUTTERED ENVIRONMENT | 171 |
| 6.1 Introduction..... | 171 |
| 6.2 Possible source of errors in cluttered environment..... | 172 |
| 6.3 ATC degradation with the speed estimation error and spectrum’s shape deformation error..... | 176 |
| 6.4 Clutter-compensated ATC system | 181 |
| 6.5 Conclusion | 190 |
| CHAPTER 7: CONCLUSIONS AND FUTURE WORKS | 193 |
| References | 198 |
| Appendix A : Antenna specification | 204 |
| Appendix B : Samples of Records from test sessions at Tilesford Airfield | 207 |

| | |
|--|-----|
| Appendix C : Example of an Html File | 208 |
| Appendix D: Example of manual on how to perform target classification..... | 209 |
| Appendix E : Explanation of software | 212 |
| Appendix F: Publications | 256 |

List of Figures

CHAPTER 1: INTRODUCTION

| | |
|---|---|
| Figure 1.1 The concept of the FSR micro-sensors radar network | 5 |
| Figure 1.2 Forward Scatter Micro-Radar network block diagram | 7 |
| Figure 1.3 FSR target classification flowchart | 9 |

CHAPTER 2: TECHNOLOGY OVERVIEW

| | |
|---|----|
| Figure 2.1 Radar systems | 13 |
| Figure 2.2 Shadowing field | 18 |
| Figure 2.3 Plane wave incidence on the rectangular plate /aperture | 19 |
| Figure 2.4 RCS of a sphere | 22 |
| Figure 2.5 Different car models..... | 24 |
| Figure 2.6 FS RCS of different car models | 25 |
| Figure 2.7 Doppler shift for a moving source | 26 |
| Figure 2.8 Monostatic radar geometry for moving target | 27 |
| Figure 2.9 The definition of the Doppler shift in bistatic radar..... | 27 |
| Figure 2.10 Doppler Effect for a moving target in FSR..... | 29 |
| Figure 2.11 Target signature comparisons for point and flat-shaped targets | 34 |
| Figure 2.12 Measured Doppler signatures for different cars..... | 36 |
| Figure 2.13 Experimentation layout (signals recorded using video camera) | 39 |
| Figure 2.14 Standard deviation vs. speed of the vehicle from video camera..... | 40 |
| Figure 2.15 General scheme of a pattern recognition system..... | 42 |
| Figure 2.16 Example of a feature space, X..... | 44 |
| Figure 2.17 Automatic Target Classification (ATC) block diagram..... | 45 |
| Figure 2.18 Pre-processing block diagram | 46 |
| Figure 2.19 Comparisons between target's power spectrum and clutter power spectrum..... | 47 |
| Figure 2.20 (a) Low-pass filter and (b) High-pass filter frequency response..... | 48 |

| | |
|---|----|
| Figure 2.21 (a) Time domain signature and (b) frequency domain for Nissan Almera and Land rover | 50 |
| Figure 2.22 Amplitude normalisation..... | 51 |
| Figure 2.23 The spectra of a same car at two different speeds before | 53 |
| Figure 2.24 Overall system for vehicle classification that has been used in this thesis | 54 |
| Figure 2.25 The variance and cumulative described by a given principal component | 57 |
| Figure 2.26 The location of training data of each vehicle type in the PCA-space | 58 |
| Figure 2.27 Distance between test data and training data for $k=3$ | 59 |

CHAPTER 3: TARGET SIGNATURE EXPERIMENTAL COLLECTION AND ANALYSIS

| | |
|---|----|
| Figure 3.1 Multi-frequency equipment..... | 64 |
| Figure 3.2 Example of 3N_TX block diagram | 65 |
| Figure 3.3 Antenna (a) Panorama helical (b) 9dB directional Yagi | 66 |
| Figure 3.4 3N_RX block diagram | 68 |
| Figure 3.5 Receiver detection range | 69 |
| Figure 3.6 Layout of data acquisition setting | 71 |
| Figure 3.7 Data acquisition layout..... | 72 |
| Figure 3.8 Data visualisation layout | 73 |
| Figure 3.9 Tilesford Airfield | 74 |
| Figure 3.10 Pritchatts car park, University of Birmingham | 75 |
| Figure 3.11 Spectra for Peugeot 406 (red), Volvo S40 (blue) and Land Rover (green) at various frequencies | 78 |
| Figure 3.12 The location of data for each vehicle in the 2 components in the PCA..... | 79 |
| Figure 3.13 2-D PCA ATC plots for low frequency | 79 |
| Figure 3.13 Sensor positions at Tilesford Airfield | 81 |
| Figure 3.14 Pritchatts car park (a) experimental layout (b) sensors moved 5m from original position and (c) sensors moved 10m from original position..... | 81 |
| Figure 3.15 Spectra for different positions at Tilesford airfield..... | 82 |
| Figure 3.16 Spectra for different positions at Pritchatts car park..... | 83 |
| Figure 3.17 Topology for different baseline lengths | 84 |

| | |
|---|----|
| Figure 3.18 Spectra for different baseline distances | 85 |
| Figure 3.19 Experiment set up for different baseline distances. | 86 |
| Figure 3.20 Time domain signal for all baseline distances at 64 MHz | 87 |
| Figure 3.21 Experimental received spectrum for four different baseline lengths, BL | 88 |
| Figure 3.22 Experiment layout for different crossing points..... | 89 |
| Figure 3.23 Spectra for different crossing points | 90 |
| Figure 3.24 Experiment layout for different crossing angles | 92 |
| Figure 3.25 Spectra for different crossing angles at various frequencies..... | 93 |
| Figure 3.26 Experiment layout for different obstacle positions | 94 |
| Figure 3.27 Spectra for different obstacle positions..... | 96 |
| Figure 3.28 The location of data for the different obstacle positions in the 2D PCA plots | 97 |

CHAPTER 4: CLUTTER SIMULATION

| | |
|--|-----|
| Figure 4.1 Measured clutter..... | 103 |
| Figure 4.2 Power spectrum density of measured clutter | 103 |
| Figure 4.3 Clutter envelopes..... | 104 |
| Figure 4.4 PSD envelope | 105 |
| Figure 4.5 Distribution of measured clutter at 64 MHz, 135 MHz, 173 MHz and 434 MHz channels..... | 106 |
| Figure 4.6 Clutter (a) short period (1 min) and (b) long period (20 min) | 107 |
| Figure 4.7 Clutter simulation model block-diagrams..... | 108 |
| Figure 4.8 Stationary Clutter developments | 110 |
| Figure 4.9 Simulated Stationary clutter | 110 |
| Figure 4.10 Signal pre-envelope creation block diagram..... | 111 |
| Figure 4.11 Envelope (output of Hilbert transform) after low-pass filter | 112 |
| Figure 4.12 Envelope normalisation..... | 113 |
| Figure 4.13 Scaled envelope..... | 114 |
| Figure 4.15 Modulation process | 115 |
| Figure 4.15 Non-stationary Clutter..... | 115 |

| | |
|--|-----|
| Figure 4.16 Non-stationary clutter statistical characteristic for 434 MHz (a) clutter with envelope (b) power spectrum density for envelope (c) power spectrum density for clutter and (d) Weibull distribution for clutter..... | 116 |
| Figure 4.17(a) Envelope and (b) PSD for different level clutter condition at 434 MHz..... | 117 |
| Figure 4.18 (a) Envelope and (b) PSD of simulated clutter for different frequencies when clutter is very strong..... | 118 |
| Figure 4.19 Distribution of simulated clutter at 64 MHz, 135 MHz, 173 MHz and 434 MHz channels..... | 119 |
| Figure 4.20 Envelope comparisons between measured and simulated clutter | 120 |

CHAPTER 5: TARGET MOTION PARAMETER ESTIMATION AND DOPPLER SIGNATURE NORMALISATION AS THE MAJOR PROCESSING ROUTINE FOR ATC

| | |
|---|-----|
| Figure 5.1 FSR configuration in two-ray propagation | 123 |
| Figure 5.2 Example of target signature resembles a two-sided chirp signal | 124 |
| Figure 5.3 Simulated signatures of targets with different velocities | 125 |
| Figure 5.4 Example of Pulse compression waveform | 128 |
| Figure 5.5 Processing scheme | 128 |
| Figure 5.6 Example of a received signal and reference signal that provide maximum cross correlation | 129 |
| Figure 5.7 The extracted information that can be obtained as a result of matching filtering. | 129 |
| Figure 5.8 Compressed output of a matched filter with a correct matching reference signal | 130 |
| Figure 5.9 Compressed output of a matched filter with an incorrect matching reference signal | 131 |
| Figure 5.10 Histogram of speed values after introducing a 5% error to the true speed of 8.3 m/s..... | 137 |
| Figure 5.11 The 2-D PCA components (for a frequency of 151 MHz) when considering: ... | 138 |
| Figure 5.12 Classification accuracy with error in speed estimation for all frequencies..... | 139 |
| Figure 5.13 Generation of clutter and SCR definition..... | 141 |
| Figure 5.14 ATC with speed estimated using various levels of SCRs at different frequencies. | 144 |

| | |
|--|-----|
| Figure 5.15 Signals for different baseline at 64 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling to baseline of 250m | 148 |
| Figure 5.16 Target position for signal duration of 20s, 40s and 60s. | 149 |
| Figure 5.17 PSD after resampled to baseline when the signal's duration is increased from 20seconds to (a) 40seconds and (b) 60seconds..... | 149 |
| Figure 5.18 Signals for different baselines at 151 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling, for a baseline of 250m | 150 |
| Figure 5.19 Signals for different baselines at 434 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m | 151 |
| Figure 5.20 RCS radiation pattern for an extended rectangular shape target when crossing a baseline of 100m at a speed of 15m/s. | 153 |
| Figure 5.21 Square targets with 1mx1m dimensions for different baselines at 64 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m..... | 155 |
| Figure 5.22 Square targets with 1mx1m dimensions for different baselines at 151 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m | 156 |
| Figure 5.23 Square targets with a 1mx1m dimensions for different baselines at 434 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m | 157 |
| Figure 5.24 A rectangular plate target with dimensions of a Land Rover..... | 158 |
| Figure 5.25 A complex target with dimensions of 4.71mx2.2m for different baselines at 64 MHz (a) time domain signals before resampling, (b) frequency domain signals | |

| | |
|--|-----|
| before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m | 159 |
| Figure 5.26 A complex target with dimensions of 4.71mx2.2m for different baselines at 151 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m | 160 |
| Figure 5.27 A complex target with dimensions of 4.71mx2.2m for different baselines at 434 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m | 161 |
| Figure 5.28 The RCS radiation pattern for an extended rectangular shaped target when crossing a baseline of 100m with speed of 15m/s..... | 163 |
| Figure 5.29 Dead zone for different baseline lengths..... | 164 |
| Figure 5.30 Measured signals at 64 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling | 165 |
| Figure 5.31 Measured signal at 151 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling | 166 |
| Figure 5.32 Measured signal at 434 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling | 167 |
| Figure 5.33 2D PCA plots for different baseline lengths using measured signals at (a) 64 MHz, (b) 151 MHz and (c) 434 MHz | 168 |

CHAPTER 6: ATC PERFORMANCE DEGRADATION IN CLUTTERED ENVIRONMENT

| | |
|--|-----|
| Figure 6.1 Spectra of the same signal with three different level of clutter strength:..... | 173 |
| Figure 6.2 Signal processing block diagram with added simulated clutter | 174 |
| Figure 6.3 Spectra of different types of errors with SCR=10dB at 151 MHz..... | 175 |

| | |
|--|-----|
| Figure 6.4 PCA plots for different level of SCR when only shape deformation error presence at frequency 151 MHz when SCR (a) 5dB (b) 15dB and (c) 30 dB | 178 |
| Figure 6.5 PCA plots for SCR=10 dB when only shape deformation error presence at various frequencies (a) 64 MHz, (b) 151 MHz and (c) 434 MHz | 179 |
| Figure 6.6 Target classification accuracy | 180 |
| Figure 6.7 Block diagram of the multi-environment model..... | 183 |
| Figure 6.8 Example of feature extraction | 184 |
| Figure 6.9 PCA plots for different level of SCR when only shape deformation error presence at frequency 151 MHz when SCR (a) 5dB (b) 15dB and (c) 30 dB | 186 |
| Figure 6.10 Comparison of the classification obtained using the clutter-uncompensated and clutter-compensated ATC system | 188 |
| Figure 6.11 An example of the target signal with indicated part of signal used for estimation of the clutter energy | 189 |

List of Tables

CHAPTER 3: TARGET SIGNATURE EXPERIMENTAL COLLECTION AND ANALYSIS

| | |
|---|----|
| Table 3.2 Transmitter and receiver modules and antennas | 65 |
|---|----|

CHAPTER 4: CLUTTER SIMULATION

| | |
|--|-----|
| Table 4.2 Characteristics of real measured clutter | 109 |
|--|-----|

CHAPTER 6: ATC PERFORMANCE DEGRADATION IN CLUTTERED ENVIRONMENT

| | |
|--|-----|
| Table 6.1 Target classification accuracy achieved by the clutter-compensated ATC system for the ideal and real case scenarios when using frequency 64, 151 and 434 MHz | 190 |
|--|-----|

Glossary of Abbreviations

| | |
|----------------|--|
| ADC | Analogue to D igital C onverters |
| ATC | A utomatic T arget C lassification |
| AWGN | A rbitrary W hite G aussian N oise |
| BS | B ack S cattering |
| CW | C ontinuous W ave |
| CST | C omputer S imulation T echnology |
| DEW | D istant E arly W arning |
| EM | E lectrom m agnetic |
| FFT | F ast F ourier T ransform |
| FS ISAR | F orward S catter I nverse S ynthetic A perture R adar |
| FS | F orward S cattering |
| FSR | F orward S cattering R adar |
| FSSC | F orward S catter C ross S ection |
| GMM | G aussian M ixture M odelling |
| HPF | H igh P ass F ilter |
| HMM | H idden M arkov M odel |
| KNN | K - N earest N eighbour |
| LPF | L ow P ass F ilter |
| MTI | M oving T arget I ndication |
| NN | N eural N etwork |
| OTH | O ver T he H orizon |
| PC | P rincipal C omponent |
| PCA | P rincipal C omponent A nalysis |

| | |
|-------------|---|
| PDF | Probability Density Function |
| PEC | Perfect Electrical Conductive |
| PSD | Power Spectrum Density |
| RAM | Radar Absorbing Material |
| RCS | Radar Cross Section |
| RMS | Root Mean Square |
| RSSI | Received Signal Strength Indicator |
| SCR | Signal to Clutter Ratio |
| TRP | Two-ray Path |
| UAV | Unmanned Aerial Vehicle |
| USB | Universal Serial Bus |
| VHF | Very High Frequency |
| VP | Vertical Polarized |

CHAPTER 1

INTRODUCTION

1.1 Background

The electromagnetic wave forward scattering (FS) phenomenon, broadly referred to as shadowing, has been known for many years. It was first investigated as a part of geometrical optics and later on in physical optics. The phenomenon of radio wave scattering in a forward direction was identified around the turn of the last century, when the first experiments with radio wave propagation were undertaken [2]. By the time World War II started, several countries including the US, UK, Japan, Russia had developed and deployed radar based on FS, known as Forward Scatter Radar (FSR) [2]. However, despite general interest in FSR, it was not until the Cold War that wider research into radio frequency and microwave FS began in earnest, fuelled by a need to develop radars capable of detecting stealth aircraft. Today, the detection of stealth targets remains a major challenge with the introduction of additional

stealth targets such as ballistic missile heads, cruise missiles, ships, jet-skis and ground vehicles. FSR [3-7] possesses a number of fundamental characteristics. Some of these can be considered as advantages, for example: enhanced target radar cross-sections (RCS); a robustness to a stealth target's shape and coating; an absence of target phase fluctuations that make coherent signal processing possible over the full visibility time of the target¹; and relatively simple hardware required for implementation. Some of the characteristics might be viewed as drawbacks and limitations, for example, the bistatic nature of FSR, operating within a relatively narrow corridor adjacent to the baseline and the absence of range resolution. Over the past decade there has been increased interest in bistatic, multistatic and netted radar, and forward scatter radar in particular. The traditional functions of radar are target detection, parameter estimation, tracking and Automatic Target Classification (Recognition) (ATC). Of these functions, the core focus of our research will be on Automatic Target Classification in Forward Scatter Radar [1] for ground-target vehicles in particular.

Effective target classification in FSR was used in ground-based air defence radars in the mid-80s [8]. Chapursky developed a new approach for FSR analysis known as radio holography and appropriate radars referred to as Forward Scatter (Shadow) Inverse Synthetic Aperture Radar (FS ISAR) [8]. In FS ISAR, the stability of the received signals can be achieved due to the absence of phase fluctuation and long coherent integration time; thus, ATC algorithms could be effectively applied.

The effectiveness of FSR for ground-target (humans and vehicles) detection and vehicle classification has recently been investigated [9-17]. A wireless network for situational

¹ That may be in the order of several seconds or even minutes.

awareness was proposed, where each node of the network has a transceiver and a pair of two nodes which form the FSR. When a target is crossing the radar baseline, it is partially blocking and disturbing the radio waves and the Doppler signature is registered at the receiver. This signal is then used for target detection, parameter estimation, tracking and potentially ATC.

1.2 Motivation for the Present work

Automatic target classification is perhaps the most complex area of the major radar functions: detection, parameter estimation and tracking and ATC. Some research has been done in this area for air targets [18-22] . However, there are key differences between air defence and ground-sensor radars when it comes to ATC [18-22]. FSR air defence:

- ✓ The use of directional antennas positioned on tall masts in order to minimise ground reflections (multipath) and clutter from the underlying vegetation;
- ✓ Air targets are presumed to be moving at a high-speed compared with ground targets and hence their spectrums are better separated from the clutter spectrum;
- ✓ Air target operations are over a range of many kilometres, and the targets do not come within close range of the antennas.

Thus it is difficult, if not impossible, to use the results of air target automatic classification and apply these results directly for ground targets.

Some elementary study on ground-target detection has been undertaken by Raja in his PhD thesis [1]. His research was first demonstrating the feasibility of ground target classification in FS sensors. However the research was conducted for ideal conditions, namely:

- ✓ A high frequency of about 1 GHz was used, i.e. a strictly optical region for a typical vehicle;
- ✓ The directional antennas were facing each other and were positioned at about one metre above the ground;
- ✓ The transmitter and receiver were separated only by about 20 m;
- ✓ All data collections were done in clutter-free conditions, partly because of the directional antennas and partly because the experiments were conducted above the concrete surface of a road;
- ✓ The target trajectories were known and strictly linear;
- ✓ Speed was estimated using a playback video camera.

In our research we are considering a more realistic, and hence a more complex, scenario for ATC in a wireless network for situational awareness based on FS sensors. An example of such a network, proposed for a defence application and studied in [23], is shown in Figure 1.1. This is perhaps the most complex type of network with sensors delivered by air. Other configuration and installation methods should definitely be considered. The area of research that we will focus on is where only a pair of sensors is considered. This is highlighted in yellow.

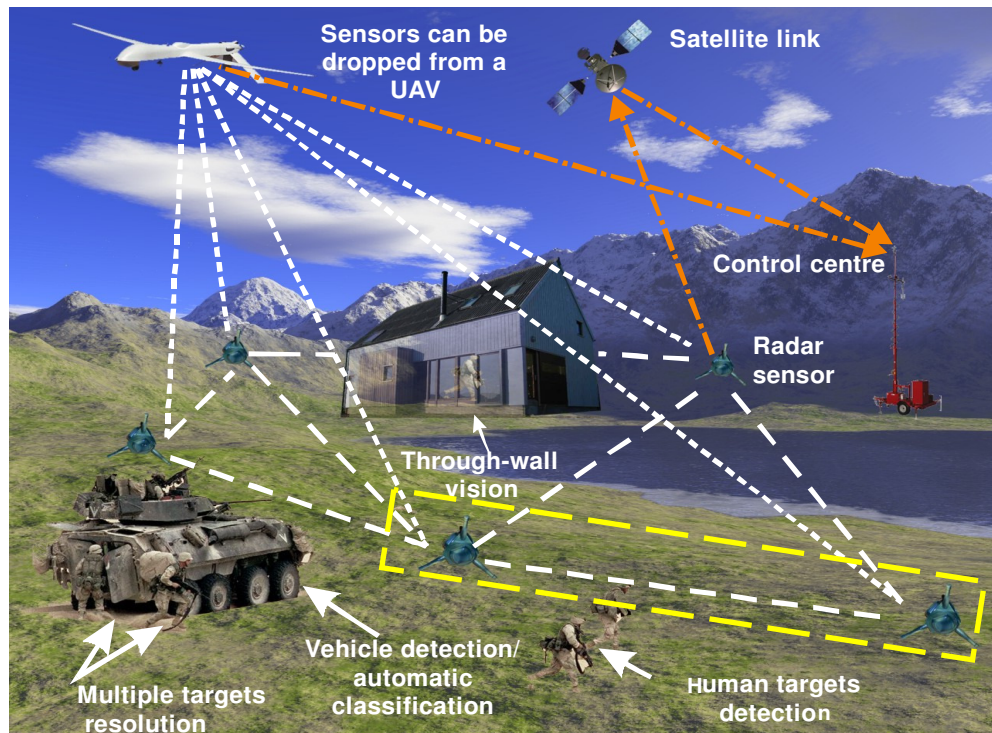


Figure 1.1 The concept of the FSR micro-sensors radar network

The system should be able to perform a special task: to detect and classify targets that enter the network coverage area.

It is assumed that the FS micro-sensors radar network can be used on any type of terrain, in any weather conditions and for detecting different targets for example vehicles and humans. The sensors are specially designed to be easily deployed from any moving platform such as an Unmanned Aerial Vehicle (UAV) or, if possible, a ground vehicle. Consequently, the sensors should be able to position themselves directly on the ground in random order and orientation in space. To operate in these conditions, as shown in [23-25], the sensors should utilise omnidirectional antennas and operate at relatively low frequencies, assumed to be around 100 MHz. This low frequency is derived as a compromise between the target RCS (most

importantly that of a human), the vegetation clutter level, low diffraction loss in a complex terrain and favourable propagation conditions (providing a reduction in ground reflection) and a practical antenna size [23].

Thus, in contrast to the research mentioned earlier, the conditions for ATC in our network are different:

- ✓ The system operates at the edge of the optical region, where the target dimensions are comparable with the operational wavelength;
- ✓ The antennas are omni-directional, positioned directly on the ground and hence the system operates in a strong multipath environment;
- ✓ Low target speed leads to the overlapping of clutter and the target's spectrum and hence clutter levels are observed in the signal;
- ✓ The speed and trajectory are unknown and subject to automatic estimation by the system;
- ✓ The baseline length can be up to 200m and the target can be in the vicinity of a transmitter or receiver antenna.

This new scenario introduces a vast number of problems for effective vehicle ATC in the FS sensors network.

The overall signal processing procedure is shown in Figure 1.2 in a simplified manner. This particular figure assumes that the signal processing is made at the command centre side but it can also be done by a processor in the node. When a target has crossed a transmitter-receiver baseline, the pre-processed target signature from the receiver is sent to a computer. The first step of the signal processing is Automatic Target Detection (ATD). Assuming that the target

is detected, the next step is a rough target speed and size automatic estimation. If the speed and the size of the target correspond to human target parameters, the processing comes to the end. Otherwise, if the target is vehicle-like, the signature of a vehicle target is used for the ATC.

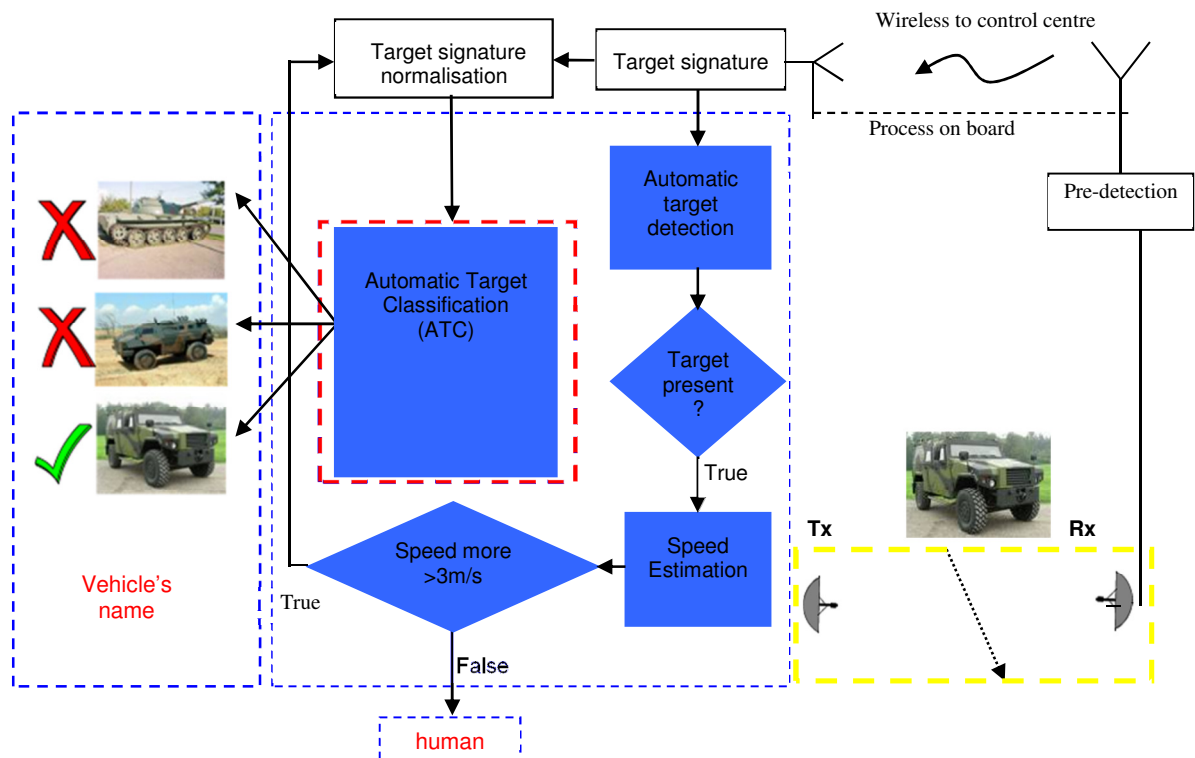


Figure 1.2 Forward Scatter Micro-Radar network block diagram

Once such a system is developed and optimised for detecting and classifying targets using freely dropped nodes, the following questions will be asked: what is the system performance like from an ATC perspective and how can we maximise this performance?

1.3 Problem Statement and Original Work

The aim of this thesis is to estimate and improve the performance of ATC operating in practical conditions for ground-based FS sensors².

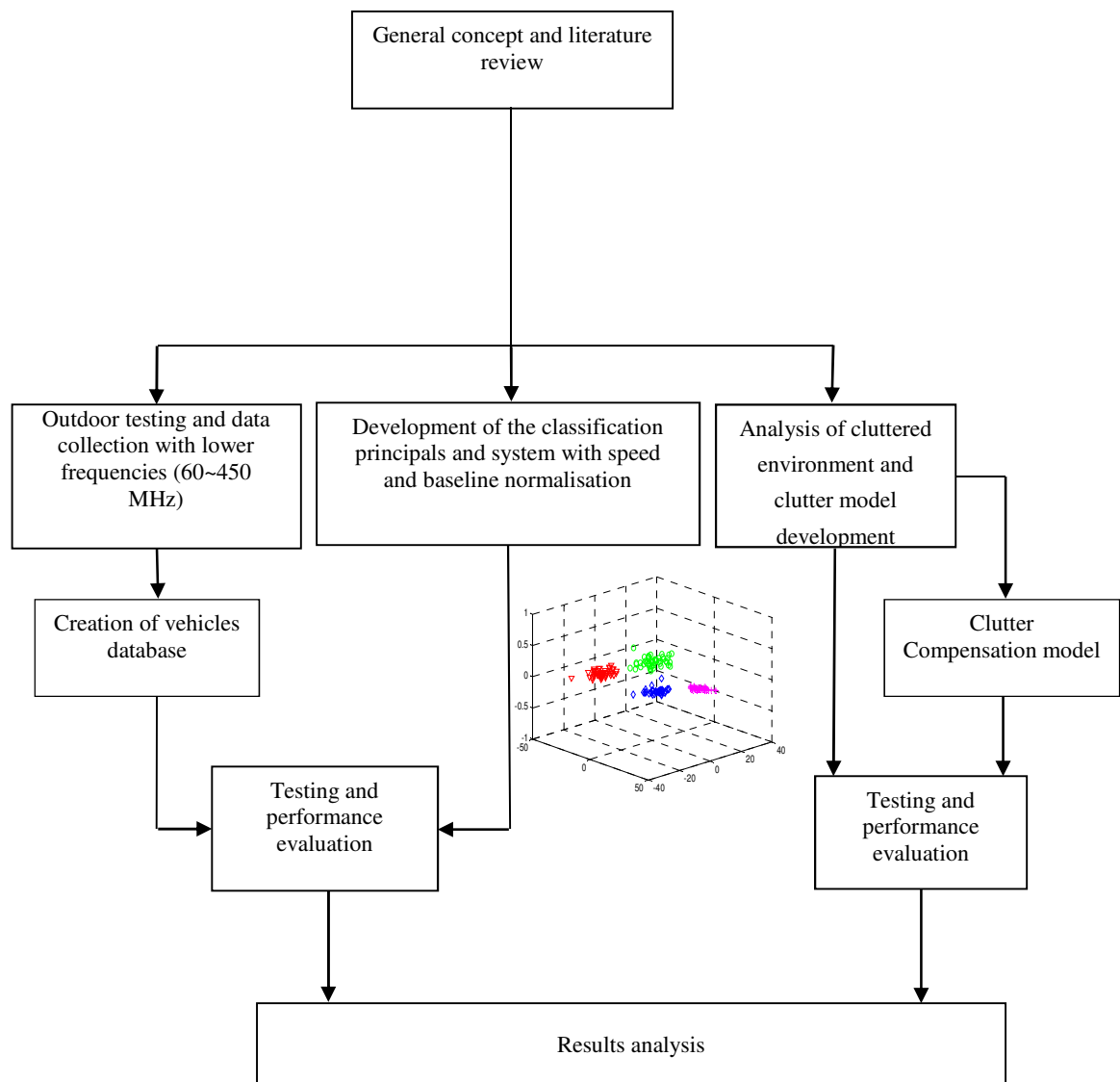
To achieve this aim we have conducted the following new studies:

- **Automatic Targets Classification using low frequencies (i.e. where the wavelength is comparable with the target dimensions).**
- **Automatic Target Speed estimation within the system.**
- **Investigation of speed estimation accuracy on ATC.**
- **Investigation of the influence of clutter from vegetation on ATC.**
- **New ATC algorithm development based on clutter compensation that improves ATC performance in a cluttered environment.**

A range of traditional scientific approaches have been used as part of this research: computer models have been developed on the basis of an analytical description; a wide range of experiments and experimental data collection has been completed for the model verification; following this, practical algorithms for ATC have been developed and tested on real data obtained during numerous experiments.

The research work flowchart is shown in Figure 1.3.

² The research has been done within the scope of EMRC DTC project 2/65 and is oriented on the parameters specified in this project. However, the fundamental and generic results from this study could be used in much wider applications.

**Figure 1.3 FSR target classification flowchart**

1.4 Thesis Organisation

The first part of Chapter 2 provides an overview of the fundamentals of radar technology, followed by relevant work published in the area of FSR, specifically how the signal (sometimes referred as ‘signature’) is obtained in FSR and its mathematical presentation in both the optical and the quasi-optical region. The second part of Chapter 2 reviews the method that has been used and the results obtained from previous research on ATC. Chapter 3 describes the data collection process. This includes hardware, data collection methods and experimental data descriptions. Furthermore, Chapter 3 contains the initial analyses on factors that affect the target signature, i.e. low frequencies, different baseline lengths and crossing angles. A brief description on foliage clutter and a detailed explanation on clutter modelling can be found in Chapter 4. In Chapter 5, an appropriate approach for estimating the speed of a target is presented and the results are analysed to see how errors can affect speed regardless of clutter presence. Speed and baseline normalisation methods are also introduced in this chapter in order minimize the effect of speed and baseline to target spectra. The analysis of ATC performance degradation in cluttered environments is continued in Chapter 6. Two potential sources of errors behind the degradation are described: error due to speed estimation and error due to amplitude shape deformation. Later, the classification performance improvement using clutter compensation model is presented at the end of the chapter. Finally, in Chapter 7, the work completed is summarised, important conclusions are drawn and future research is discussed.

For clarity, in the remainder of this thesis, equations are defined for continuous variables, as opposed to the discrete forms which are used in computation.

CHAPTER 2

TECHNOLOGY OVERVIEW

2.1 Introduction

The first part of this chapter provides the reader with a general introduction to radar technology and some background to Forward Scatter Radar (FSR). It emphasises the important areas that the reader needs to know and understand before reading the following chapters. In particular, the development of radar technology and FSR will be reviewed. This will include geometry, principles, basic equations such as FSR Radar Cross Sections (RCS) and the Doppler Effect in FSR. In the second part of this chapter, relevant research undertaken by other authors into Automatic Target Classification (ATC) will be reviewed. The classification system and classification algorithm are also described.

2.2 Radar Technology

The term RADAR is an acronym for Radio Detection and Ranging [26]. Radar is a system that uses electromagnetic waves to detect and locate a target of interest. Radar can identify the range, altitude, speed and direction of targets such as aircrafts, ships, vehicles, personnel, terrain and weather formations. When the radar transmitter transmits an electromagnetic wave, and the target (for example an aircraft) is present within the radar coverage area, the wave will be scattered back to the receiving antenna and the target can be detected. The information about the target can then be determined.

Radar configurations can be classified as monostatic, bistatic and multistatic. In monostatic radar the transmitter (TX) and receiver (RX) are collocated and often use the same antenna for transmitting and receiving a signal as shown in Figure 2.1(a). In contrast, the bistatic radar transmitter and receiver are separated by a distance comparable to the maximum range of a target. The angle between the transmitted R_T and reflected R_R rays is a bistatic angle, β (Figure 2.1(b)).

Monostatic and bistatic radars usually have narrow-beam TX and RX antennas to obtain a longer target detection range and better angle resolution. In a conventional bistatic radar configuration, when the bistatic angle significantly differs from 180° , the transmitted signal is not received by the RX antenna due to narrow-beam antenna patterns (Figure 2.1(b)). Radar only receives the signal scattered (reflected) by a target.

Bistatic radar has been on the radar scene for decades. More than 200 types of bistatic radar have become operational since the late 1930s [2].

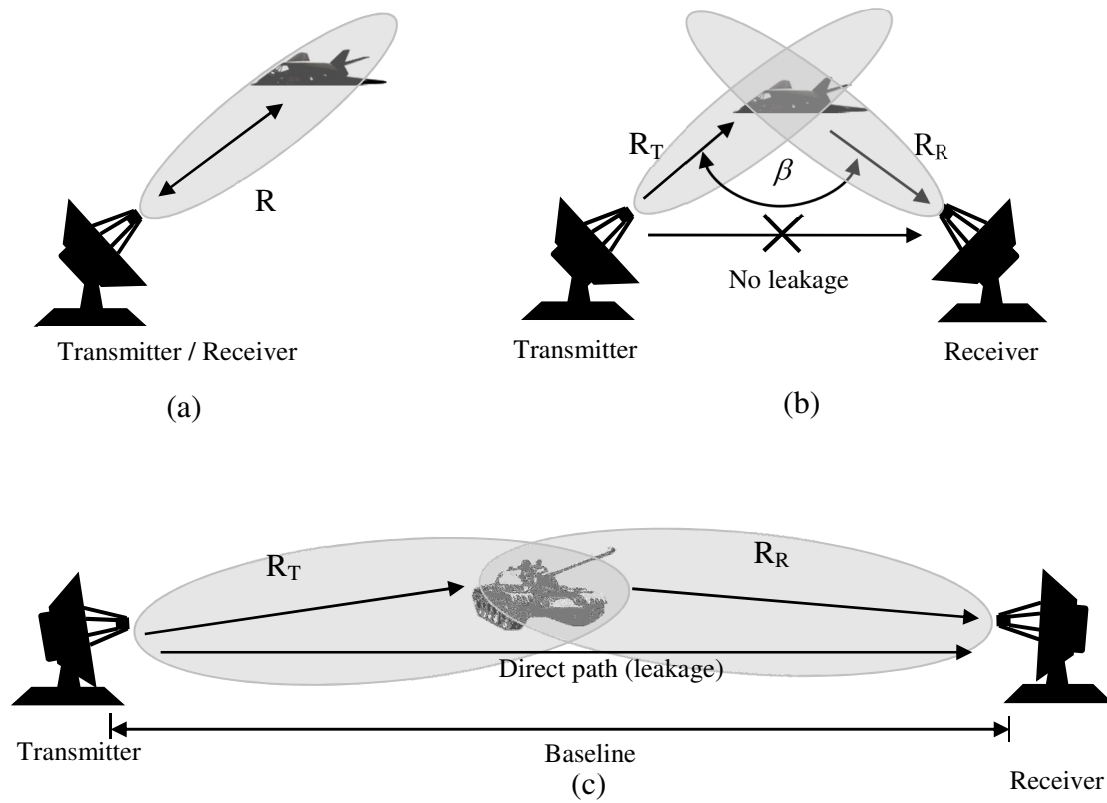


Figure 2.1 Radar systems

(a) Monostatic radar (b) Bistatic radar and (c) Forward Scatter radar

2.3 Forward Scatter Radar

Forward scatter radar (FSR) is a special type of bistatic radar. The main difference is in the operational region. FSR operates near the baseline (distance from TX to RX), where TX and RX antennas face each other, and a significant part of transmitted power (a leakage) is received by the RX antenna. A target situated on the baseline (or very close to it - see Figure

2.1(c)) blocks the direct path creating a shadow and introducing an important signal increase compared to monostatic and bistatic configurations, which improves radar sensitivity at the bistatic angle $\beta \approx 180^\circ$. It has been confirmed experimentally that the RCS for FS direction is greater than backscatter by 30-40 dB depending on the frequency band [7]. More accurate numbers can be specified only when relevant to a particular target and wavelength. This effect is called forward scattering [2, 4, 27] and the radar system which utilises it is known as Forward Scatter Radar.

FSR offers a number of advantages including:

- reasonably simple hardware, continuous wave (CW) radiation mode and the use of less radiated power;
- long coherent intervals of the received signal and a reduction of signal fluctuation compared to monostatic radar [28], which makes it capable of using inverse shadow synthetic aperture for automatic target recognition;
- FS RCS is practically independent from the radar absorbing material (RAM) coating [29] which means it can easily detect stealth targets.

As with any other system, FSR has some limitations. It does not have range resolution [12], but, if a target is moving in a uniform trajectory and is observed for a comparatively long time, its speed and direction can be estimated by special signal processing. The FSR system also has a limited operational area (estimated as $\beta = (180 \pm 10 \dots 20)^\circ$ at narrow bistatic angles).

2.4 The Forward Scatter Radar Equation

The FSR equation [24, 28] is the same as for any bistatic radar:

$$P_R = P_T G_T G_R \frac{4\pi\sigma}{\lambda^2} L_{T-tg} L_{tg-R}, \quad (2.1)$$

where P_R and P_T are received and transmitter powers, G_T and G_R are transmitter and receiver antenna gains in the direction of the target, respectively. λ is the wavelength and σ is the target radar cross section (RCS), and L_{T-tg} and L_{tg-R} are the transmitting and receiving propagation losses.

For free space propagation, the losses describe the degradation of the power flow density within the range:

$$L_{T-tg}^{FS} = \left(\frac{\lambda}{4\pi R_T} \right)^2 \text{ and } L_{tg-R}^{FS} = \left(\frac{\lambda}{4\pi R_R} \right)^2, \quad (2.2)$$

where R_T and R_R are the distances between the transmitter-to-target and target-to-receiver (Figure 2.1(c)).

Recent publications in this field have discussed forward scatter radar for ground vehicle detection and automatic classification [12, 13, 15, 23, 25, 30, 31]. With such radar, the transmitting and receiving antennas are placed directly on the ground and low frequencies (UHF, VHF) are used. The practical range of this radar is in the order of hundreds of metres.

In this case a more complex two-ray path (TRP) propagation model should be used [24], in which the reflected ground waves are multiplied with the ground reflection coefficient Γ . In the model of Perfect Electrically Conducting (PEC) ground, $\Gamma = -1$, and the TRP propagation losses are easily defined [32, 33]:

$$L_{T-tg}^{TRP} \approx \frac{h_T^2 h_{tg}^2}{R_T^4}, \text{ and } L_{tg-R}^{TRP} \approx \frac{h_{tg}^2 h_R^2}{R_R^4}. \quad (2.3)$$

They depend only on the distances to the target, the transmitting h_T and receiving h_R antenna heights, as well as target height h_{tg} , and not on the radar wavelength (frequency) as in the case of free space propagation. A PEC ground model is applicable for high radar frequencies (in GHz band), where the ground characteristics are very close to this model. It has been shown [24, 32, 33] that for lower frequencies in VHF and lower part of UHF bands (50-500MHz), the ground parameters are closer to a dielectric model with dielectric permittivity $\epsilon_g = 7...25$ and conductivity $\sigma_g = 0.005...0.05$, depending on different ground surface types. In this case, ground reflectivity becomes dependent on radar frequency. A more complex calculation for the propagation loss for dielectric ground properties shows great dependence of the losses from radar frequency: the higher the frequency, the greater the losses. This tendency has been experimentally confirmed [24]. The short-range ground-based FSR radar should use lower frequencies, for example for vertically polarised (VP) waves, TRP losses at 869 MHz may exceed the losses at 64 MHz on 20dB in the same conditions.

The choice of radar frequency depends on many other factors (antenna sizes, environmental noise, vegetation clutter, propagation in rough terrain and target reflectivity), but such an

investigation is outside of the goal of this thesis. In the following analysis we will use a number of different frequencies in VHF (64, 135, 144, 151 and 173 MHz) and lower UHF bands (434 and 869 MHz), on which the experimental tests were made, and compare the results for these different frequencies. The possibility (and quality) of automatic target classification at different frequencies is also an important factor in the choice of system carrier.

2.5 The Forward Scatter Radar Cross Section

In equation (2.1), the target RCS is the ratio showing how much of the incident power is reradiated (scattered) to the radar receiver [34] and may be evaluated as

$$\sigma = 4\pi \frac{P_s}{P_i}, \quad (2.4)$$

where P_s is the power per unit solid angle scattered in the radar receiver direction and P_i is the power per unit area in a plane wave incident on a target from the radar transmitter. The units for the RCS are in square metres.

Strictly speaking, a target does not reflect the incident wave in forward scattering. In Figure 2.2, a target (black body) is placed in the way of plane wave propagation. The body blocks the incident wave in its direction and the propagation fully or partially forms a shadow behind it.

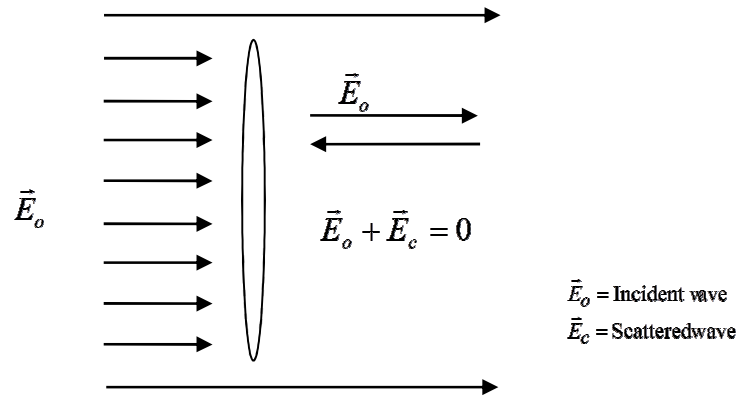


Figure 2.2 Shadowing field

2.5.1 Optical approach to FS RCS estimation

In FSR theory [2, 27, 28] a so called geometric optics (or physical optics) hypothesis is mostly used for the description of the scattering mechanism. This hypothesis can be applied if the following conditions are met:

- i. The dimension of the target is greater than the wavelength, i.e. the target should be electrically large. For example, for a sphere with a radius a , the incident wavelength λ must satisfy the equation $2\pi a / \lambda > 10$ [26]
- ii. Both the incident and scattered waves are considered as plane waves, so that the distances to the target from the transmitter and receiver sites are much longer than the size of the target.

Under these conditions we can neglect the wave diffraction around the target's edges, and suppose that the target fully blocks the incident electric field in this shadow region to zero. This can be mathematically modelled as there is a certain field reradiated by the target,

$\vec{E}_c = -\vec{E}_0$, which is equal in amplitude and opposite in phase to the incident wave field and compensates it. The target is presented as an equivalent plane antenna having the same shape as a target cross-section in the perpendicular direction to the incident wave – the target shadow contour or silhouette. The intensity and phase of the incident and reradiated fields are supposed to be the same for all antenna shapes.

In an optical case, the target FS RCS is:

$$\sigma_{fs} = \frac{4\pi S^2}{\lambda^2}, \quad (2.5)$$

where S is the target silhouette square, which corresponds to the planar antenna that derives power $P_i \cdot S$ from the primary field and reemits it with the planar antenna gain $G = \frac{4\pi S}{\lambda^2}$.

This maximal FS RCS value is achieved when the target is situated exactly on the line of sight (LOS). When the target is observed at some angles in elevation ϕ and azimuth θ (Fig. 2.3), the value of FS RSC decreases according to the equivalent antenna radiation pattern.

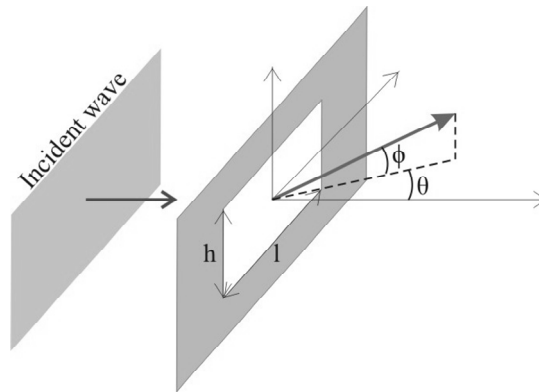


Figure 2.3 Plane wave incidence on the rectangular plate/aperture [35]

For the rectangular aperture presented in the figure the FS RCS is defined by [35]:

$$\sigma_{fs}(\theta, \phi) = \frac{4\pi(lh)^2}{\lambda^2} \left[\frac{\sin(\frac{\pi l}{\lambda} \sin \theta)}{\frac{\pi l}{\lambda} \sin \theta} \right]^2 \cdot \left[\frac{\sin(\frac{\pi h}{\lambda} \sin \phi)}{\frac{\pi h}{\lambda} \sin \phi} \right]^2 \quad (2.6)$$

In FSR, when the baseline distance (of hundreds of metres) which is greater than the target height h (up to a few metres), the elevation viewing angle close to 0, and the third multiplicand (in equation (2.6)) is very close to 1. When the target is moving on the ground surface, we can suppose that the target RCS pattern only varies in the azimuth plane:

$$\sigma_{fs}(\theta) = \frac{4\pi(lh)^2}{\lambda^2} \left[\frac{\sin(\frac{\pi l}{\lambda} \sin \theta)}{\frac{\pi l}{\lambda} \sin \theta} \right]^2 \quad (2.7)$$

We will use the optical approach to a rectangle target FS RCS prediction in the following analysis to simulate a target signature in different test conditions and use these simulated signatures for the testing of automatic target classification algorithms. Of course, this simulated signal does not contain any information about differences in shape for real vehicles, but it is modelled for an exact known speed and trajectory, and can be used for the estimation of ATC algorithm stability with regards to external noise and clutter, to real target speed estimation errors and some other factors.

2.5.2 FS RCS prediction for electrically small targets

In ground-based FSR, working in VHF and lower UHF bands with wavelengths in the order of metres, the targets of interest (personnel, animals, small cars and larger vehicles) may have a size which is smaller than or comparable to the wavelength. These targets cannot be considered as electrically large, and the optical scattering approach cannot be applied to their RCS estimation.

There are several solutions methods for calculating the forward scattered field. The universal method is given by Maxwell's equation [35]. This equation can be applied to almost any target shape or dimension with any material properties. For example, we can use this equation if the target dimension is compatible with or less than the wavelength and the diffraction wave over the target is not negligible.

Unfortunately, Maxwell's equation does not have a common analytical solution for all types of target shape. It is only suitable for a class of simple objects with central or axial symmetry in their body geometry, for example a sphere, a cylinder, an ellipsoid, a flat symmetrical plate (rectangle or ellipse) and a cone [35, 36]. There are various well established computer programmes that can be applied to make a numerical simulation of Maxwell's equation (a so-called electromagnetic simulation) and to predict the RCS for more complex targets (with certain limitations with regard to the environment, the material used etc). The accuracy of the simulation is limited by the power and memory volume of the computer and also the calculation time.

In Figure 2.4 the comparison between the analytic solution and numeric calculations by Computer Simulation Technology (CST) is shown for a standard target: a PEC sphere with radius R . In the figure, both the analytic and simulated RCS values are normalised to the sphere projected area (its optical back scatter RCS value, $\sigma_0 = \pi R^2$) and given in dB as a function of the ratio of the sphere's circumference to the wavelength $p = 2\pi R/\lambda$.

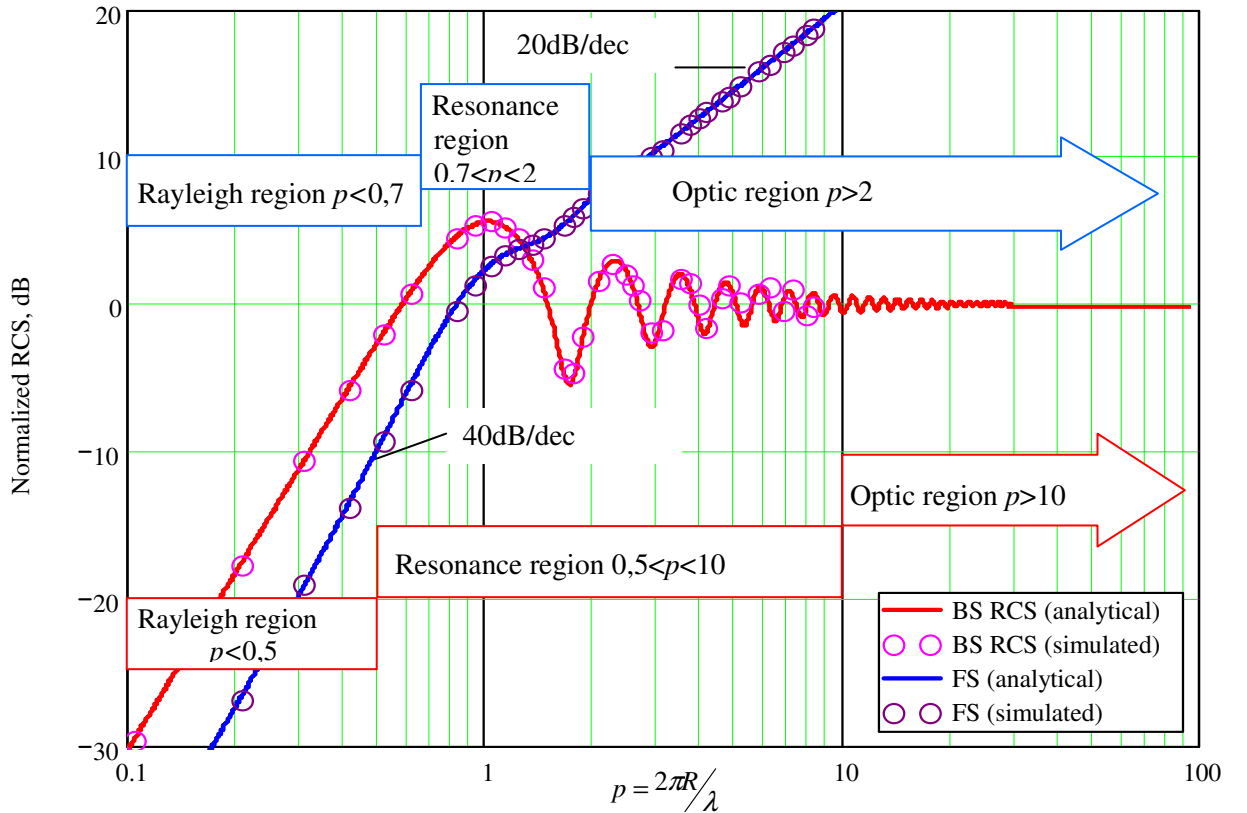


Figure 2.4 RCS of a sphere [26]

In the monostatic (back scatter) case, the reflection from the sphere is well known [4]; its analytical solution is shown in the figure as a solid red line. The dots show the results of the numerical simulation of Maxwell equations by CST which coincides well with the analytical solution. There are three regions (in red for the back scattering) of diffraction pointed out here – optic, resonance and Rayleigh regions. In the optic region ($p > 10$) the RCS is $\sigma = \pi R^2$ and

does not depend on the wavelength. In this case the normalised value is $\sigma_{-dB}(p) = 0$ dB (see Figure 2.4). In the resonance region (or Mie region, where $0.5 < p < 10$) the RCS oscillates near this value with a maximum amplitude of about 11dB.

Finally, in the Rayleigh region at $p < 0.5$ the RCS dramatically falls down in p^4 or 40dB/decade proportion.

In the case of forward scatter, the analytical solution is given in [17] (shown in the figure by a solid blue line), and the results of the electromagnetic simulation (purple dots) are presented in [24]. For forward scattering, all three regions are also presented (marked in blue in the figure), but the resonance region is narrower ($0.7 < p < 2$) since FS RCS has practically no resonance oscillations and it is in good accordance with the optical presentation for shorter wavelengths ($p > 1$). For high frequencies, partly in the resonance region and in the optical region ($p > 1.3$) the FS RCS exceeds the BS RCS.

In the Rayleigh region the FS RCS has the same slope of 40dB/decade, but its value is about 8dB less than the BS RCS. This is caused by the wave diffraction around the small sphere at low frequencies. Diffracted waves partially illuminate the shadow region behind the target and thus decrease the FS RCS. This fact suggests a difference to the commonly held view that the FS RCS is greater than the BS RCS. This is the case only for short wavelengths. If the wavelength is longer than the target's characteristic dimension (a sphere circumference in our example), the FS RCS is smaller than the BS RCS.

We can see that the simulated RCS values fully coincide with the analytical results in all diffraction regions, both for back scatter and forward scatter directions.

For vehicle recognition we will estimate different vehicle lengths from 3m (a very small car) to 5m (a large car) and even up to 8-12m for buses and trucks. In our experiments we have used wavelength bands from 4.7 to 0.7m corresponding to frequency bands from 64 to 434MHz. Thus, for higher frequencies the scattering from all possible vehicles may be considered as being in the optical region, and for the lowest frequency as in the resonance region.

An example of FS RCS calculation was presented in [17]. In Figure 2.5, three car models of a similar size (length 4m, width 2m and height 1.6m) are shown. The cars have different shapes.

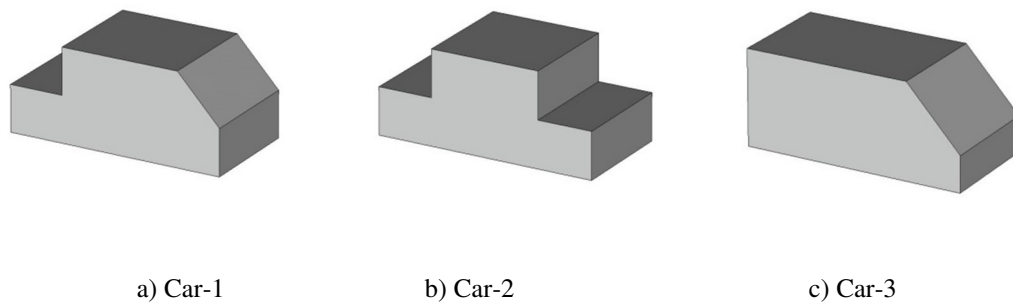


Figure 2.5 Different car models

A FS RCS of these cars was simulated by CST Studio for a radar frequency of 434MHz and is shown in Figure 2.6. We can see from the figure that the main lobe width of the RCS pattern is practically the same for all cars and it is defined by the same car length. The side-lobes of the RCS pattern are very different for each model giving the possibility of target recognition, if the difference in side-lobe could be extracted from the received target signature.

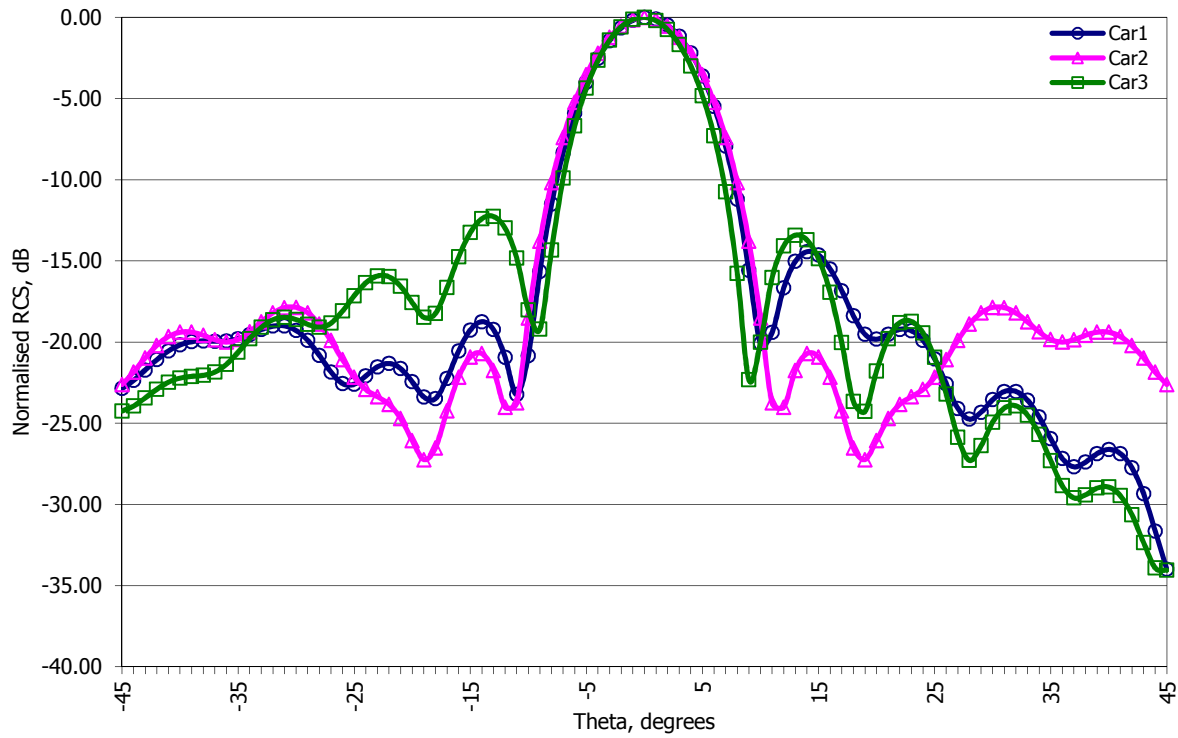


Figure 2.6 FS RCS of different car models

2.6 Doppler Effect

The Doppler Effect is the change in frequency of a wave for an observer moving relative to the source of the wave. It is commonly heard when a vehicle sounding a siren or horn approaches, passes, and recedes from an observer. The received frequency is higher (compared to the emitted frequency) during the approach, it is identical at the instant of passing by, and it is lower during the recession [37] (Figure 2.7).

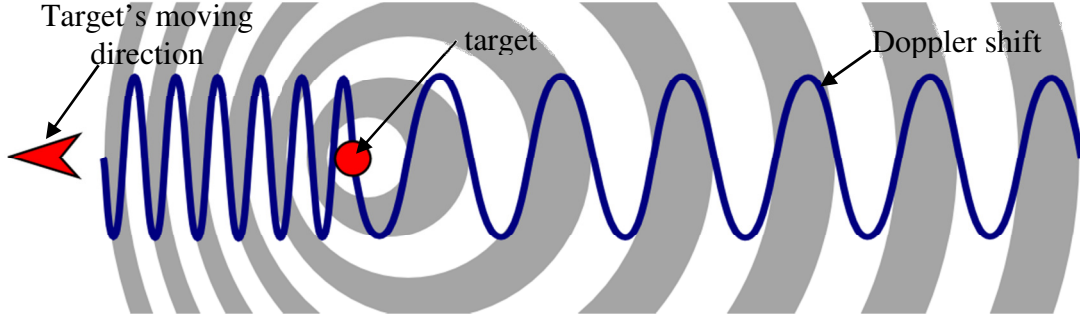


Figure 2.7 Doppler shift for a moving source [37]

In radar technology, the Doppler Effect is a frequency shift of a wave reflected and received from a moving target in comparison to a radiated wave frequency. The Doppler shift by the common definition is the time rate of change of the total path length of the scattered signal, normalised by the wavelength λ [2]:

$$f_d(t) = \frac{1}{\lambda} \cdot \frac{d}{dt} [R_T(t) + R_R(t)] = \frac{1}{\lambda} \left[\frac{dR_T(t)}{dt} + \frac{dR_R(t)}{dt} \right], \quad (2.8)$$

For monostatic radar (Figure 2.8) the Doppler shift is defined as:

$$f_d = \frac{2V_R}{\lambda} = f_o \frac{2V \cdot \cos \alpha}{c}, \quad (2.9)$$

where V is the target speed, $V_R = V \cos \alpha$ is the radial target speed, λ is the radar wavelength, f_o is the radar radiated frequency and c is the speed of light [28].

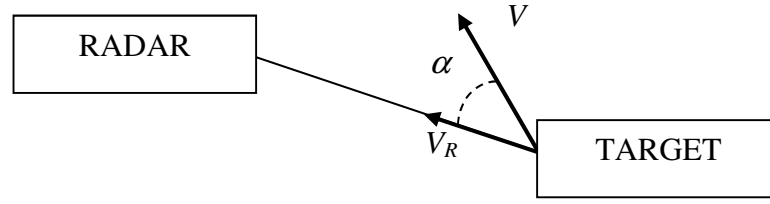


Figure 2.8 Monostatic radar geometry for moving target [12]

In bistatic and FS radar configurations the Doppler Effect is also present. Figure 2.9 shows the geometry of a bistatic system used to calculate the Doppler shift [2]. In this configuration, the transmitter and the receiver are stationary and the target is crossing the transmitter-receiver baseline with a velocity vector, \bar{v} .

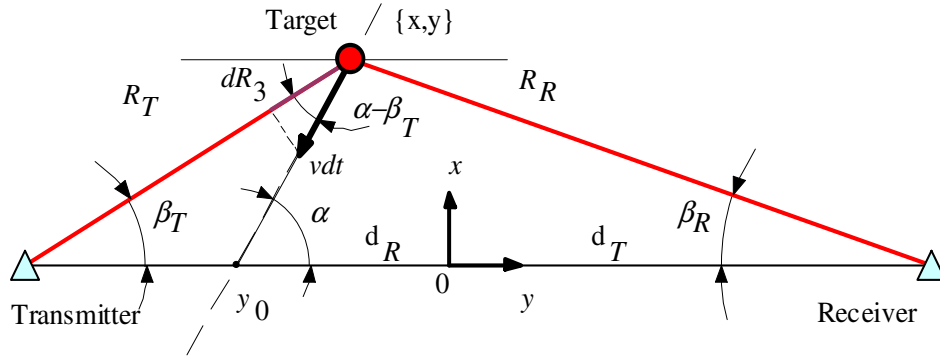


Figure 2.9 The definition of the Doppler shift in bistatic radar

The known equation for the Doppler shift in bistatic radar [2] follows from the figure:

$$f_d(t) = \frac{2v}{\lambda} \sin \left[\alpha - \frac{\beta_T(t) - \beta_R(t)}{2} \right] \cdot \sin \left[\frac{\beta_T(t) + \beta_R(t)}{2} \right]. \quad (2.10)$$

The zero Doppler shift $f_d(0) = 0$ occurs at zero time $t = 0$, when $\beta_T(0) = \beta_R(0) = 0$. The maximal value of the Doppler shift (far away from the FSR position) is $f_{d\max} = \frac{2v}{\lambda}$ which is the same as for the monostatic system.

Equation (2.10) is true for a bistatic radar where the target is considered to be an independent source of radiation due to the absence of direct leakage between the transmitter and the receiver. The creation of a target signature in FSR on the background of the leakage is peculiar and can be described as follows.

2.7 Target Signature in FSR

The Doppler signature creation mechanism in FSR is described in [24] for a point target, i.e. a small size target, the dimensions of which are smaller than the FSR geometry dimensions. The target still has some definite RCS value independent from the viewing angles (an isotropic RCS pattern).

A simplified analysis of a target Doppler signature has been done in [38]. Let us consider the FSR geometry presented in Figure 2.10 (a), where the point target is moving. Let us also suppose, for the sake of simplicity, that this model is two-dimensional (2-D). The target height is equal to 0, as well as both antenna heights, and the wave propagation conditions correspond to free space. The received Doppler signature is a result of the interference between the constant direct leakage signal u_L and the target signal $u_{tg}(t)$ which changes in time while a target is moving:

$$u_R(t) = u_L + u_{tg}(t). \quad (2.11)$$

Usually the leakage signal is about 20-40dB greater than the target signal [24] and the equation can be rewritten as:

$$u_R(t) = u_L \left[1 + \frac{u_{tg}(t)}{u_L} \right] = U_L e^{j(\omega_0 t + \varphi_L)} \left\{ 1 + \frac{U_{tg}(t) e^{j[\varphi_{tg}(t) - \varphi_L]}}{U_L} \right\}, \quad (2.12)$$

where $U_{tg}(t)$, U_L are magnitudes and $\varphi_{tg}(t)$ and φ_L are phases of target and leakage signals, respectively, and ω_0 is radar carrier frequency.

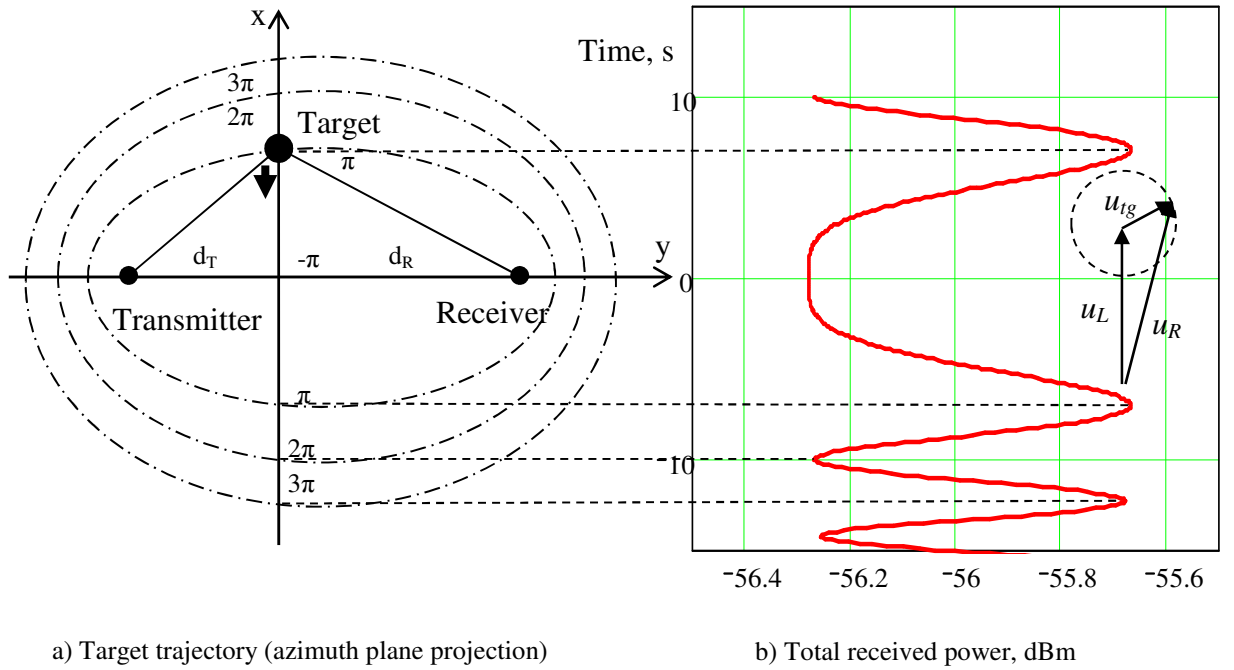


Figure 2.10 Doppler Effect for a moving target in FSR

We can see from equation (2.12) that the target signal creates a small modulation of the larger leakage signal.

When the target is set exactly on the baseline at the same height as the antennas, the path phase difference between the leakage signal and the target signal is equal to 0, but the target signal's phase is opposite to that of the leakage signal. The total signal amplitude is at its minimum at this point, $U_R = U_L - U_{tg}$.

As the target continues to move, the path phase difference increases, reaching π . At this moment, the target's signal is in phase with the leakage signal, and the total signal amplitude reaches its maximum of $U_R = U_L + U_{tg}$. As time progresses, the path phase difference is further increased, and reaches 2π at another point where the total signal becomes minimal again, and so on. This is illustrated in Figure 2.10(b). The points of equal path phase lie on an ellipsoid surface (the projections of these surfaces in the azimuth plane are shown in the figure as ellipses). As the total signal is the vector sum of the leakage signal and the target signal (Figure 2.10(b)), it has both amplitude and a phase modulation when the target is moving.

This modulation can be extracted in the receiver, even if it has a very small ratio. The receiver hardware must have a phase or amplitude detector for this operation.

Assuming unity transmitted power $P_T = 1W$ and isotropic antennas ($G_T = G_R = 1$) in the radar equation, the leakage and target signal magnitudes can be obtained as

$$U_L = \sqrt{L_{T-R}}, \quad (2.13)$$

$$U_{tg} = \sqrt{\frac{\sigma}{4\pi\mathcal{K}}} L_{T-tg} L_{tg-R} , \quad (2.14)$$

where transmitter-to-receiver loss L_{T-R} , transmitter-to-target L_{T-tg} and target-to-receiver L_{tg-R} loss can be calculated for free space and TRP models (depending on the conditions) by the use of equations (2.2) and (2.3) accordingly.

For a moving target, its signal amplitude is obviously the function of the target position, or time:

$$U_{tg}(t) = \sqrt{\frac{\sigma}{4\pi\mathcal{K}}} L_{T-tg}(t) L_{tg-R}(t) . \quad (2.15)$$

Let us also suppose for simplicity that this model is two-dimensional (2-D) and the wave propagation conditions correspond to free space. In these conditions, the equations (2.13) and (2.14) may be simplified:

$$U_L = \lambda/4\pi D , \quad (2.16)$$

where D is baseline distance, and

$$U_{tg}(t) = \frac{\lambda}{(4\pi)^2 R_1(t) R_2(t)} \sqrt{\frac{\sigma}{4\pi}} , \quad (2.17)$$

where both the TX-to-target (R_1) and target-to-RX (R_2) path lengths are functions of time as the target is moving. Therefore, the amplitude of the target signal depends on time.

The phases of the target and leakage rays can easily be found from the system geometry (Figure 2.10):

$$\varphi_L = 2\pi D/\lambda, \text{ and} \quad (2.18)$$

$$\varphi_{tg}(t) = 2\pi [R_1(t) + R_2(t)]/\lambda \quad (2.19)$$

The phase of the target signal also changes with time.

After quadratic amplitude detection and Doppler filtering in the receiver we can see the measured target signature as:

$$s_R(t) \approx [u_R(t)]^2 = U_L^2 \left\{ 1 + \frac{U_{tg}(t) e^{j[\varphi_{tg}(t) - \varphi_L]}}{U_L} \right\}^2 \approx U_L^2 + 2U_L U_{tg}(t) e^{j[\varphi_{tg}(t) - \varphi_L]}, \quad (2.20)$$

i.e. the received signature has a constant value which is proportional to the leakage power and a small additional time-varying Doppler component which is proportional to a moving target signal. Finally, considering only the Doppler component of the signature, we can estimate that

$$s_{tg}(t) \approx K U_{tg}(t) e^{j[\varphi_{tg}(t) - \varphi_L]} = K \sqrt{\frac{\sigma}{4\pi\lambda^2}} \cdot \sqrt{L_{T-tg}(t) L_{tg-R}(t)} \cdot e^{j[\varphi_{tg}(t) - \varphi_L]}, \quad (2.21)$$

where K is a scaling factor, which depends on the exact value of transmitter power, transmitter and receiver antenna gains, receiver gain and detection characteristics and some other factors.

The calculated signature for a point target crossing a baseline with a length of $D = 100m$ at midpoint is presented in Figure 2.10(b). The FSR carrier is supposed to be 64MHz; the target has a RCS of $\sigma = 10dBsm$ and moves with a constant velocity of $v = 1m/s$ in this example. In the received signature, the Doppler component oscillates slightly around a median value of -56dBm, which is the constant leakage power. The maximums and minimums in the Doppler signature correspond to the $n\pi$ -shifted phase of the target signal.

The RCS for a target where the shape size is extended does not have to be omni-directional. It depends also on the viewing angles, which have changed in time according to the target moving. In this case we have to take into account in equation (2.21) the dependence of viewing RCS on time for a moving target:

$$s_{tg}(t) \approx K \sqrt{\frac{\sigma(t)}{4\pi\lambda^2}} \cdot \sqrt{L_{T-tg}(t)L_{tg-R}(t)} \cdot e^{j[\varphi_{tg}(t) - \varphi_L]}, \quad (2.22)$$

For example, the RCS for a rectangular shape is given by equation (2.6) or (2.7). The viewing angles are dependent on the position of the target (i.e. on time) if the target moves. All these parameters can be calculated from FSR geometry for any baseline distance, antenna height, target dimension and trajectory [24].

In Figure 2.11, an example of target signature calculation is presented for two types of targets: a point and a flat rectangle target. The FSR has a baseline distance of $D = 100m$, a carrier frequency of 64 MHz and the transmitter and receiver antenna heights are $h_T = h_R = 0.3m$. The target crosses the baseline in the middle at perpendicular direction. The target speed is 10m/s. The RCS of the target is $\sigma = 16dBsm$, which correspond to the value of a large car,

similar to that of a Land Rover Discovery. The flat-shaped target has a height of $h = 2$ m and a width of $l = 4.5$ m. The point target has a height of 1 m, according to the position of the phase centre of the flat-shaped target. The transmitter power is 1 mW (0 dBm), and the antennas have a unity gain of $G_T = G_R = 1$. The propagation loss is calculated³ by using equations presented in [24] for a dielectric ground model with average surface parameters of $\epsilon_g = 15$, $\sigma_g = 0.005$.

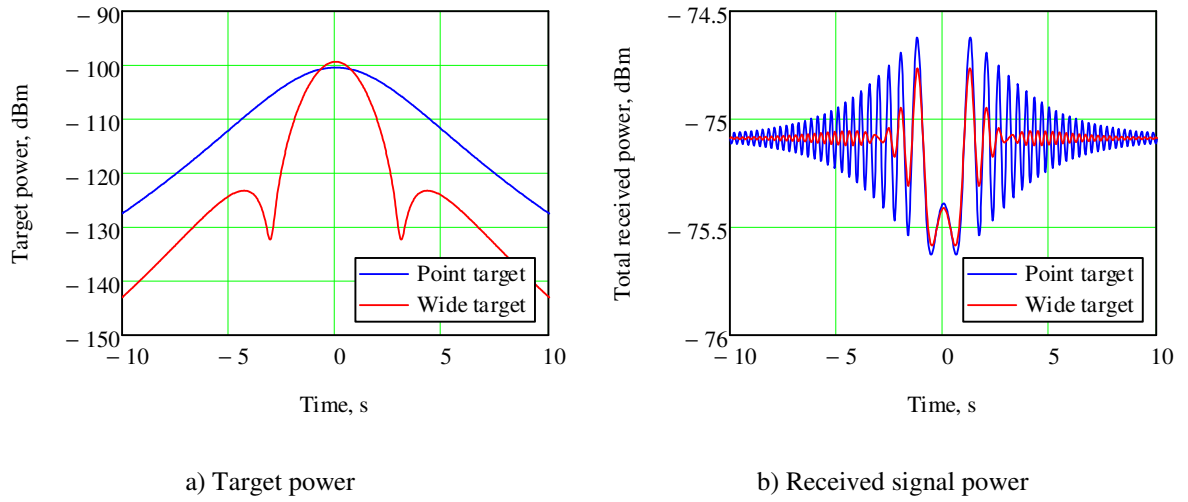


Figure 2.11 Target signature comparisons for point and flat-shaped targets

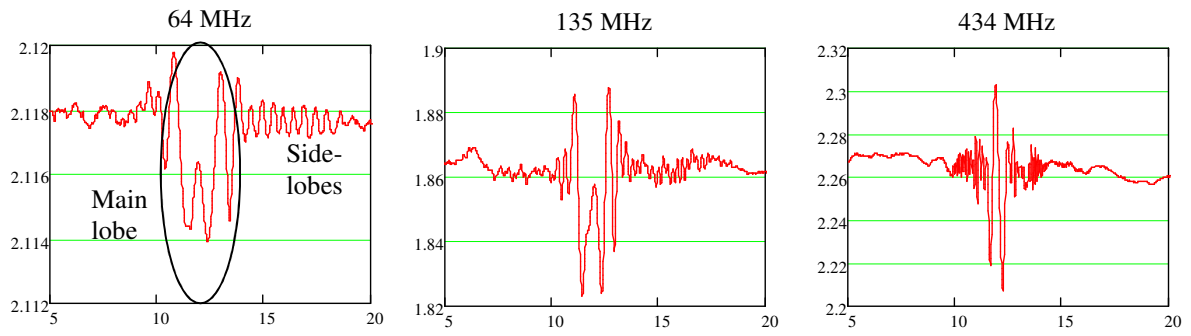
We can see from the figure that maximal target power is observed at the baseline, which then decreases as the target is moving. For the point target with an isotropic RCS pattern, the received power (Figure 2.11(a)) has only changed by the increase of propagation loss due to the increase in transmitter-to-target and target-to-receiver ranges, in accordance to equation (2.21). For the flat target there is extra modulation by form of the target RCS pattern, according to equation (2.22).

³ A programme in MathCAD, developed by Dr. V. Sizov at the University of Birmingham, was used for the calculations.

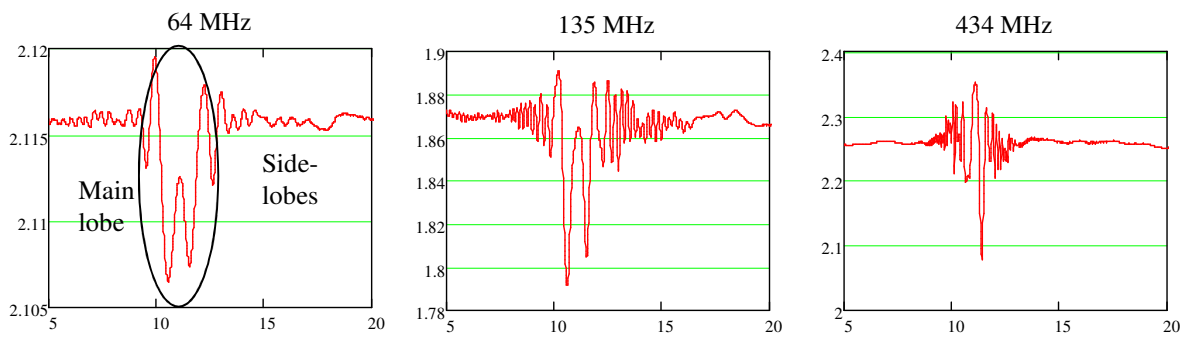
In the full received target signatures (Figure 2.11(b)) we can see the same Doppler shift for both targets, but it has additional phase modulation due to the flat shape RCS pattern. In the main lobe, the phase is the same for both targets. In the first side-lobe the phase of the flat-shaped target is opposite to the point target phase.

The extended size target received signature therefore has some information on the target shape, and once it is extracted it can be used for target recognition.

Real targets have more complex shapes than a simple rectangular plate. Other factors, such as resonance diffraction, can also make the received signature more complex. Figure 2.12 shows an example of measured time domain signatures for two cars with different sizes (the smaller one is a Peugeot 206 and the larger one is a Land Rover) at different carrier frequencies. The baseline distance for this test was 50m; the cars drove across the middle of the baseline in a perpendicular direction with a speed of approximately 20miles/hour. In the figure, the vertical axis is the voltage output of the receiver, and the horizontal axis is the observation time in seconds.



a) Small car (Peugeot 206)



b) Big car (Land Rover)

Figure 2.12 Measured time domain signatures for different cars

We can see a similarity in the main lobe part of the Doppler signature envelope (which is defined mostly by the length of the car), but the side-lobes are different. The Doppler waveform will be the input to the classification system. In the forthcoming section, we will describe how information can be extracted from the received Doppler signature.

2.8 Literature Review on Forward Scatter Ground Target Classification

Earlier research in FSR has mainly concentrated on air target detection, the estimation of their coordinates and automatic target classification [8, 19, 20, 39]. Recently, a set of publications on forward scatter for ground target vehicle detection and especially automatic classification has been discussed in [1, 30, 31, 40-42].

Previous research on FSR for ground targets was operated using a CW transmitting waveform with around 900 MHz carrier frequencies for a short baseline of about 17m. In this type of radar, the directional antennas are placed at about 1m above the ground. It was demonstrated [1], and numerous experimental confirmations show that at this frequency the system can be reliably used for detection and classification of a ground target.

The concept of a micro-sensor wireless network for situation awareness has recently been presented [23]. The sensors are installed directly on the ground, where the utilisation of VHF band is much more effective due to a smaller propagation loss. It was experimentally proven [24] that due to real ground properties, the two-ray path propagation loss is heavily dependent on the frequency. For example, using a 70 MHz carrier frequency instead of 900 MHz, may give an advantage of about 30dB in the target signal's received power [24]. It is also well-known that a longer wave with a lower frequency diffracts around ground obstacles and penetrates foliage more effectively. On the other hand, at high frequencies, it is practically impossible to build directional antennas of a reasonably small size and only omni-directional antennas can be used (even they should be designed to be small in size).

Obviously, the radar carrier frequency cannot be too low (implying a long wavelength) or the target RCS decreases significantly. According to [43, 44], a human target (as the smallest of ground targets of interest for situational awareness) has a resonance scatter in the FSR configuration for waves with vertical polarisation and carrier frequencies in the Very High Frequency (VHF) band (around 70-120 MHz), giving an additional advantage of approximately 10 dB in the FSR power budget. Furthermore, an investigation into the nature of foliage clutter [45] shows a significant dependence of the received clutter power on the operational frequency f , approximately proportional to f^4 . On the other hand, we might need to decrease the radar wavelength in ATC because we can sense the differences between targets better if the wavelength is smaller than the characteristic dimensions of the target's shape. Previous experiments on car recognition were done on carrier frequencies of around 870 MHz [1]. These experiments showed that FSR can be effectively used for ATC and targets can be classified accurately.

The knowledge of speed estimation is essential to the ATC system. The width of the target's spectrum main lobe is dependent on it. However, speed information is in general not available from the radar signal itself. It is therefore necessary to develop an algorithm to estimate the speed. A number of methods were considered to estimate the speed of a vehicle.

In [1], the speed of a vehicle was estimated based on the relationship between the time domain signal and the speed of the vehicle from the capture video footage. Two markers/posts were placed within the scene of the experimentation as shown in Figure 2.13 [1]. In order to establish this relationship, the speed was associated with a number of zero crossings present between the start and the end of the target signature, and then normalised over the observation period.

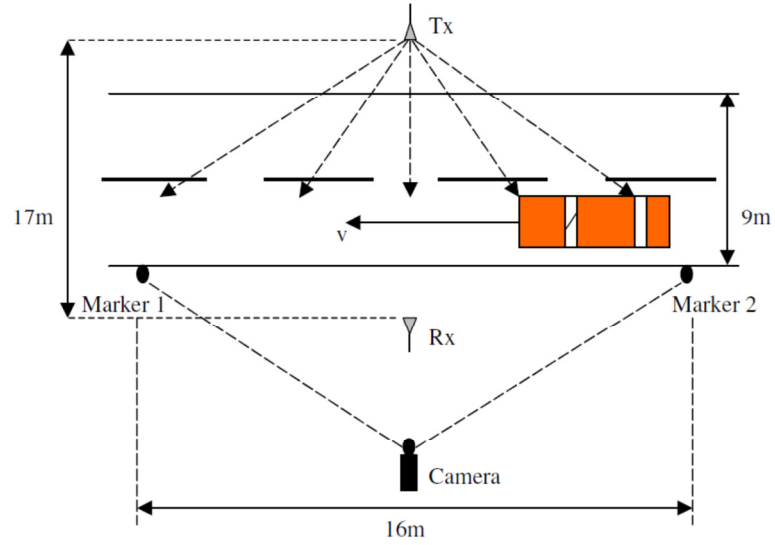


Figure 2.13 Experimentation layout (signals recorded using video camera)[1]

Later, the above method was improved by finding the relationship between the standard deviation, σ , of the time domain signal and the speed from the video camera. The system is divided into two phases: training and testing. In the training phase, the standard deviation of each time domain signal is calculated. In order to find the relationship between the standard deviation value and the speed of the target estimated from the video camera, the value of the standard deviation versus the speed is plotted (for example in Figure 2.14 [46]). A line of ‘best fit’ is determined in order to estimate an unknown vehicle speeds and this is given by:

$$\text{Vehicle Speed} = 20 \times \sigma - 1.5 \quad (2.23)$$

In the testing phase, the standard deviation of the unknown signal is also calculated. This value is then substituted in equation (2.23) and the speed of the unknown target can be estimated.

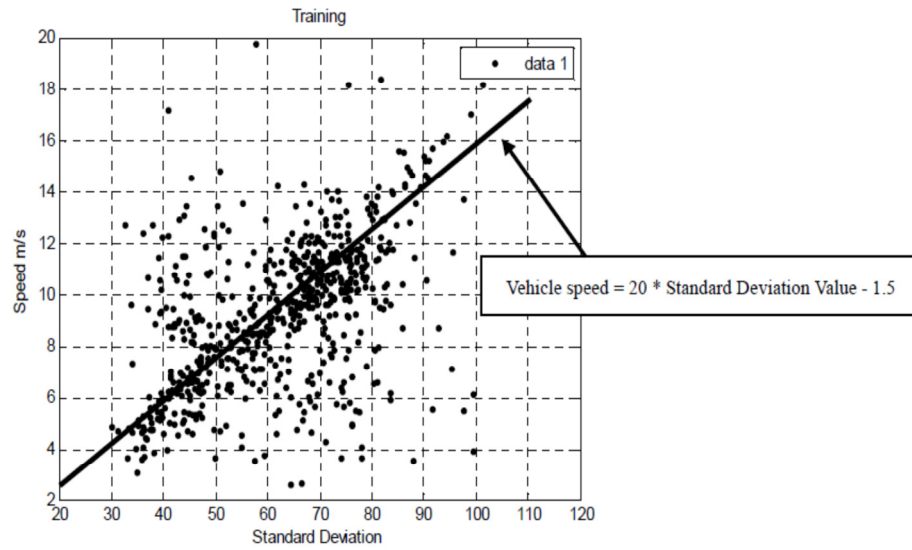


Figure 2.14 Standard deviation vs. speed of the vehicle from video camera[46]

In [44], the speed is estimated using a set of reference functions. This set of reference functions can be designed by using different combinations of parameters: speed and target trajectories. Each reference function is then correlated with a target signal and the maximum correlation between both waveforms, indicating the target's estimated velocity. Further details of the speed estimation procedure will be described in Chapter 5.

Various techniques can be used for modelling in pattern recognition, for example: Neural Networks and the K-Nearest Neighbour classifier [47]. In the earliest work done in [1], the system used the K-Nearest Neighbour (KNN) classifier. KNN is a classifier that is based on calculating the distance between the vectors of test and training data [48]. Examples of where KNN has been used for radar application can be found in [1, 30, 31, 40, 41, 49, 50]. The test data is classified to a class which is the most common amongst its k nearest neighbours. The classification is considered a success if the assigned group and the actual group of the test data are match.

In [41], it was shown that the K-Nearest Neighbour classifier with $k=3$ gave a very good performance even though there was a limited amount of data for both testing and training.

Subsequently on, the improvement of the classification performance using Neural Networks was discussed in [42]. This method imitates the concept of the human brain. In [51], the input of the network is the features from the vehicle signature which are extracted based on signal theoretical analysis and the common characteristics inherent in the signatures. Three types of features extracted from the spectra have been used as the input of the neurons: first main lobe width, second main lobe width and number of lobes. The authors implemented a Multi-layer Perceptron (MLP) back propagation neural network. The network consists of three layers:

- 1) The first layer accepts the defined features and distributes them to all neurons in the second layer.
- 2) The second layer is known as the hidden layer. The number of neurons in this layer is associated with the weights of the neurons in the input patterns.
- 3) In the third layer the number of neurons depends on the number of classes in the classification process.

The activation function used for the second and the third layer is a log-sigmoid transfer function. The presented results suggest that neural networks can be effectively used in FSR as an automatic classifier.

2.9 Pattern Classification

The time domain target received signal has hidden information that characterises this signal and hence needs to be extracted.

Pattern classification/recognition is a process where we take raw data, examine a pattern and take an action based on the “category” of the pattern [47]. Figure 2.15 shows a general scheme of a pattern recognition system.

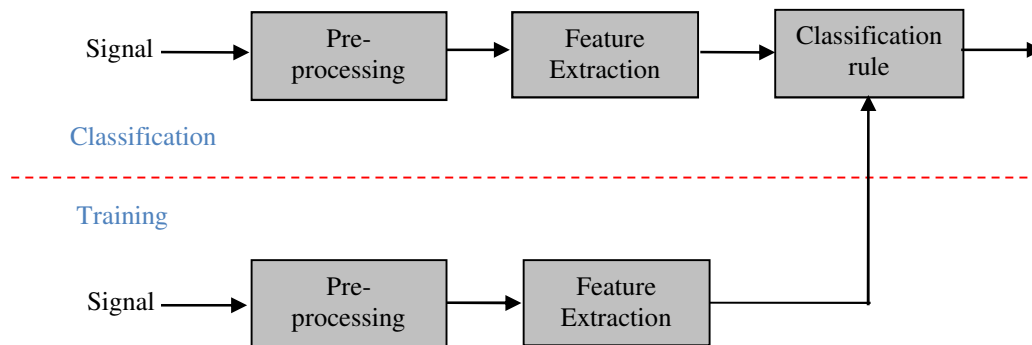


Figure 2.15 General scheme of a pattern recognition system

Signals from the sensor depend on the characteristics and limitation of the sensor used, such as frequency, target resolution and signal-to-noise ratio. A signal needs to be segmented from the background by removing clutter and normalising it prior to any further processing. The pre-processing output becomes the input of the ‘Feature Extraction’ block. Typically, we need to extract features which capture the information in the signal relevant for the given classification task as the input to the classification process. For example, if someone is interested in classifying a vehicle that passes through a security border, the length and height of the vehicle could be used as features.

The output of the ‘feature extraction’ is a feature vector, denoted by $x = [x(1), x(2), \dots, x(N)]$.

The feature extractor will reduce each vehicle to a feature vector x in a N-dimensional feature space, x , where:

$$x = \begin{bmatrix} x(1) \\ \vdots \\ x(N) \end{bmatrix} \quad (2.24)$$

The information about the target needs to be chosen appropriately so that the similarity of objects in each class results in the closeness of their features in the feature space. By using the selected features, we can partition the objects into different classes as in Figure 2.16. However, perfect separable features are often impossible. This is due to certain factors such as existence of noise in the signal and the choice of features for classification.

The recognition system is operated in two phases: training and testing (classifying) (see Figure 2.15). In the training mode, the appropriate features are extracted to represent the input patterns. For each type of target, a model describing the distribution of the features in the feature space is obtained. In the testing mode, the input pattern is assigned to one of the classes by a classifier, based on its measured features. In Figure 2.16, for example, a new signal is received, and it falls within region A. This new signal is automatically classified as a ‘triangle’ (Δ). This task is known as classification. The performance of the system can be statistically analysed by finding how many testing data are correctly classified. Details of the classification method used in our research will be described in section 2.10.2.

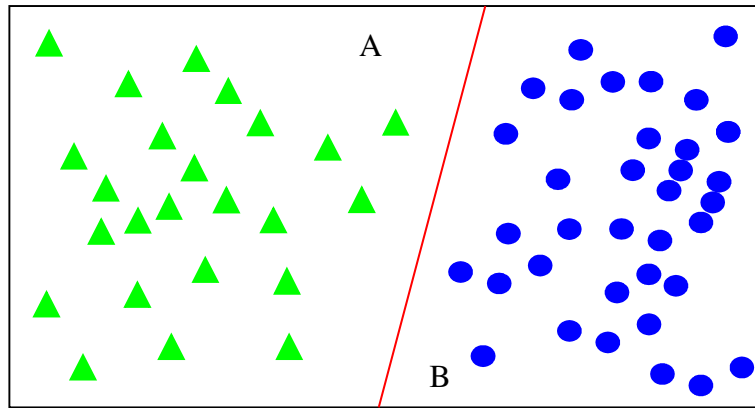


Figure 2.16 Example of a feature space, X

2.10 Automatic Target Classification (ATC) procedure

An overview of the ATC system that we used is depicted in Figure 2.17. In general, it comprises of three stages: pre-processing, feature extraction and classification. The first stage consists of a number of sub-processes to analyse the time domain signals. First, the signal is transformed to the frequency domain. The power spectrum is then normalised to its maximum in order to avoid the effect of different levels of signal power. The spectrum of the signal is then resampled to the reference speed. This is because the signal (and its spectrum) is affected by the speed of the target. The spectra of a car of various speeds will have similar shapes, but with a different scaling-like factor which is proportional to the speed. The output of this stage, which provides the features that characterise the target, is then used as the input in the second stage.

In the feature extraction stage, before any further processing, the selection of the frequency range to be used should be chosen in order to extract only useful information. Based on practical experience, the difference in spectra for different car is observed at higher part of the

power spectrum and signal degradation below 20 dB from the maximum power level can safely be ignored, since it does not improve the classification performance but can only clutter the useful information [1]. This selection of spectra is still of a very high dimension and we need to decorrelate the features. In our analysis we employed Principal Component Analysis (PCA).

In the third stage, the signal can be used both as training or testing data.

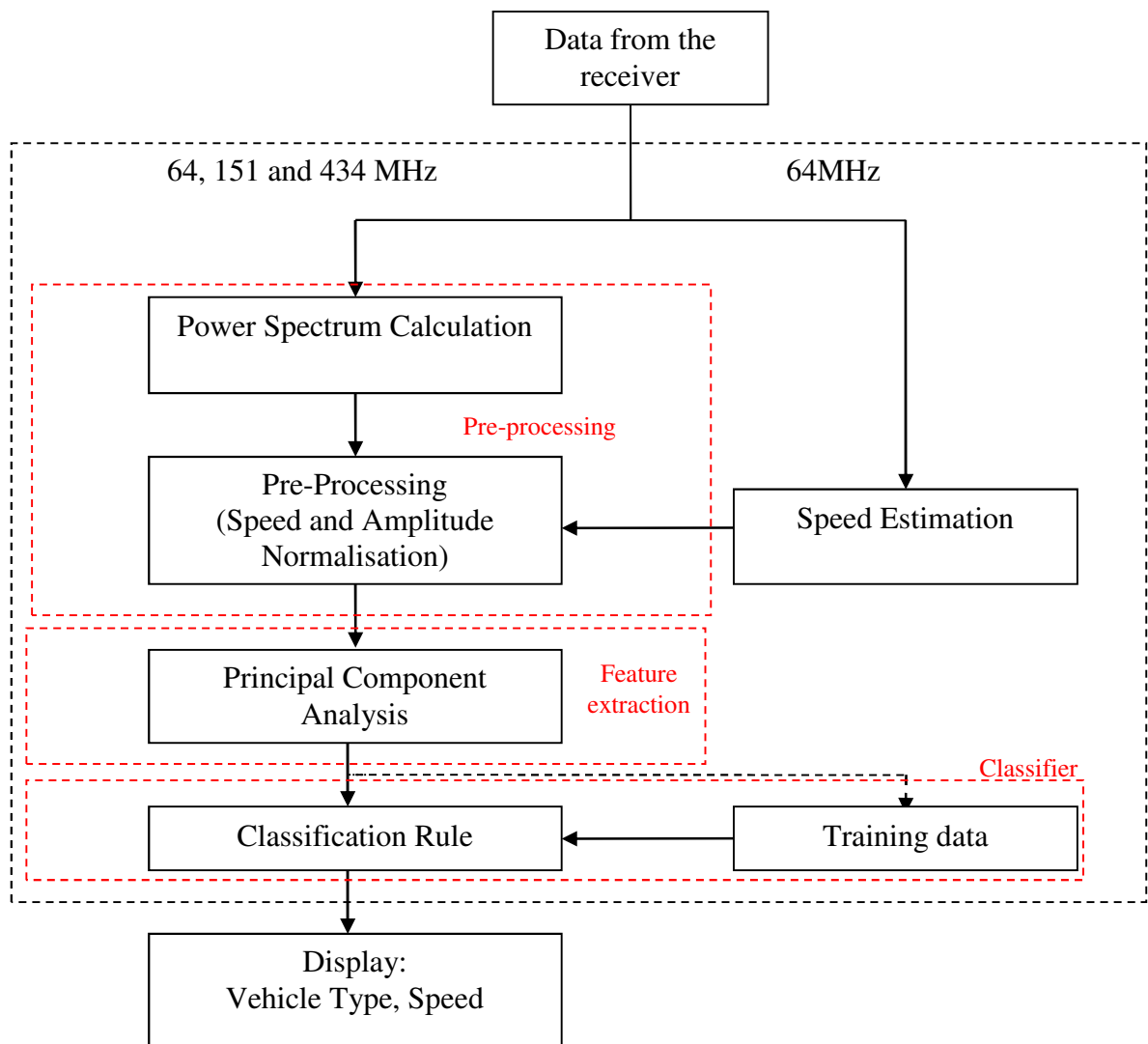


Figure 2.17 Automatic Target Classification (ATC) block diagram

2.10.1 Pre-processing

Figure 2.18 illustrates a block diagram of a pre-processing module. The aim of this process is to produce a target's 'feature vector' in the frequency domain that will later be used by PCA to reduce its dimension.

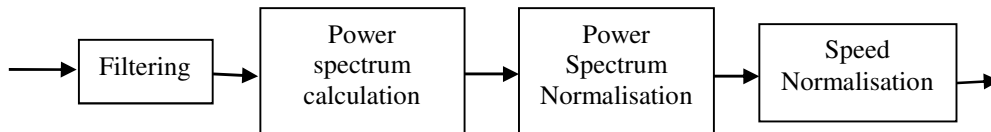


Figure 2.18 Pre-processing block diagram

- **Signal filtering**

A recorded signal comprises the reflection from all objects surrounding the radar including slow-moving or swaying objects, e.g. trees and bushes that act as clutter and noise. We assume here that clutter concentrates in the lower part of the Doppler signal spectrum. This unwanted signal needs to be removed by passing the signal through a high-pass filter (HPF). For each frequency, the HPF cut-off frequency has a different value irrespective of the speed of the target. This is due to the fact that the spectrum of clutter is more or less the same at all frequencies [45], whilst the width of the target's spectrum varies with frequency. For low frequencies the clutter spectrum largely overlaps with the spectrum of the target signature, whereas for high frequencies only a small part of the clutter spectrum overlaps with that of the target signature (see Figure 2.19). If the larger value of cut-off frequency is used for a low frequency carrier, most of the important information about the target will be removed; therefore a small value of cut-off frequency should be used. However, if we apply the same

small value cut-off frequency to a high carrier frequency, large amounts of clutter could still remain in the target signature.

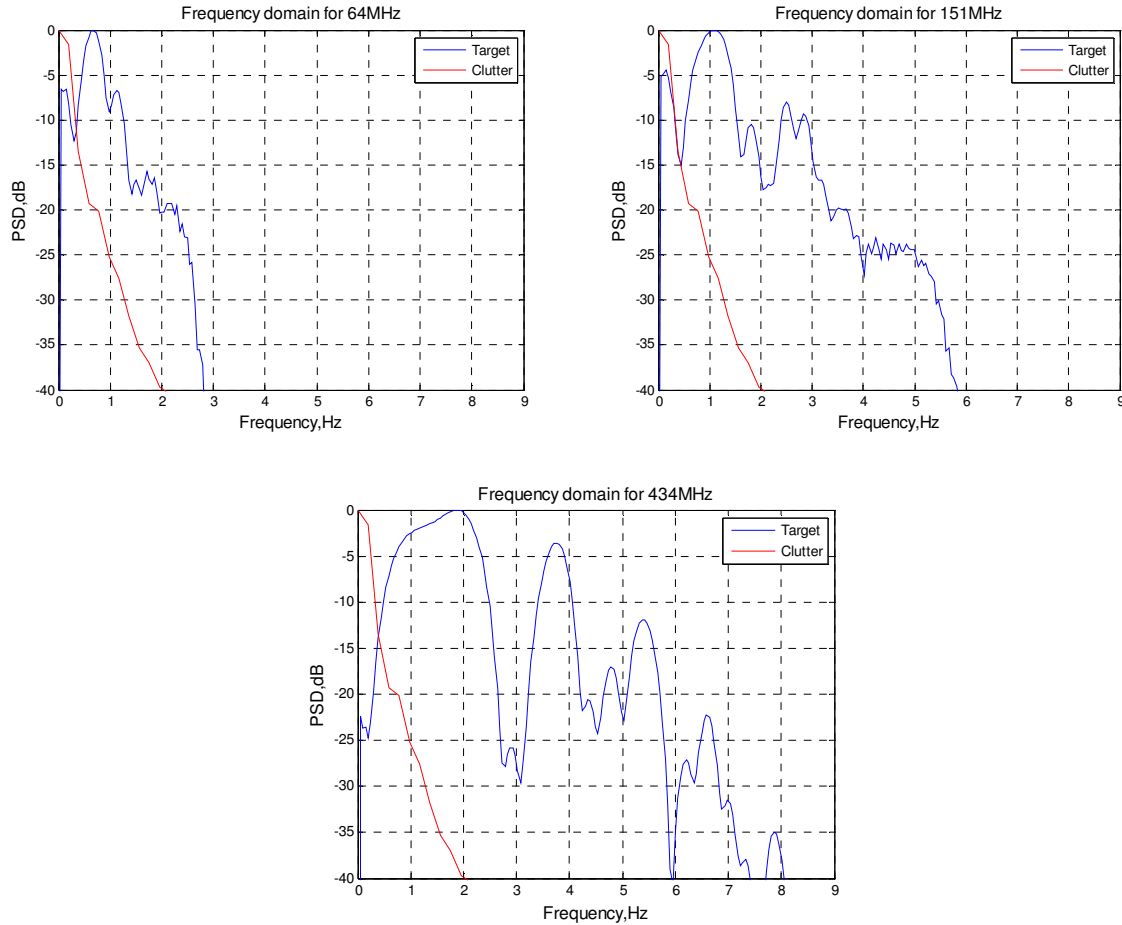
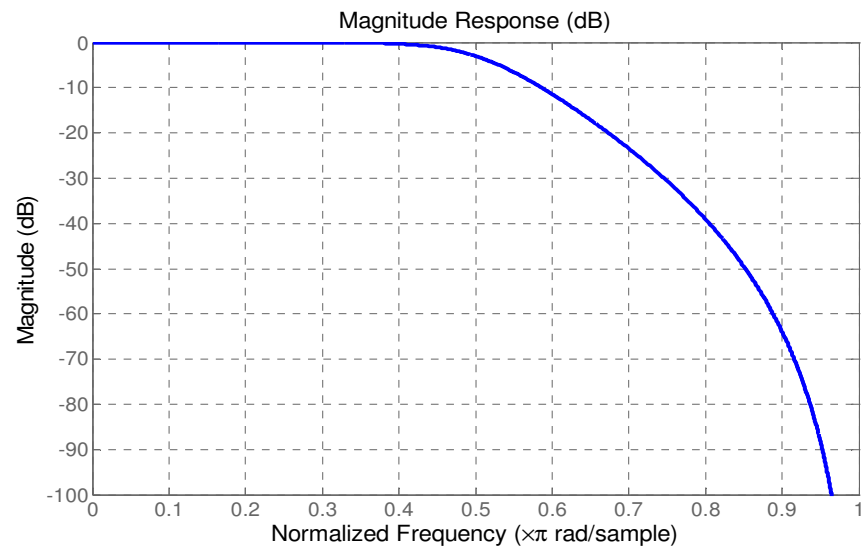
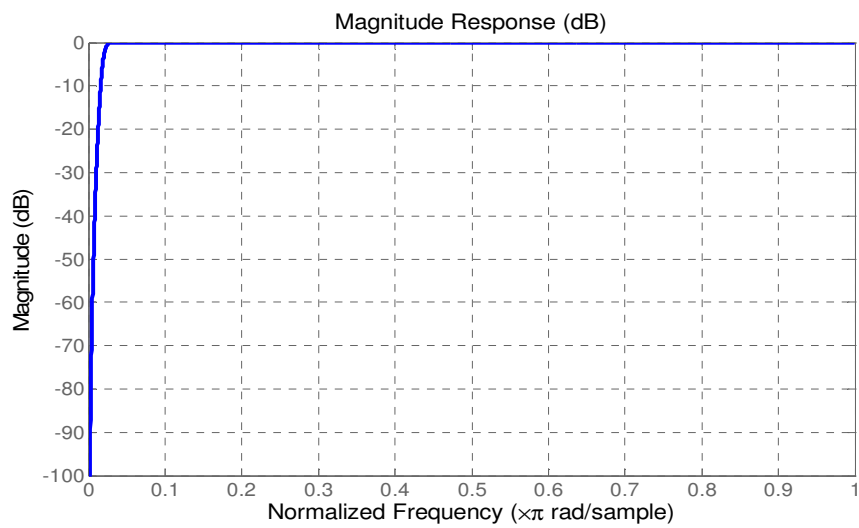


Figure 2.19 Comparisons between target's power spectrum and clutter power spectrum

Figure 2.20 (a) and (b) show an example of the magnitude frequency response of LPF and HPF, respectively, for 151MHz. To reject any signals outside the Doppler frequency band, a 5th order Butterworth LPF with a cut-off frequency of 25Hz was used for all carrier frequencies.



(a)



(b)

Figure 2.20 Clutter suppression filter frequency response (a) Low-pass filter and (b) High-pass filter

- **Power Spectrum Calculation (Frequency Domain transformation)**

The signal sampling rate is set during the experimentation.

The filtered time domain signal is transformed into the frequency domain using Fourier Transform (FT) and the power spectrum, $S_x(f)$ of the signal, $x(t)$ is estimated using:

$$S_x(f) = \lim_{T \rightarrow \infty} \frac{1}{T} |X_T(f)|^2, \text{ where } X_T(f) = \int_{-T}^T x(t) e^{-j2\pi ft} dt \quad (2.25)$$

The spectrum represents the Doppler components contained in the received signal. The frequency resolution in the power spectrum is evaluated by the ratio of the sampling rate, F_s and signal length, n :

$$\Delta f \approx \frac{F_s}{n} \quad (2.26)$$

For example, if the signal length is set to be 2000 and the sampling rate is 100Hz, the power spectrums will have a frequency resolution of 0.05Hz.

Examples of measured time domain signals and their corresponding power spectral densities for two different types of vehicles (a Land Rover and a Nissan) are shown in Figure 2.21. The differences between the spectra main lobe and side lobes are visible for both vehicles as highlighted.

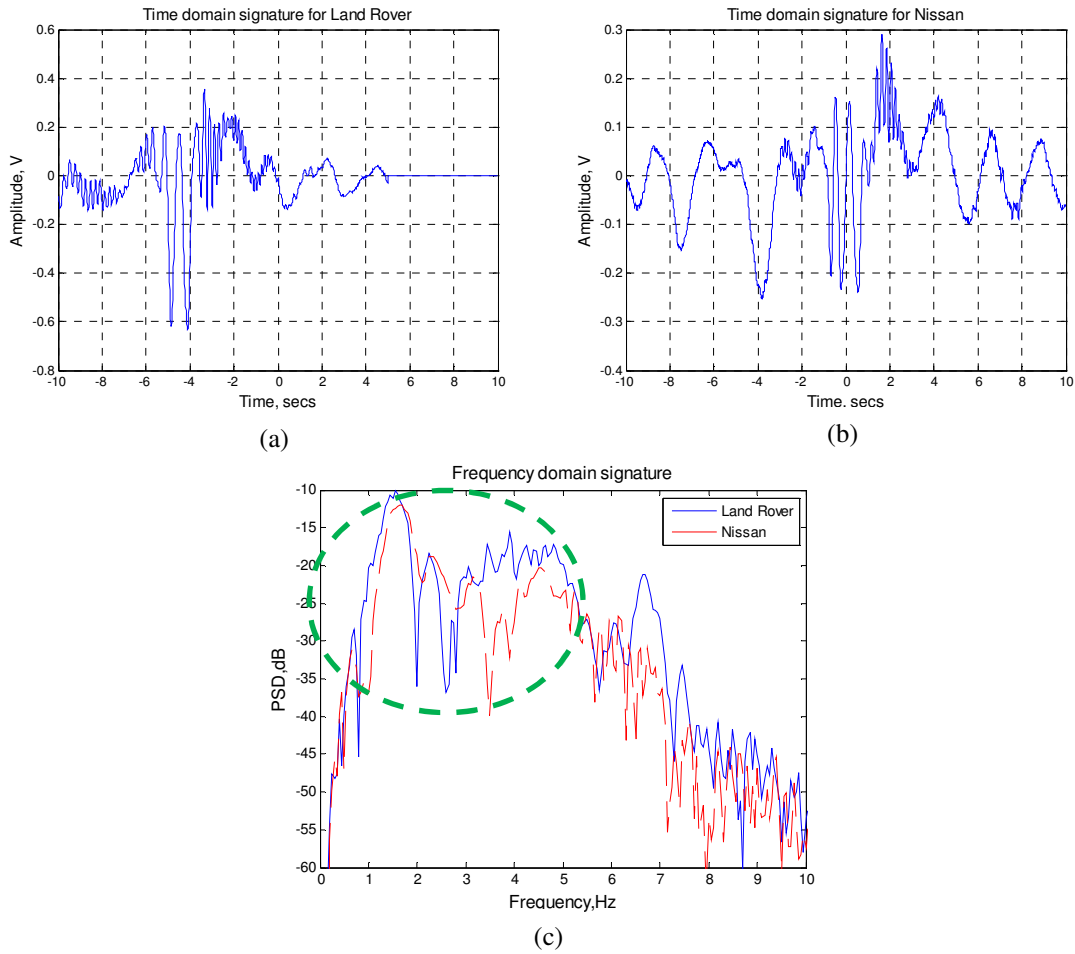


Figure 2.21 Time domain signature a for (a) Land rover (b) Nissan and (c) frequency domain for both vehicles

- **Power Spectrum Normalisation**

After transforming the signal to its spectrum, the next step is to normalise the amplitude of the spectrum to its maximum power:

$$PSD_{\text{normalised}} = 10 * \log_{10}(S_x(f)/\max(S_x(f)));$$

The absolute level of the receiving signal fundamentally depends on the baseline crossing point. For any target, the receiving signal power could vary by around 20-30 dB when crossing the baseline (either closer to the transmitter or receiver).

In our analyses, we have only considered the shape of the spectrum to describe the target's properties. Hence, a normalisation process is applied in order to avoid any misclassification caused by the absolute power level of the corresponding signal. The spectra of the Land Rover before and after the normalisation process are shown in Figure 2.22.

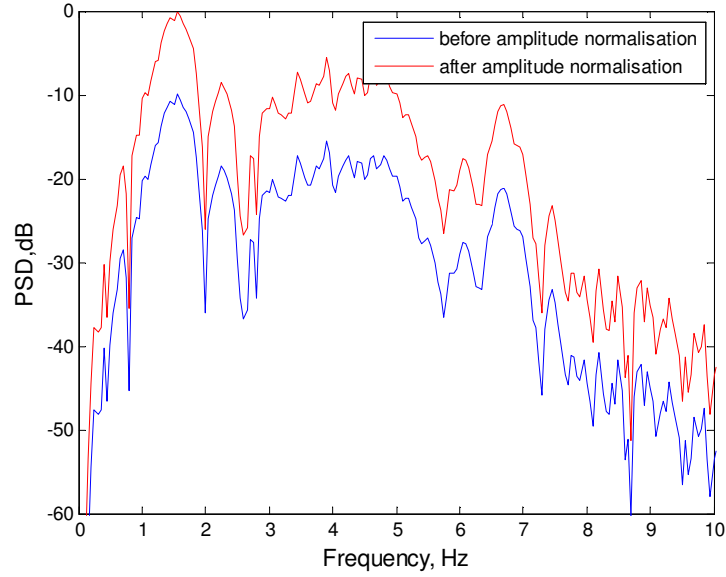


Figure 2.22 Amplitude normalisation

- **Speed Normalisation**

Since the signal (and its spectrum) is affected by the speed of the target, the obtained spectrum needs to be processed before it is used as a feature vector in the classification system. In general, the bandwidth of the main lobe of the spectrum Δf depends on the speed v , and the length of the target l and is given by [8]

$$\Delta f = \frac{v}{l} \quad (2.27)$$

Figure 2.23 (a) shows the spectrum of the same car passing through the baseline at two different speeds (8.3 m/s and 8.1 m/s). We can see that both spectra have a similar shape, however, with a different scaling-like factor which is proportional to the speed.

The speed can be estimated using different methods such as an automatic speed estimation algorithm, a speedometer or manually by using a video camera. When we use this information about the speed, the spectra in Figure 2.23(a) can be “resampled”. At this stage a resampling ratio is calculated. The resampling ratio is the ratio of the estimated speed of the target to the reference speed. The spectra of the target are either linearly compressed or expanded based on the resampling ratio. Figure 2.23 (b) shows the spectrum of the same vehicle after it has been normalised to the reference speed. Details of the speed estimation algorithm will be described in Chapter 5.

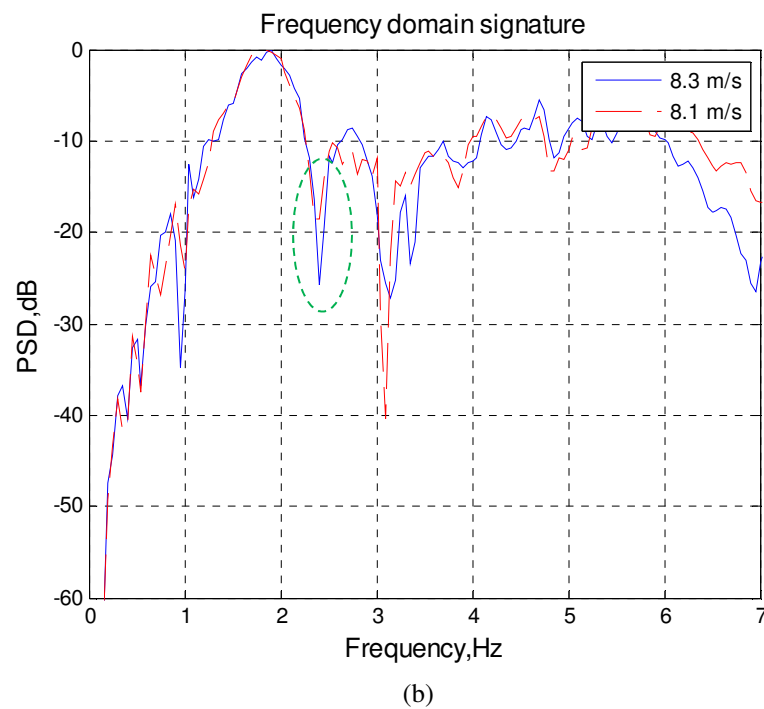
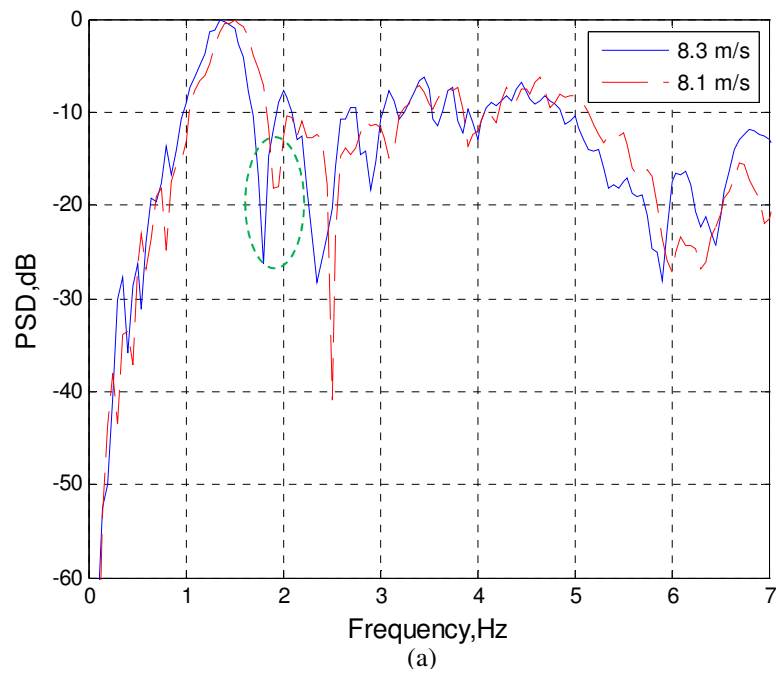


Figure 2.23 The spectra of a same car at two different speeds before (a) and after (b) the speed normalisation spectrum

2.10.2 Classification System

Figure 2.24 shows the overall system for vehicle classification that has been used in our work. In order to perform the classification, the first step is to convert the received signal to a frequency domain signal so that the distinct differences between the vehicles become visible.

The spectra feature vector obtained from the pre-processing block, which consists of the processing steps as described in section 2.10.1, is of a high dimension and the features are highly correlated. Principle Component Analysis (PCA) was employed in order to reduce the dimensionality of the spectra feature vector by exploiting the correlation between the features.

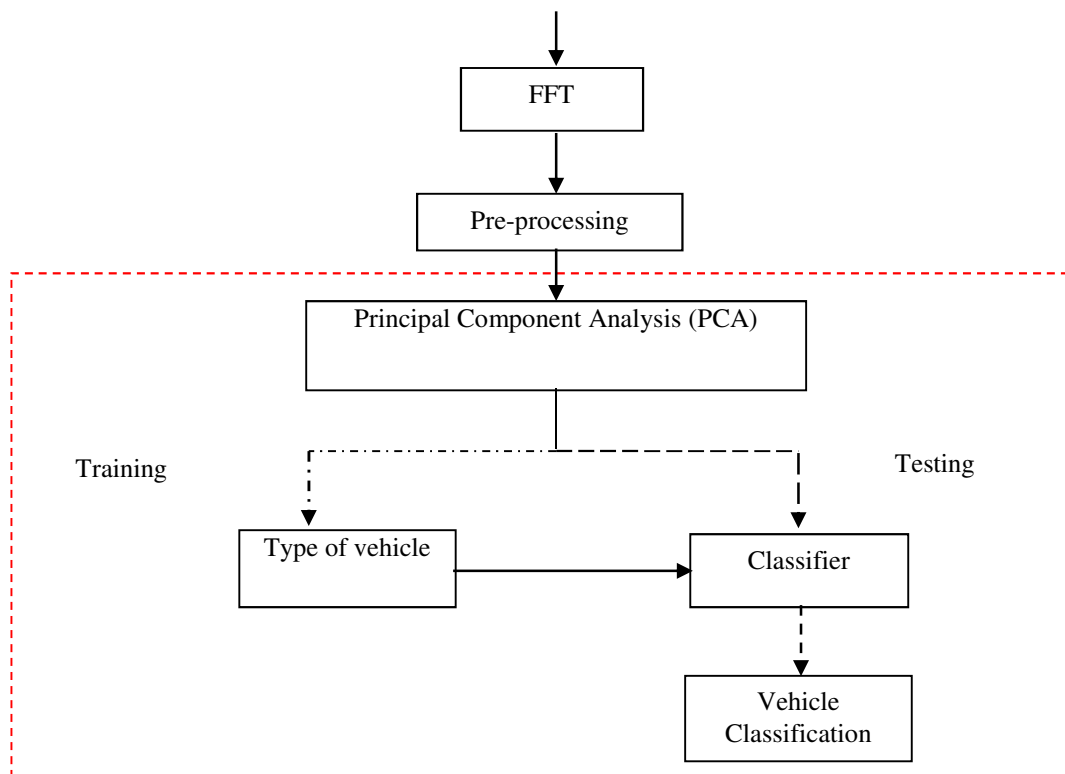


Figure 2.24 Overall system for vehicle classification that has been used in this thesis

- **Principal Component Analysis (PCA)**

Principal Component Analysis (PCA) was invented by Pearson (1901) [52] and was first introduced by Hotelling in 1933[53]. PCA can be used for exploratory data analysis and data pre-processing to reduce data dimensionality in many applications, for example short-term weather prediction [54], face recognition [55] and target recognition.

The extracted feature is known as Principal Component (PC). The PCs are arranged so that the amount of variance of the data explained by each PC is non-increasing; the first PC represents the largest amount of variance in the data, the second represents the second largest and so on [56]. Normally, only the first few principal components are necessary to represent the information contained within the data.

The PCA technique for a given set of vectors , $\{x_i\}$ $i=1 \dots M$ can be summarised as below[57]:

- i. Find the mean of a given set of signals feature vectors.

$$\bar{x} = \frac{\sum_{i=1}^M x_i}{M}$$

where x_i is a feature vector and n is a number of vectors

- ii. Subtract the mean feature vector from each feature vector. This produces a data set that has a zero mean.

$$R = (x_i - \bar{x})$$

- iii. Calculate the covariance matrix for a given set of signal feature vectors

$$S = \frac{\sum_{i=1}^M (x_i - \bar{x})(x_i - \bar{x})}{M - 1}$$

- iv. Calculate the eigenvectors, e_i and eigenvalues, λ_i of the covariance matrix

$$(S - \lambda_i I)e_i = 0$$

- v. Choose the eigenvectors corresponding to the highest value of eigenvalues. The eigenvectors are the PC of the data set i.e. only the first p eigenvectors

$$W = [e_1; e_2; e_3; \dots; e_p]$$

- vi. Transform the signals into a new space

$$NewData = W^T \times R$$

There is no mathematical formula to determine the number of PCs to retain. One simple method is to select a (cumulative) percentage of the total variation to which the selected PCs should contribute high amount of variation. The required number of PCs is then the smallest value for which this chosen percentage is exceeded [56].

Figure 2.25 shows the amount of variance retained by the first 10 PCs for our database. As we can see, the percentage of variance is $> 80\%$ for the first three PCs and increase at the PCs number increase.

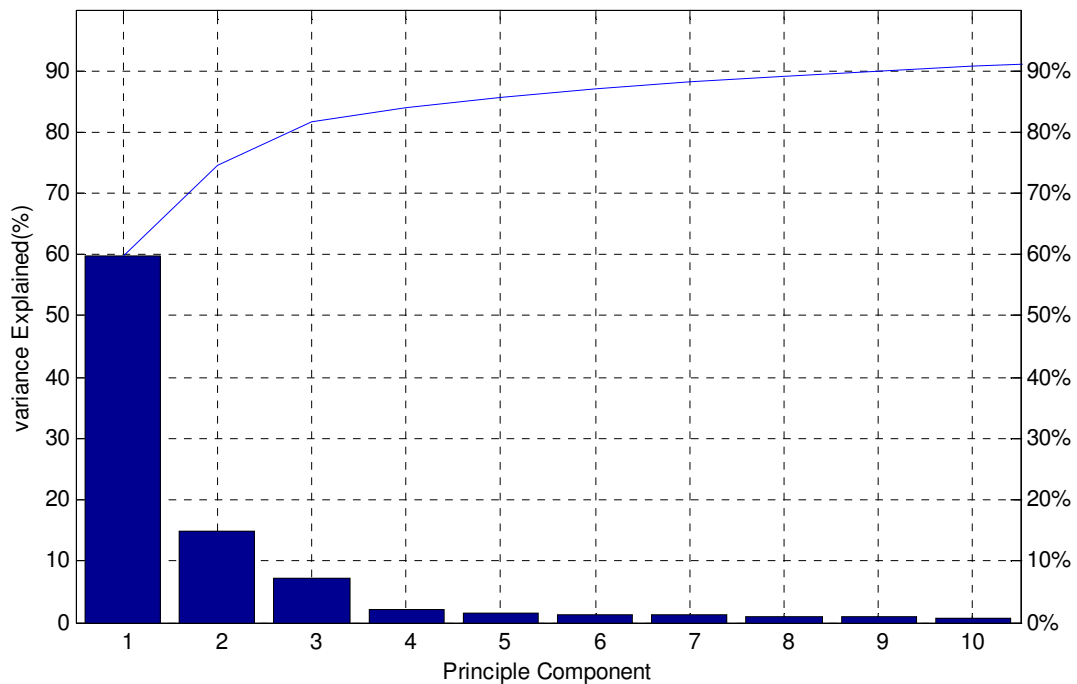


Figure 2.25 The variance and cumulative variance described by a given principal component

- **Training and testing phase**

In general, there are two phases in constructing a classification system: training phase and testing phase.

The aim of the training phase is to create a target model of each class. The training data is converted into multidimensional vectors in the PCA feature space, each with a class label (for example, type of car).

The classification system can be based on modelling the distribution of the features for each class or on discrimination between classes. The discrimination-based approach, such as neural networks, has a potential advantage over the modelling approach; however, it requires a large

amount of training data. The distribution modelling approach may employ a parametric or a non-parametric model of the distribution. Assuming some parametric form of the distribution, for instance Gaussian distribution model or a mixture of Gaussian distributors, the parameters of the model are estimated during the training phase using the training data. An accurate estimation of the parameter may be difficult in a case of a small amount of data. Therefore, in this work a non-parametric approach was employed, specifically the k-Nearest Neighbour (k-NN) classifier. Despite its simplicity, the k-NN is a powerful pattern recognition technique.

The training phase in the case of the k-NN classifier consists of simply storing all the feature vectors for each class according to the labels of the training data. Figure 2.26 shows an example of 2-dimensional PCA features of the training data of three different types of cars.

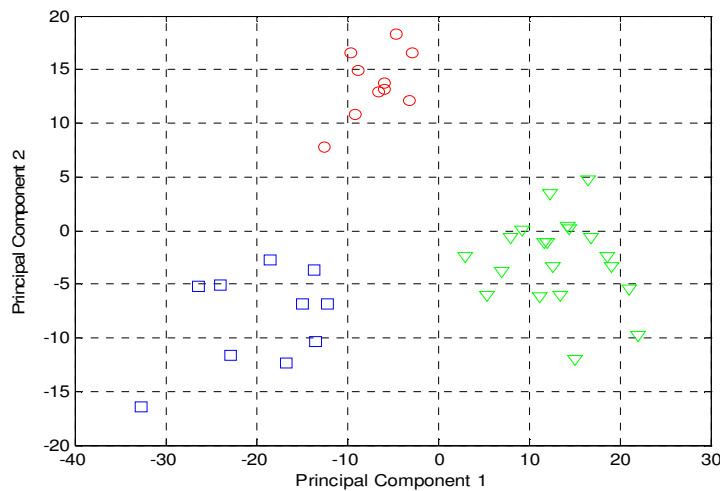


Figure 2.26 The location of training data of each vehicle type in the PCA-space

In the classification phase, a feature vector is extracted from a given unknown signal. In the case of the k-NN classifier, the Euclidean distance of this test vector is calculated for all the stored training data feature vectors of each class. The unknown signal is then classified as belonging to the class which has the majority of the training feature vectors within the k-

nearest training feature vectors to the test feature vector. There are several slight modifications to this conventional k -NN classifier rule. In this work, the overall distance of the test feature vector to each class is computed by taking the average of the Euclidean distances to the k -nearest feature vectors from the corresponding training set. Then the test feature vector is classified according to the minimum average distance across all the classes. K is a constant defined by a user. The best choice of K depends on the classification task itself. Generally larger values of K can reduce the effect of noise on the classification, but they also make the boundaries between the classes less distinct. The accuracy of the k -NN algorithm can be severely degraded if the scales of individual features are not consistent with their importance.

An example of how a target is classified using KNN is illustrated in Figure 2.27. In this plot, we have three different target classes: target A='red circle' (○), target B='blue square' (□) and target C='green triangle' (△). The unknown target is represented by a 'black circle' (●). Using $k=3$, the test data (●) would be recognised as the 'red circle' (○).

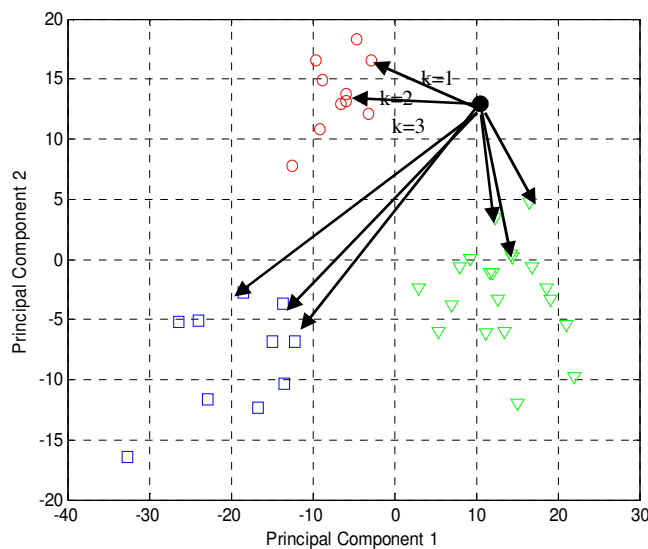


Figure 2.27 Distance between test data and training data for $k=3$

2.11 Limitation

In literature on the subject, FSR demonstrates its competent performance in the classification of vehicles. Several aspects could, however, be investigated to explore the potential of FSR for further improvement as a means of vehicle classification.

The early ATC system that was built used a high frequency of 1GHz, directional antennas and covered a short range. This became a limitation to the system since higher frequencies will cause higher propagation losses (higher power budget) and reduce battery life. Apart from this we also require a sensor that could be freely dropped at any position and at any distance. With a directional antenna this is impossible to achieve since its transmitter and receiver need to be positioned directly facing each other.

It is shown in [58] that the classification performance depends on the speed estimation accuracy. Previously, speed was estimated by playing back a recorded video – a time-consuming process which was done manually. It is therefore worthwhile to study and develop an intelligent method for estimating speed accurately, at the same time improving the accuracy of the classification.

Previous research concentrated on feasibility studies in ideal cases using directional antennas and fixed FSR topology. It did not take into account the effects of the changes of FSR topology such as the length of the baseline and the crossing angle. These changes affect the Doppler signature of the target and might degrade the classification accuracy. Furthermore, research has so far been conducted with practically no clutter presence.

2.12 Summary

In the first part of this chapter (subsections 2.1-2.7), the basic concept of FSR has been discussed. A survey of the literature review on FSR reveals that there is a lack of research and analysis of the factors that influence the accuracy of classification for ground targets and low (VHF) frequencies in particular.

The second part of the chapter 2 (subsections 2.8-2.11), highlighted the techniques for estimating the speed and for extracting the required features, as well as the classification technique used within the classification system. The classification system that is currently being used for ground target FSR system was reviewed and the limitations of the current techniques were discussed.

CHAPTER 3

TARGET SIGNATURE EXPERIMENTAL COLLECTION AND ANALYSIS

3.1 Introduction

In this research we will concentrate on collecting and analysing target signatures in a real case scenario. We carried out an outdoor experiment in order to see and analyse the influence that the target trajectories and the environment present to the system.

The outdoor experimental data collection and description are described in detail in this chapter, at the end of which we will present the preliminary experimental results.

3.2 Experimental data collection

The experiment was performed using a prototype FSR system developed by the Microwave Integrated System Laboratory (MISL) at the University of Birmingham [59] . The objective of the experiment was to collect signals from ground moving targets crossing the baseline. The targets were of different dimensions, shapes and sizes.

The experiments were carried out at Tilesford Airfield in Pershore, Worcestershire and Pritchatts Road car park, University of Birmingham. The airfield has a large level runway, with occasional trees and bushes far from the baseline on the outskirts of the runway, so that minimum vegetation clutter is produced. The car park, on the other hand, is an open space with some big trees located behind the transmitter.

3.2.1 Hardware description

Hardware is one of the important components in the experiment. A transmitter is used to transmit signals and a receiver is used to capture them. Figure 3.1 shows the multi-frequency equipment that we used for the FSR system. It consists of:

- a) a 3-channel transmitter (3N_TX) and
- b) a 3-channel receiver (3N_RX).



Figure 3.1 Multi-frequency equipment

The transmitter and the receiver both have six channels (three Doppler channels and three raw signal channels). The transmitter (3N_TX) transmits three continuous wave (CW) carrier frequencies simultaneously. The receiving antennas capture the signals, which are a combination of a shadowed signal from the moving target, a direct signal from the transmitter, and a scattered signal from the surroundings and clutter.

The transmitter and receiver channels are built from standard short-range radio modules made by Radiometrix⁴ (see Table 3.1). The use of unified transmitting and receiving modules make it possible to change carrier frequencies in the equipment. In case there is an interference with other narrow-band signals nearby, the modules have a built-in carrier frequency adjustment

⁴ <http://www.radiometrix.com/our-products/narrow+band>

* The receiving modules are modified to increase the dynamic band of the input signal.

system of around 1-2 MHz in each band. The frequency selection is used to install the requisite frequency carriers.

Table 3.1 Transmitter and receiver modules and antennas

| Channel, MHz | TX module | RX module | Antenna | Output power, dBm |
|--------------|------------|-------------|--------------------------|-------------------|
| 64 | TX1M-64-5 | RX1M-64-5 | MXSK-BNC (64.5 MHz) | 20 |
| 151 | LMT1-151-5 | LMR1-154-5* | MFX-H5-BNC (151 MHz) | 10 |
| 434 | TX2M-433-5 | LMR2-434-5 | Directional Yagi antenna | 20 |

A more detailed block diagram of the transmitter is shown in Figure 3.2.

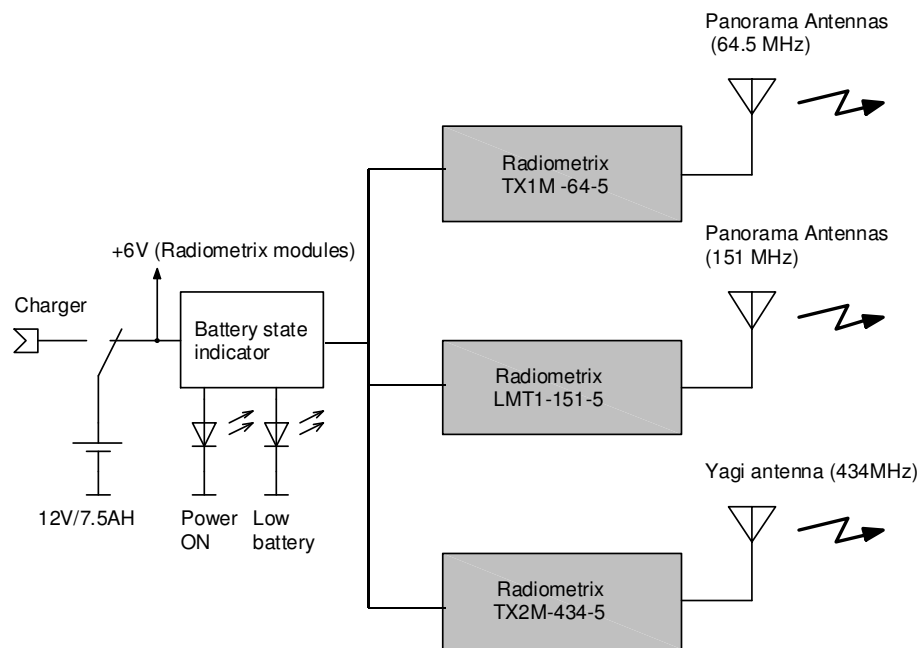


Figure 3.2 Example of 3N_TX block diagram

The transmitter comprises a 12V-power supply, a battery state indicator, frequency selector units, and Radiometrix' modules and antennas. The battery state indicator (LED) shows when

the power is on and the flashing LED indicate when the battery is low. The battery can be charged using a suitable charger connected to an external socket.

In this reported experimentation, we used directional Yagi antennas with 9dB gain at 434MHz and standard Panorama helical monopole antennas at other frequencies for both the transmitting and receiving antennas⁵. Figure 3.3 shows photos of the antennas. Detailed specifications of the antennas can be found in Appendix A .

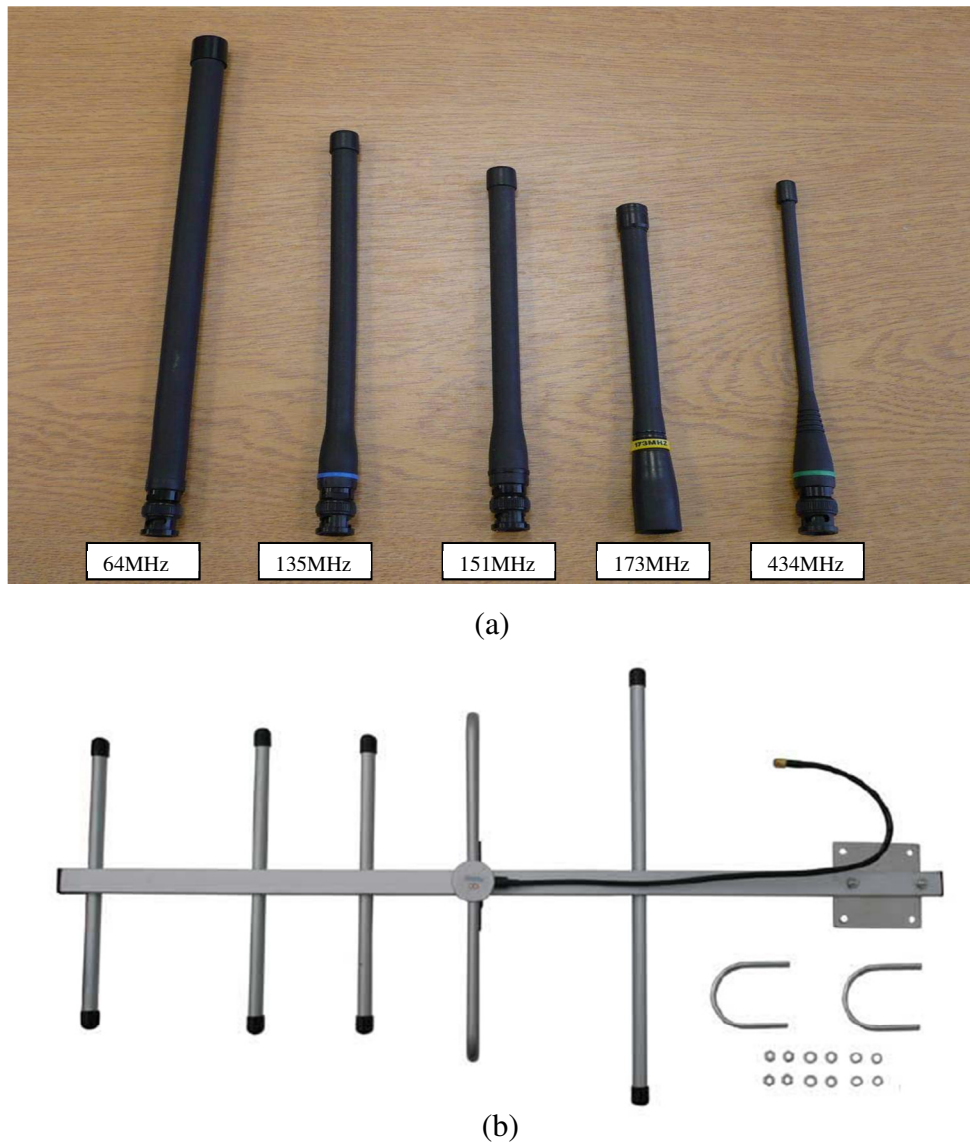


Figure 3.3 Antenna (a) Panorama helical (b) 9dB directional Yagi

⁵ The choice of antennas was not specified and only available antennas were used.

Since the transmitter and the receiver are installed directly on the ground, the effective heights of the antennas are dependent on the length and module dimensions of each antenna. The total height for each antenna is listed as follows:

- 3N_RX: 64 MHz-32cm; 151 MHz -29cm
- 3N_TX: 64 MHz-20cm; 151 MHz -15cm
- The used of Yagi antennas with 9dB gain for 434 MHz on tripod – 1m

In some situations, especially when the baseline is small, we needed to attach an extra BNC-type attenuator to the antenna in order to reduce the amplitude or power of the signal. Each extra BNC-type attenuator that we attached added another 5cm to the total height of the antenna from the ground.

At the receiving site, we added additional parts to the hardware such as filters, an amplifier and a Universal Serial Bus (USB) extender. The receiver was then connected to a personal computer by extended USB cables to get data from analogue-to-digital converters (ADC) and to store the received data files in the hard drive of the computer) for further processing. A block diagram of the receiver is shown in Figure 3.4.

The received signals were fed to the Radiometrix' modules. The received signal strength indicator (RSSI) output of the modules has a logarithmic dependence of output voltage from input power in a dynamic band (as in Figure 3.5). The average value of the output voltage corresponds to the direct leakage of power. It is converted by the ADC and transferred to the computer. Since the obtained signals are a mixture of target signals, clutter and noise, and are lower (up to 20-40 dB) than the leakage power, we can see a small variation in the RSSI

voltage. In order to fully utilise the ADC resolution for the target Doppler signature, additional Doppler channels were created. The DC component in each RSSI output was rejected by 1st order HPF with cut-off frequency of 0.1 Hz before it was sent to the amplifier. Then Doppler variations were then amplified with a constant gain of 30dB. After amplification, the signal was filtered by a 40 Hz LPF before being sent to the ADC. The ADC is a 16-bit USB-1608FS. The output of the ADC is sent to a USB extender, which is connected to a PC.

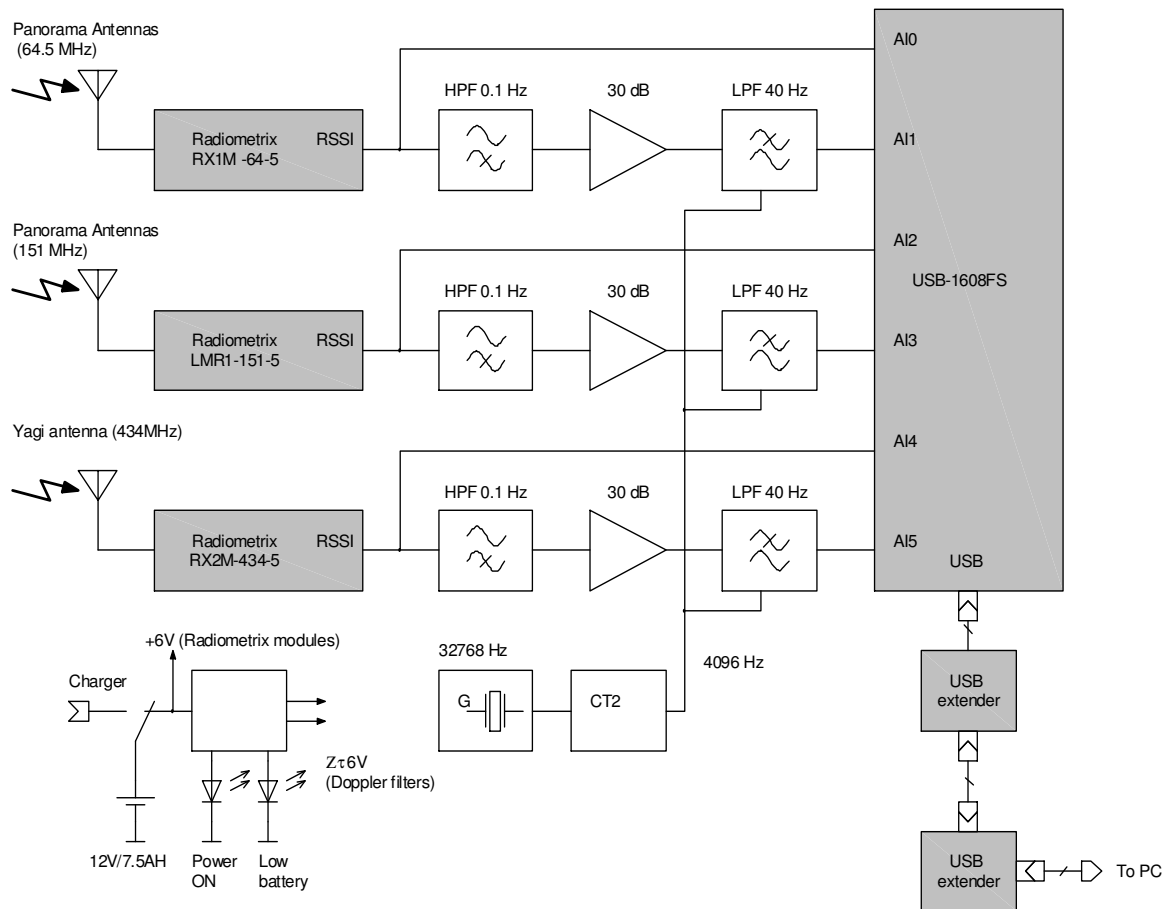


Figure 3.4 3N_RX block diagram

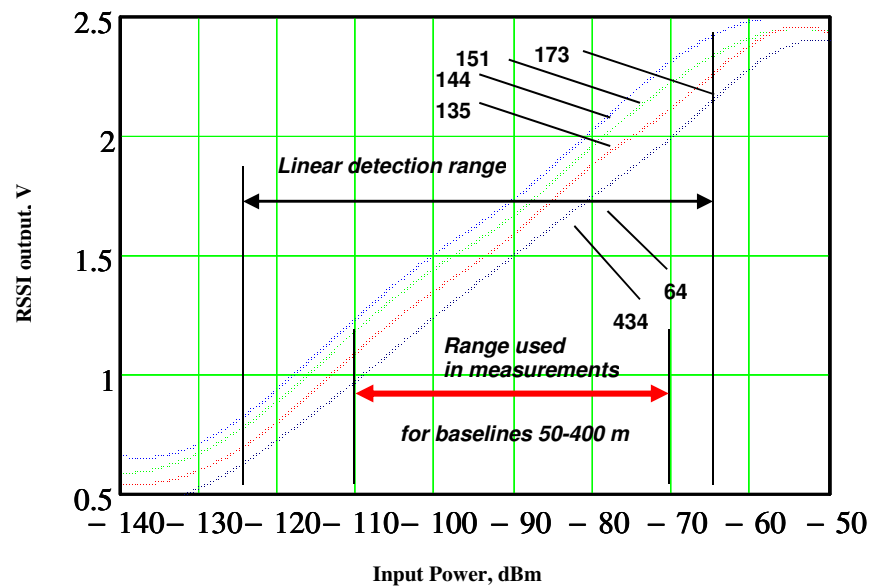


Figure 3.5 Receiver detection range

The amplified signal is passed to the anti-aliasing LPF with a cut-off frequency of 40 Hz (5th order switched capacitor filters). The filtered signal is then sent to the ADC converter and stored in a computer where the signal can be processed and analysed further in the MATLAB. The signal at the output of the ADC contains the Doppler signal and this signal is referred to as the ‘target signature’, which became the input to the classification method.

3.2.2 Recording data format

There are two MATHCAD programs that are used to extract and display data from the hardware. The programs, listed below, are developed by Dr Vladimir Sizov for EMRC DTC project 2/65:

- a. Data acquisition: 3N_receiver.mcd

This program is used to get the data from the ADC and store it in the hard drive.

- b. Data visualisation: 3N_rceiver.mcd

This program is used to visualise the raw data from the stored files.

The data acquisition program automatically adds prefixes “3N_filename.bin” to all data file names collected in the measurements by this variant of the equipment. For example “3N_landrover.bin” means binary data file with main name Land Rover obtained by 3N_RX.

The data acquisition procedure is done in the following few steps:

- i) Connect USB cable from the receiver to a computer.
- ii) Start the program “Instacal” to install the driver for the ADC board. Close the program once the ADC board has been found.
- iii) Start the data acquisition software and set the initial value of the sampling frequency, the acquisition time, the number of samples and the sampling rate for each ADC channel.

The sampling frequency is given by $f_s > 2B$ where $B = \frac{c}{2vf_c}$. For example, if the frequency, $f_c = 150MHz$, speed of the target $v = 20m/s$, the sampling rate must be bigger than 40.

The rate depends on the Channel string. The equipment that we used has a 6-channel string that consists of three RSSI channels and three Doppler Signal channels. Hence, the rate in Figure 3.6 is given by $\frac{f_s}{(\text{number of channels})}$. In our case, for each experiment, we set the sampling frequency $f_s = 100Hz$, number of

samples=12000 and rate=16.66667. The layout of the setting is shown in Figure 3.6.

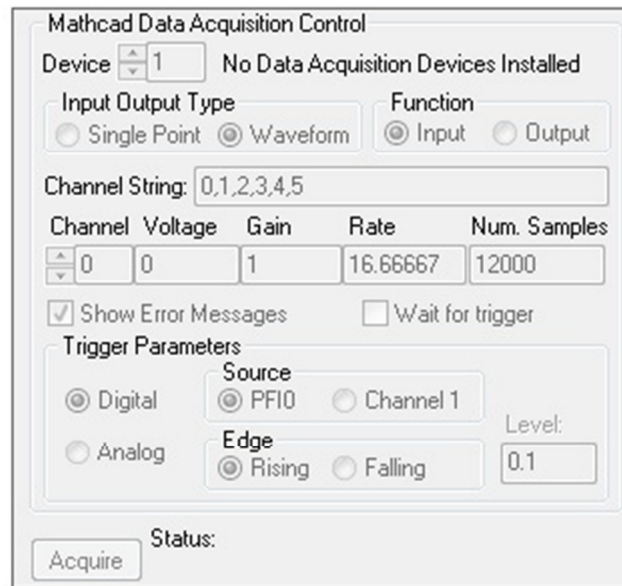


Figure 3.6 Layout of data acquisition setting

- iv) Insert the name of the measurement. The signal will be saved as **.bin** file.
- v) Start the acquisition by pressing Ctrl-F9.

The layout for the MATHCAD data acquisition program is shown in Figure 3.7.

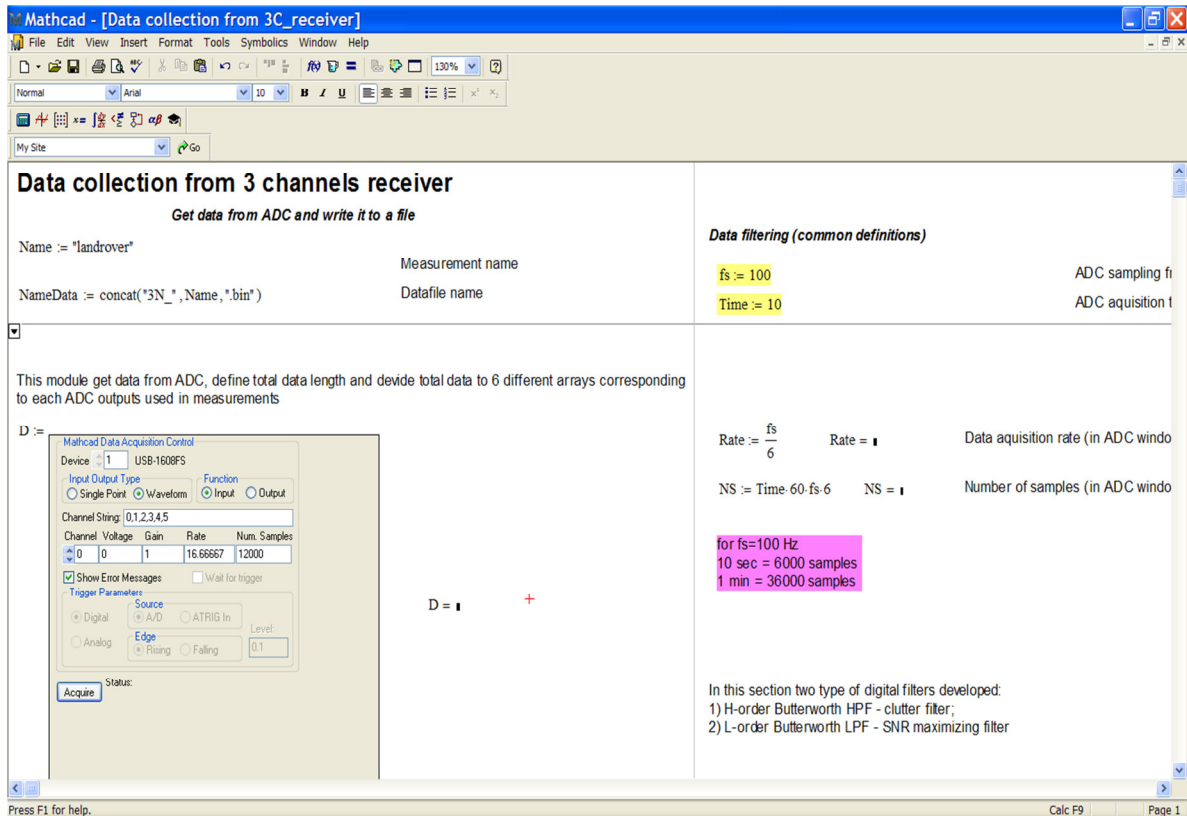


Figure 3.7 Data acquisition layout

For data visualisation software (as shown in Figure 3.8) the procedure is as follows:

- i) Enter the name of the required measured data.
- ii) Press Ctrl-F9 to start the visualisation program.

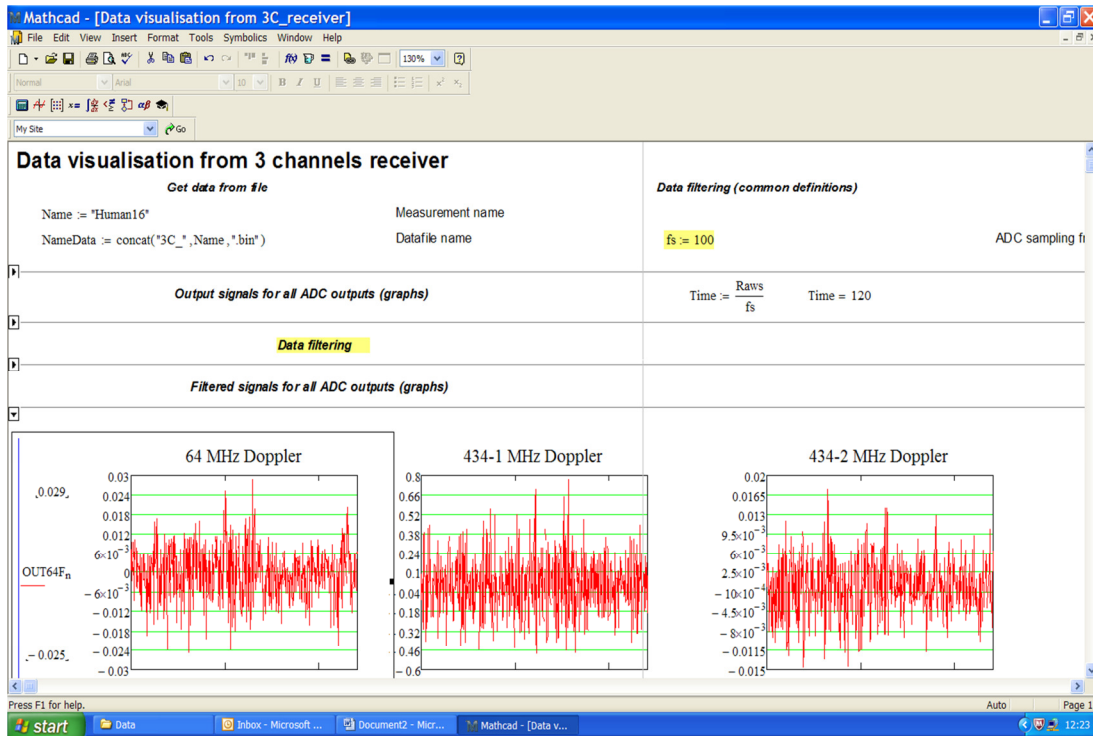


Figure 3.8 Data visualisation layout

3.2.3 Data collection method

The outdoor experiments are, as we have already mentioned, carried out at the Tilesford Airfield in Pershore, and at Pritchatts car park, University of Birmingham. Both experiment fields are practically flat with some vegetation around the baseline. Figure 3.9 and Figure 3.10 show the area of the airfield and the car park, respectively. The measurement site is highlighted in yellow. The airfield has no trees nearby whilst there are rows of trees behind the transmitter and in the middle of the baseline in the car park.

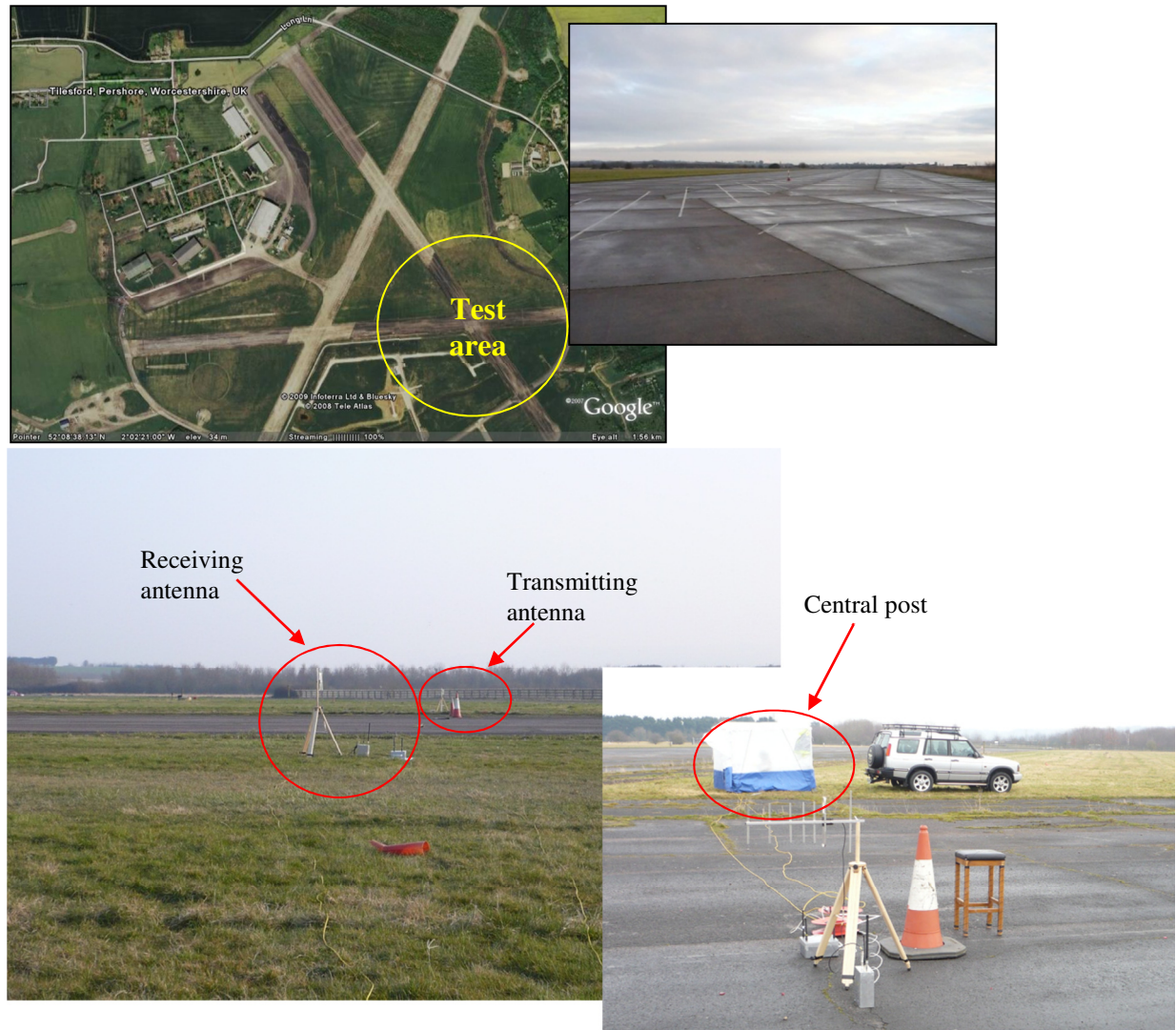


Figure 3.9 Tilesford Airfield

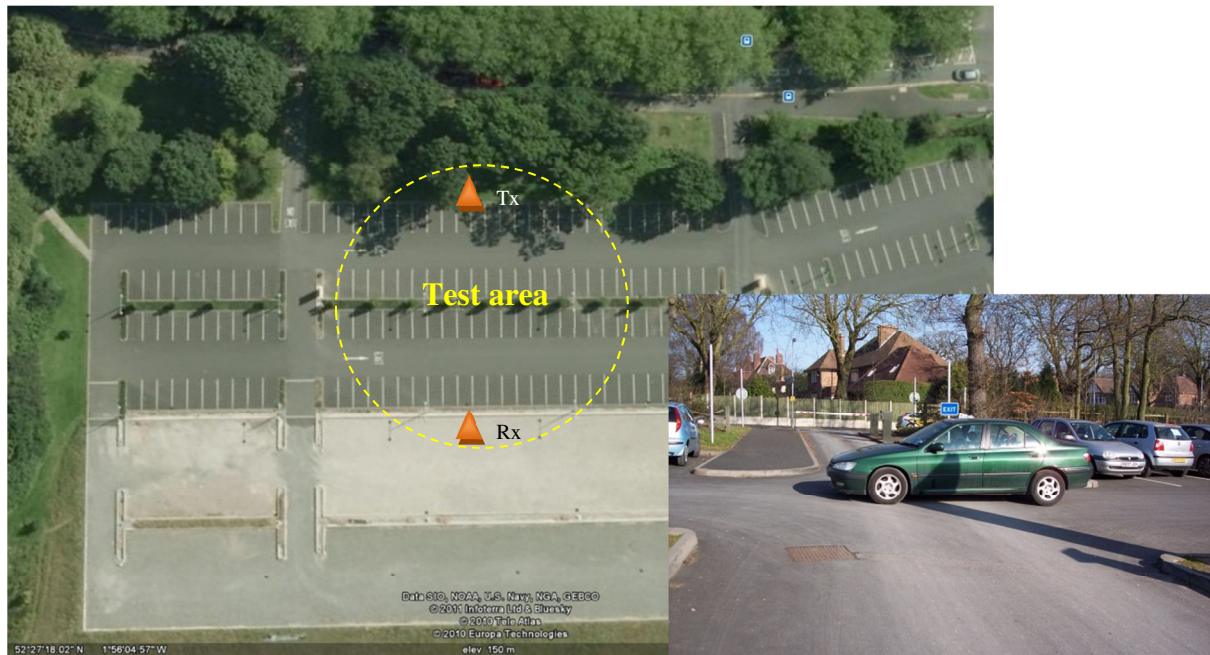


Figure 3.10 Pritchatts car park, University of Birmingham

For both measurements, two pairs of omni-directional antennas and one pair of directional antennas are used for both transmitting and receiving. The antennas faced each other on opposite sides with a separation distance (i.e. FSR baseline) of 50m-200m.

3.3 Experimental data description

As mentioned earlier in section 3.2.2, the recorded signal is saved in a .bin file. The data is in binary format where it contains six channels (three for RSSI and three for Doppler signal). All signals are stored in folder named 'database' and the list of signals is known as 'Vehicle record' (see Appendix B). This list consists of a detailed description of the signals, for example the length of the baseline, the positions of the sensors on the ground, the crossing points and the number of attenuators used.

Throughout this thesis, the ATC system performance evaluations were performed on a database consisting of four types of vehicle target: Land Rover (4.8mx2.1m), Nissan (4.0x1.4m), BMW 316i (4.5mx1.4m) and Ford Focus (4.4mx1.5m). The baseline was set at 50m. The ATC systems used three PCA coefficients and three nearest neighbour classifiers for the classification process. The speed was estimated using a low frequency of 64 MHz. Due to the small size of the database, the classification process employed a leave-one-out procedure rather than splitting the database into a training part and a testing part.

The attached CD has collections of target signals and training models for each frequency. In order to process the signals, MATLAB codes are established, which consist of extracting features of the signal, estimating the speed and classifying the target. In this section, we will explain the process of how to extract the data from the attached CD.

Different kinds of analyses have been done throughout this thesis. For each analysis, we created different MATLAB codes in order to avoid mistakes during processing. An example of a manual on how to perform target classification is given in Appendix D.

3.4 Preliminary Study

Although the concept of FSR for ground-target classification has been developed, the system is far from complete. In this section, we will investigate the effects of a low frequency and of different target trajectories to the target's spectrum.

The experiments are divided into sub-sections. The first subsection inspected the probability of target classification using a low-carrier frequency. As in [1], the author proved that by using a high frequency ($\sim 1\text{GHz}$) with optical approximation, we were able to classify different types and classes of targets. However, to date there is no publication dedicated to ground-target detection and recognition using a low frequency. The second sub-section investigates the effect of the target trajectories, which includes different baseline lengths, crossing points, sensors positions and crossing angles. In this investigation, we will study the similarity and stability of the target signature. Simulation and outdoor experiments have been carried out to see the effect of different target trajectories both in ideal and real cases. For all analysis, we have collected around 5-10 signals for each measurement condition.

3.4.1 Low frequency

In our first experiment [9], the target crossed the 50m baseline at an equal distance from both transmitter and receiver. Three different cars were used: a Land rover, a Peugeot 406 and a Volvo S40. In order to analyse the effects, we used three levels of frequencies: 64 MHz, 151 MHz and 434 MHz. The targets were assumed to have a constant velocity of 9 m/s crossing perpendicular to the middle of the baseline with 10 runs for each car.

The frequency spectra of the three cars in all three frequencies are shown in Figure 3.11. We can see that each car has a different signature and this signature varies with the frequency.

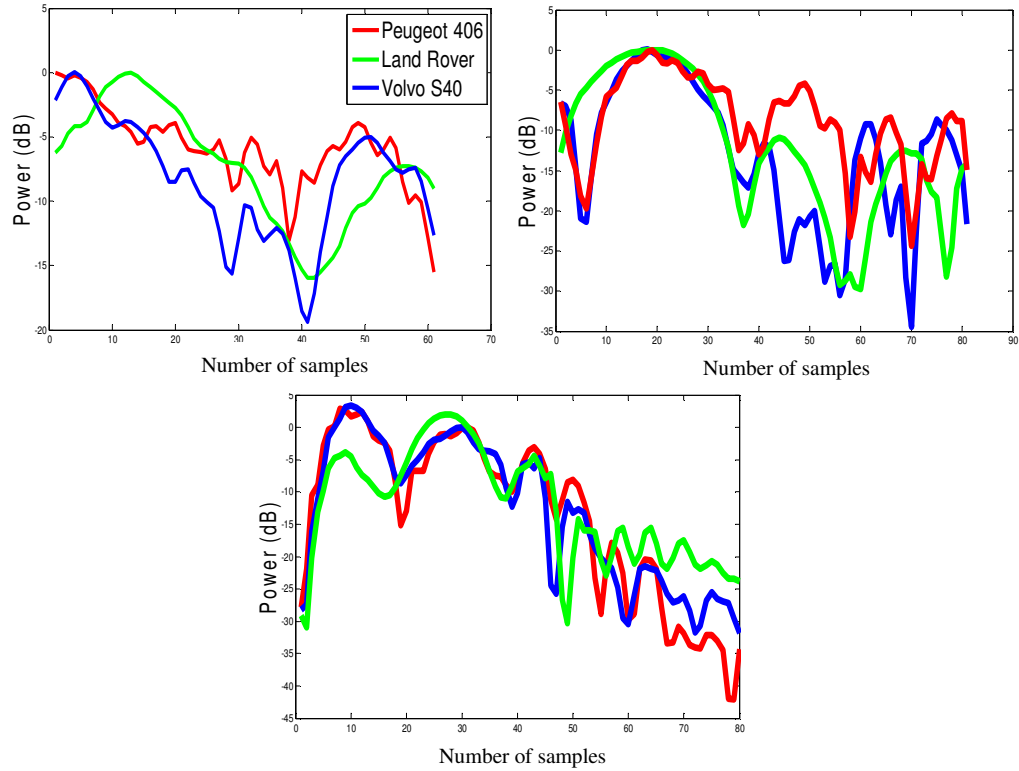


Figure 3.11 Spectra for Peugeot 406 (red), Volvo S40 (blue) and Land Rover (green) at various frequencies

Figure 3.12 shows the plots of all training data in the principal component space, for all three cars in all frequencies (red circles - Peugeot 406, blue squares - Volvo S40 and green triangles - Land Rover). A good separation between different types of cars can be seen in all frequencies, even though the optical approximation does not hold, at least not at 64 and 151 MHz frequencies. Moreover, 151 MHz seems to provide the best results. This result is in contrast to the usual optical presentation of forward scattering nature which requires the system wavelength to be shorter than characteristic dimensions of the target shape for the reliable target recognition. This can be explained by the presence of clutter, which is stronger

at 434 MHz than at 64 and 151 MHz. Furthermore, 151 MHz provides better results than 64 MHz because it gives more detailed frequency information regarding the target spectra.

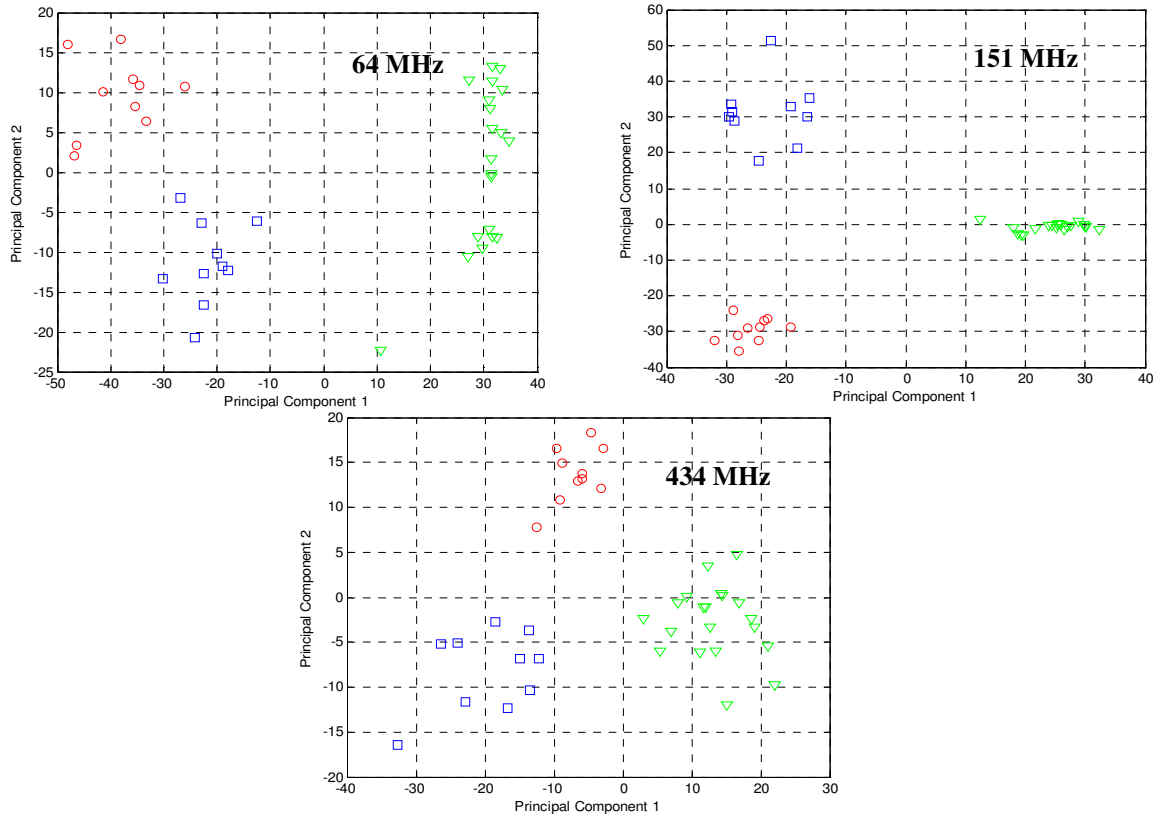


Figure 3.12 The location of data for each vehicle in the 2 components in the PCA

The above result demonstrates that we can still distinguish different target types at a low frequency.

3.4.2 Effect of different positions of sensors

Sensors are placed as in Figure 3.14 and Figure 3.15 with a baseline distance of 50m at three different positions:

- a) Position 1: original position
- b) Position 2: 5 m from position 1
- c) Position 3: 10 m from position 1

It is beneficial to repeat the same experiment in different places in order to investigate and confirm the effect of the environment and the ground reflection on the target spectrum using the same target, Land Rover. Our experiments were carried out in both Tilesford airfield and Pritchatts car park. In both places, the sensors were placed on a concrete road, but with different surroundings and environment. At the car park, the receiver was placed near a row of big trees where the branches and leaves swayed with the wind, whereas there were no big trees nearby at the airfield, only grass. Five signals were collected for each position of sensors.

Spectra in Figure 3.16 and Figure 3.17 show the effect of the different sensor positions. At the airfield, the shape of the spectrum for all positions is pretty much the same for all frequencies, although at 64 MHz the spectra is slightly expended and at 434 MHz the amplitude of the spectra is a bit distorted. At the car park, we can see bigger fluctuation in the amplitude of the spectra as their shapes differ from each other. This is probably because of vegetation and reflected signals from the nearby trees and needs to be confirmed through more analysis.

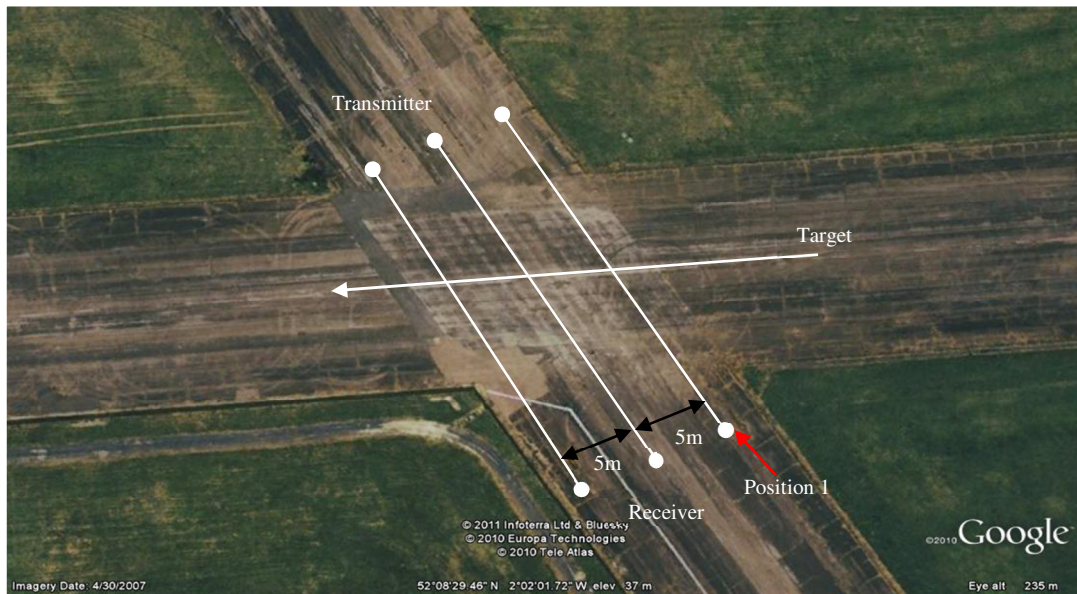
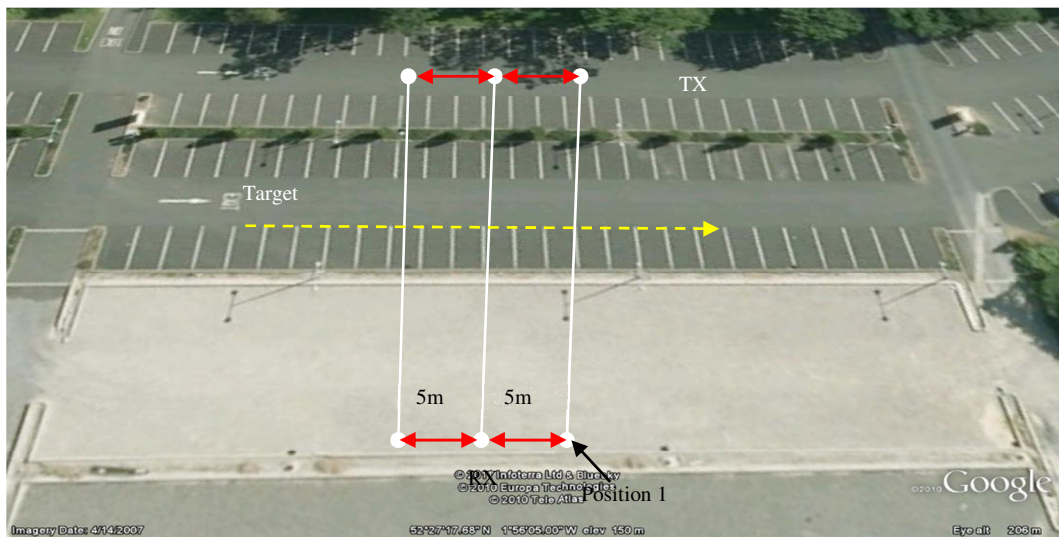
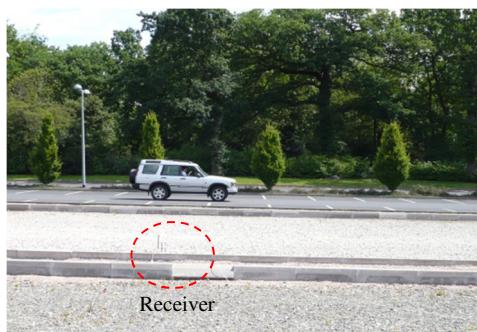


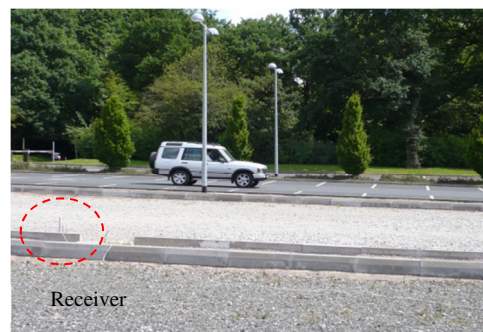
Figure 3.14 Sensor positions at Tilesford Airfield



(a)



(b)



(c)

Figure 3.15 Pritcharts car park (a) experimental layout (b) sensors moved 5m from original position and (c) sensors moved 10m from original position

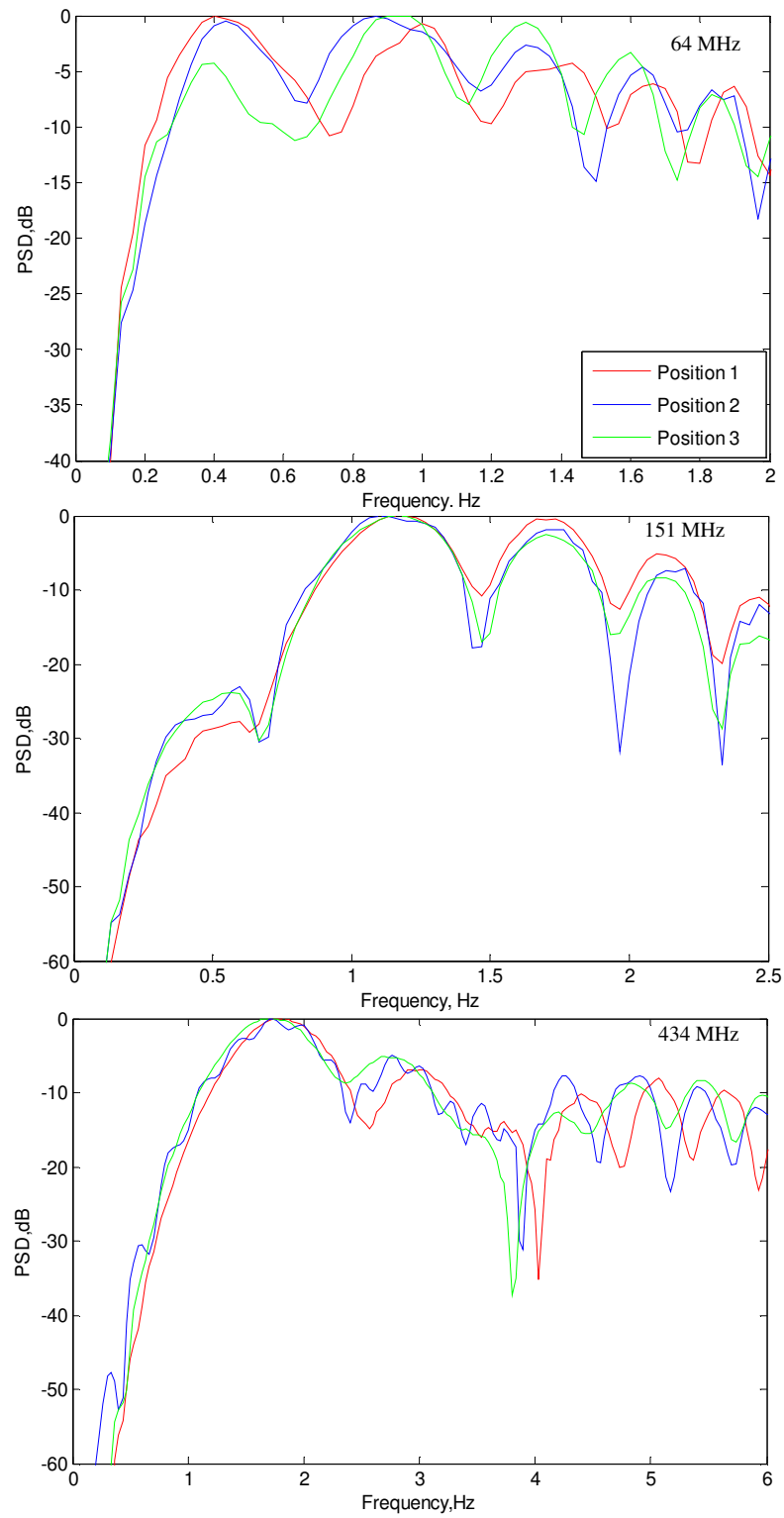


Figure 3.16 Spectra for different positions at Tilesford airfield

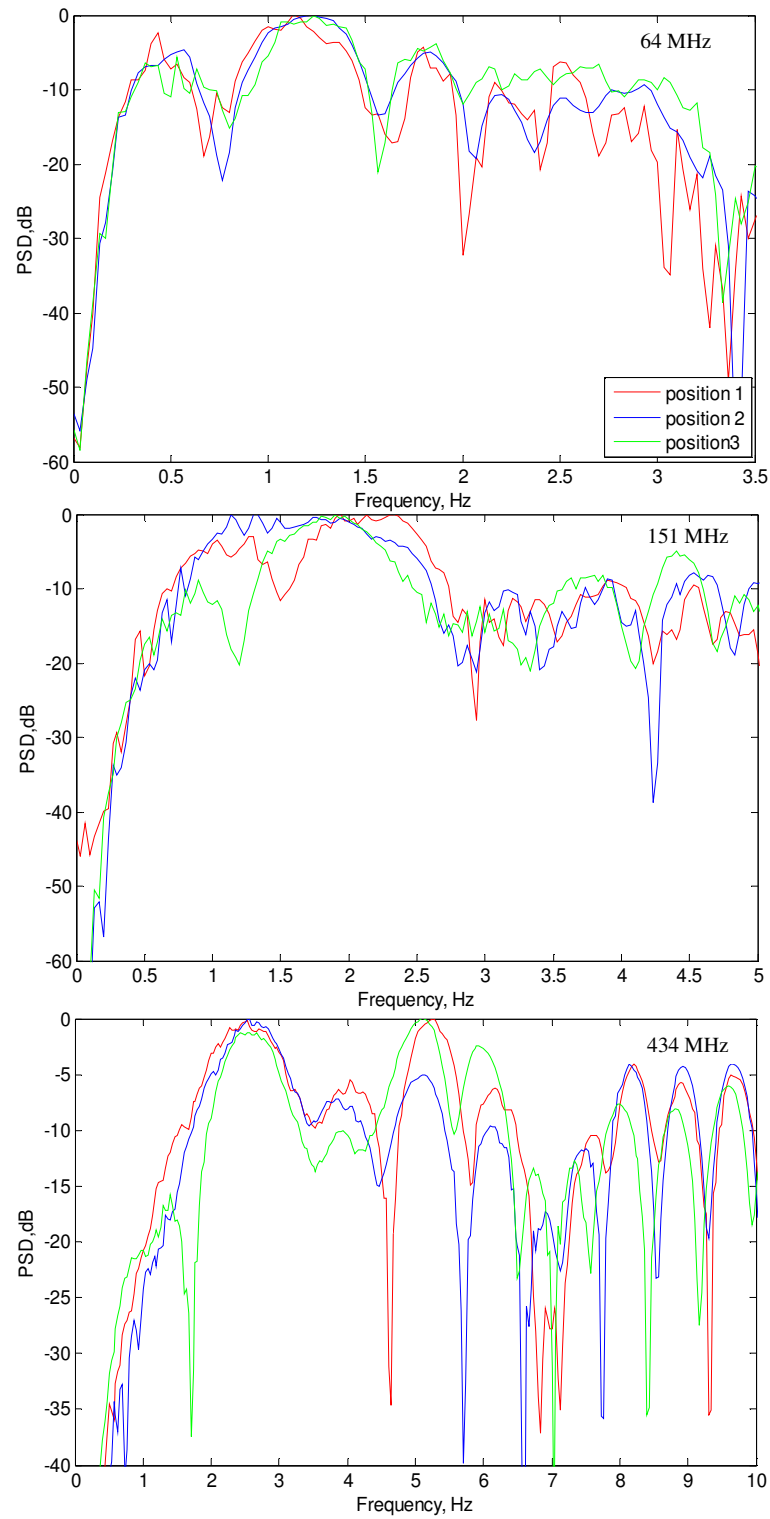


Figure 3.17 Spectra for different positions at Pritchatts car park

3.4.3 Effect of different baseline lengths

Figure 3.18 illustrates the topology of different baseline lengths. The transmitter and receiver locations are denoted by TX and RX , respectively. The point O is the origin of the co-ordinate system and denotes the point where the target crosses the baseline. The distance between O and the transmitter is d_T , and d_R is the distance between O and the receiver. The target moves with velocity, v and its motion direction is specified by the angle, ϕ . In this case, let us assume that the target moves in a direction normal to the baseline, $\phi = 90^\circ$.

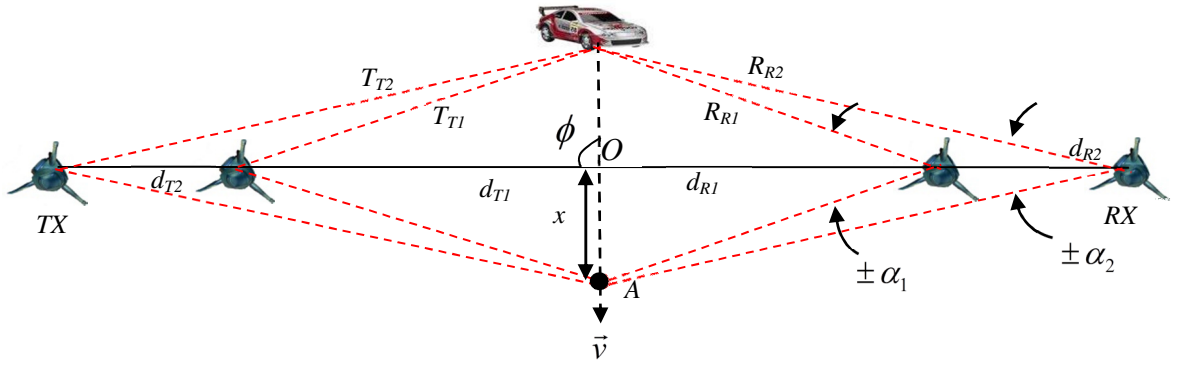


Figure 3.18 Topology for different baseline lengths

If the distance between point O and the RX is given by d_{R1} , x is the distance between point O and point A , t is the time at point A and v is the velocity of the target, thus the angle of arrival, α_1 can be defined as $\alpha_1 = \tan^{-1}(x/d_{R1})$ where $x = v \times t$. We know that the Doppler frequency correlates to the velocity and the angle of arrival.

In order to see the influence of different baseline lengths (50m and 100m) on the target spectrum, signals are simulated using MATLAB at a carrier frequency of 151 MHz with the

same velocity and target dimension (rectangular plate with the Land Rover's dimensions). As shown in Figure 3.19 (after amplitude normalisation) the shapes of the spectra shows some similarity for both cases, with the exception of the spectrum for the 50m baseline which is expended. However, this simulated signal is ideal case with no vegetation nearby. Analysis using measured clutter is therefore necessary to investigate the influence of vegetation.

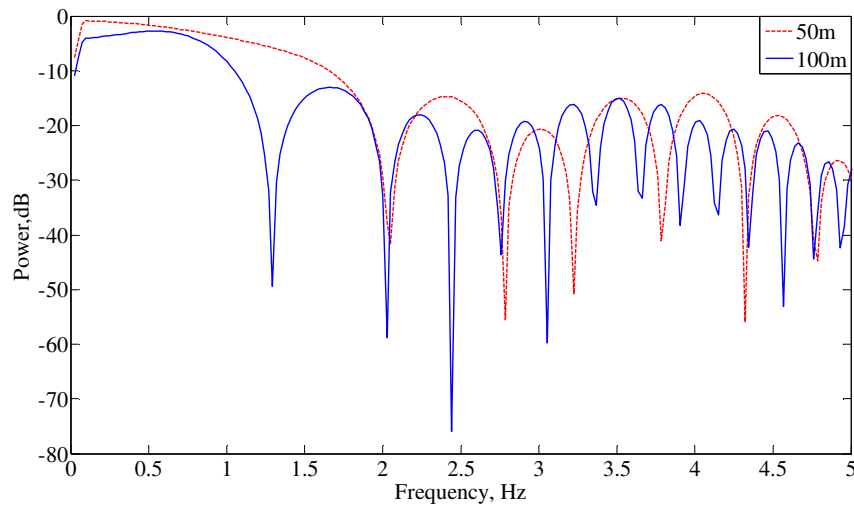


Figure 3.19 Spectra for different baseline distances

The measurement is conducted as in Figure 3.20, where the baseline lengths are set for 50m, 100m, 150m and 200m across the runway at Tilesford Airfield. The transmitter output power is set at 20dBm for 64 and 434MHz and 10dBm for 151MHz. Helical monopole (omni-directional) antennas are used for all channels. The target was crossing perpendicular in the middle of the baseline with a constant velocity at ~20mph (based on a speedometer) with around five runs for each baseline lengths.

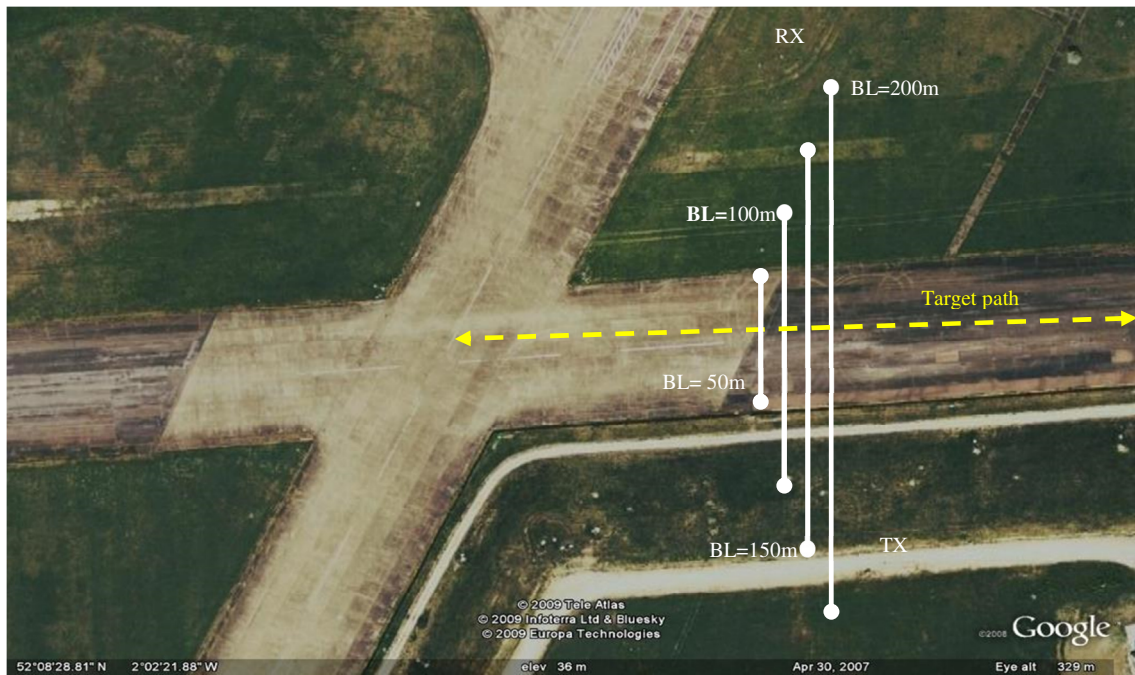


Figure 3.20 Experiment set up for different baseline distances.

Figure 3.21 and Figure 3.22 show an example of a time domain signal at a carrier frequency of 64 MHz and a normalised Doppler spectrum for all frequencies, respectively. The time domain received signals for each baseline distance at 64 MHz look undistinguishable. The spectra suggest that as the length of the baseline increases, the main lobe width of the spectrum decreases and the spectra become more sensitive to the presence of clutter.

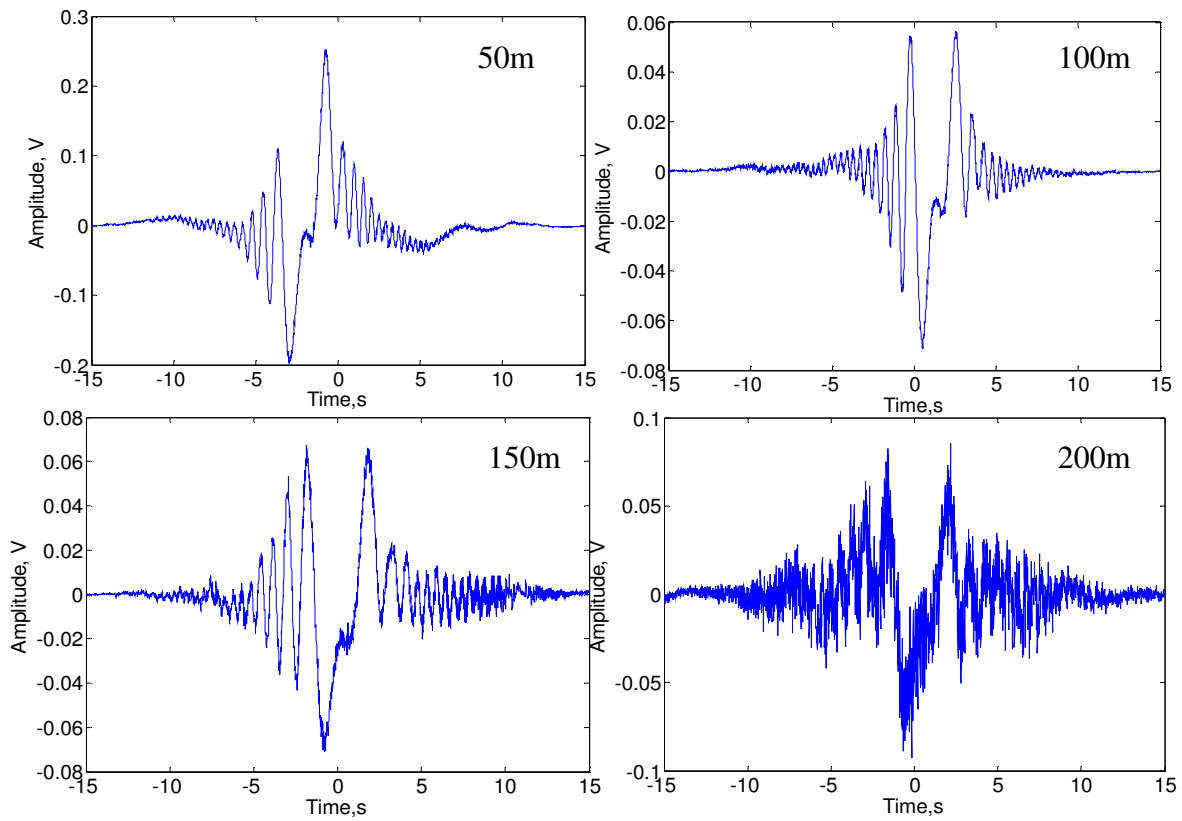


Figure 3.21 Time domain signal for all baseline distances at 64 MHz

A larger database might be needed in order to accommodate all different baseline distance signals (even for a same car) from the simulation and experimental results, with the shifting of the lobes and nulls of target spectra, or the classification accuracy will decrease. However, we are able to reduce this influence by normalising the shifted nulls to a reference baseline. Further discussion on how to normalise the baseline can be found in Chapter 5.

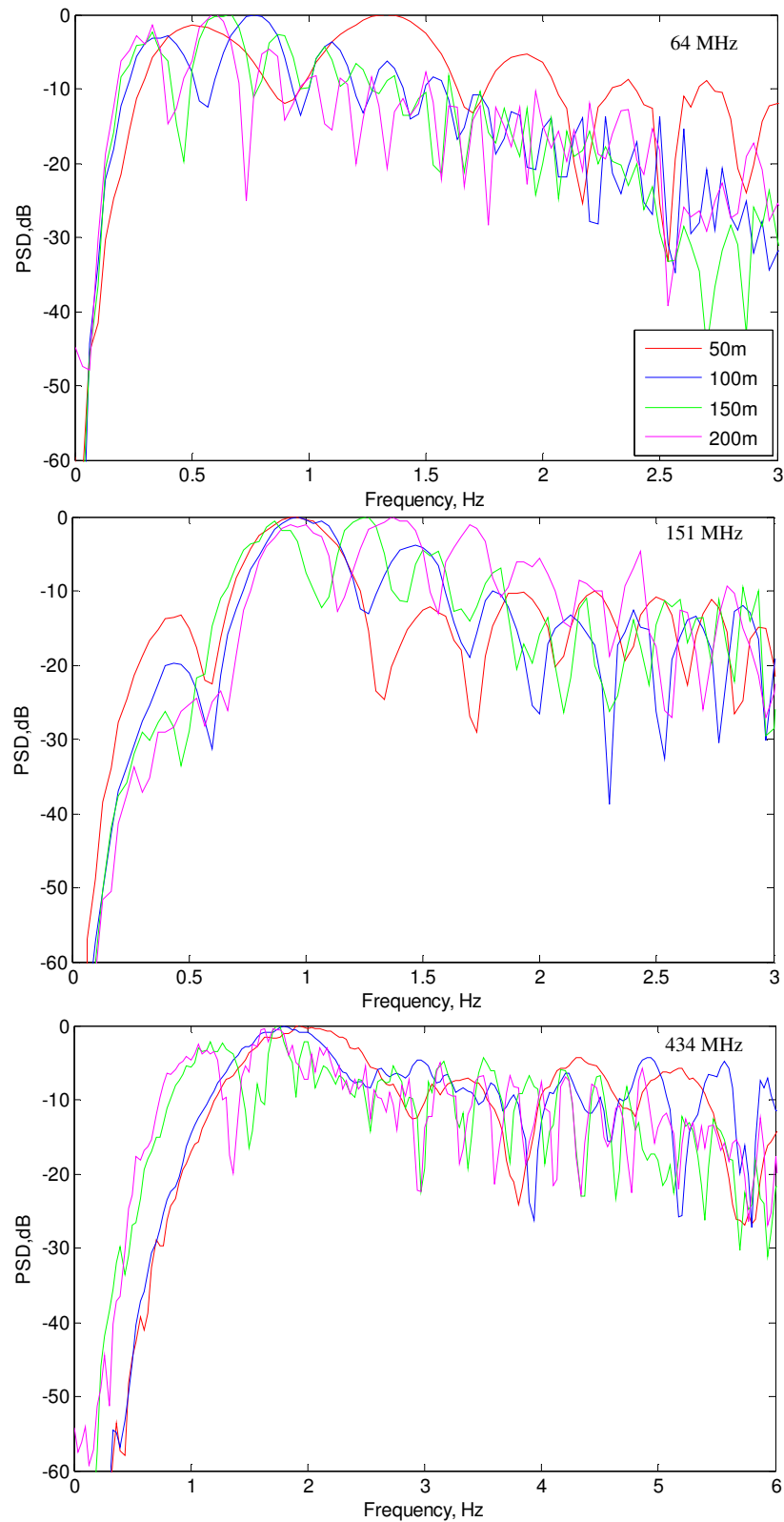


Figure 3.22 Experimental received spectrum for four different baseline lengths, BL

3.4.4 Effect of different crossing points

The layout of different crossing points is illustrated in Figure 3.23. The baseline length was 100m and the target was crossing perpendicularly to the baseline at every 10% distance from the receiver (until the middle of the baseline) with a velocity of around $v = 6 \text{ m/s}$. Land Rover was used in this measurement and five signals were collected for different crossing points.



Figure 3.23 Experiment layout for different crossing points

Figure 3.24 depicts the spectra for different baseline lengths. For each frequency, the shapes of the spectra are comparable to each other but the width of the main lobes is dissimilar especially when the target is crossing at 10m from the receiver. When a target is crossing near the receiver, the presence of the target blocks a large amount of the signal waveform from the receiver compared to when the target crossing point is far from the receiver. However, the effect of different crossing point is subject to further investigation.

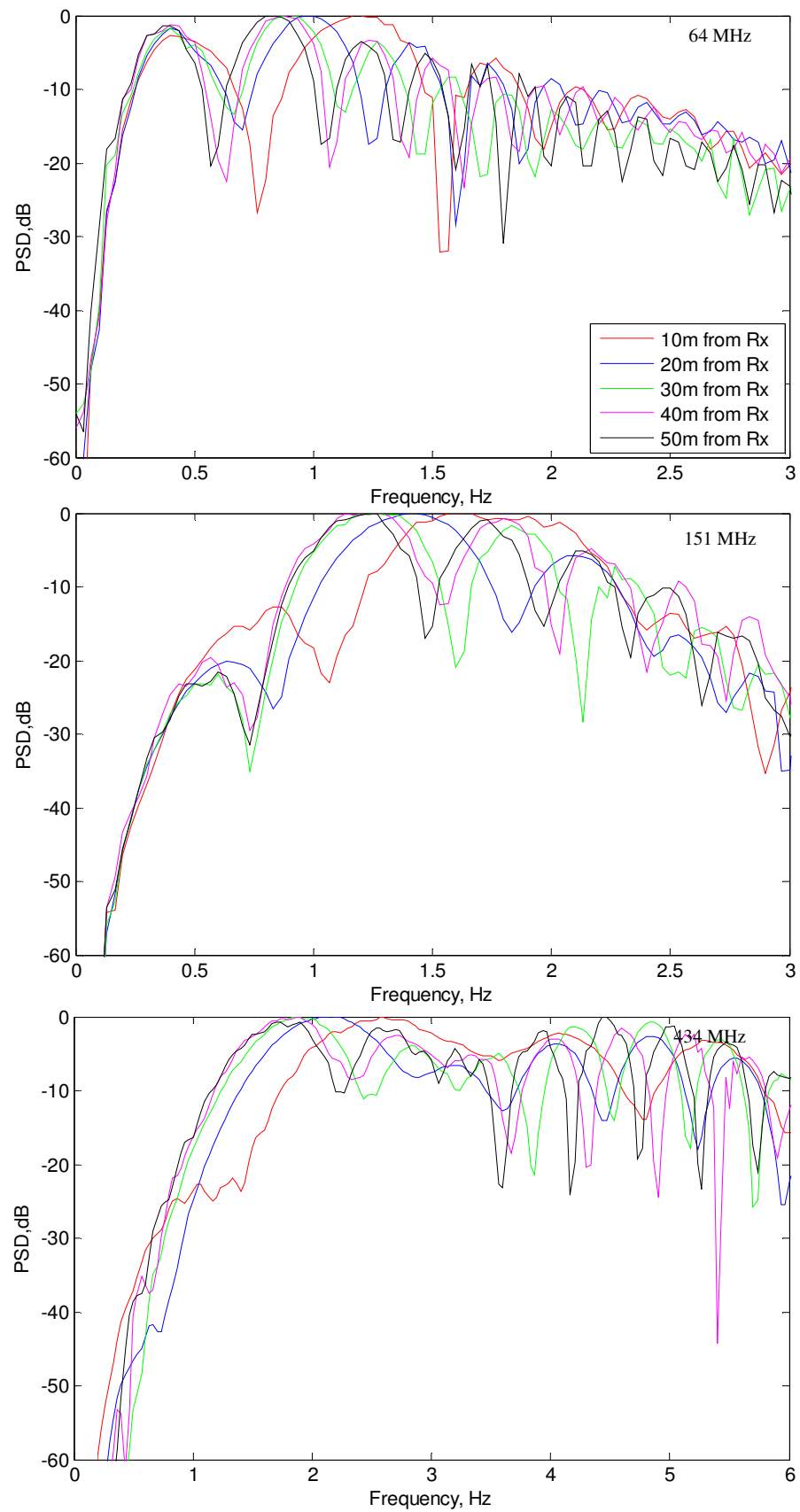


Figure 3.24 Spectra for different crossing points

3.4.5 Effect of different trajectory angles

As mentioned earlier (in Chapter 2), in forward scatter the shadow field occurs when the target partly blocks the signal wave front from the transmitter. The pattern of the shadow depends on the target silhouette and not on the target surface shape. Different target silhouette areas give different target RCS [29]. This suggests that RCS fluctuates are also dependent on the angular properties of the target and the direction in which the target is viewed [4].

It is essential to investigate the effect of small changes in target crossing angles to the target RCS in real conditions. Hence the experimentation was carried out in an open space at Pritchatts car park using a set-up illustrated in Figure 3.25. The black dots represent the positions of the sensors with a baseline of 50m. Four different trajectory angles to the transmitter-receiver baseline (30° , 45° , 75° and 90°) are analysed. Due to space limitation, instead of the target crossing five times at the same baseline and only changing its direction, the sensors were moved to different positions, but the required crossing angle from the baseline was maintained. The target signal might therefore be affected by different sensor positions as in section 3.4.2.

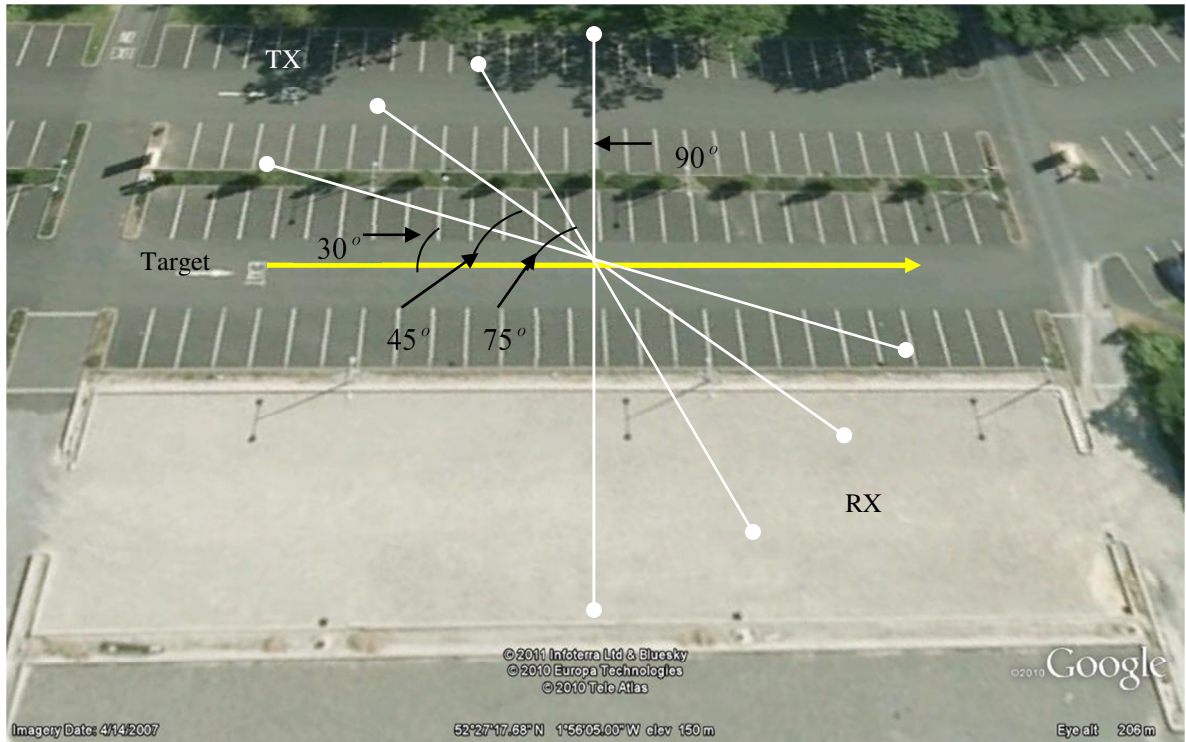


Figure 3.25 Experiment layout for different crossing angles

Figure 3.26 shows the target spectra for four different trajectory angles. We can see from the figure that the differences between spectra for different crossing angles become more visible as the target crossing is closer to the baseline (with a smaller crossing angle). As the frequency increases, the spectra become more distorted. A small percentage of the differences might be due to the effect of the different sensor positions. However, more investigation needs to be conducted in order to confirm this.

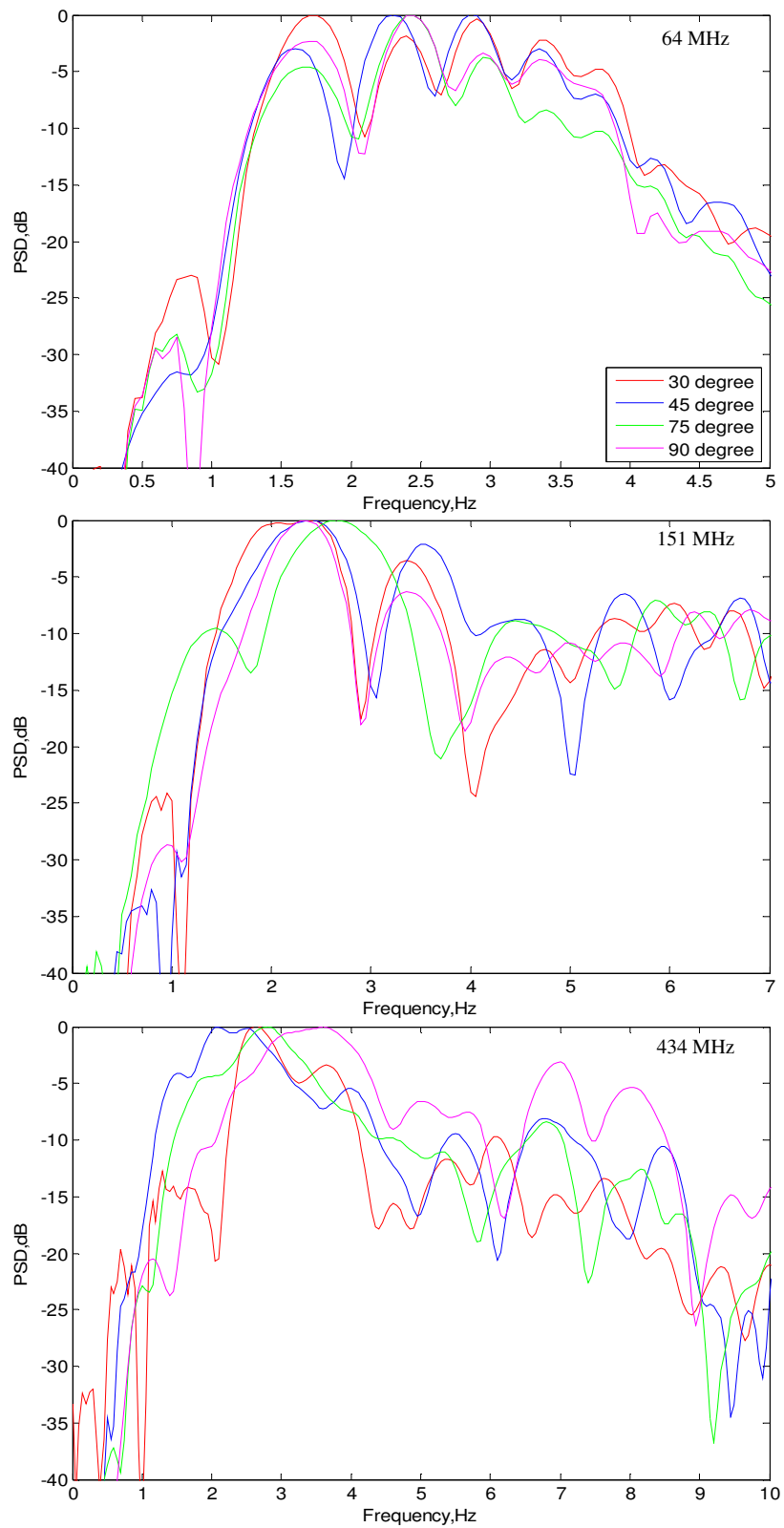


Figure 3.26 Spectra for different crossing angles at various frequencies

3.4.6 Effect of obstacles

The signals were recorded using the same car (a Land Rover) with four different obstacle settings at Pritchatts car park as seen in Figure 3.27

- a) No obstacle between the baseline
- b) A Ford car 5m from the baseline
- c) A Ford car in between the transmitter and the receiver
- d) A Ford car 10m from the baseline

The aim of this analysis was to determine if an obstacle can create a multipath effect and thus affect the signal. Again for each setting, the measurements were repeated for five times.

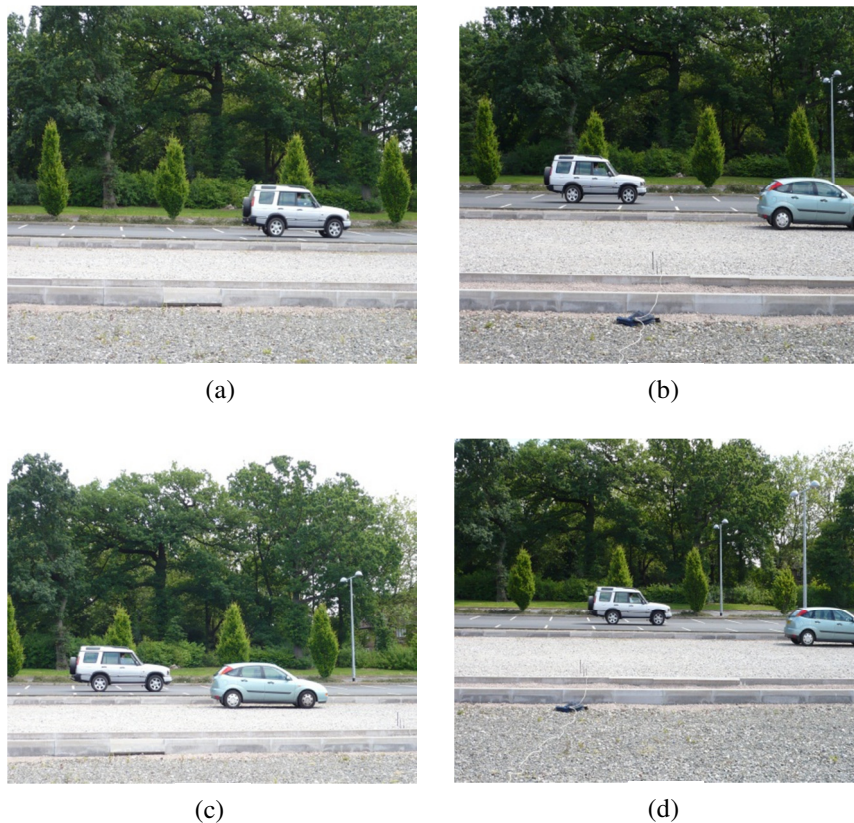


Figure 3.27 Experiment layout for different obstacle positions

The combination of the Doppler spectra for four different obstacle positions is shown in Figure 3.28. The results clearly demonstrate that the spectra have a similar shape in all cases, although compared to conditions with no obstacles around; the side lobes are significantly unmatched. It is likely that there is some influence from the reflections of the obstacle. However, the stability and consistency of the target spectra show a high degree of similarity.

The illustrations in Figure 3.29 are the PCA plots for different obstacle positions. The PCA points are concentrated in one area in all frequencies, especially at a carrier frequency of 151 MHz. The scattering of the points in 434 MHz are probably due to the effect of clutter and multipath fading, but this subject needs further analysis.

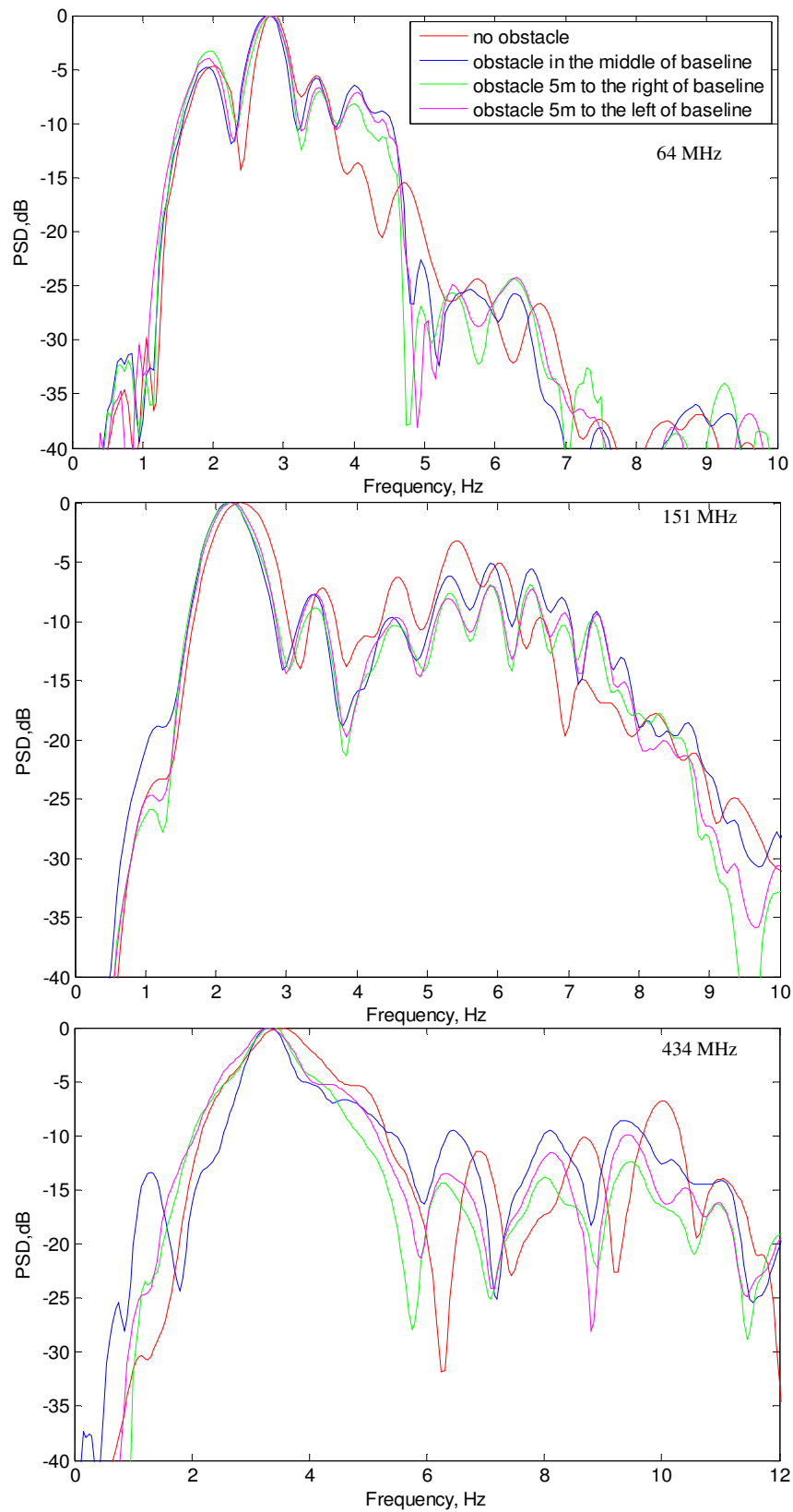


Figure 3.28 Spectra for different obstacle positions

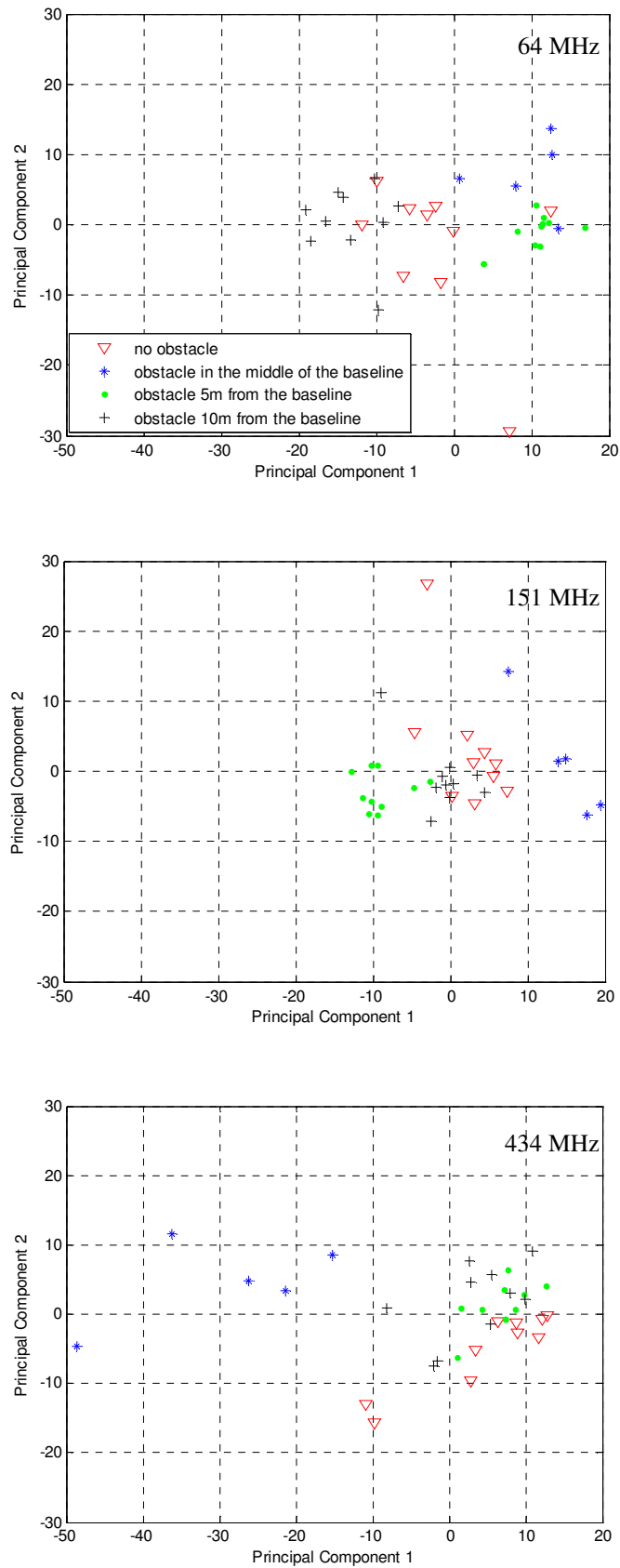


Figure 3.29 The location of data for the different obstacle positions in the 2D PCA plots

3.5 Conclusion

In this chapter we described the hardware used for the experimental set-up and the data collection process. We collected a reasonably large database which is more than 200 signals (~50 runs for each car) using four types of cars (Land Rover, Ford Focus, BWM 316i and Nissan). The signals were collected under ideal condition where minimum amount of clutter present, car speed is around 20mph and the cars were crossing perpendicular to the middle of the baseline. For preliminary analyses, only the Land Rover was used and the measurements were repeated five times for each trajectory setting. Again, ideal condition was assumed in all cases.

In section 3.4, we presented the preliminary results, including the probability of classifying targets using a low frequency; the influence of different target trajectories and the influence of the sensor positions and obstacles to target spectra. The results suggest that we can both distinguish and classify targets even at low frequencies. It is also clear that there are some effects from different target trajectories to the spectra of the target.

Chapter 4

CLUTTER SIMULATION

4.1 Introduction

In ground-based FSR, the target signal is embedded into a strong background signal caused by interference and ground clutter. Research dedicated to clutter has been conducted in order to define its characteristics and features and to develop a simulation model of clutter for different conditions [34, 38, 45, 60-62].

In this chapter we will present a brief description of ground clutter including its types and sources. We will also explain the practical procedure of simulating clutter (stationary or non-stationary) in detail and discuss the results.

4.2 Foliage clutter

Noise is a random and unwanted signal. It can be generated either from an internal or an external source. When the source is internal, the noise is generated from electronic equipment while an external noise source is normally the natural thermal radiation produced by the area surrounding the target. Noise is the limiting factor of systems which use a low signal level, such as the weakly received echo-signal in a radar receiver, where it is difficult to distinguish the signal from the noise.

In our case of ground-based FSR, such noise signals are significantly lower than the signals from the target, which means that the main limiting factors of such systems are clutter or/and interference distorting the signal from the target.

In our case, clutter refers to the unwanted radio frequency echoes returned from surrounding objects. It could be vegetation and precipitation such as rain, snow or hail. Multipath interference from buildings, the ground and the sea are also considered. For ground applications, the main source of clutter is vegetation. The movement of foliage and branches blowing in the wind results in the presence of strong multipath components in the received signal. Therefore the clutter signal is dependent on conditions such as wind strength and the season of the year, since trees generally have more foliage in summer. This leads to seasonal variability in their effect. More clutter means that their masking effect with regard to the target signal is greater, and may result in more significant errors in target classification.

In a FS micro-sensor network, the sensors are positioned on the ground with a baseline length of hundreds of metres and have omni-directional antennas with a very low elevation. If the

sensors are surrounded by vegetation, the foliage and the branches swaying with the wind can be considered as volume distributed scatterers. The use of non-directional antennas and the absence of range resolution cause the clutter to collect from a larger volume around the FSR position.

These different aspects of vegetation clutter merit further investigation. An approach which uses foliage modelling has been suggested in [38]. In this chapter we present the analytical approach for simulation of non-stationary clutter as formulated in [45]. Simulated clutter should demonstrate the characteristics of real measured clutter obtained as a result of statistical analysis. Later, simulated clutter will be compared with real measured clutter to show the accuracy of the modelling.

4.2.1 Non-stationary Clutter

In order to develop a simulation model for vegetation clutter, we need to investigate and determine the characteristic parameters of non-stationary clutter by analysing the measured clutter. A large number of clutter signals were recorded at a number of different sites with terrain profiles varying from a concrete runway to a dense wood⁶. The signals were recorded for more than 20 minutes with a baseline of 50-100 m in order to see the variation in the clutter properties.

⁶ The work was conducted for project DTC 2-65 by Dr Marina Gashinova and PhD student, Nor Ayu Zalina

The properties of clutter can be found by estimating the signal power spectral density, time variation of average clutter power, wind speed and direction, and the distribution of clutter amplitudes [45].

An example of recorded clutter signals made in very strong wind conditions (15-25km/h) is presented in Figure 4.1. Figure 4.2 shows the clutter power spectral density corresponding to different carrier frequencies. We can see that the clutter spectrum is independent from the radar carrier frequency (from 64 to 434 MHz) and corresponds approximately to a predicted decay of 40 dB/decade. The clutter bandwidth is defined by a 10 dB power drop and is 0.3-0.5 Hz with a slight increase depending on the radar carrier frequency.

The clutter power spectral density (PSD) decrement is proportional to the increment of the carrier frequency. The total power increases in about the fourth power of the carrier frequency with a difference of approximately 5dB between 64 MHz and 135 MHz and 15 dB between 173 MHz and 434 MHz.

At 64 MHz (lowest frequency) the clutter power is very small. This signal can be defined as receiver thermal (in situations where there is no wind) and environment noise with a probability density function close to Rayleigh. It therefore reflects mostly the Gaussian nature of clutter at a low frequency. For higher frequencies clutter is defined by wind distributed vegetation and becomes a Weibull distributed and partly correlated [16].

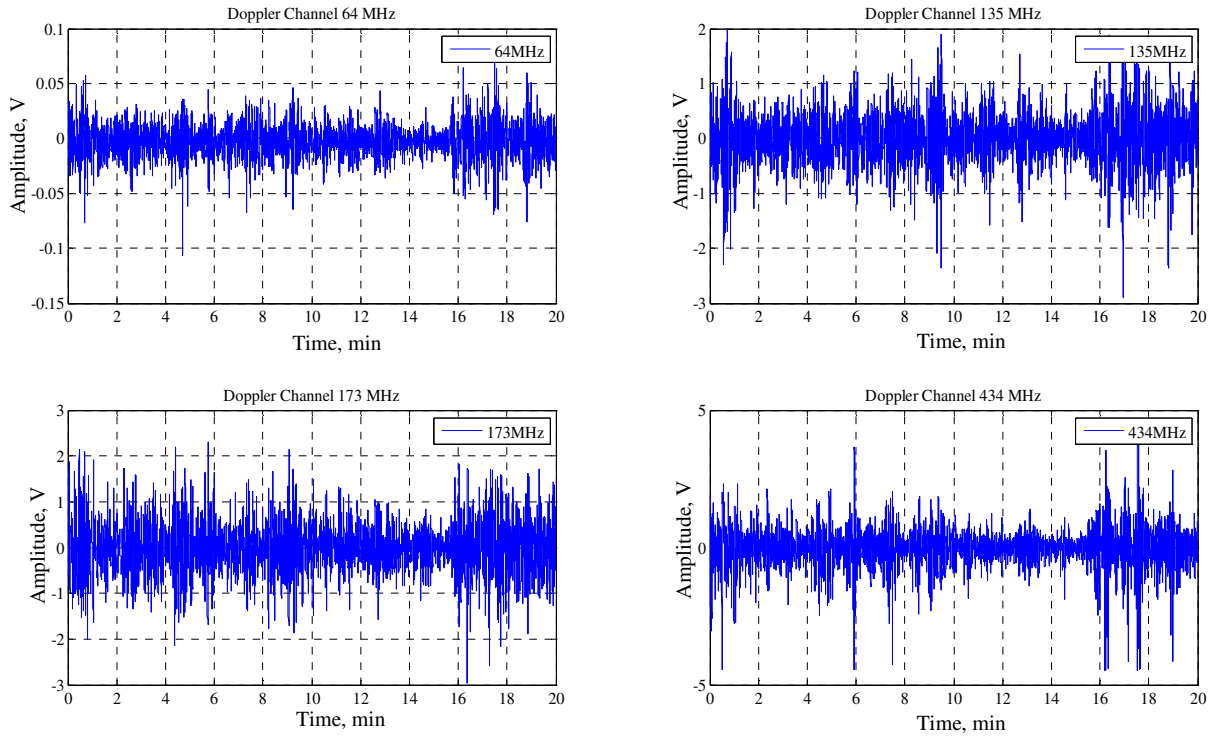


Figure 4.1 Measured clutter

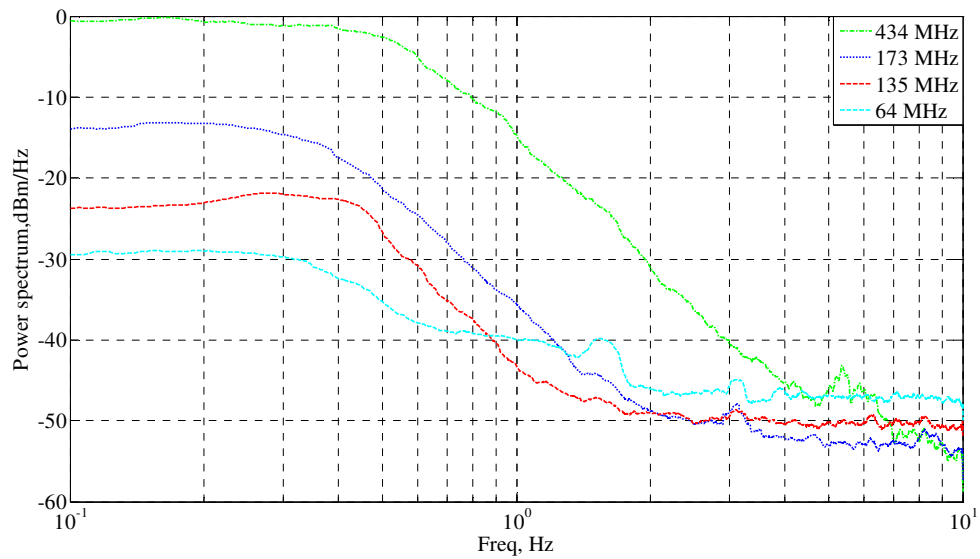


Figure 4.2 Power spectrum density of measured clutter

Figure 4.3 demonstrates the envelopes of the measured clutter. The envelopes mainly replicate each other, which indicate that the clutter is from the same source for each different carrier frequency. Again, this indicates that the amplitude of the envelope increases with the increment of the carrier frequency.

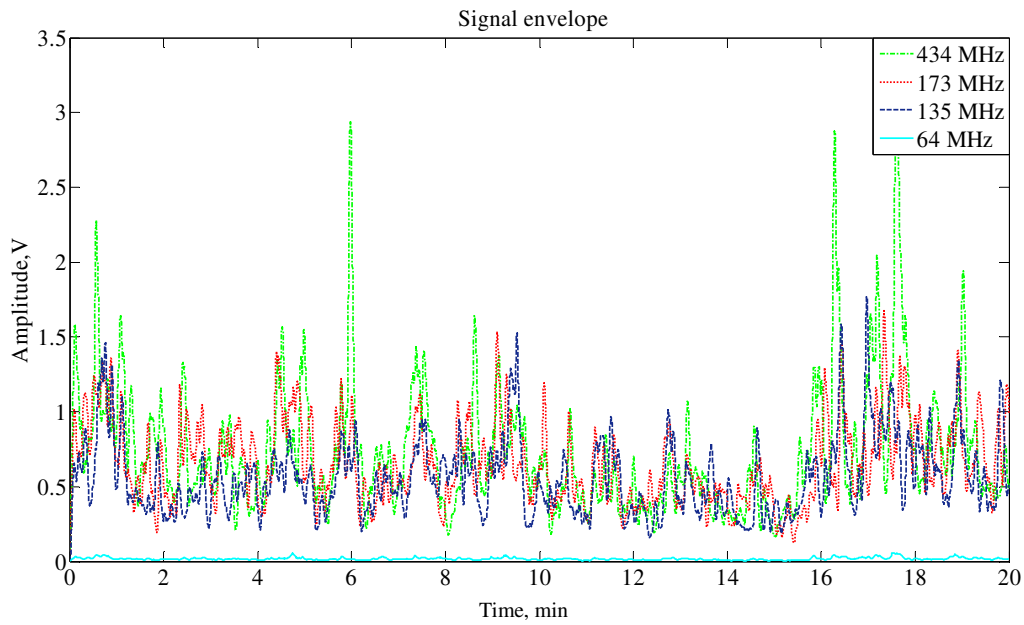


Figure 4.3 Clutter envelopes

Non-stationary clutter can be analysed in terms of the relative power homogeneity period where the clutter is stationary within a certain length of time [45]. The period of power homogeneity is roughly defined as a length of a signal section where the deviation of RMS subsections within the period is less than 20% [45]. By analysing measured clutter signal where the clutter strength is comparatively very strong, we found that the spectrum of the clutter envelope has a maximum frequency of 0.01-0.02 Hz, which corresponds to 50-100 seconds of relative power homogeneity. This can be seen in Figure 4.4.

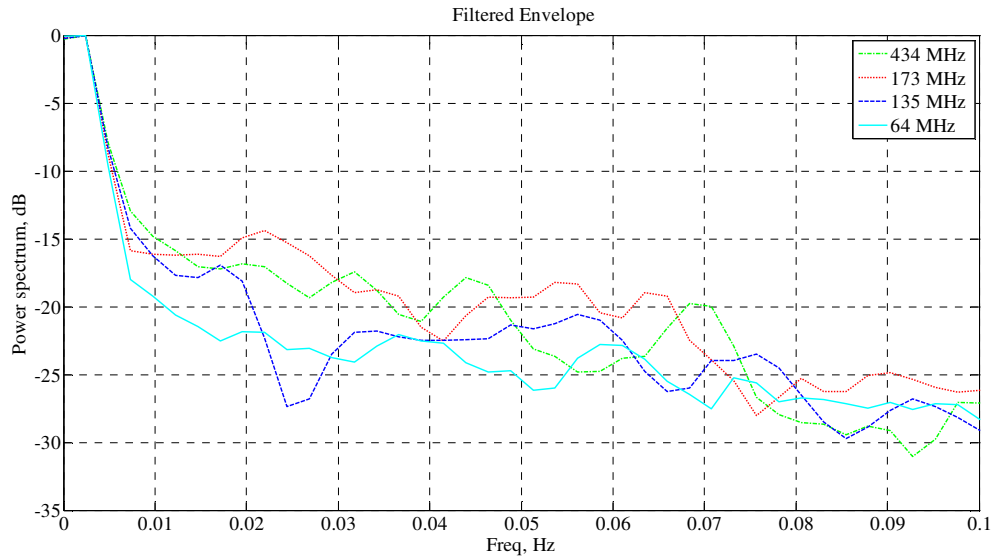


Figure 4.4 PSD envelope

To estimate the radar performance it is essential to quantify the radar clutter amplitude distribution. Several PDFs have been used to describe the clutter distributions. Among these functions are lognormal distribution, Weibull distribution, the general chi-squared distribution, and Rayleigh distribution [63]. Work undertaken for environment clutter in [45] demonstrated a certain invariance in the PSD. The period of clutter power variance and its PDF have been found close to Weibull distribution with shape factor 1.1-2.

In Figure 4.5 the amplitude distribution of signals presented in Figure 4.1 and the Weibull PDF fit is shown. At the 64 MHz channel, the clutter record has a Weibull distribution with a shape factor of 1.7. As the frequency increases, the degree of similarity to the Weibull PDF lessens and the shape factor reduces from 1.7 for 64 MHz to 1.6 for 434 MHz channel. The lower the wind, the closer the distribution to a Rayleigh distribution, inherent to normally distributed thermal noise [45].

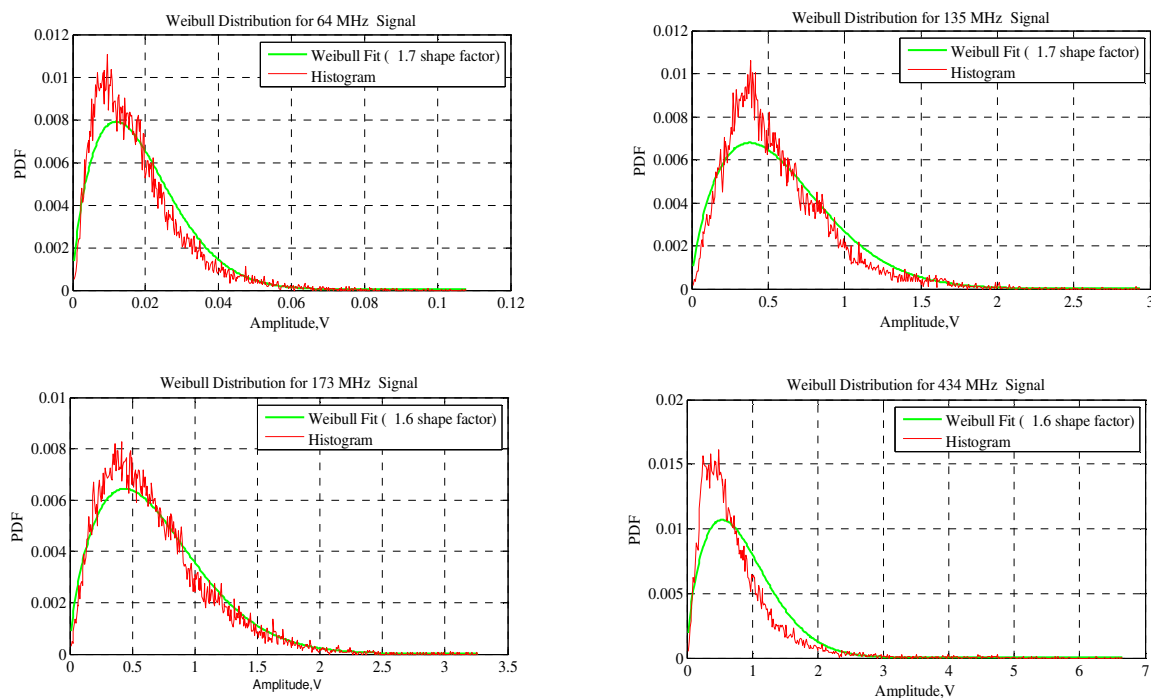
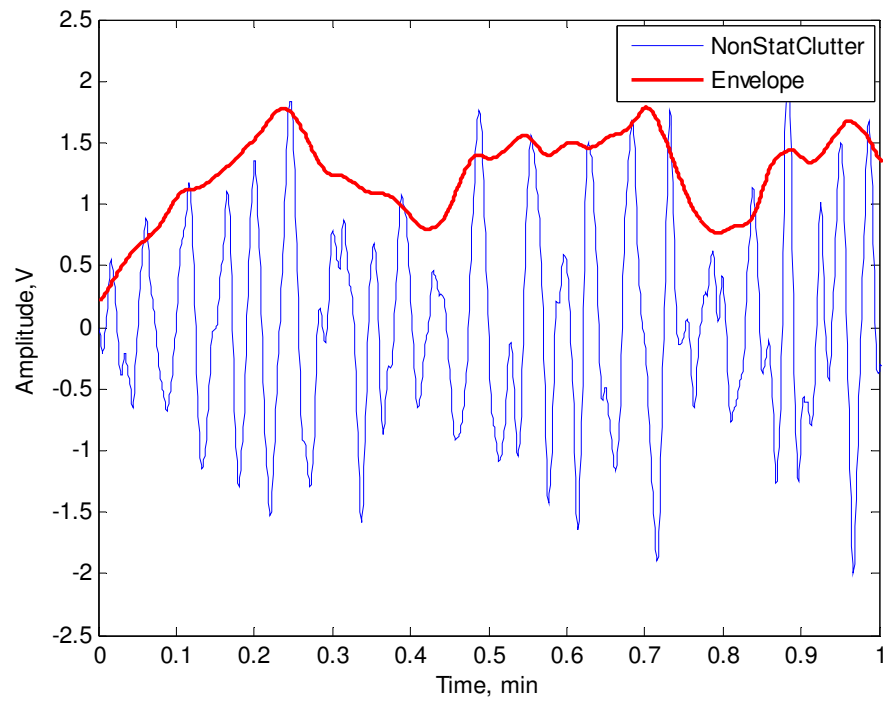
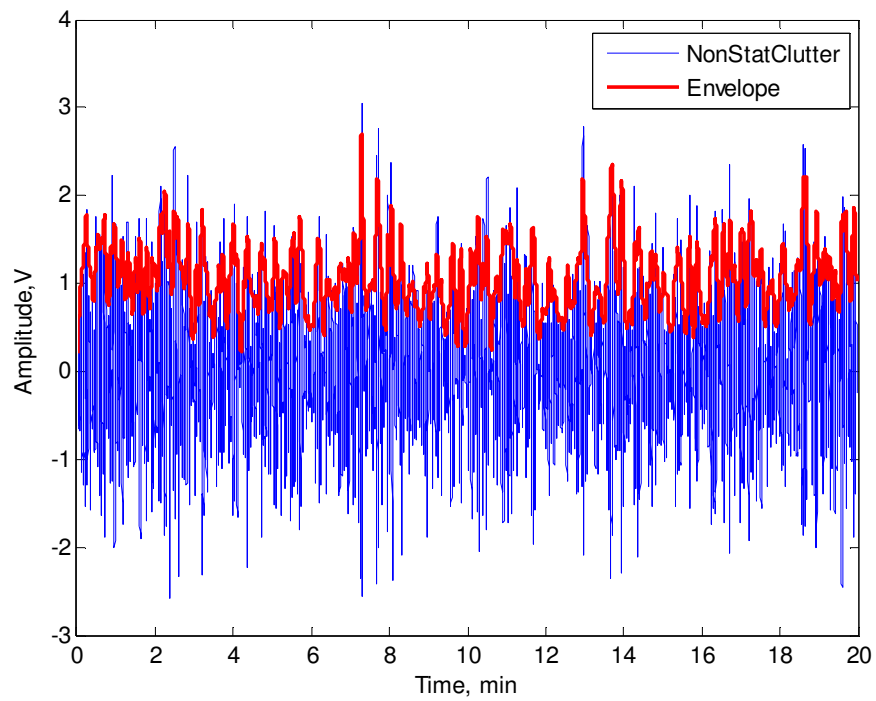


Figure 4.5 Distribution of measured clutter at 64 MHz, 135 MHz, 173 MHz and 434 MHz channels

For land clutter the amplitude is almost constant within short periods and visibly non-stationary within longer periods. This is shown in Figure 4.6 (a) and Figure 4.6 (b), respectively. When we compare the two signals, we see that in the short-period signal the amplitude of the signal hardly varies, while for the longer periods we can clearly see fluctuations in the signal. In our analysis we will therefore focus on how to simulate non-stationary clutter from a random noise.



(a)



(b)

Figure 4.6 Clutter (a) short period (1 min) and (b) long period (20 min)

4.3 Non-stationary Clutter Simulation

The overall clutter simulation block-diagram is depicted in Figure 4.7. In general, the simulation comprises of two phases: the generation of coloured noise and the modulation process.

In order to generate stationary clutter, a so-called ‘coloured Gaussian noise’ is generated by passing a set of random Arbitrary White Gaussian noises (AWGN) through a LPF. The power of the generated stationary clutter is then defined from the measured clutter power, and a cut-off frequency of a low-pass filter is defined by the bandwidth of the measured clutter which typically is around 0.25-1Hz.

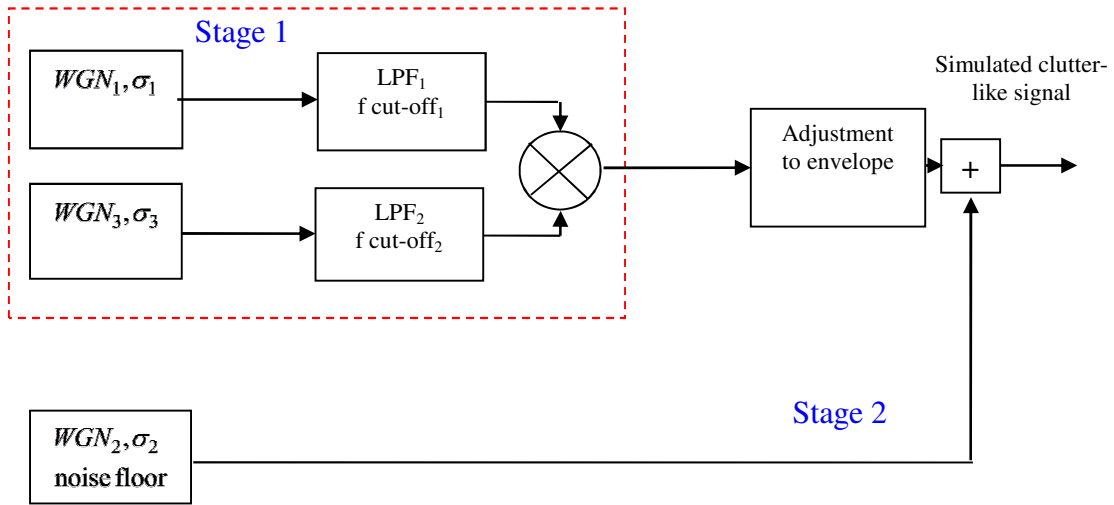


Figure 4.7 Clutter simulation model block-diagrams

For higher frequencies and in stronger wind conditions, clutter becomes mainly non-stationary. Thus, the procedure to generate clutter should include a modulation of stationary clutter which implies that coloured Gaussian noise must be time-variant according to a period of variance which is found in the measured non-stationary clutter. Modulation requires the

generation of another random Gaussian noise signal, passing it through a low-pass filter with a cut-off frequency of 0.001-0.05 Hz and scaled to the measured envelope of the clutter. This modulation procedure is in conformity with the measured envelope and makes the clutter both non-Gaussian and non-stationary. Later, a noise floor is to be added.

For the measurements in this chapter, the clutter signal was simulated for 20 minutes with a sampling frequency of 20Hz. For each frequency, the clutter conditions were found to have their own characteristics. These characteristics⁷ are based on real measured clutter and are presented in Table 4.1 . Practically, at 64 MHz frequency, the envelope is between 0.0024-0.2694 whilst for 434 MHz it is in between 0.0137-4.2691. The standard deviation for all frequencies and different wind strengths also varies from 0.0016-0.9493. Based on this characteristic, we will concentrate on modelling very strong clutter at 434 MHz.

Table 4.1 Characteristics of real measured clutter

| Frequency, MHz | Weak clutter | | | | Medium clutter | | | | Strong clutter | | | | Very strong clutter | | | |
|-------------------|-------------------|--------------------------|----------|--------|-------------------|--------------------------|----------|--------|-------------------|--------------------------|----------|--------|---------------------|--------------------------|----------|--------|
| | std of clutter | std of noise floor | envelope | | std of clutter | std of noise floor | envelope | | std of clutter | std of noise floor | envelope | | std of clutter | std of noise floor | envelope | |
| | | | min | max | | | min | max | | | min | max | | | min | max |
| 64 | 0.0016 | 0.001 | 0.0024 | 0.0041 | 0.1446 | 0.0015 | 0.0148 | 0.0405 | 0.0247 | 0.0015 | 0.0085 | 0.0939 | 0.03658 | 0.006 | 0.0173 | 0.2694 |
| 135 | 0.0106 | 0.001 | 0.0095 | 0.0228 | 0.2518 | 0.002 | 0.0861 | 0.8593 | 0.4471 | 0.002 | 0.1312 | 1.8141 | 0.6673 | 0.003 | 0.1544 | 2.7789 |
| 173 | 0.0252 | 0.001 | 0.0118 | 0.0911 | 0.3068 | 0.0025 | 0.0912 | 1.0618 | 0.5536 | 0.0025 | 0.143 | 2.4551 | 0.7259 | 0.003 | 0.1533 | 3.0868 |
| 434 | 0.0447 | 0.001 | 0.0137 | 0.152 | 0.3234 | 0.003 | 0.0773 | 1.3007 | 0.6561 | 0.003 | 0.1634 | 3.2627 | 0.9493 | 0.003 | 0.0834 | 4.2691 |

⁷ The characteristics analysis was undertaken by PhD student, Nor Ayu Zalina

4.3.1 Generation of Coloured noise/stationary clutter

A set of random noise is generated with a standard deviation of 0.9493 and passed through both a high-pass and a low-pass filter. Orders and cut-off frequencies of the Butterworth filter are defined for each frequency channel and for different strengths of clutter. In most cases the filters are second/third orders with cut-off frequencies of 0.4-0.5 Hz. In this chapter, we used a second order filter with a cut-off frequency of 0.5Hz. Figure 4.8 shows the process of generating stationary clutter and Figure 4.9 portrays a simulated stationary clutter signal.

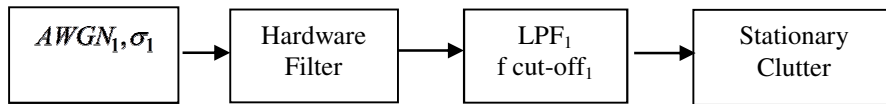


Figure 4.8 Stationary Clutter developments

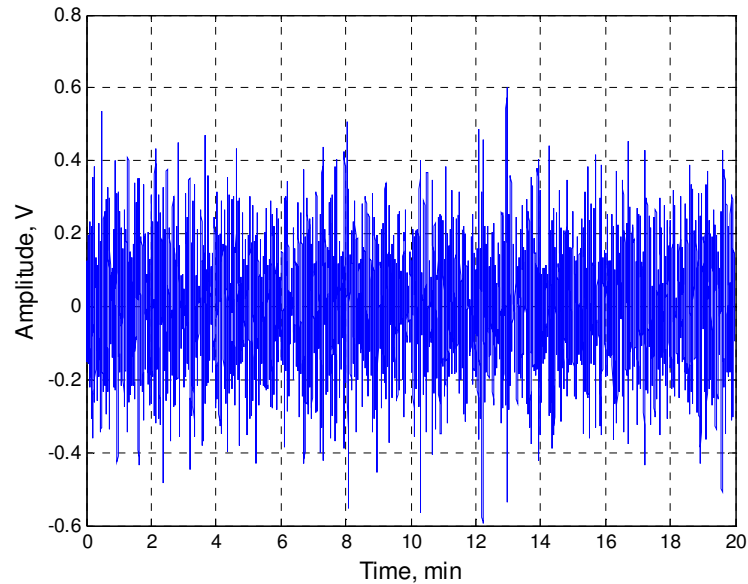


Figure 4.9 Simulated Stationary clutter

4.3.2 Envelope creation

Another set of random noise is generated using AWGN with a standard deviation of 1 used as the input of the pre-envelope process. By using a block diagram, shown in Figure 4.10, the amplitude of the envelope is calculated by employing a Hilbert transform. The Hilbert transform can be considered to be a filter by simply shifting phases by $-\pi/2$ radians for all frequency components of the input signal, a created imaginary phases [64]. The combination of the original signal and the Hilbert transformation forms a signal known as the ‘analytical signal’.

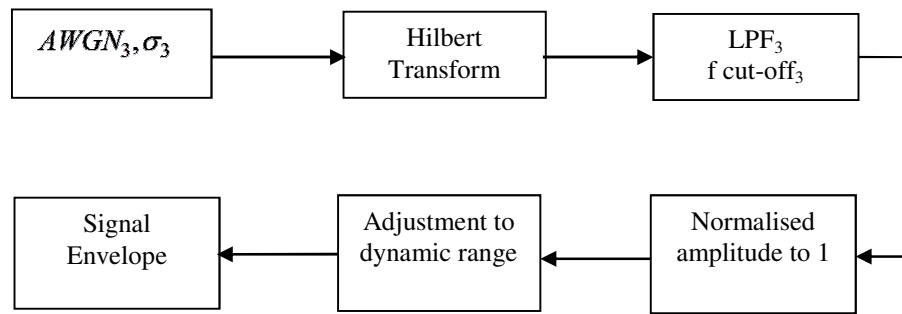


Figure 4.10 Signal pre-envelope creation block diagram

This analytical signal is a complex signal where the real part is the input signal and the imaginary part is the Hilbert transform of the input signal. It can be defined as:

$$Y(t) = y(t) * jh(t) \quad (4.1)$$

where

$$Y(t) = \text{analytical signal}$$

$y(t)$ = the real input signal

$h(t)$ = the Hilbert transform of the real input signal

The envelope of the signal can be found by taking the absolute value of the analytic signal $Y(t)$. To create an envelope of non-stationary clutter, the output is subjected to a low-pass filter. The low-pass filter used for this envelope is a fourth-order Butterworth filter with a 0.05 Hz cut-off. The output is presented in Figure 4.11, where we can clearly see the envelope successfully extracted from the noise signal.

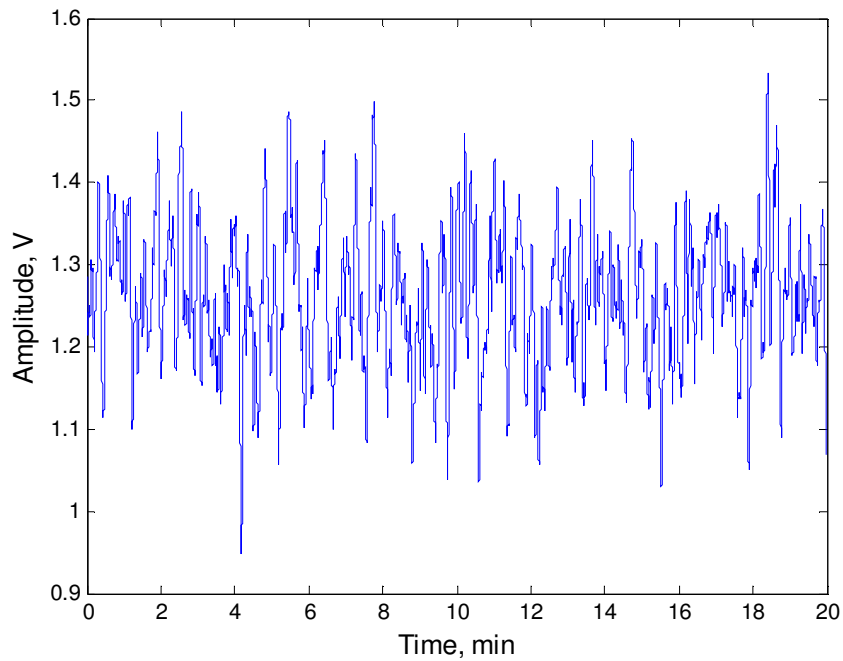


Figure 4.11 Envelope (output of Hilbert transform) after low-pass filter

The extracted envelope is then normalised to minimum and maximum amplitudes before being rescaled into an appropriate envelope voltage range. Figure 4.12 shows the plot of Figure 4.11 after normalisation to its maximum and minimum amplitudes.

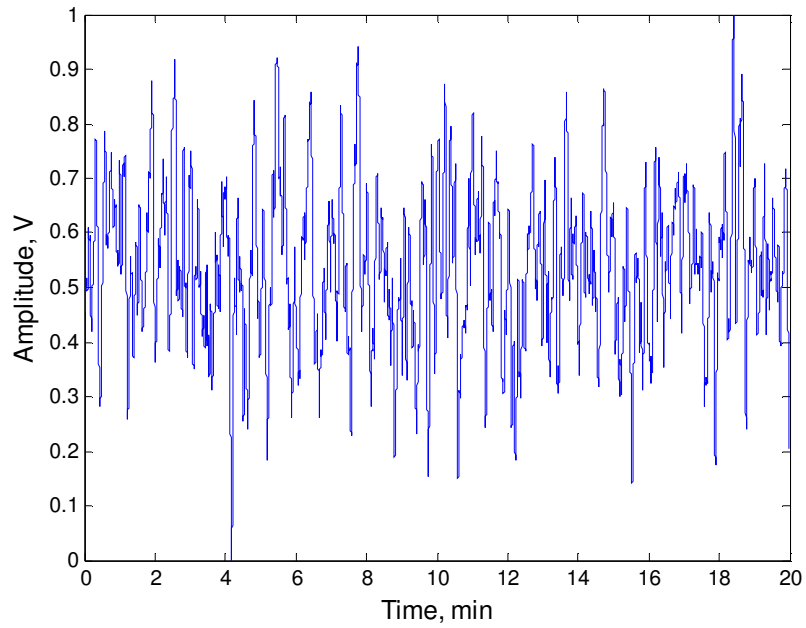


Figure 4.12 Envelope normalisation to minimum-0 and maximum-1

As mentioned earlier, our aim is to develop non-stationary clutter which has the characteristics of real measured clutter. Based on the envelope values in Table 4.1, we rescaled our normalised simulated signal to the appropriate amplitude close to the real clutter.

For very strong real measured clutter, the calculated envelope for the minimum value is 0.0834 and the maximum value is 4.2691. The modified envelope is presented in Figure 4.13 and became the input of the modulation process.

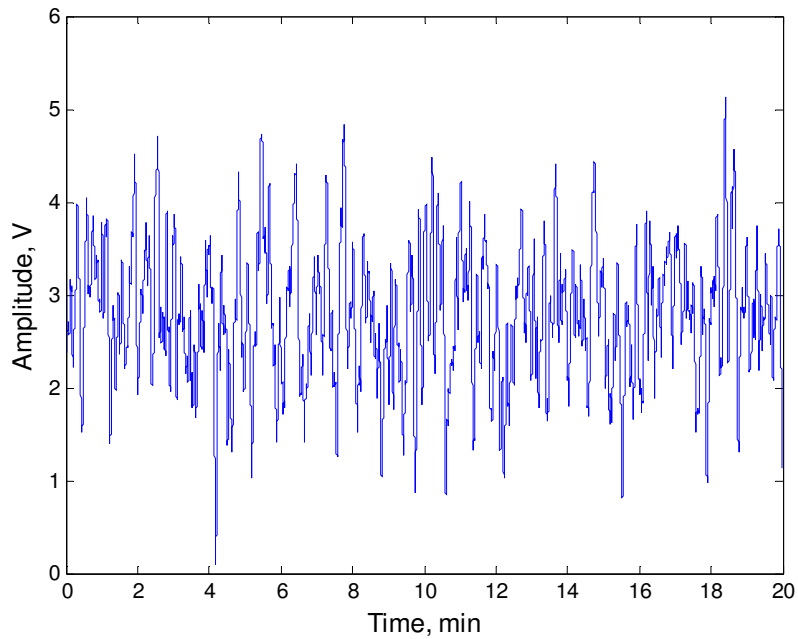


Figure 4.13 Scaled envelope

4.3.3 Modulation Process

Modulation is the process of varying the amplitude of simulated non-stationary clutter with respect to the amplitude of the envelope. The modulation process is highlighted in red in Figure 4.14. The amplitude of a non-stationary clutter signal is multiplied with the envelope.

The result of the modulation of stationary clutter by multiplying it to envelope is presented in Figure 4.15; the clutter amplitude distribution varies with time and within the defined envelope. Figure 4.16 shows the statistical properties of the simulated clutter: the PSD and distribution of non-stationary clutter and the PSD of the clutter envelope. Simulated clutter distribution has a shape factor of 1.6 which agrees with appropriate parameter of measured clutter in Figure 4.5.

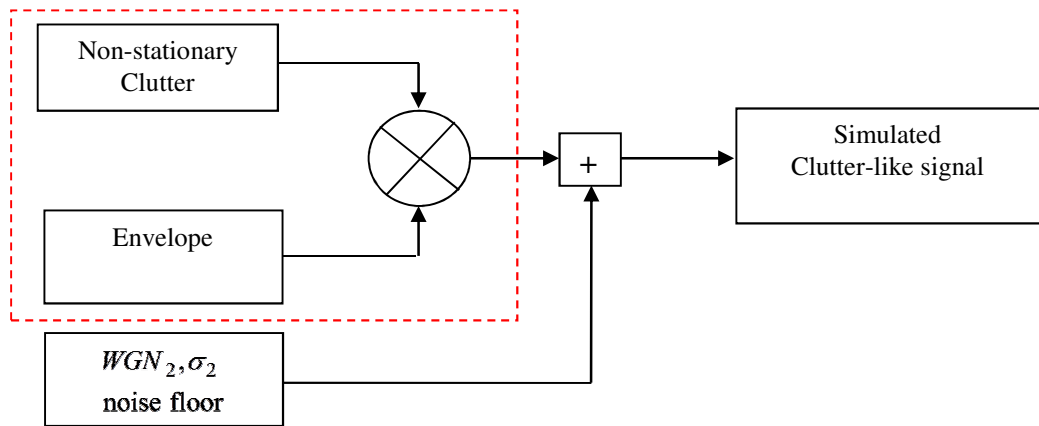


Figure 4.14 Modulation process

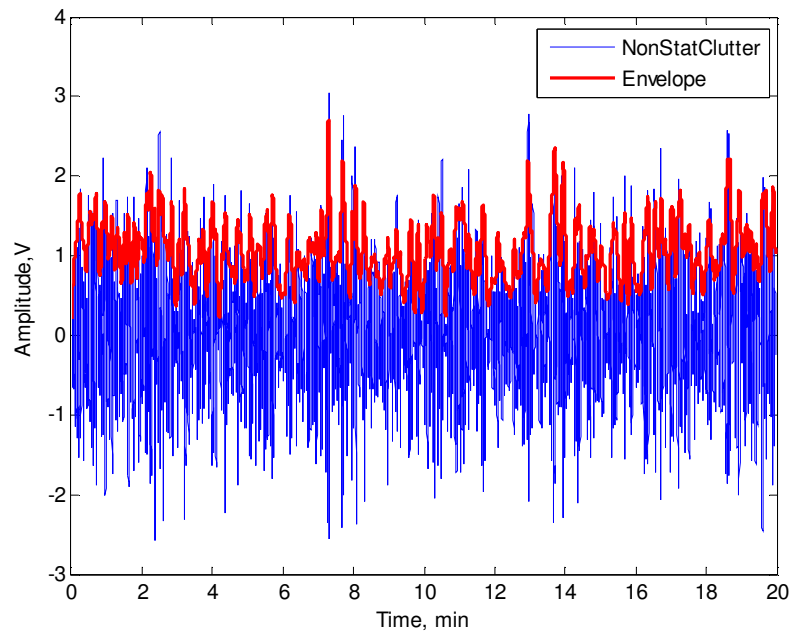


Figure 4.15 Non-stationary Clutter

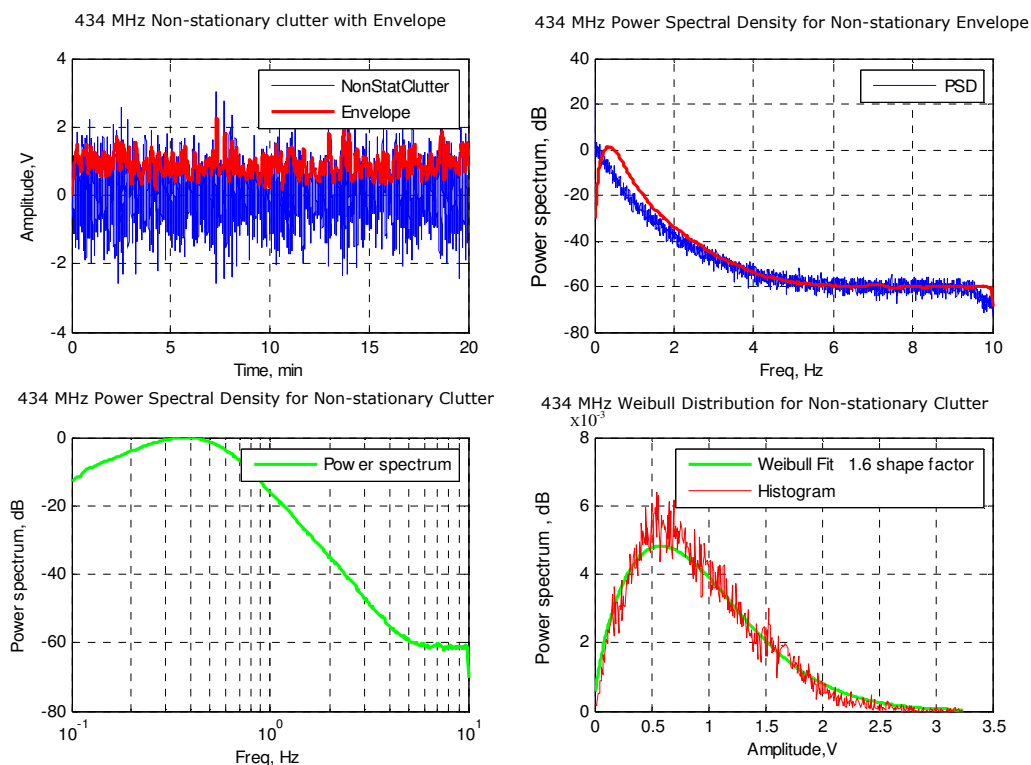
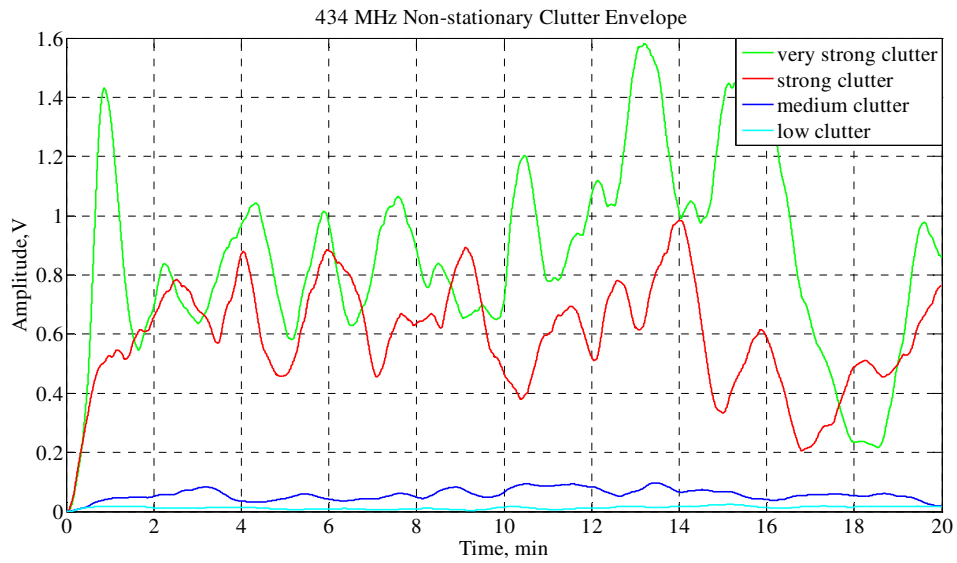
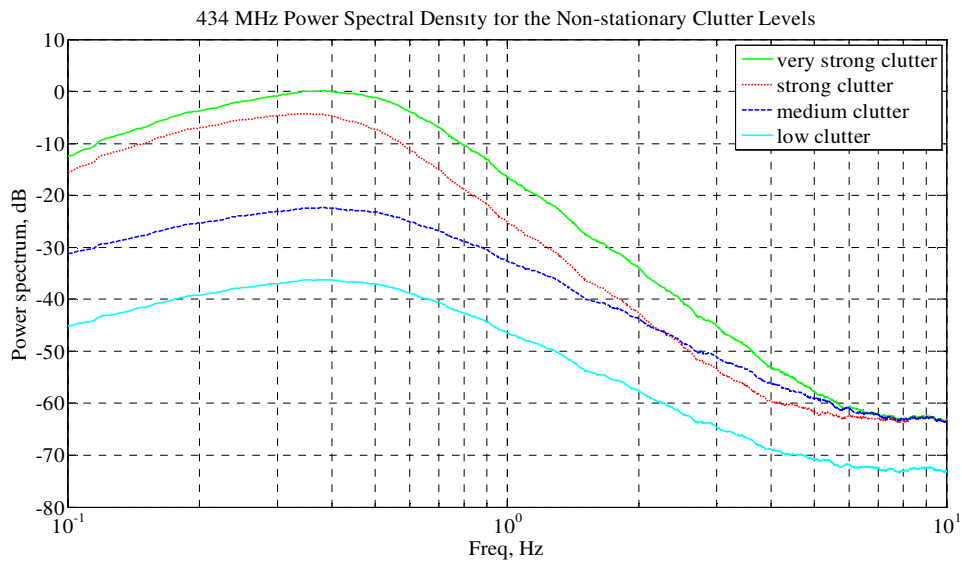


Figure 4.16 Non-stationary clutter statistical characteristic for 434 MHz

Figure 4.17 shows different levels of clutter for a carrier frequency of 434 MHz. The amplitude of the clutter envelope increases as the clutter conditions vary from low to very strong. We can see that the PSD drop corresponds to approximately 50 dB per decade in all conditions and the spectrum width, defined by a 10 dB power drop, is about 0.3-0.5 Hz.



(a)



(b)

Figure 4.17 (a) Envelope and (b) PSD for different level clutter condition at 434 MHz

Figure 4.18 and Figure 4.19 demonstrates the dependence of the clutter power on the frequency. The differences between the maximum amplitudes of the clutter for each frequency become more observable as the frequency increases. The total power increases around the fourth power of carrier frequency. Thus all characteristics presented are similar to measured clutter.

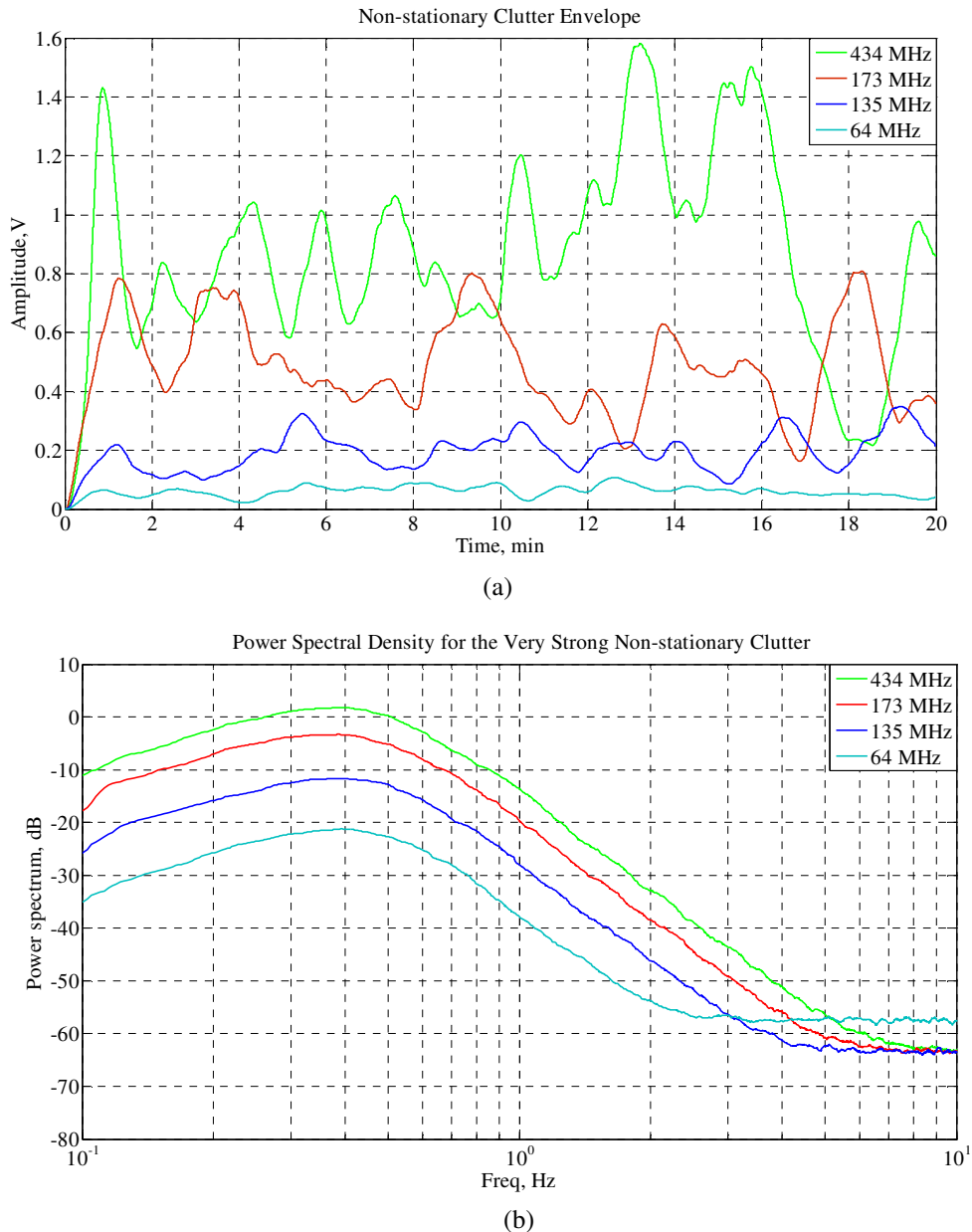


Figure 4.18 (a) Envelope and (b) PSD of simulated clutter for different frequencies when clutter is very strong

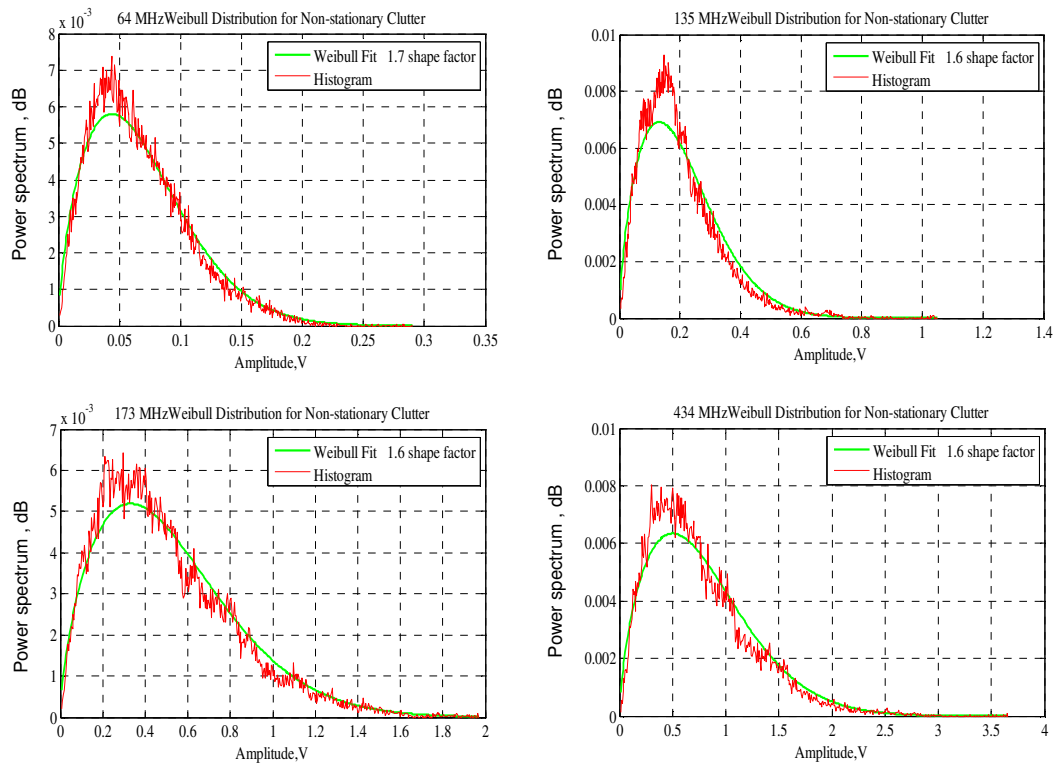


Figure 4.19 Distribution of simulated clutter at 64 MHz, 135 MHz, 173 MHz and 434 MHz channels.

In a real environment, the stability of clutter level varies with time. When the measurement period is long, the strength of the wind and the swaying of foliage and branches can differ, resulting in different clutter levels within recorded signals. This can be seen in Figure 4.20. For obvious reasons simulated clutter is never exactly like real clutter. However, we can still consider the simulated clutter at 434 MHz as comparable to measured clutter at a certain stage. The envelope of simulated clutter is within a defined envelope, between 0.245 and 3.6 (as in Table 4.1).

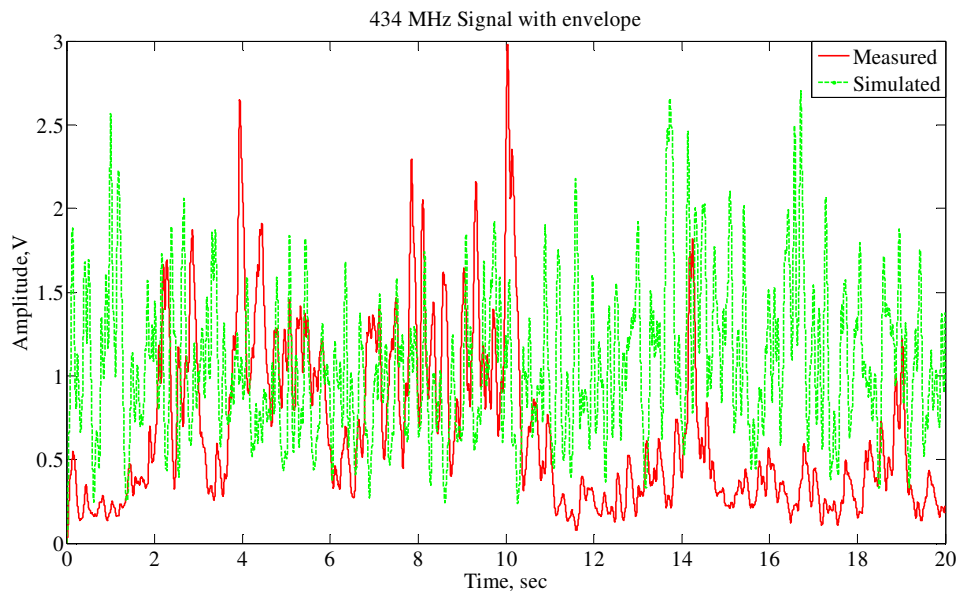


Figure 4.20 Envelope comparisons between measured and simulated clutter

4.4 Conclusion

In this chapter we have analysed the characteristics of measured clutter. We have then described a practical approach to simulate non-stationary clutter using measured clutter characteristics. Finally, comparisons between different clutter strengths and frequencies have been made, followed by comparisons between simulated and measured clutter signals.

CHAPTER 5

TARGET MOTION PARAMETER ESTIMATION AND DOPPLER SIGNATURE NORMALISATION AS THE MAJOR PROCESSING ROUTINE FOR ATC

5.1 Introduction

In Chapter 2, it has been stated that extraction of target motion parameters is an important part of data processing. The Bistatic Doppler shift $f_d = \frac{2v}{\lambda} \cos \delta \sin\left(\frac{\beta}{2}\right)$ depends on radar topology, the baseline in particular, and target motion parameters such as velocity and crossing point. Assuming that the baseline length is given, while speed and trajectory are not known a priori, we have to start with the processing algorithm enabling motion parameter extraction. In this chapter the coherent processing procedure developed by Dr Marina

Gashinova will be presented, in order to estimate the speed, and it will be shown that estimated speed corresponds quite well to the actual speed data. We will also analyse how clutter affects speed estimation accuracy. Then estimated speed will be used for Doppler target signature normalisation in order to apply PCA for normalised spectra of Doppler signatures.

Finally the Doppler spectra normalisation for different baselines and estimated speeds will be presented and conclusions will be formulated.

5.2 Target Motion parameter estimation

The configuration of FSR is shown in Figure 5.1. The target is moving with a velocity defined by projections v_x and v_y , crossing the baseline at the point $(0, y_c)$ with motion direction specified by angle, φ . As the target approaches the baseline, the distance between the TX-to-target (R_1) and target-to-RX (R_2) is changing with time and the received signal has a phase (Doppler) shift. The received signal is expressed as:

$$S_r(t) = A(t) \sin[\psi(t)] \quad (5.1)$$

where $A(t)$ is the instant amplitude of the received signal and $\psi(t)$ is the phase signature of a moving target. The envelope is given by:

$$A(t) = \sqrt{P_T G_T G_R \frac{4\pi\sigma(t)}{\lambda^2} L_T(t) L_R(t)} \quad (5.2)$$

where $G_{R/T}$ is the antenna gain for receiver and transmitter, P_T is the transmit power, $L_T(t), L_R(t)$ are the propagation losses along the transmitter-target and target-receiver paths

and target RCS which is entirely specified by the target geometrical cross section (silhouette), σ and the signal wavelength, λ

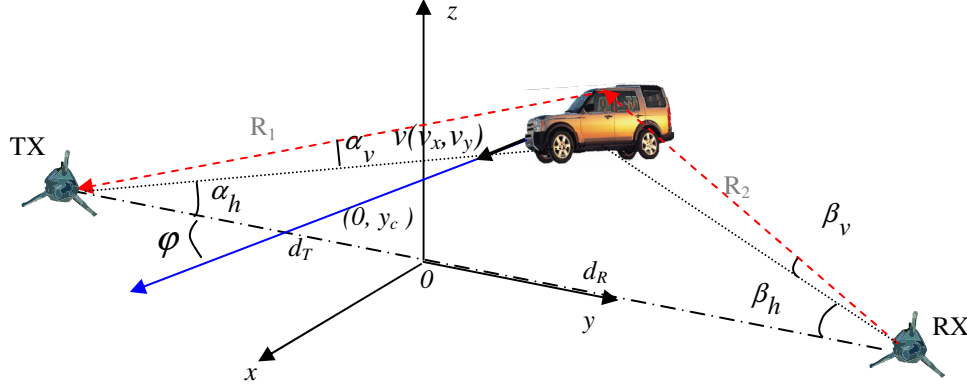


Figure 5.1 FSR configuration in two-ray propagation

The phase signature $\psi(t)$ can be written as

$$\psi(t) = \omega_0 \frac{R_T(t) + R_R(t)}{c} \quad (5.3)$$

where

$$R_T = \frac{\sqrt{(vt \cos \varphi + d_T)^2 + (vt \sin \varphi)^2}}{\cos \alpha_v}$$

$$R_R = \frac{\sqrt{(vt \cos \varphi + d_R)^2 + (vt \sin \varphi)^2}}{\cos \beta_v}$$

and ω_0 and c are the carrier angular frequency and speed of light in free space, respectively.

The receiver power can be calculated using:

$$P_r(t) = P_T G_T G_R \frac{4\pi\sigma(t)}{\lambda^2} L_T(t) L_R(t) \quad (5.4)$$

In a real environment, the target signature could be corrupted by noise, clutter and any possible interference along the baseline. In order to detect a target's signal in the presence of

interference and clutter, coherent signal processing has to be performed. By doing this, we can also extract some information on the target such as its speed, crossing point and target trajectories which will be used during the classification process.

The target signal in equation (5.1) resembles a two-sided chirp-like signal as shown in Figure 5.2. Let us assume that the signal is corrupted by noise/clutter. In order to minimise the effect of noise, the signal is to be correlated with a matched waveform or, in other words, with the reference function. This reference function is sin wave signal with variation in amplitude and frequency over time which depends on the target's parameters such as speed, RCS and trajectory. These parameters are unknown a priori.

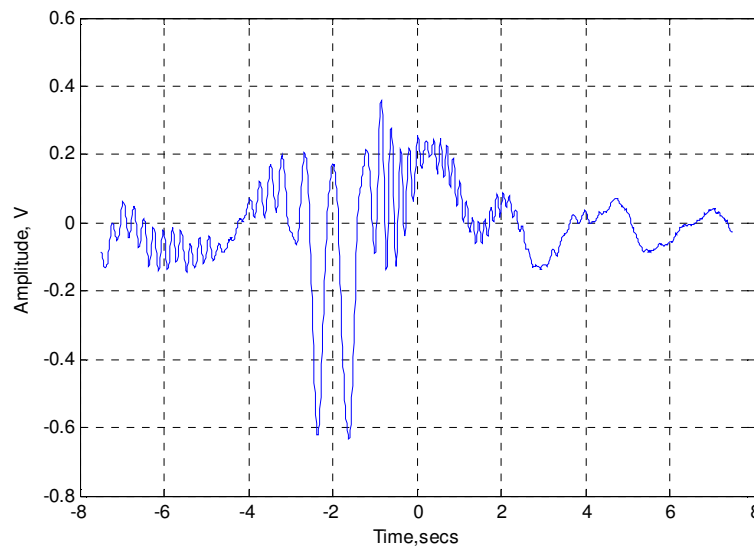


Figure 5.2 Example of target signature resembles a two-sided chirp signal

Unlike conventional radar, where the reference function is the delayed and frequency shifted transmitted signal, the received signal in FSR depends on the speed of a target, its RCS and trajectory [44]. All of these parameters are unknown in advance and thus we need to find the correlation between the received signal and the set of pre-defined reference waveforms:

$$S_{opt}(\tau) = \int_{-T/2}^{T/2} S_r(t) S_{ref}(t - \tau) dt \quad (5.5)$$

where T is integration time and S_{ref} is the reference function.

As mentioned earlier in Chapter 2 and 3, targets with different velocities, baseline distances, motion directions and baseline crossing points, produce different target signatures. This can be seen in Figure 5.3. For example, two simulated Land Rover signatures with two different velocities (3 m/s and 3.5 m/s) are moving perpendicular to the baseline ($\varphi = 90^\circ$). The targets cross in the middle of baseline length of 50m. The carrier frequency is 151MHz. For target speeds that vary only by 0.5 m/s, we can still see a significant difference between two signatures. Thus we have to determine the tolerance of accuracy for the speed estimation to enable correct classification.

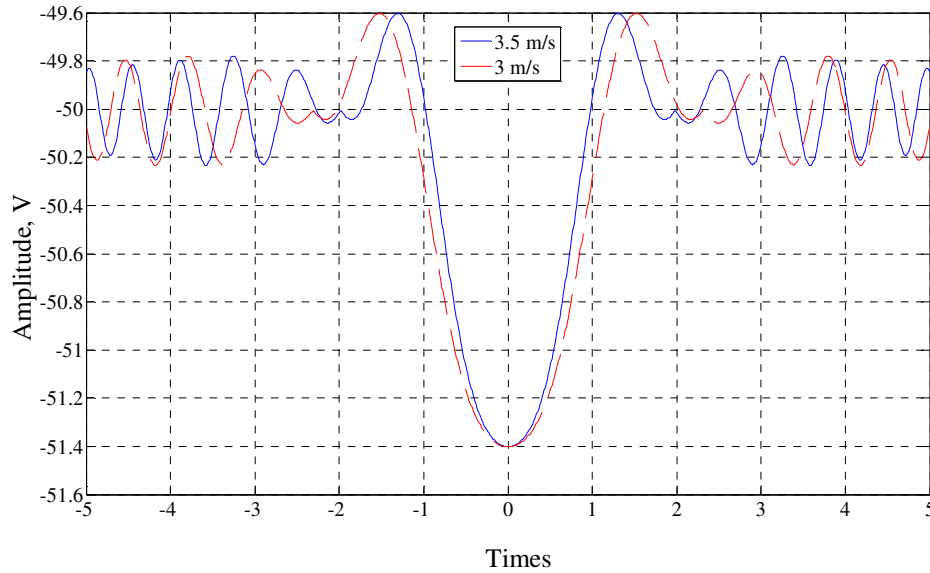


Figure 5.3 Simulated signatures of targets with different velocities

5.2.1 Coherent signal processing algorithm

From equation (5.2), it is clearly stated that the envelope of the signature depends on the target RCS. However, in the signal processing process, a knowledge of the signal envelope is not necessary as the correlation process is more sensitive to Doppler phase signature than to amplitude modulation [44]. Therefore, we can use a windowed rectangular signature to replace the signal envelope since the construction of $S_{ref}(t)$ is based on estimation.

The first step in constructing a reference signal $S_{ref}(t)$ is to determine the target's velocity and target direction. If the target has a uniform linear motion, its trajectory is defined by v_x^i, v_y^j and y_c^k (the baseline crossing point) [44] as in Figure 5.1:

$$\begin{aligned} v_x^i &= v_x^o + i \cdot \Delta v_x; \\ v_y^j &= v_y^o + j \cdot \Delta v_y; \\ y_c^k &= y_c^o + k \cdot \Delta y_c; \end{aligned}$$

where $i=0 \dots N_x, j=0 \dots N_y, k=0 \dots N_c$ and $\Delta v_x, \Delta v_y, \Delta y_c$ are corresponding to increment.

Let us assume that the target is crossing the baseline, y_c with velocity, v where $v = \sqrt{v_x^2 + v_y^2}$. A set of reference function signals can be designed by using different combinations of the target speed, $v_{x,y}$, crossing point y_c and time duration T . This signal is a sinusoidal signal with variation of amplitude and frequency over time:

$$\begin{aligned} S_{ref}(v_{x,y}, y_c) &= -\sin \frac{2\pi f}{c} (R_T(v_{x,y}, y_c) + R_R(v_{x,y}, y_c) - BL) \\ S_{ref}(v_{x,y}, y_c) &= \omega(a) \cdot S_{ref}(v_{x,y}, y_c); \quad \omega(w) = e^{-\frac{1}{2} \left(\alpha \frac{w}{W/2} \right)^2} \end{aligned}$$

$$R_T(v_{x,y}, y_c) = \sqrt{[(y_c^k + v_x^i + BL/2)^2 + (v_y^j)^2]}$$

$$R_R(v_{x,y}, y_c) = \sqrt{[(y_c^k + v_x^i - BL/2)^2 + (v_y^j)^2]}$$

where R_T and R_R are the TX-target and target-RX respectively which depends on $(v_{x,y}, y_c)$ and BL is the baseline length. S_{ref} is then passing through a Gaussian window in order to improve correlation by reducing the side lobes of correlation product where $-\frac{W}{2} \leq w \leq \frac{W}{2}$ and α is inversely proportional to the standard deviation of a Gaussian random variable and defined by shape factor.

Each reference function's signal is then correlated with the received signal [16, 44, 65], and the result is integrated over the interval T :

$$F(T, v_{x,y}, y_c) = \int_{-T/2}^{T/2} S_r(t) S_{ref}(v_{x,y}, y_c) dt \quad (5.6)$$

The function $F(T, v_{x,y}, y_c)$ has a maximum at some point where the global maximum of the correlation surface is situated.

The signal for the proper detection and parameter estimation should have maximum possible S/N ration. The maximum S/N is achieved at the output of the matched filter or an appropriate correlator. In this case, the signal processing will be optimal. For example, if the waveform (irrelevant to the transmitting waveform) is a chirp signal; the compression is known as a pulse compression waveform (wave shaping process) i.e at the output of the matched filter or correlator, we are observing pulse compression effect. The example of pulse compression effect can be seen in Figure 5.4.

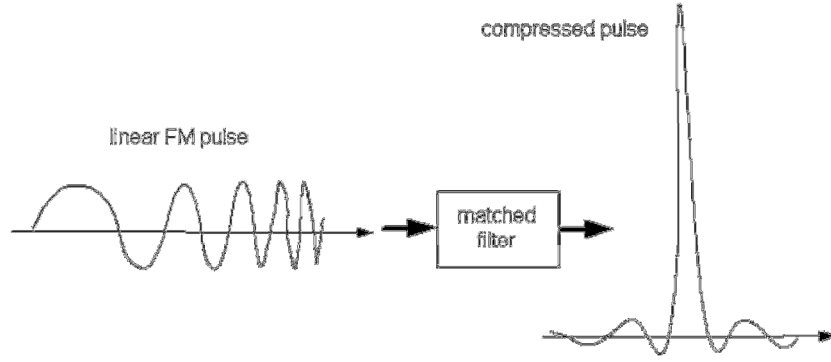


Figure 5.4 Example of Pulse compression waveform [66]

The simplified processing scheme for a 2-D array of reference signatures is illustrated in Figure 5.5; we assume that the crossing point is fixed. Once we have determined this maximum, the information about the target's trajectory and speed estimation can be obtained.

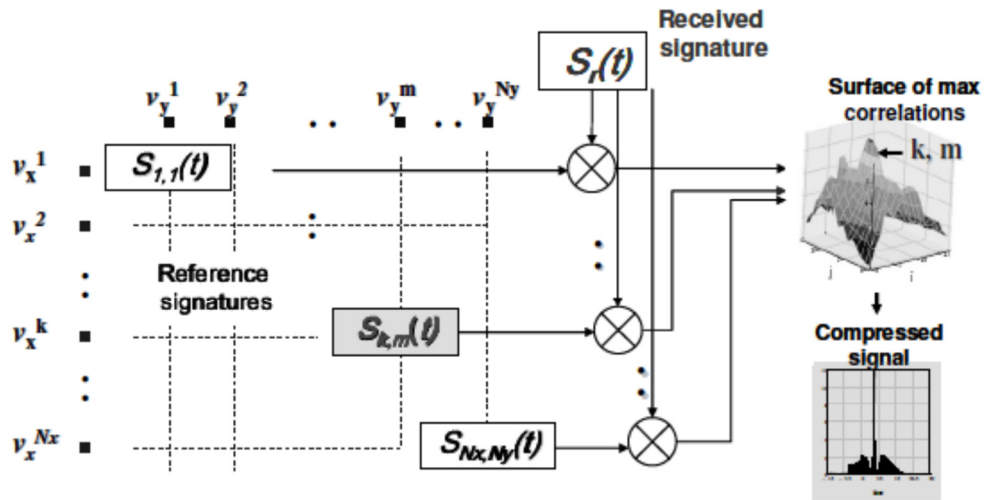


Figure 5.5 Processing scheme [44]

Let us to consider an example where the target is crossing perpendicular to the middle of a 50m baseline, with speed of 15 m/s and the simulated target signal passes through the signal processing process to extract the information about its trajectory.

Figure 5.6 shows the simulated received signal and the reference signal which provide the maximum correlation. The reference signal automatically gives indices of the specified target velocity vectors and vector of crossing points, so we can reconstruct the target speed and trajectory parameter information as shown in Figure 5.7. As presented, the speed, crossing point and crossing angle's values correspond to the trajectory of the received signal.

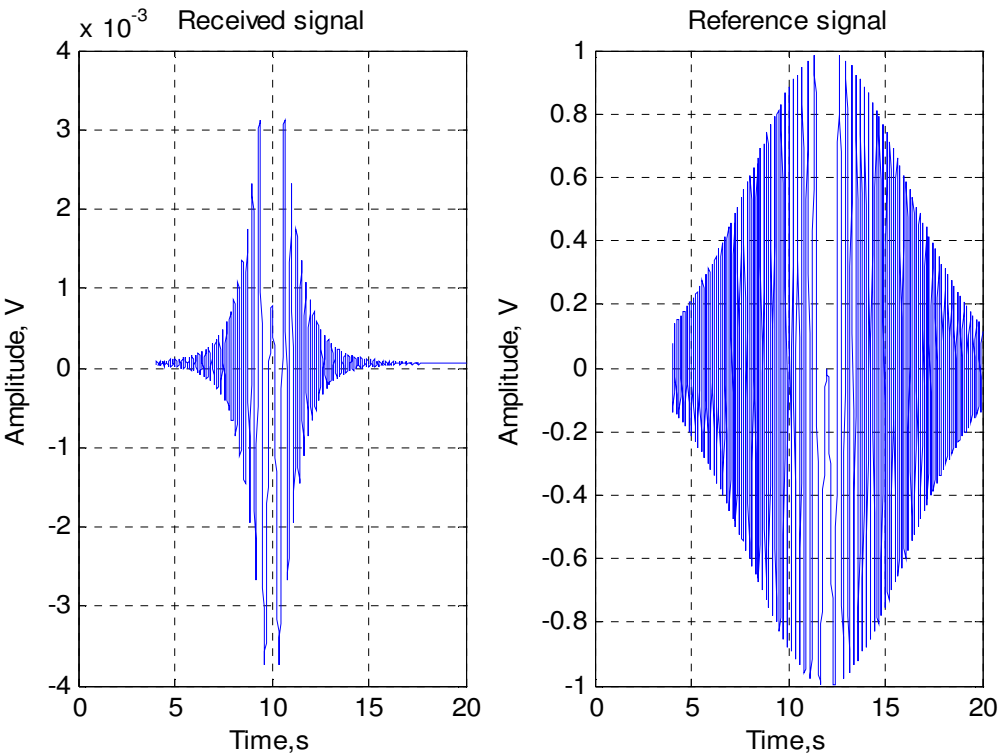


Figure 5.6 Example of a received signal and reference signal that provide maximum cross correlation

Figure 5.7 is a screenshot of a MATLAB variable viewer showing the 'velocityEst' structure. The structure contains the following fields and values:

| Field | Value |
|------------|-----------------------|
| fileName | 'RSSIsigSinBL50m.mat' |
| velValue | 15 |
| CrossPoint | 0 |
| AngleT | 90 |
| RefSig | <1600x1 double> |
| Baseline | 50 |

Figure 5.7 The extracted information that can be obtained as a result of matching filtering.
(Speed of target: VelValue, Crossing point: CrossPoint, Crossing angle: AngleT and Baseline)

If the optimal signal processing is successful, the signal compression gives maximum of S/N ratio where the received target signal compression gain is up to 10-20 dB (Figure 5.8). For an unmatched case, the received signal will not be compressed accordingly (Figure 5.9).

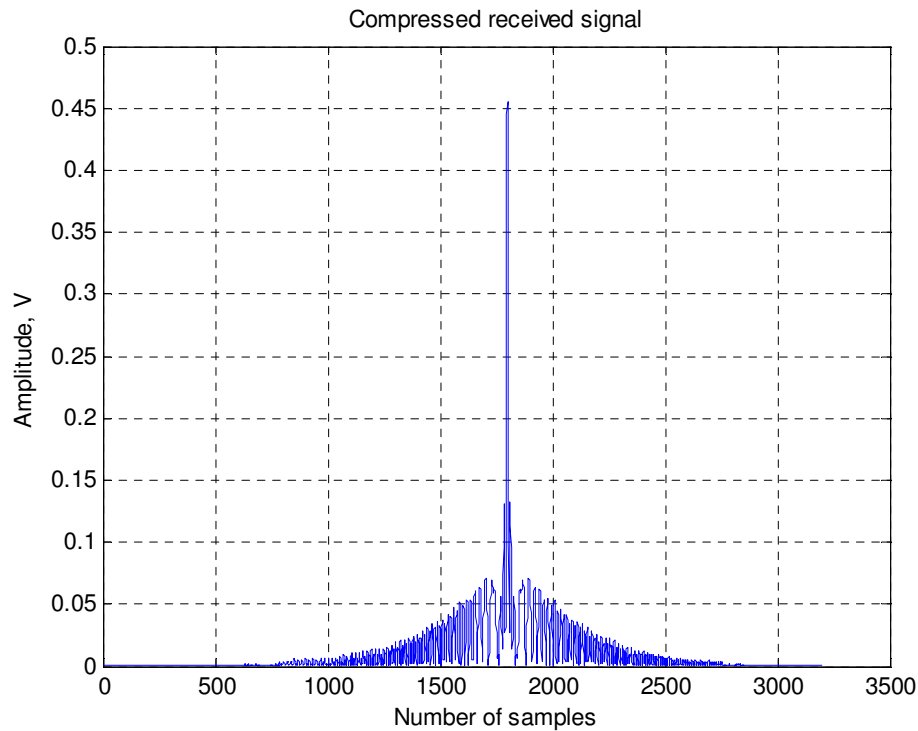


Figure 5.8 Compressed output of a matched filter with a correct matching reference signal

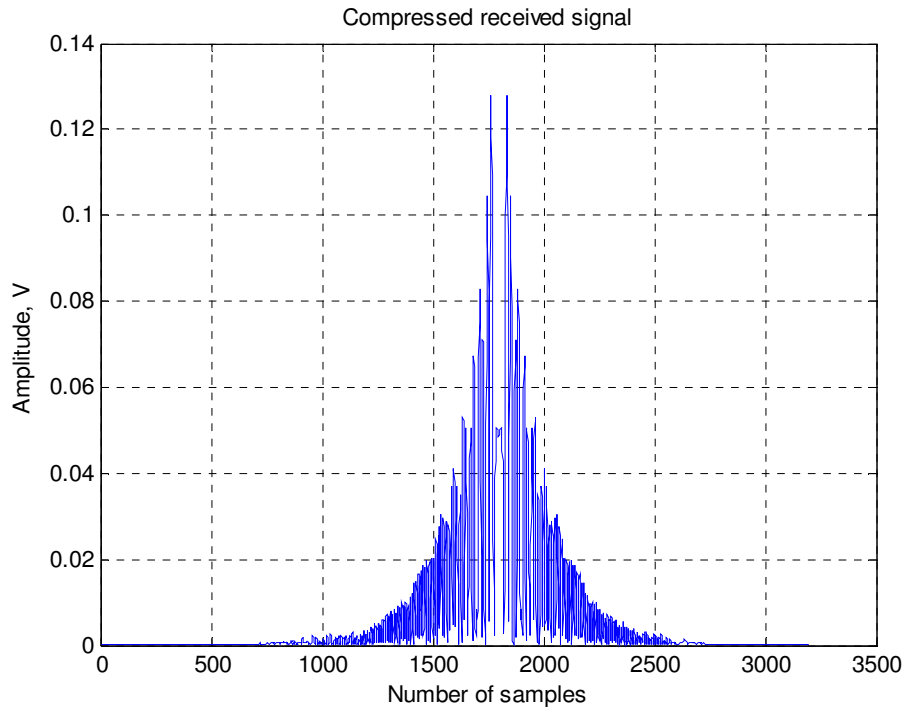


Figure 5.9 Compressed output of a matched filter with an incorrect matching reference signal

5.2.2 Algorithm verification in clutter free conditions

In order to understand the performance of the speed estimation algorithm, the evaluation is conducted using simulated signals and real-world measurements at all frequencies in “clutter free” conditions. For simulated signals, the signals are modelled using two-ray path propagation approximation [44]. The purpose of using a simulated signal is to avoid other possible factors that can affect the speed estimation’s efficiency. So, the speed estimation algorithm is evaluated using measured signals from the Pritchatts car park.

Signals are simulated using a rectangular target with the dimensions of a Land Rover and a point-like target with RCS=1 dBsm. Each signal is set to have a speed of 15 m/s and to cross perpendicular to the middle of various baseline lengths (50m, 100m, 150m and 200m). The

reference signals' speed was set to have a range between 0-20m/s with 1m/s interval between speeds.

The estimated speed for the simulated target is presented in Table 5.1. The results suggest that at all frequencies the algorithm gave an accurate estimated speed, even when the baseline is long. However, this is only an ideal case, thus further investigation is required, especially using a real measured signal.

Table 5.1 Estimated speed for rectangular target (Land Rover) and point like target at all frequencies.

| File name | True speed | 64MHz | 151 MHz | 434 MHz |
|---|------------|--------|---------|---------|
| RSSIsigSinBL50m.mat (Land Rover) | 15 m/s | 15 m/s | 15 m/s | 15 m/s |
| RSSIsigSinBL100m.mat (Land Rover) | 15 m/s | 15 m/s | 15 m/s | 15 m/s |
| RSSIsigSinBL150m.mat (Land Rover) | 15 m/s | 15 m/s | 15 m/s | 15 m/s |
| RSSIsigSinBL200m.mat (Land Rover) | 15 m/s | 15 m/s | 15 m/s | 15 m/s |
| RSSIsigSinBL50m.mat (point like target) | 15 m/s | 15 m/s | 15 m/s | 15 m/s |
| RSSIsigSinBL100m.mat (point like target) | 15 m/s | 15 m/s | 15 m/s | 15 m/s |
| RSSIsigSinBL150m.mat (point like target) | 15 m/s | 15 m/s | 15 m/s | 15 m/s |

| | | | | |
|---|--------|--------|--------|--------|
| RSSIsigSinBL200m.mat (point like target) | 15 m/s | 15 m/s | 15 m/s | 15 m/s |
|---|--------|--------|--------|--------|

Even though the results above give a good indication of the algorithm's accuracy, performance must be verified under real-environment conditions (where clutter, reflections and interference from surroundings can exist). In a real-life scenario, the speed is unknown, uncontrollable and inconsistent (it can accelerate or decelerate over time). For our purposes we assumed that the vehicle speed is known via video captured during the measurement. The signals were collected in Pritchatts car park where the target is crossing in the middle of 50m baseline. The speeds varied between 4-11 m/s.

The estimated speeds for the measured signals are presented in Table 5.2. The results suggest that the estimated speeds using the speed algorithm have an accuracy of more than 90% at all frequencies compared with speeds from the video. For example, the speed has a difference of only 0.14 m/s (which is around 2.5%) at 64 MHz and 434 MHz, and a difference of 0.24m/s (4.5%) for 151 MHz. The fact that the algorithm did not achieve 100% accuracy is probably due to the noise, clutter, reflection and interference in the signal. Furthermore, there might be some discrepancies with the calculation of speed using captured video.

In order to check the precision of the algorithm, Root Mean Square Error (RMSE) is used:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (V_{simulated,i} - V_{measured,i})^2} \quad (5.7)$$

where:

N = number of signal

$V_{simulated}$ = speed estimated using speed estimation algorithm

$V_{measured}$ = speed measured using video camera

As calculated at the end of Table 5.2, at all frequencies, the RMSE is less than 0.25 and the percentage of RMSE error is less than 4%. This gives a good indication that the procedure achieved high accuracy in estimating the speed of the target.

Table 5.2 Estimated speeds for measured signals at all frequencies

| File name | Speed from video | 64 MHz | 151 MHz | 434 MHz |
|------------------|-------------------------|---------------|----------------|----------------|
| | m/s | m/s | m/s | m/s |
| 4LT_1.bin | 5.36 | 5.50 | 5.60 | 5.50 |
| 4LT_2.bin | 4.49 | 4.30 | 4.60 | 4.60 |
| 4LT_3.bin | 4.95 | 4.80 | 5.10 | 5.10 |
| 4LT_4.bin | 4.33 | 4.20 | 4.40 | 4.40 |
| 4LT_5. bin | 4.08 | 4.00 | 4.30 | 3.30 |
| 4LT_6.bin | 5.74 | 5.50 | 6.40 | 5.90 |
| 4LT_7.bin | 5.88 | 5.70 | 5.90 | 6.00 |
| 4LT_8.bin | 6.03 | 5.80 | 6.00 | 6.10 |
| 4LT_10.bin | 6.29 | 6.10 | 6.30 | 6.40 |
| 4LT_11.bin | 7.23 | 7.00 | 7.50 | 7.30 |
| 4LT_12.bin | 8.12 | 7.80 | 7.80 | 8.20 |
| 4LT_13.bin | 8.12 | 7.70 | 8.20 | 8.30 |
| 4LT_14.bin | 8.12 | 7.70 | 8.20 | 8.40 |
| 4LT_15.bin | 7.86 | 7.50 | 8.00 | 8.10 |
| 4LT_17.bin | 10.18 | 9.70 | 10.20 | 10.50 |
| 4LT_20.bin | 10.48 | 9.90 | 10.50 | 10.80 |
| 4LT_21.bin | 3.87 | 3.90 | 4.00 | 3.80 |
| 4LT_22.bin | 4.16 | 4.20 | 4.40 | 4.30 |

| | | | | |
|-----------------|-------|-------------|-------------|-------------|
| 4LT_23.bin | 4.02 | 4.00 | 4.10 | 4.00 |
| 4LT_24.bin | 3.89 | 3.80 | 4.00 | 3.60 |
| 4LT_25.bin | 4.18 | 4.20 | 4.30 | 4.00 |
| 4LT_26.bin | 5.93 | 5.80 | 6.00 | 5.80 |
| 4LT_27.bin | 6.03 | 5.90 | 6.10 | 5.90 |
| 4LT_28.bin | 5.93 | 5.80 | 6.00 | 5.80 |
| 4LT_29.bin | 5.88 | 5.90 | 6.10 | 5.90 |
| 4LT_30.bin | 6.08 | 6.00 | 6.30 | 6.10 |
| 4LT_31.bin | 6.89 | 6.70 | 6.90 | 6.60 |
| 4LT_32.bin | 7.95 | 7.80 | 8.10 | 7.90 |
| 4LT_33.bin | 8.12 | 7.90 | 8.20 | 8.20 |
| 4LT_34.bin | 8.13 | 7.90 | 8.30 | 8.00 |
| 4LT_35.bin | 7.86 | 7.70 | 8.00 | 8.00 |
| 4LT_36.bin | 10.63 | 10.50 | 11.10 | 10.40 |
| 4LT_37.bin | 10.48 | 10.50 | 11.20 | 10.80 |
| 4LT_38.bin | 10.95 | 10.60 | 11.00 | 10.80 |
| 4LT_39.bin | 10.79 | 10.50 | 11.00 | 10.10 |
| 4LT_40.bin | 10.48 | 10.20 | 10.90 | 10.20 |
| RMSE,m/s | | 0.24 | 0.24 | 0.25 |
| RMSE,% | | 3.44 | 3.46 | 3.69 |

5.3 Doppler spectra normalisation using estimated speed of target

Results in Table 5.2 suggest that errors may occur in the system which may have the potential to degrade the performance of ATC. Hence, in this section we will evaluate how speed estimation accuracy can affect the classification in both “clutter free” cases and in cases where clutter is present.

5.3.1 Analysis of speed estimation error in “clutter free” environment

We assume that the measured signals in our database (Land Rover, Nissan, BMW316i, and Ford Focus) are “clutter free” (no simulated clutter is added) and the speed is estimated accurately using a low frequency (64 MHz).

The evaluation is performed by introducing an error to the speed estimation block diagram. The error introduced to the estimated speed is considered to be a Gaussian random variable with zero mean and the variance corresponding to the percentage of the estimated speed. The analysis was performed with the percentage of speed errors being from 1% to 10% in 1% steps.

Figure 5.10 depicts an example of the histogram of the speed error values when the true speed is 8.3m/s and the speed percentage error is 5%. The system will randomly record the speed with the error within a given range.

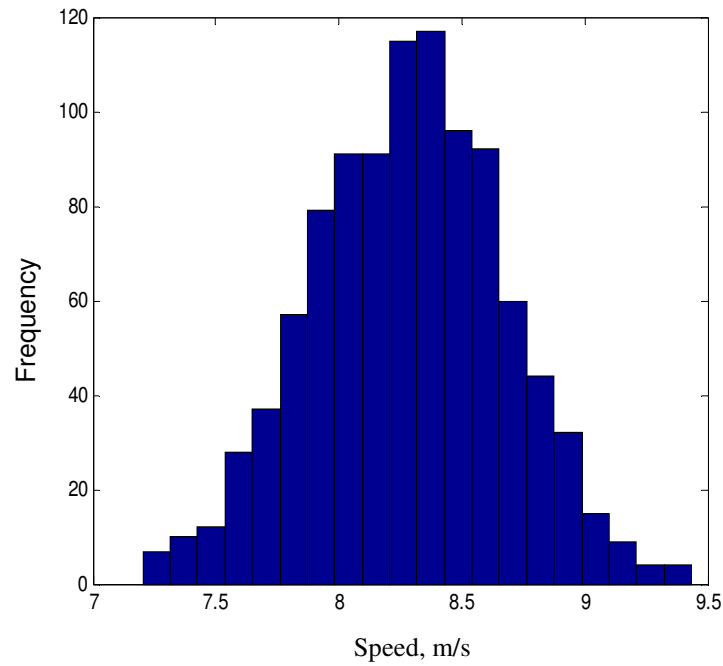


Figure 5.10 Histogram of speed values after introducing a 5% error to the true speed of 8.3 m/s

The effect of the error in speed estimation in the 2-D PCA components space is plotted in Figure 5.11. We can see that when the percentage of error is very small, the plots for each type of target are concentrated within the group and are well separated from the other type of targets. However, as the error in speed estimation increases, the spread of the PCA components is wider.

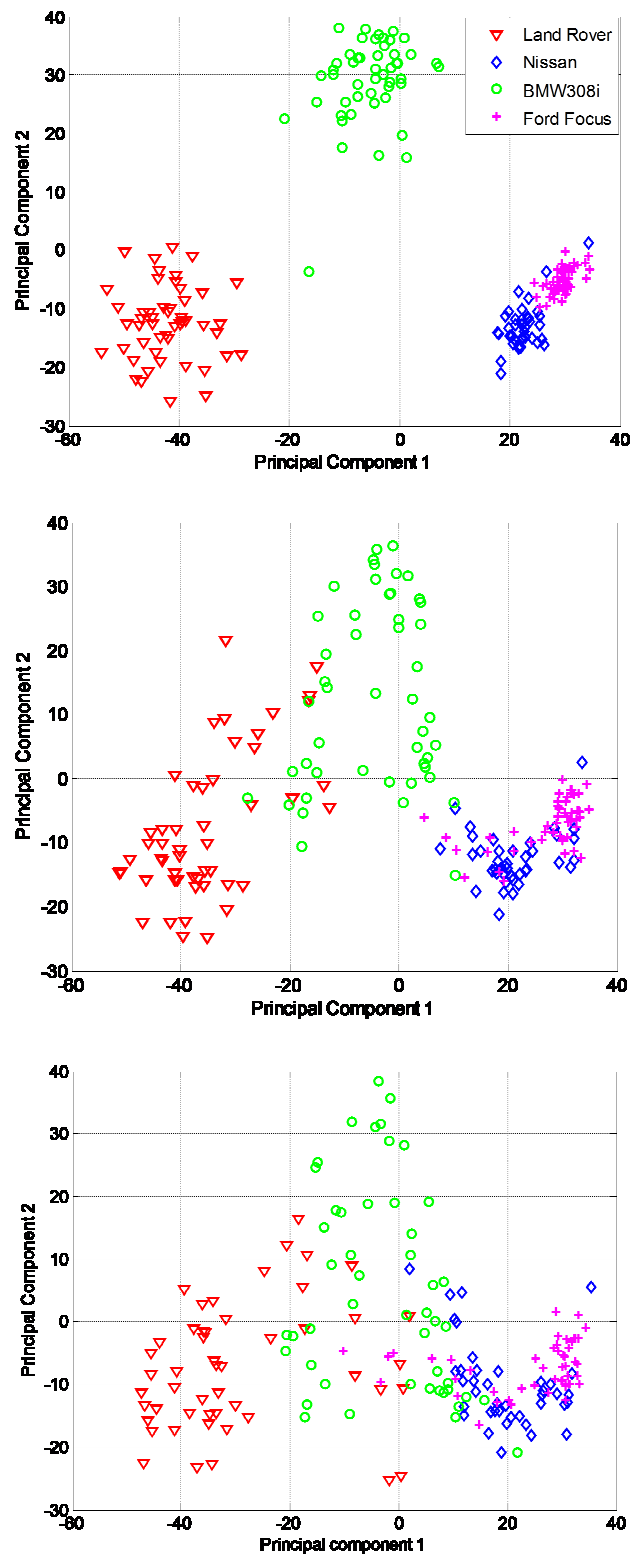


Figure 5.11 The 2-D PCA components (for a frequency of 151 MHz) when considering:
(a) 1% speed estimation error; (b) 5% speed estimation error; and (c) 10% speed estimation error.

The results of the recognition experiments are shown for different frequencies in Figure 5.12 as a function of the speed estimation error. The classification accuracy decreases sharply as the speed estimation error increases. For instance, the classification accuracy drops from 100% to 67% at the 10% error in speed estimation at 151 MHz.

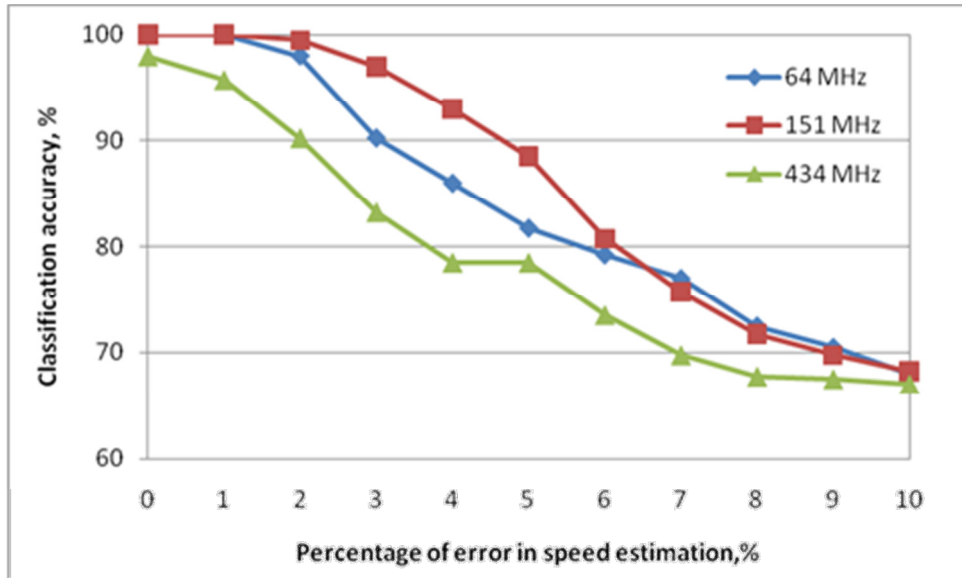


Figure 5.12 Classification accuracy with error in speed estimation for all frequencies

In our case where the target is moving around 5-15 miles/hour, the calculated percentage of RMSE value in Table 5.2 is approximately 4%. Hence, by comparing this value to Figure 5.12, we see that for a 4% speed estimation error, the classification system achieves an accuracy of more than 75% at all frequencies. For higher speed, different classification accuracy might be achieved.

5.3.2 The influence of clutter on the speed estimation

The influence of clutter on the speed of the target is investigated at various levels of SCRs where artificial clutter is added to the received signals. In this investigation, instead of only using 64 MHz for estimating the speed, the speed is estimated at all frequencies. In this section, we will first define SCR. Then, we will present the degradation of classification accuracy in the presence of different levels of clutter.

- **Definition of SCR (Signal to Clutter Ratio)**

The process of simulating artificial clutter is described in Chapter 4.

The effect of clutter is introduced by adding in time domain simulated non-stationary clutter to the received target signature for various levels of SCRs at three different frequencies. In our case, we define the SCR as:

$$SCR = 10 \log(P_s / P_c) \quad (5.8)$$

where P_s and P_c are the total power of the signal and total power of the clutter, respectively, after being passed through a high-pass clutter rejection filter (hardware filter) with a cut-off frequency of 1Hz. The process of generating the clutter at a given SCR is depicted in Figure 5.13. The output of the clutter rejection filter is then passed through the software filter where the cut-off frequency of the filter varies with the carrier frequency. The SCR calculated here is considered to be an improved SCR (SCR^*).

It is important to note that for all analyses throughout this thesis, we only use the SCR defined after passing through a 1Hz clutter rejection filter.

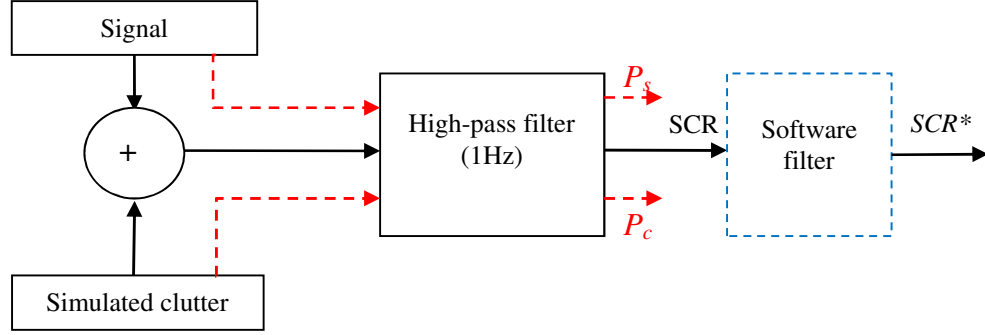


Figure 5.13 Generation of clutter and SCR definition

With the purpose of analysing the effect of different levels of SCRs on our system, a so-called “required SCR” signature is used, against which additional clutter is added. In order to obtain the “new SCR”, method below is used:

1. Calculate the SCR of a signal by finding the ration between signal power to clutter power:

$$SCR_{calculated} = 10 \log \left(\frac{P_{signal}}{P_{clutter}} \right)$$

where:

P_{signal} = power of signal

$P_{clutter}$ =power of clutter

2. Calculate the constant between the calculated SCR and “required SCR”. This constant will be used to modify the ratio between signal power to clutter power so that “required SCR” value can be achieved.

$$\text{constant} = \sqrt{10^{\left(\frac{SCR_{\text{calculated}} - SCR_{\text{required}}}{10}\right)}}$$

3. Find the SCR value as the “required SCR” value

$$SCR = 10 \log \left(\frac{P_{\text{signal}}}{P_{\text{clutter}}} \times \text{constant} \right)$$

Once a value of the “required SCR” is obtained, the signal corrupted at the “required SCR” level is then used in the speed estimation algorithm.

- **ATC degradation in the presence of clutter**

The effect of clutter on the speed is analysed using measured signals with added simulated clutter. The speed is estimated at three different frequencies: 64 MHz, 151 MHz and 434 MHz. Four different types of vehicles are used: BMW306i, Nissan, Land rover and Ford Focus. The baseline is set to be 50m. Simulated clutter is added to the signal with the SCR ranging from 0 dB to 30 dB with incremental steps of 5 dB.

In order to monitor the effect of speed estimation errors on the classification performance, the analysis is performed using speeds estimated from signals with varying levels of clutter.

The evaluation results for ATC at three different frequencies and for four types of cars are shown in Figure 5.14. Overall, the classification accuracy indicates good performance for higher SCR signals; however the accuracy decreases when the SCR drops below 5 dB. This shows that the classification accuracy is significantly affected by the speed error caused by

clutter, especially for 64 MHz. As explained earlier in Chapter 2, large amounts of clutter remain in the signal due to the small value of the cut-off frequency used in the software filter.

The choice of a suitable frequency used in the FSR classification will be a trade-off between resolution, classification performance and resilience to clutter. In the case of a FS micro-radar network employed for the detection and/or recognition of different sized targets (such as a vehicle, animal or human) in high profile terrain, 64 MHz is most suited to detection and speed estimation due to its superior robustness to clutter. A small, slow-moving target might be visible at a low frequency whilst at a higher frequency; a target behind the trench will be in a shadow.

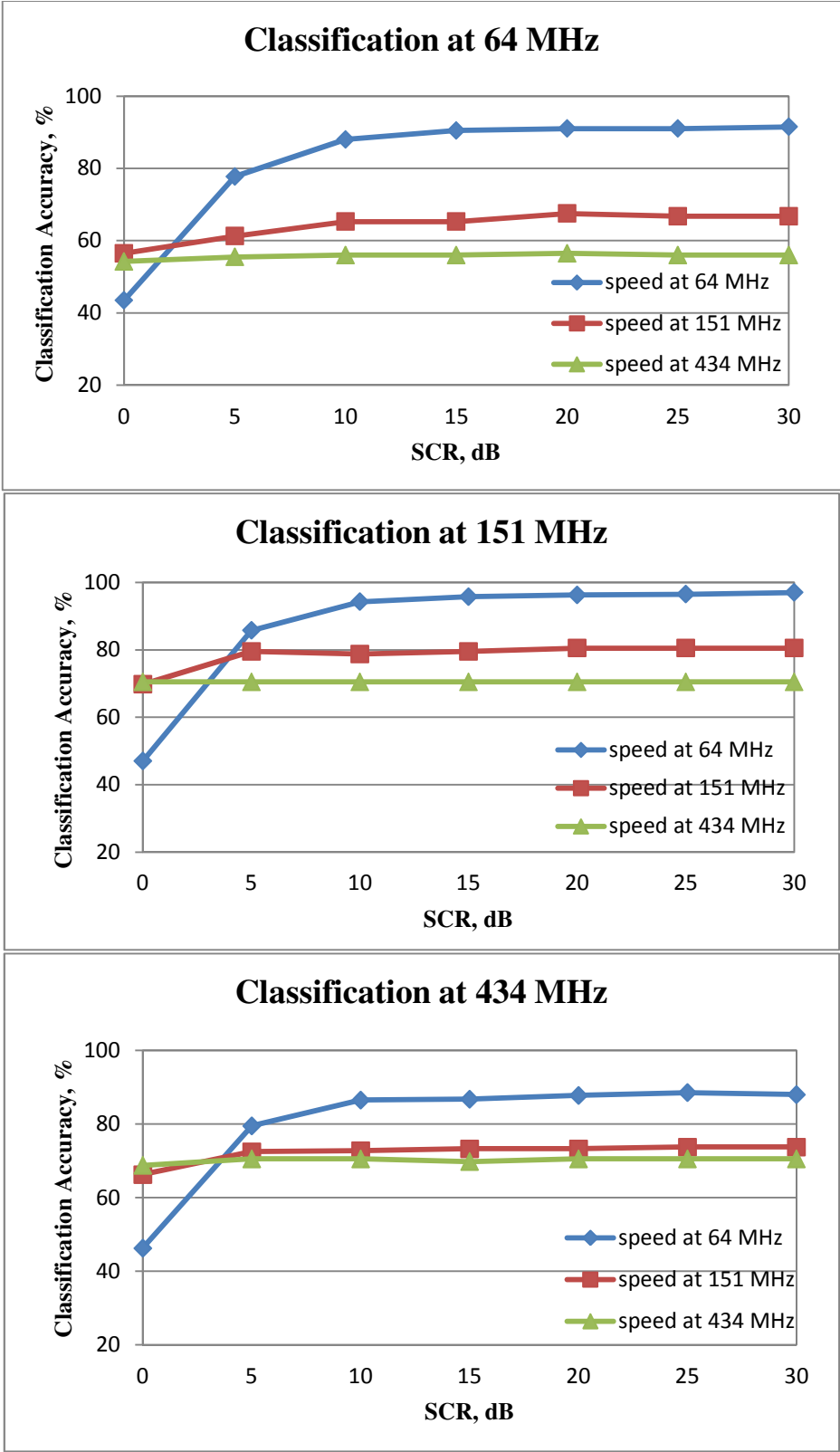


Figure 5.14 ATC with speed estimated using various levels of SCRs at different frequencies.

5.4 Doppler spectra normalisation for different baselines

As mentioned earlier, target signature is defined by speed, target trajectories and RCS of the target. Clutter, noise and interferences are factors affecting the target signature and defining Signal-to-noise/Signal-to-clutter/Signal-to-interference (SNR/SCR/SIR) ratios.

Let us consider a situation when the same car is crossing perpendicular to the middle of baseline.

The signals are simulated according to the two-ray path propagation model for the targets, with the same RCS and a speed of 15 m/s for a period of 20 seconds. We will consider various baseline lengths.

Three different mechanisms of scattering affect the target's RCS. For example for an electrically relatively small target, where the wavelength is bigger or comparable to the size of the target, the scattering mechanism is said to be in the Rayleigh and Resonance/Mie region. Our targets of interest have effective dimensions of the order of a few metres, and the scattering mechanism can range from resonant for 64 MHz, ($\lambda \approx 4.7m$) to nearly optical region for 434 MHz, ($\lambda \approx 0.67m$). Using this approximation, we can check if there is an influence of RCS on the suggested normalisation procedure.

Further consideration will be given to three main situations:

- i. The signal is simulated for a point-like target with RCS=1 dBsm
- ii. The signal is simulated for a simple target - square flat plate with side, $s=1$ m

- iii. The signal is simulated for an extended target such as a car but using a simple rectangular aperture where the size is much larger and comparable with the wavelength.

The baseline affects the waveform because the ‘chirp-like target’ inherently depends on the range. As the Doppler changes, the phase signature also changes. Therefore, prior to any further processing, a time domain signal with a different baseline length is normalised to a reference baseline; in our case we used 250m. In order to do this we need to find an appropriate resampling coefficient.

The procedure of calculating the ratio is as follows:

- i. Find $f_1 = \frac{f_c}{baseline}$ and $f_2 = \frac{f_c}{reference\ baseline}$
- ii. Next find a constant value , $Const = \frac{f_1}{f_2}$
- iii. Before resampling the signal, we need to find a rational approximation for Const so that $Const=N/D$. N and D are the integers providing this constant.

Once we have obtained the rational approximation, we can resample the signal performing consequent up sampling and down sampling procedures for a time domain signal. The obtained resampled signal is used as the input of the signal processing procedures for detection and recognition.

5.4.1 Simulated target

A simulated target is used in order to avoid any other factors such as clutter and speed errors that may affect a target's signal. Three different electrical sizes of targets are taken into consideration: a point-like target, a square plate target with $s=1\text{m}$ and a rectangular aperture target with the same dimensions as a Land Rover (length= 4.7m and height=2.2m).

- **Point-like target**

A simulated point-like target signal can be considered as an isotropic antenna. In this evaluation, a constant value of $\text{RCS}=1\text{ dBsm}$ is applied at all frequencies for the target's speed of 15 m/s.

Figure 5.15 shows the time domain and frequency domain signal before and after the resampling process at 64 MHz. As shown in Figure 5.15 (a) the larger the baseline is, the greater visibility time so that the signal waveforms remain the same but with different timescales. This is reflected in Figure 5.15 (b), as the spectrum main lobe width varies with the baseline.

In order to place all of the signals on the same common timescale, we implemented the resampling process for the time domain signal. The result is presented in Figure 5.15 (c) and (d). The resampling process managed to set the time domain to a common timescale, and the first three lobes of the spectra for each baseline coincide. This is because the signal duration is

only 20 seconds and this mean that we can see only limited frequency bandwidth of the chirp-like signal.

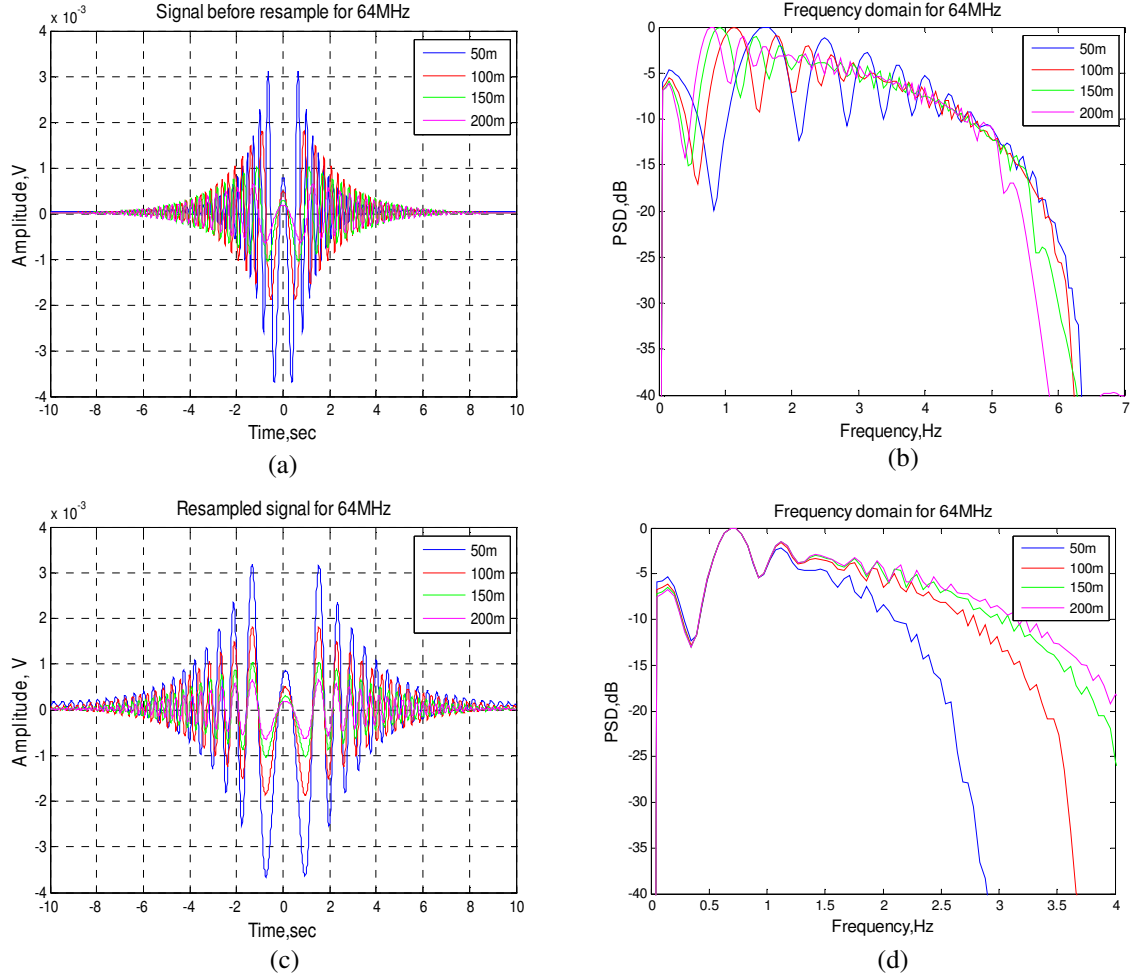


Figure 5.15 Signals for different baseline at 64 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling to baseline of 250m

In order to support the above explanation, the duration of the signal at 64 MHz when $f_s=100$ Hz is increased from 20 seconds (2000 samples) to 40 and 60 seconds (4000 and 6000 samples, respectively); the target visibility time is longer as in Figure 5.16 and the signal is effectively a longer chirp-like target signal. Hence, more precise information can be obtained. The purpose of increasing the duration is to prove that the resampling process could be improved for 64 MHz if the duration is increased.

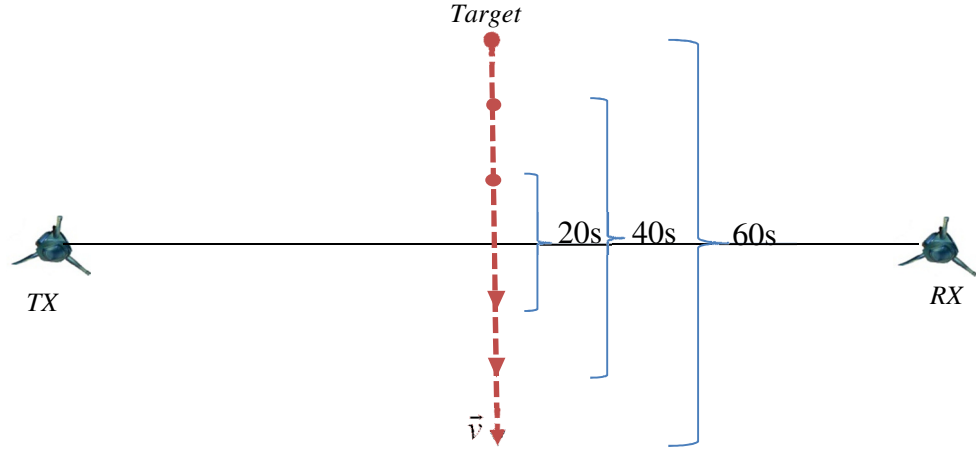


Figure 5.16 Target position for signal duration of 20s, 40s and 60s.

The resampled spectra for new signal durations are depicted in Figure 5.17. From this figure, we can see that higher repetition in the spectra lobes can be achieved between different baseline spectra as the signal duration increases. In comparison with Figure 5.15 (d), the spectra's lobes for all baselines coincide even more.

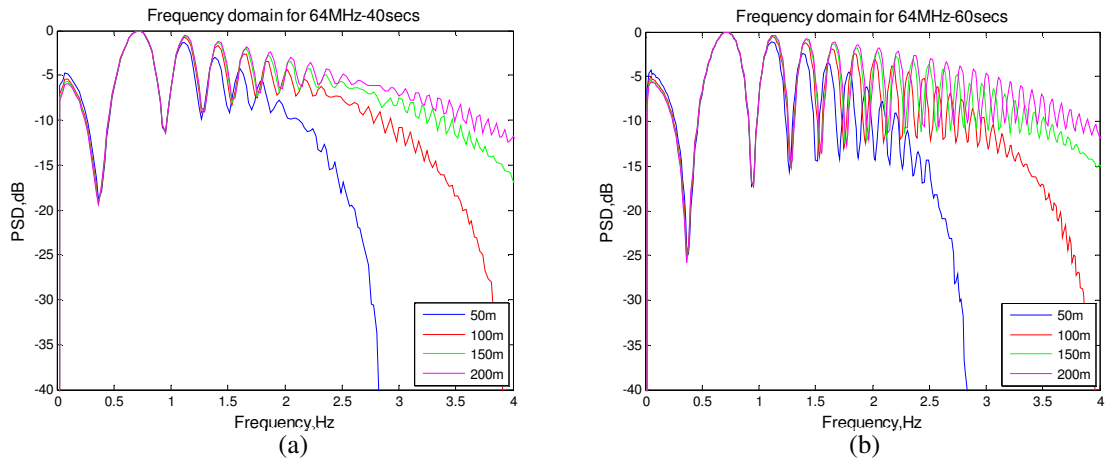


Figure 5.17 PSD after resampled to baseline when the signal's duration is increased from 20seconds to (a) 40seconds and (b) 60seconds

The same baseline influence in Figure 5.15 can be observed for carrier frequencies of 151 MHz and 434 MHz. The results are presented in Figure 5.18 and Figure 5.19. Again, the time domain and spectra are successfully resampled and this can be seen in (c) and (d). We see better repetition of the spectra's lobes especially at 434 MHz.

The resampling process at 434MHz is more successful compared to 64 MHz and 151 MHz due to the fact that more information can be obtained from a larger part of the high frequency in 434 MHz with signal duration of 20 seconds.

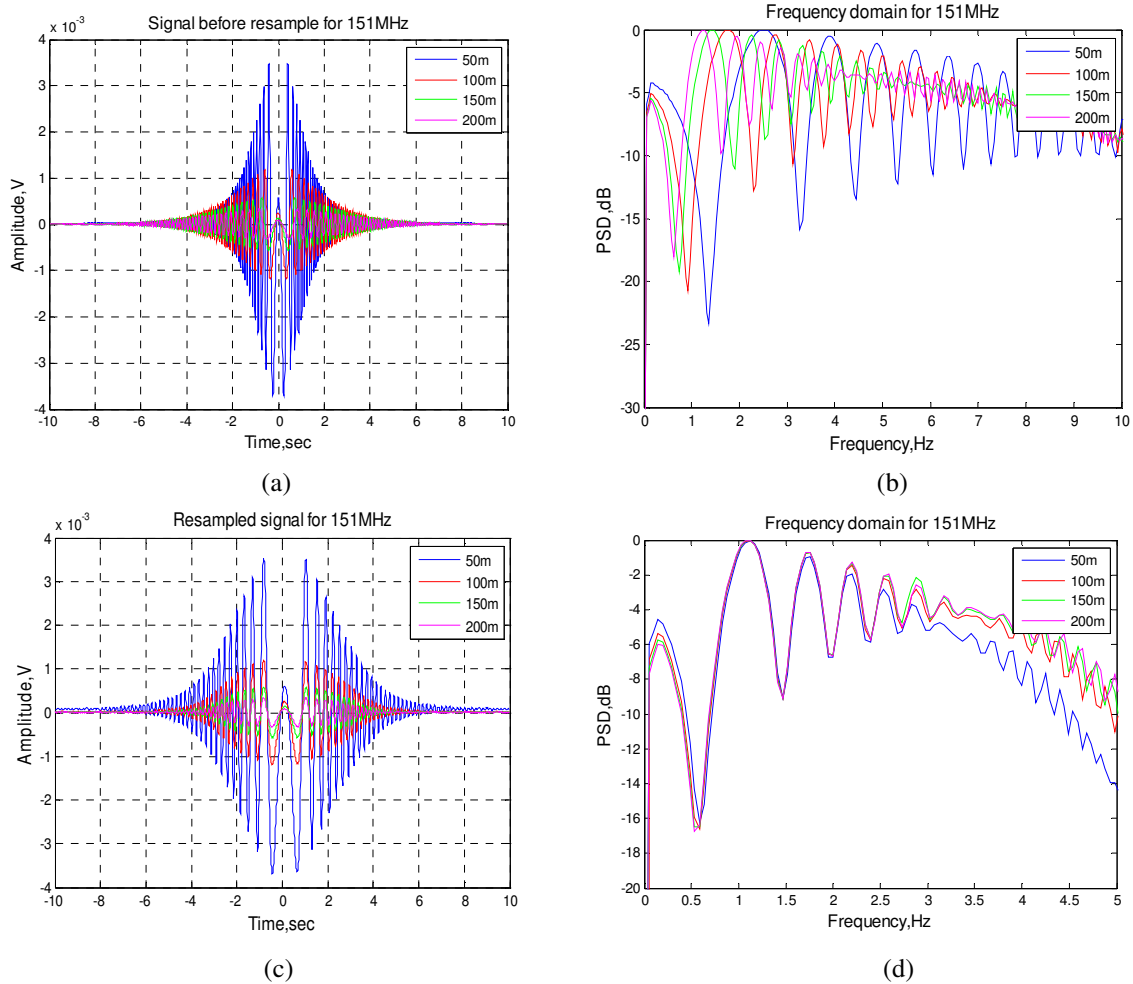


Figure 5.18 Signals for different baselines at 151 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling, for a baseline of 250m

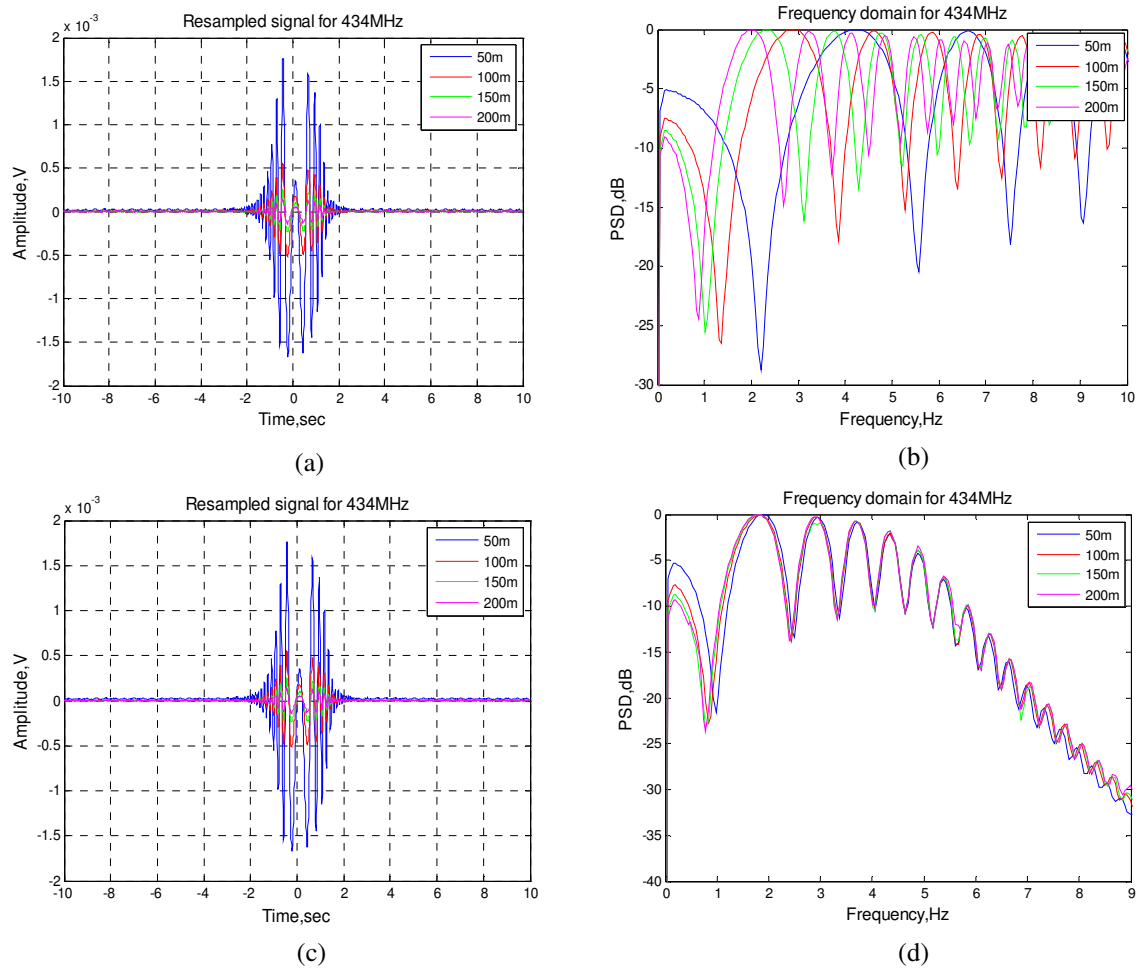


Figure 5.19 Signals for different baselines at 434 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m

- **Square target : 1m x 1m**

The second step is to evaluate the influence of the baseline when a square target with the dimensions of 1m x 1m is used. We can expect that with this electrical size of the target, the influence of the baseline and the effectiveness of the normalisation procedure become more evident.

First, the radiation pattern of the target is checked at all frequencies using our modelling software.

Simulated radiation patterns are presented in Figure 5.20. At 64 MHz, this target can be considered (to some extent) to be a point-like target because the radiation pattern is nearly isotropic – at forward the peak of RCS=0.75dB.

As the frequency increases, the RCS pattern is no longer isotropic, hence the target cannot be considered as a point-like target. For 151 MHz, the maximal RCS=1.8 dB and the same situation can be perceived at 434 MHz. At this frequency, the radiation is typically forward scattering where at the RCS=5.2dB. We shall stress here again that the electrical size of the plate corresponds to the Mei region of scattering because $\lambda \approx 0.67m$. It can be considered as optical scattering if and only if $D \gg \lambda$.

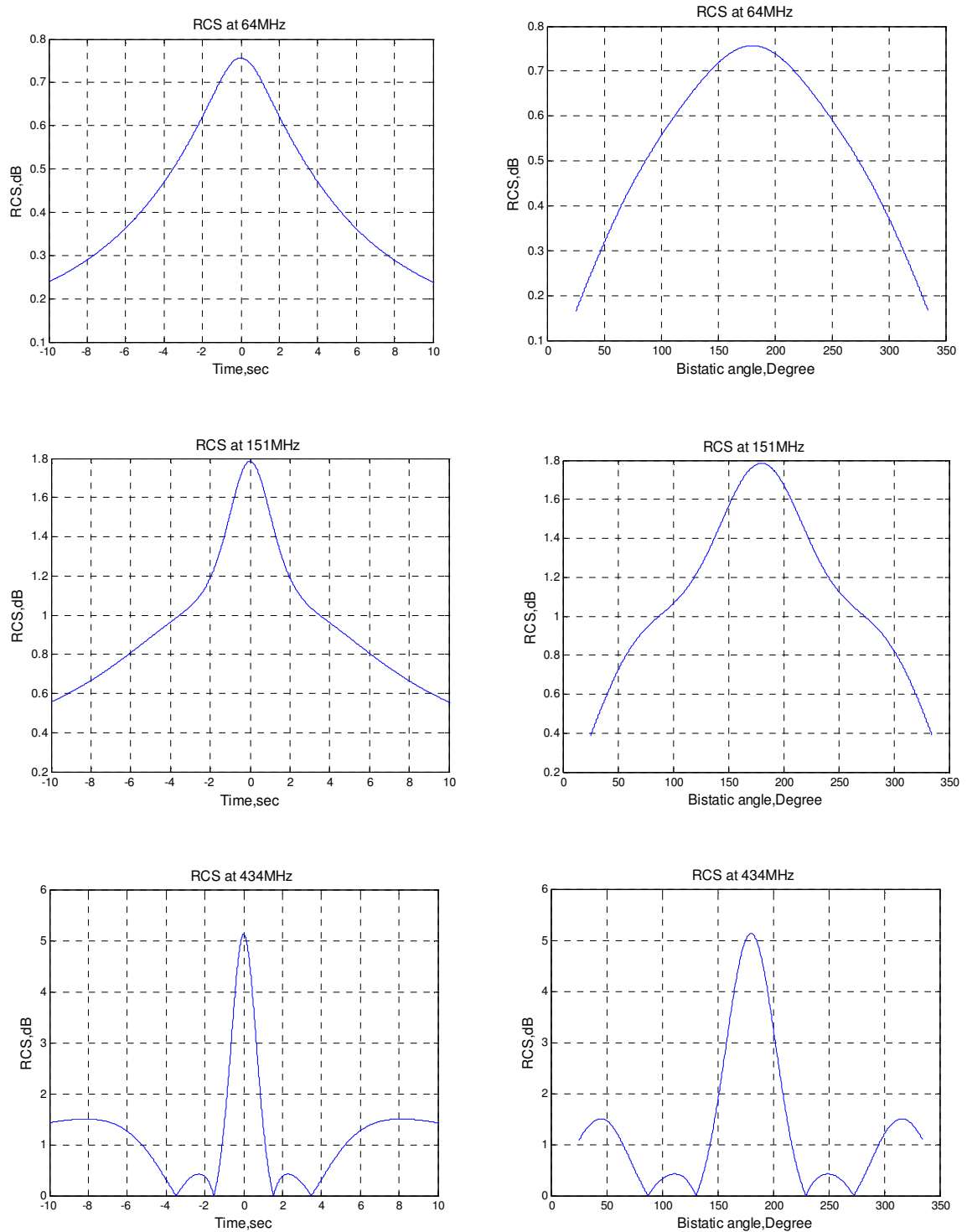


Figure 5.20 RCS radiation pattern for an extended rectangular shape target when crossing a baseline of 100m at a speed of 15m/s.

Figure 5.21, Figure 5.22, Figure 5.23 depict the time and frequency domain signals before and after the resampling process at 64 MHz, 151 MHz and 434 MHz respectively. The same result as found in the previous example can be seen for the time domain and frequency domain. We then apply the resampling process in the cases where the signal's amplitude and width varies relative to the baseline.

The spectra in Figure 5.21 (d) show the baseline's significant influence and pattern, as shown in Figure 5.15 (d), since both targets can be considered as isotropic cases. Again, due to the signal's short duration, we could not achieve full consistency in the spectra. For 151 MHz and 434 MHz, the normalisation works properly even though differences in the amplitude of spectra's side lobes can be observed.

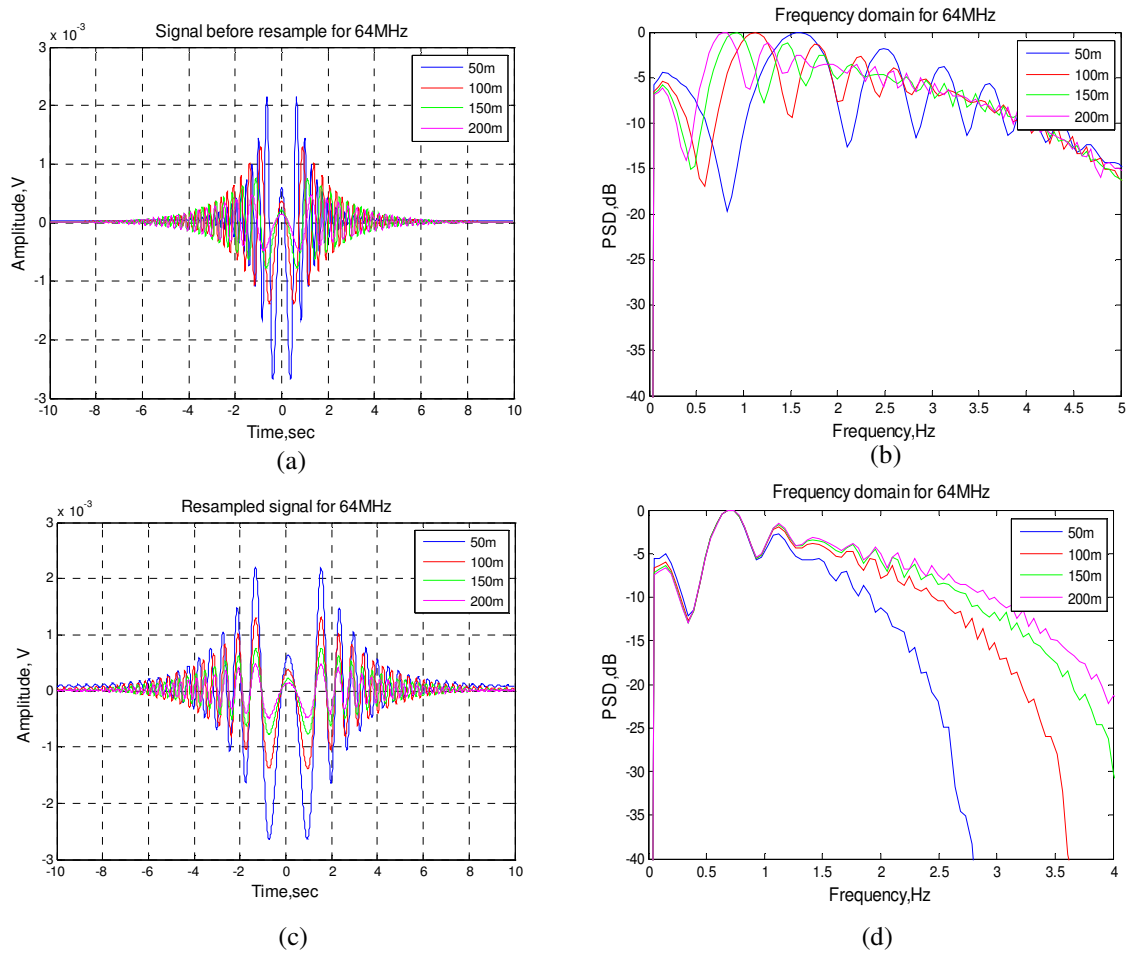


Figure 5.21 Square targets with 1mx1m dimensions for different baselines at 64 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m

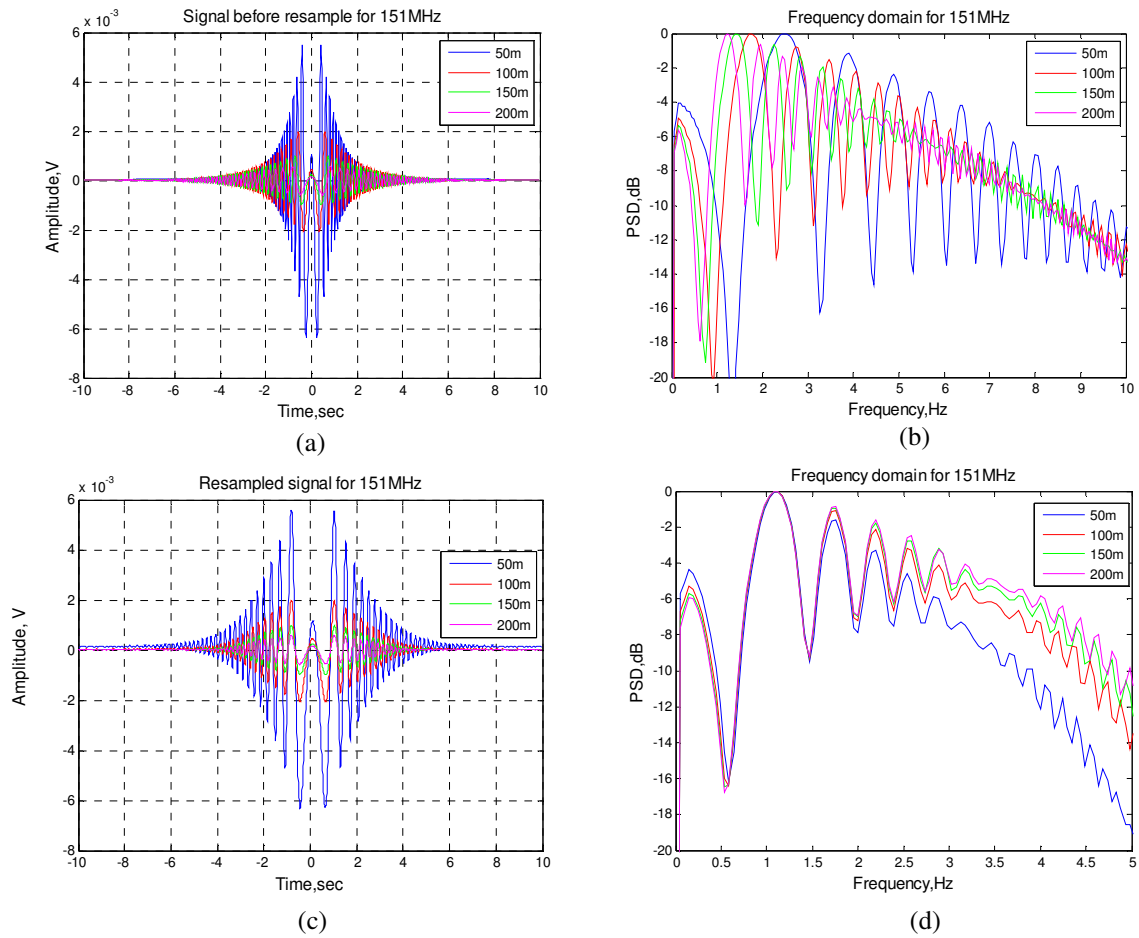


Figure 5.22 Square targets with 1mx1m dimensions for different baselines at 151 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m

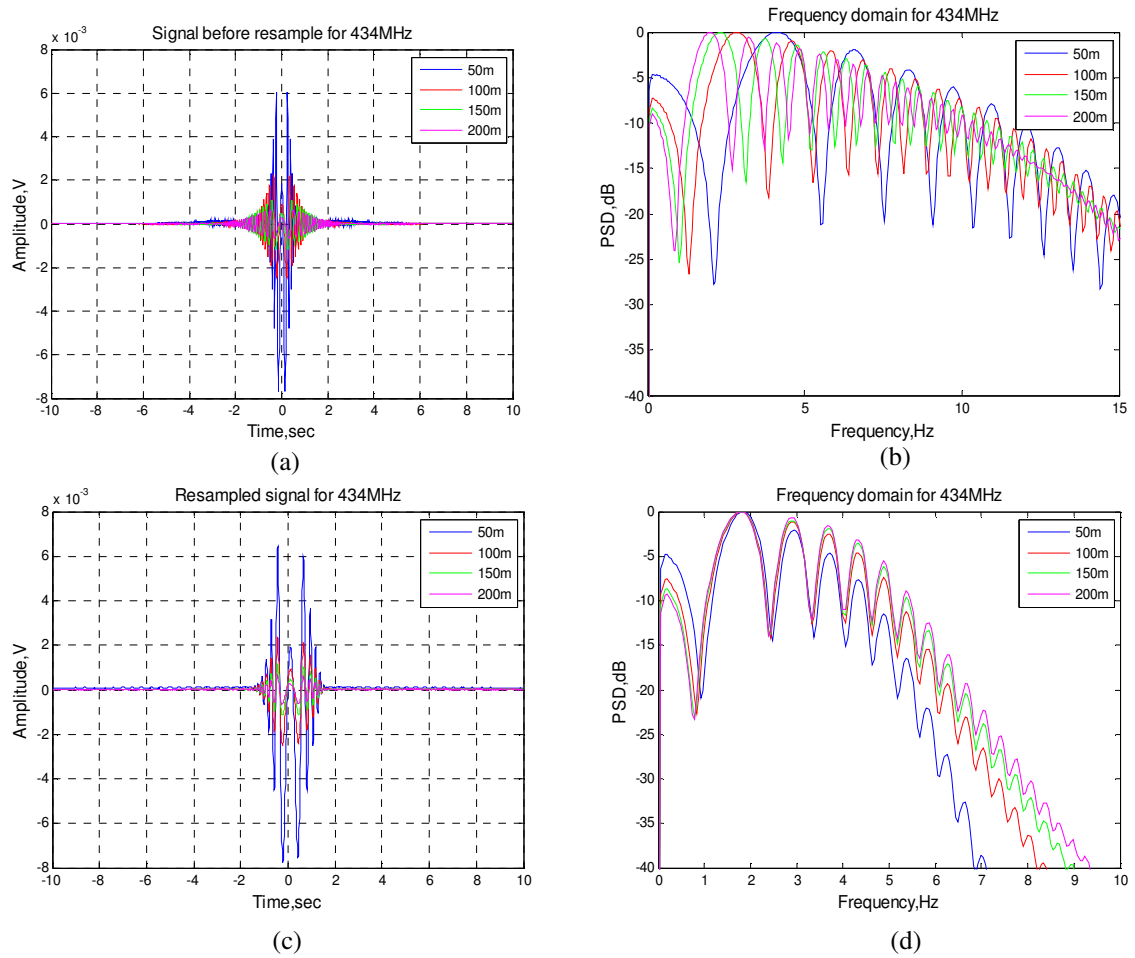


Figure 5.23 Square targets with a 1mx1m dimensions for different baselines at 434 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m

- **Simple approximation using rectangular aperture**

The influence of the baseline is evaluated once again but this time we use an extended rectangular aperture that has the dimensions of 4.8m x 2.2m (the dimensions are based on a Land Rover). The rectangular aperture target is illustrated in Figure 5.24. The speed was set to be 15 m/s with the same target trajectory as the previous two evaluations where the target is crossing in the middle of the baseline.

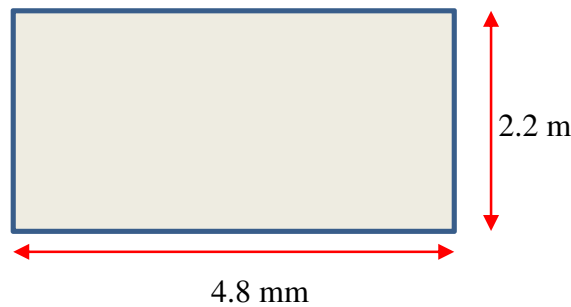


Figure 5.24 A rectangular plate target with dimensions of a Land Rover

Figure 5.25, Figure 5.26 and Figure 5.27 show the target's signatures and spectra before and after the baseline resampling process at various frequencies. If we refer to Figure 5.25 (a) and (b), at 64 MHz, it is hard to distinguish any signal difference between different baselines except for different scaling. However, after the resampling process, only the main lobe of the spectra has been resampled successfully; the power of the side lobes varies with distance, especially for 50m where the power of the first side lobe drops rapidly.

As for 151 MHz, even after the resampling process, we can see some discrepancies between resample spectrums. In the case of 434 MHz, the difference between each spectrum becomes more visible. Although the normalisation process managed to bring the trough of the main

lobe to a single point, we can still see the variation in the spectra shape and amplitude even at the main lobe as highlighted in Figure 5.27 (d).

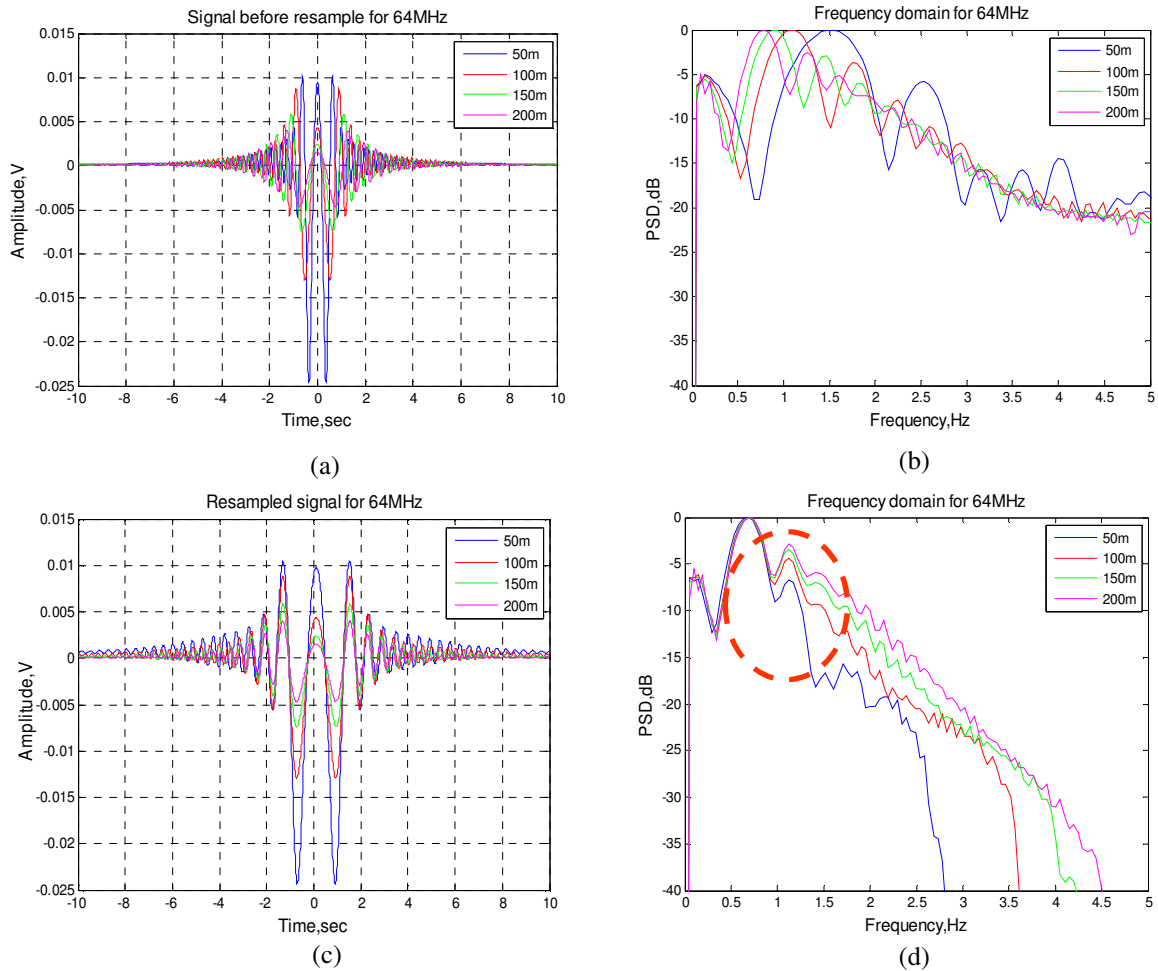


Figure 5.25 A complex target with dimensions of 4.71mx2.2m for different baselines at 64 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m

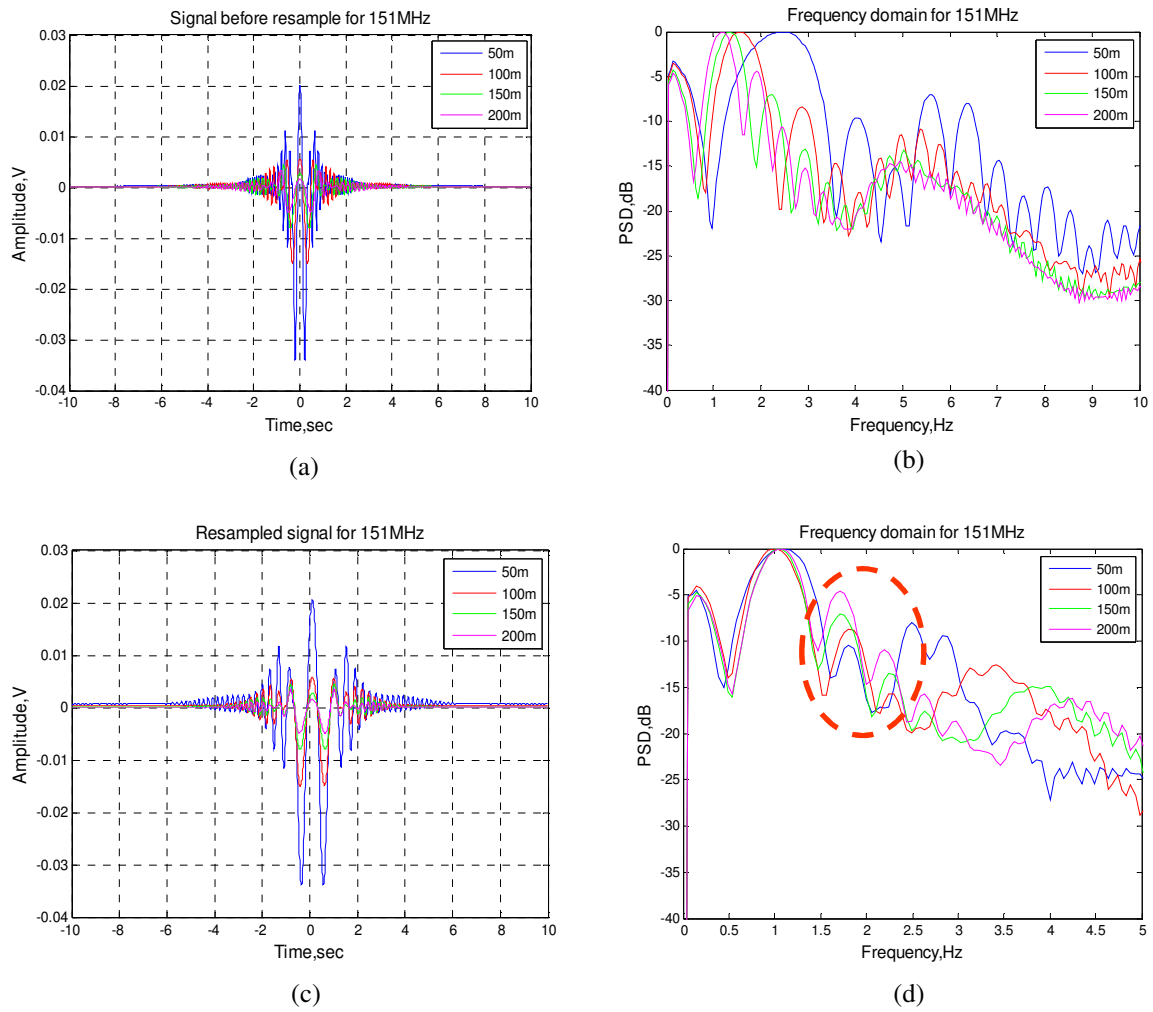


Figure 5.26 A complex target with dimensions of 4.71mx2.2m for different baselines at 151 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m

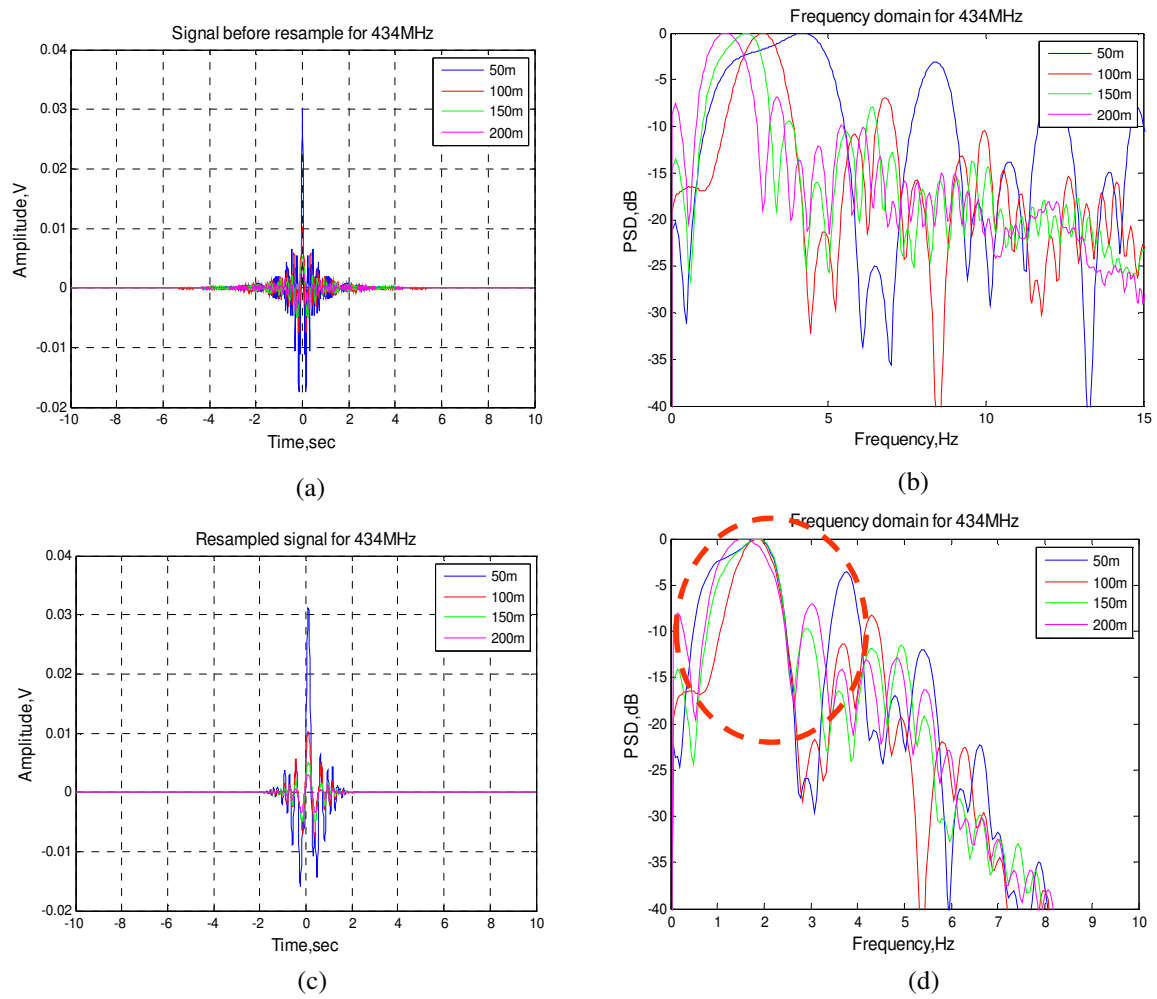


Figure 5.27 A complex target with dimensions of 4.71mx2.2m for different baselines at 434 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling for a baseline of 250m

A possible explanation can be made by looking at the RCS radiation pattern and also at the target's received power signal. In order to do this, the RCS radiation pattern for the rectangular shape is simulated using a signal modelling algorithm.

The RCS radiation pattern for all frequencies is shown in Figure 5.28. Unlike an isotropic target, the radiation pattern has a clearer main lobe and side lobes. The number of side lobes increases with frequency. Hence, when a target crosses baselines of different lengths, the receiver is most likely to receive a different part of a target's RCS.

Furthermore, the size of the 'dead zone' and the FSR detection varies with the baseline. This is illustrated in Figure 5.29. As we can see, area A is greater than area B, which means that when the target is passing through area A, the target is invisible for a longer time compared with area B.

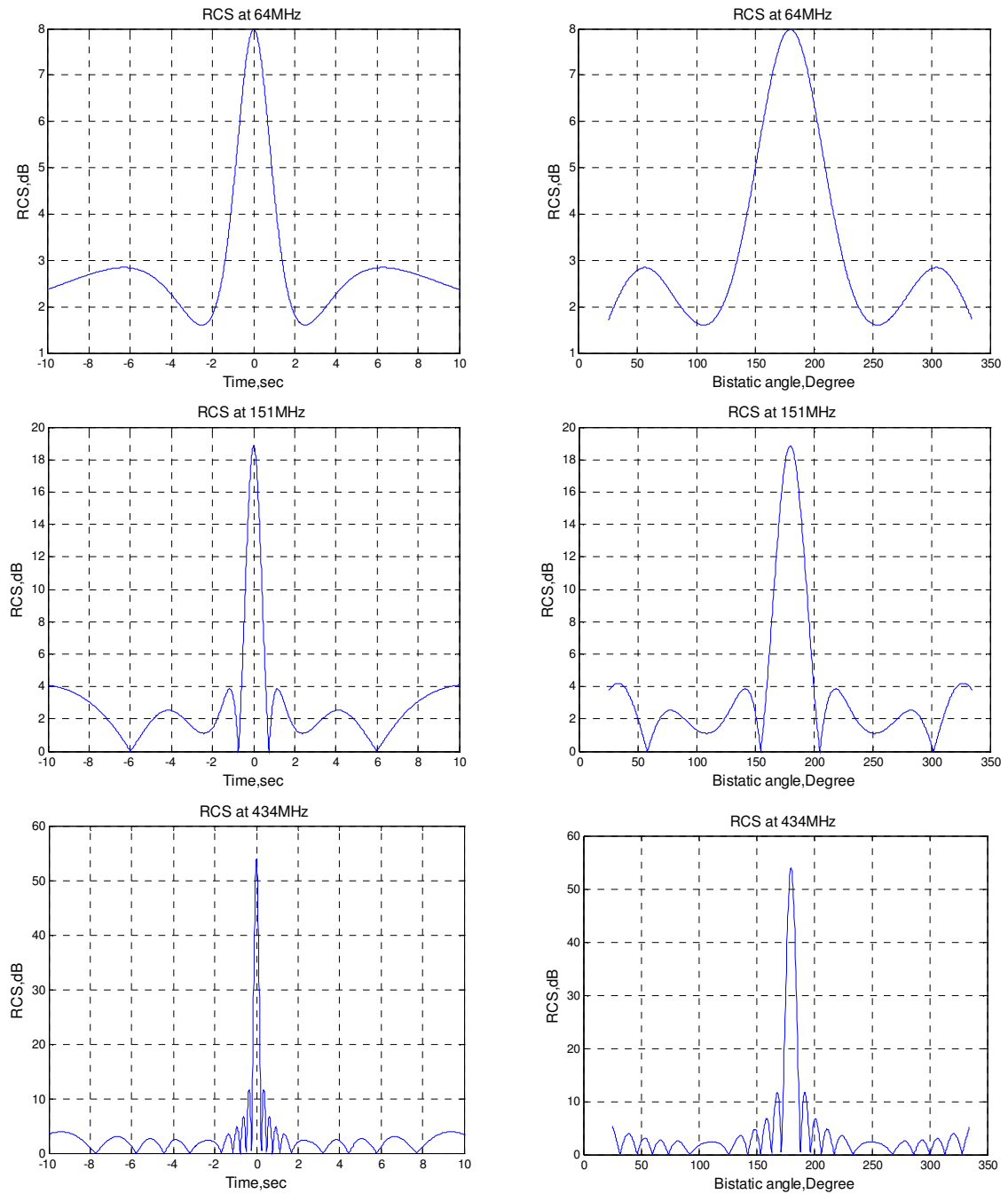


Figure 5.28 The RCS radiation pattern for an extended rectangular shaped target when crossing a baseline of 100m with speed of 15m/s.

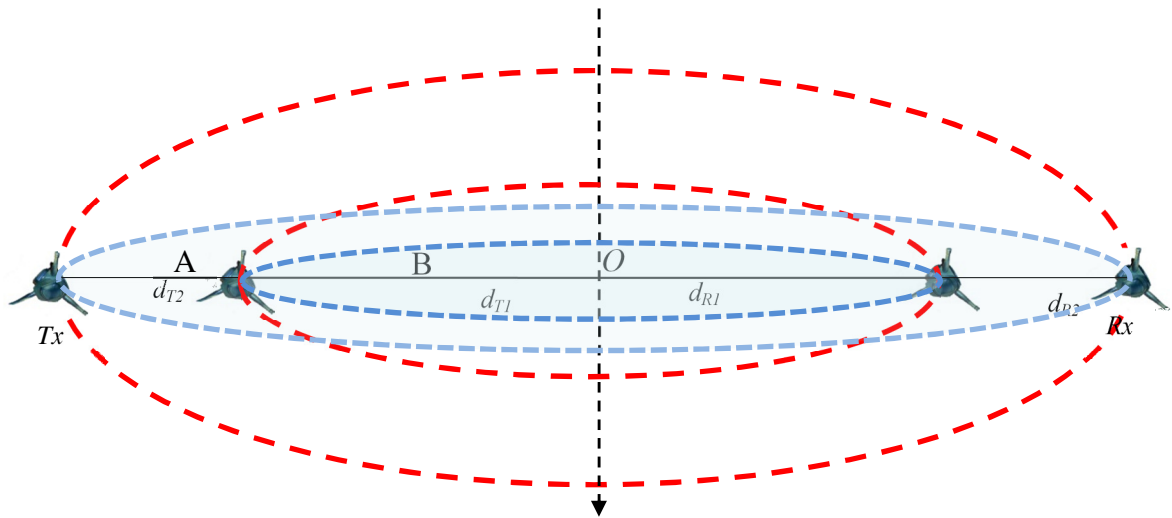


Figure 5.29 Dead zone for different baseline lengths

(— — — — — dead zone and — — — — — FSR detection zone)

5.4.2 Measured signal

The efficiency of the baseline normalisation procedure is verified using a measured signal. The experiment was conducted at Tilesford Airfield with four different baseline lengths using a Land Rover. The target crossed in the middle of the baseline with a speed of approximately 6m/s. The measured target signals in the time and frequency domain before and after the normalisation procedure are presented in Figure 5.30, Figure 5.31 and Figure 5.32 at 64 MHz, 151 MHz and 434 MHz respectively.

As we can see in (c), the time domain signals after normalisation are similar even for different baseline lengths except with time shifted and different maximum amplitude which due to the received power drop. This is reflected in the spectra in (d). The main lobe of the spectra gives a good repetition for different baseline lengths at all frequencies. This shows that the

normalisation procedure works successfully and is consistent with previous evaluation results from the simulated signal, especially for the extended rectangular plate.

However, the dissimilarity and variation in amplitude and shapes of the spectra side lobes can be clearly observed, even after the baseline normalisation procedure, especially for high frequencies. This is due to the fact that different parts of a target radar cross section radiation pattern are received by the receiver. As the baseline length increases, the system will have a more forward scattering effect. The result also suggests that the spectra are more corrupt, probably due to clutter and reflection from the surroundings.

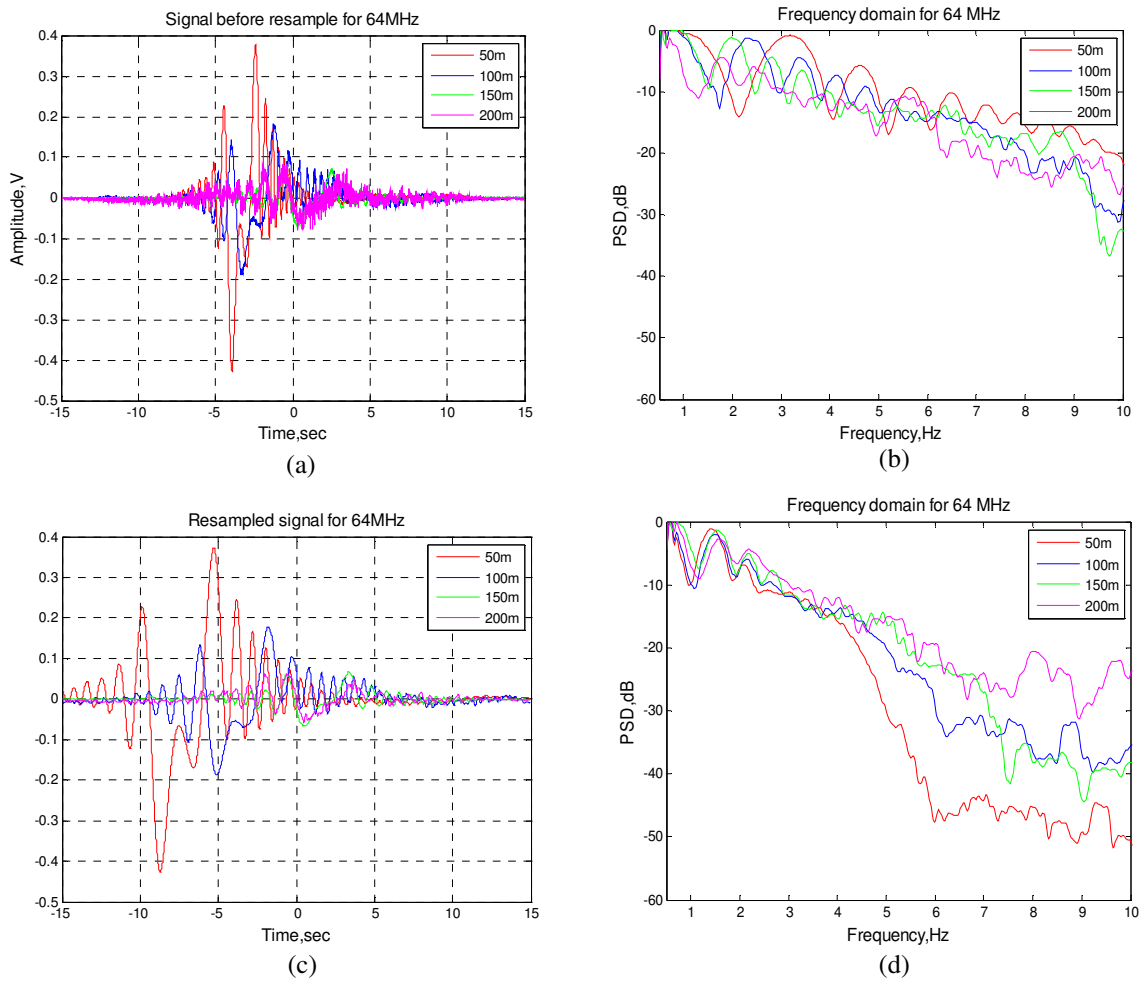


Figure 5.30 Measured signals at 64 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling

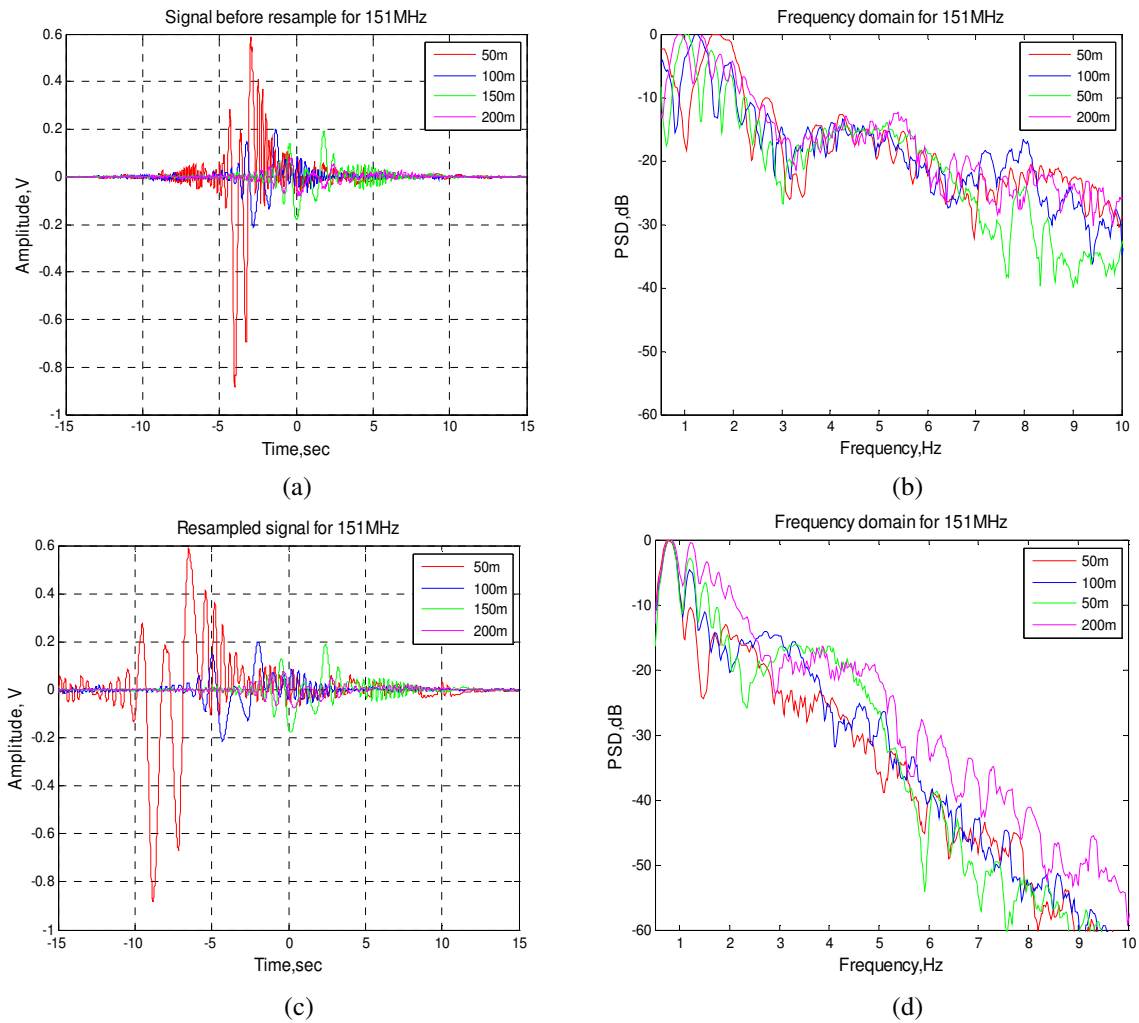


Figure 5.31 Measured signal at 151 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling

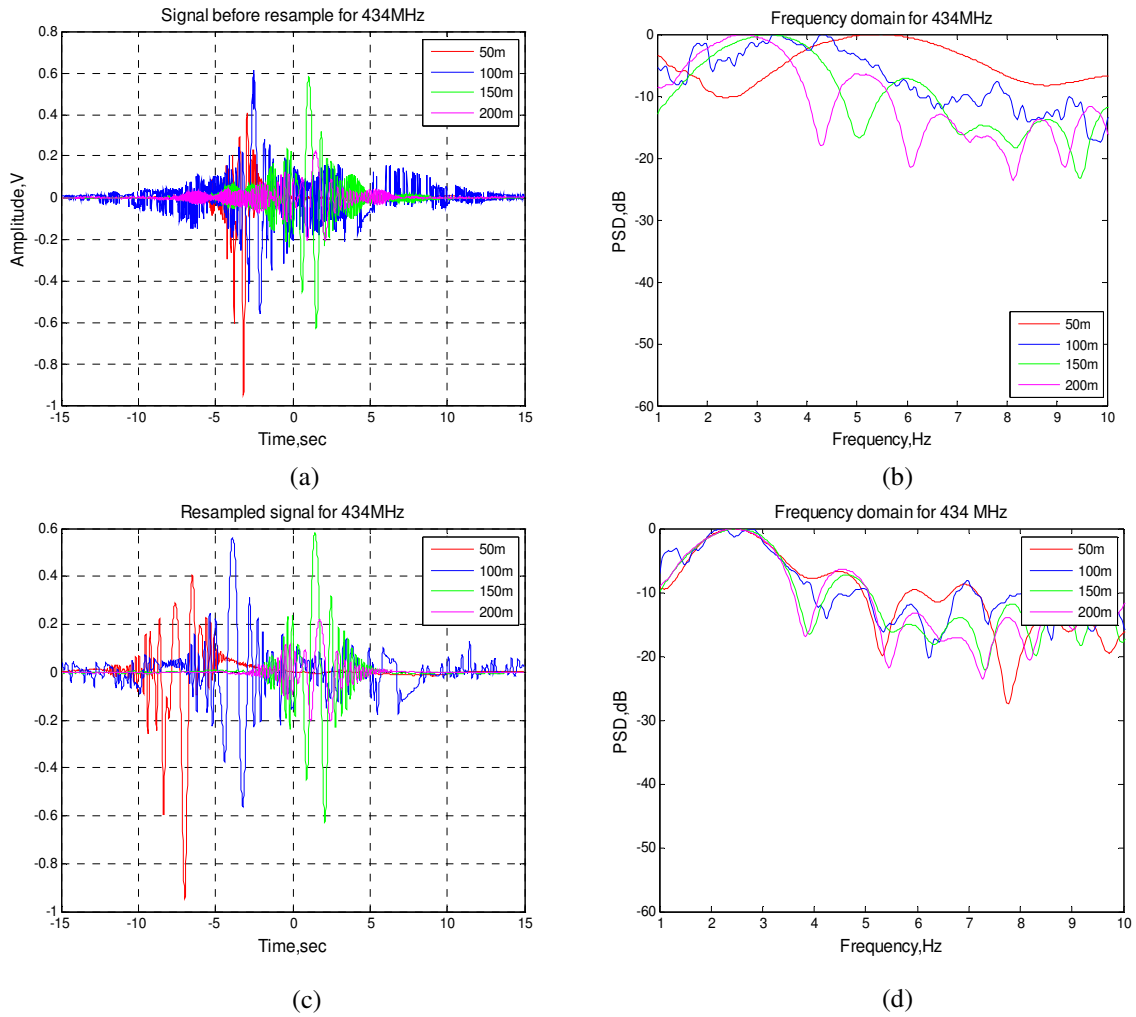


Figure 5.32 Measured signal at 434 MHz (a) time domain signals before resampling, (b) frequency domain signals before resampling, (c) time domain signals after resampling and (d) frequency domain signals after resampling

2D-PCA plots for different baseline lengths for a measured signal at all frequencies are depicted in Figure 5.33. As shown, we can see a separation between the plots for different baseline lengths. This separation is due to the small number of signals that we recorded and also the influence of other factors, such as the increment of clutter, as the baseline lengths increased.

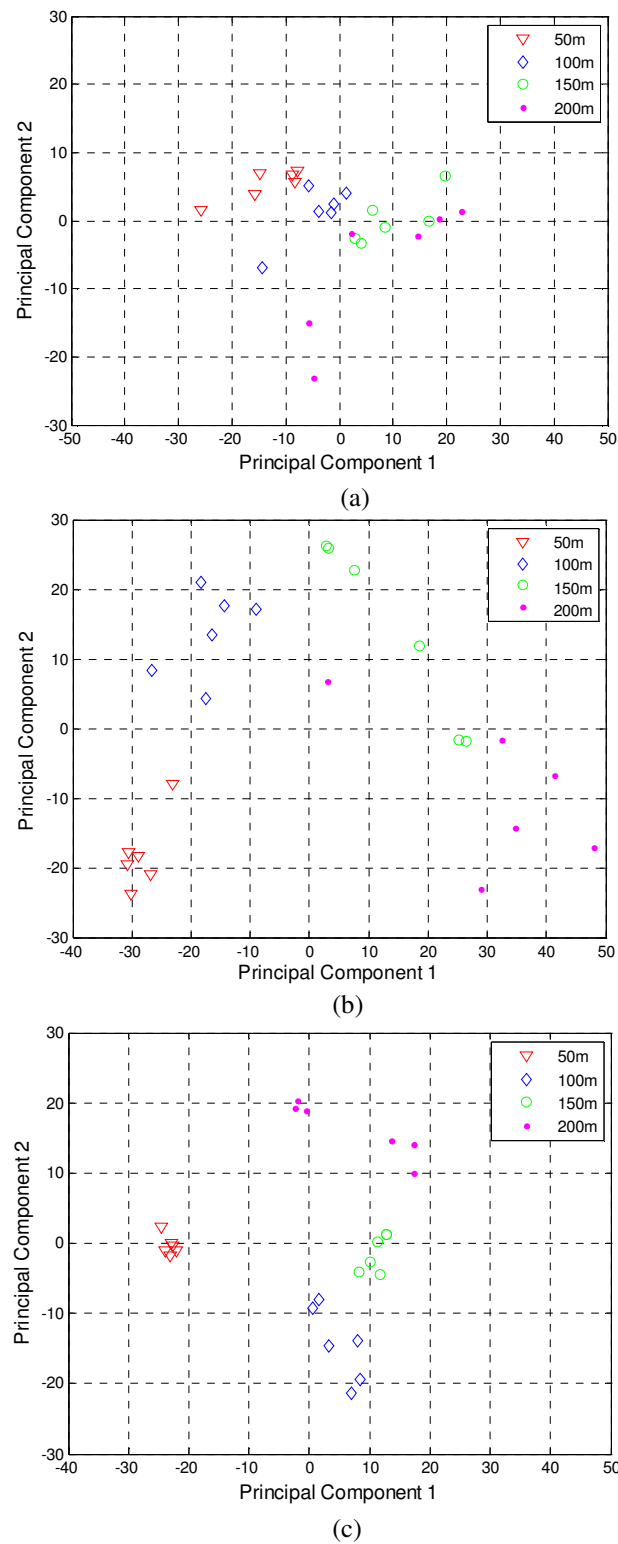


Figure 5.33 2D PCA plots for different baseline lengths using measured signals at (a) 64 MHz, (b) 151 MHz and (c) 434 MHz

Due to the limitation of data collected for different baselines, we could not proceed with the classification on the basis of statistical data. Furthermore, further research is required to investigate this effect and any possible solutions.

5.5 Conclusion

In the first part of this chapter, the process to extract the motion parameter using a coherent processing procedure was explained. Verification has been undertaken for simulated and measured signals. This showed that, in cases where the signal is assumed to be clutter free and the speed from the video is accurate, the speed estimation algorithm achieved accuracy in excess of 90%. The estimated speed was then used as an input for Doppler target signature normalisation. The effect of this normalisation on the classification process was analysed.

The effect of the speed estimation on the target classification was analysed for FSR at three different frequencies in clutter free and cluttered environment. For clutter free situations, an error is generated using Gaussian random variable and this causes the degradation in ATC performance. In a cluttered environment, the signal is embedded with simulated clutter. The evaluation results suggest that a higher frequency gives a higher level of accuracy in estimating the speed (due to the larger HPF cut-off frequency). However, in terms of classification, a speed estimated at 64 MHz gives superior classification performance. These findings suggest that, in general, classification depends on the accuracy of the speed estimation algorithm; the classification accuracy decreases proportionally with the increment of error in the estimated speed.

In the second part of this chapter, the effect of different baseline lengths was analysed. Studying the performance of the system with different baseline lengths for an isotropic target showed that, after applying a baseline normalisation process, the system can achieve spectral stability at various frequencies. However, as the target approaches optical approximation, the differences in the target's spectra for different baselines become more evident.

Without doubt the results themselves could be improved and should be the subject of further study. A better resampling factor should be found in order to be able to place the signal on a single common timescale.

CHAPTER 6

ATC PERFORMANCE DEGRADATION IN CLUTTERED ENVIRONMENT

6.1 Introduction

Doppler signal is produced when the target is crossing the transmitter-receiver baseline. This Doppler signal is a combination of scattered signal from target that may include both bistatic and forward scattering components, reflections from nearby objects and clutters. When radar site is surrounded by vegetation, the foliage and branches sway with the wind; this will create clutter masking the useful target signals. This masking will give problem and cause errors in the target detection and classification.

In our case, the omni-directional sensors are positioned on the ground with baseline of hundred(s) meters. Vegetation may present near or surrounding the baseline region, hence, clutter is picking up from a large volume.

The goal of this chapter is to analyse the performance of ATC system in the presence of clutter. This chapter will start with the description of possible sources of errors in cluttered environment. This is followed by the explanation on how ATC degrade with the presence of these possible errors. Lastly, a clutter compensated ATC system is introduced at the end of this chapter.

6.2 Possible source of errors in cluttered environment

The presence of clutter in the target signal can potentially introduce errors in two stages in the ATC system: the speed estimation error and the signature deformation. The speed estimation error is the error when the speed is wrongly estimated. As mentioned earlier in Chapter 2, the signal spectrum width depends on the speed of a target. Hence, it is important to see if the presence of clutter can affect the accuracy of the estimated speed. Therefore when speed normalisation process is applied, the spectrum might not be properly rescaled due to the erroneous ratio between the estimated speed and the reference speed. As a result, the spectrum is incorrectly normalised and affecting the classification process.

Clutter level varies with the environment and carrier frequency of the system. The higher the frequency, the more sensitive the system becomes to the presence of clutter. As the clutter level increases, the shape of the signature becomes more distorted. An example of spectra at

the receiver output with different level of clutter is shown in Figure 6.1 after passing through software filter. The same signal is corrupted by simulated clutter with different level of SCRs: 5 dB, 10 dB and 20 dB. As depicted, the fluctuations on the amplitude of the signal become noticeable at low frequency part as the SCR decreases.

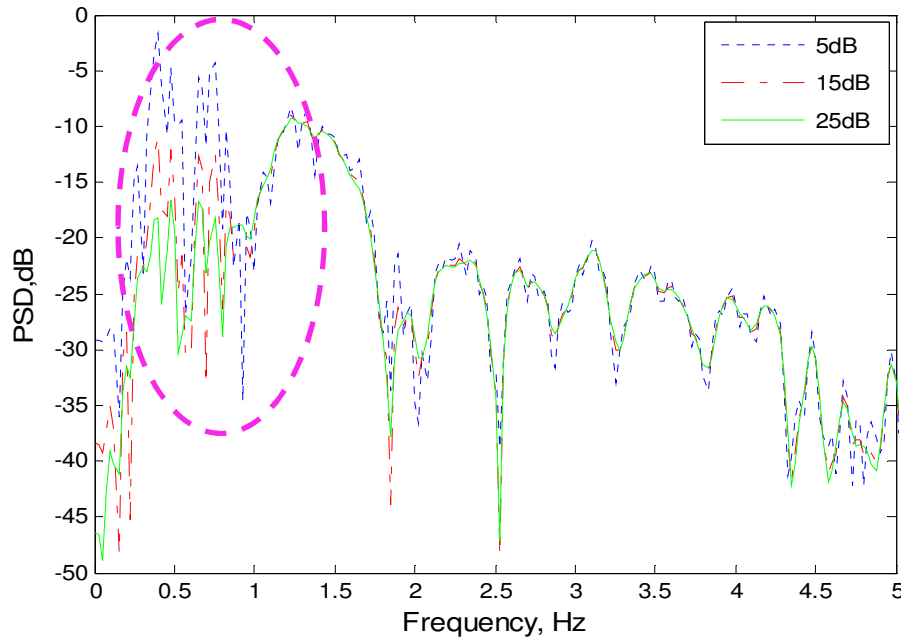


Figure 6.1 Spectra of the same signal with three different level of clutter strength: SCR=5 dB, SCR=15 dB and SCR=25 dB.

Figure 6.2 shows the signal processing block diagram with added simulated clutter. In real world scenario, cluttered received signal is used to estimate target's speed and as the input to classification process. The presence of clutter in the signal causes errors in speed estimation and shape of the target's spectrum. Therefore, in order to evaluate the effect of errors independently, simulated clutter is added separately at two different places as shown in Figure 6.2: before power spectrum calculation is made and before estimating the speed of the target. For the evaluation, we divided the errors into three categories:

- i. Error due to the estimated speed only

Clutter is added to the time domain signal only before estimating the speed. During the normalisation process, the time domain signal is used with the obtained corrupted estimated speed.

ii. Error due to spectrum's shape deformation only

In this case, we assumed that there is no error occurs during speed estimation process simulated clutter is only added before the pre-processing stage.

iii. Error due to estimated speed and shape deformation

The error is created by combining both procedures above where simulated clutter is added to time domain signal before estimating target's speed and pre-processing stage.

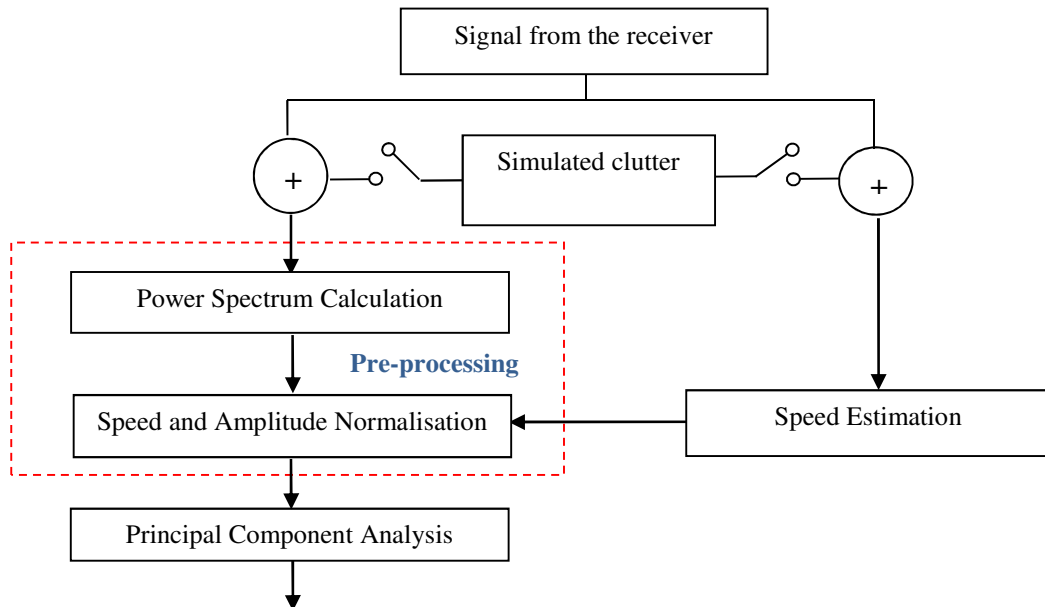


Figure 6.2 Signal processing block diagram with added simulated clutter

The original spectrum and the example of spectra for all three types of errors for SCR =10 dB at 151 MHz (calculated before passing through software filter) are shown in Figure 6.3. When the error is due to the shape deformation, the spectra's (pink and red lines) are distorted and the amplitude of the spectra is fluctuated due to the clutter presence. However, when only

speed estimation error exists (green line), the amplitude of the spectrum is not affected by clutter; the spectrum is similar to the original spectrum (blue line).

As mentioned earlier in Chapter 2, the width of the spectrum depends on the speed of the target. Hence, when an error due to the speed estimation exists, the width of the main lobe is slightly shifted from the original position. As highlighted in Figure 6.3, the trough points of green and pink lines are slightly shifted from the original signal (blue line) and only spectrum's shape deformation error exists (red line). When both types of errors spectrum's shape deformation and speed estimation exist in the spectra, there is fluctuation in the amplitude and the first null is also distorted as demonstrated by the pink line.

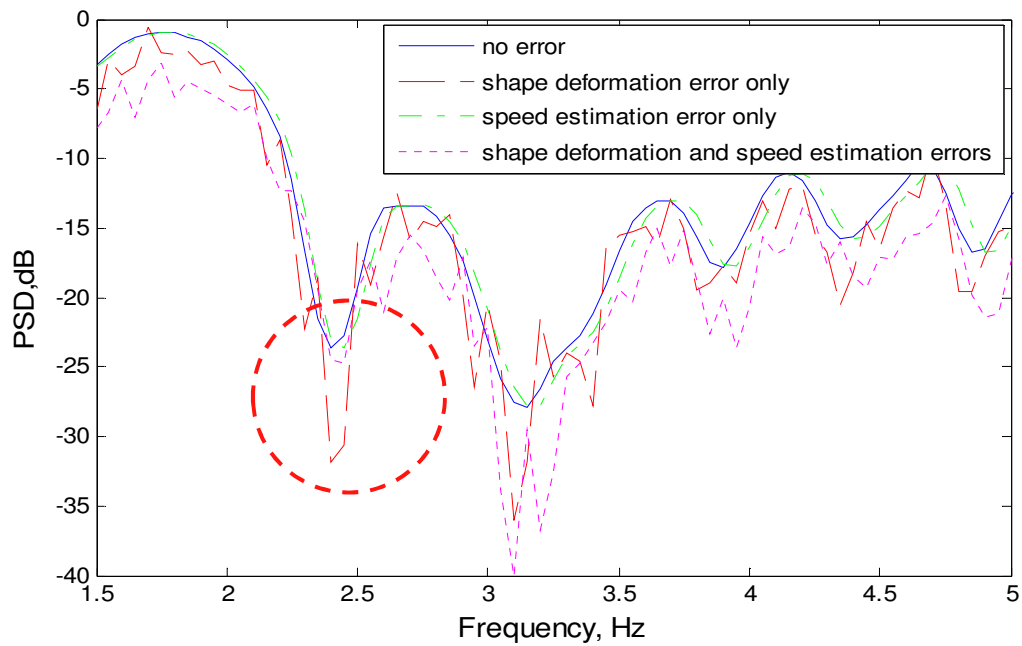


Figure 6.3 Spectra of different types of errors with SCR=10dB at 151 MHz

In the reported experiments, the ATC system employed three PCA coefficients and the three-nearest neighbour classifier. As previous chapter, clutter is simulated with SCRs from 0 dB to 30 dB, in 5 dB steps. The process of estimating the SCR has been described in section 5.3.2 where the ratio of signal and noise energy is defined after filtering both signals.

Figure 6.4 shows the effect of the clutter (in this case we are using the signal with error in spectrum's shape deformation only) on the PCA components at 151 MHz. As the SCR level decreases the spread of the PCA components for each type of targets become wider and consequently causing more overlap between the different targets. For the sake of better visualization we are using 2 PCA components and the ATC is based on 3 PCA components. The examples of PCA plots when the clutter causes the spectrum's shape deformation error at 64 MHz, 151 MHz and 434 MHz for SCR= 10 dB are depicted in Figure 6.5.

6.3 ATC degradation with the speed estimation error and spectrum's shape deformation error

This section presents the classification accuracy results of ATC system for all frequencies when subjected to the speed estimation and spectrum's shape deformation error.

Figure 6.6 (a) presents the results obtained when the clutter affected only the speed estimation algorithm. It can be seen that the classification accuracy is nearly the same when SCR is in the range from 5 dB to 30 dB and decreases more only when the SCR is below 5 dB. This demonstrates that the speed estimation algorithm is reasonably robust to the clutter.

Classification results obtained when the clutter causes the spectrum's shape deformation error only, and both the spectrum's shape deformation and the speed estimation error are depicted in Figure 6.6 (b) and Figure 6.6 (c), respectively. In both cases, the obtained results are very similar and have the same slope trend, which shows that the spectrum's shape deformation error is the main cause of degradation of ATC performance. It can be seen that frequency 151 MHz provides the best performance; the classification accuracy is 97.25% at SCR of 30 dB and is nearly not affected till the SCR of 15 dB but then decreases sharply when SCR is below 10 dB. The frequencies 64 MHz and 434 MHz provide in most cases much worse performance than the frequency 151 MHz. However, for SCR=0dB, the results suggest that 434 MHz give the best result for all cases. This is probably because when bigger clutter is added to the signal, the spreading of the cluttered data increases. Despite that, as shown in Figure 6.5, the plots for different classes of target are not perfectly separable in 434 MHz in comparison with 64 MHz and 151 MHz. Hence, the chances of the testing signal are classified correctly especially when the number of k is smaller. However, this is subject to further investigation.

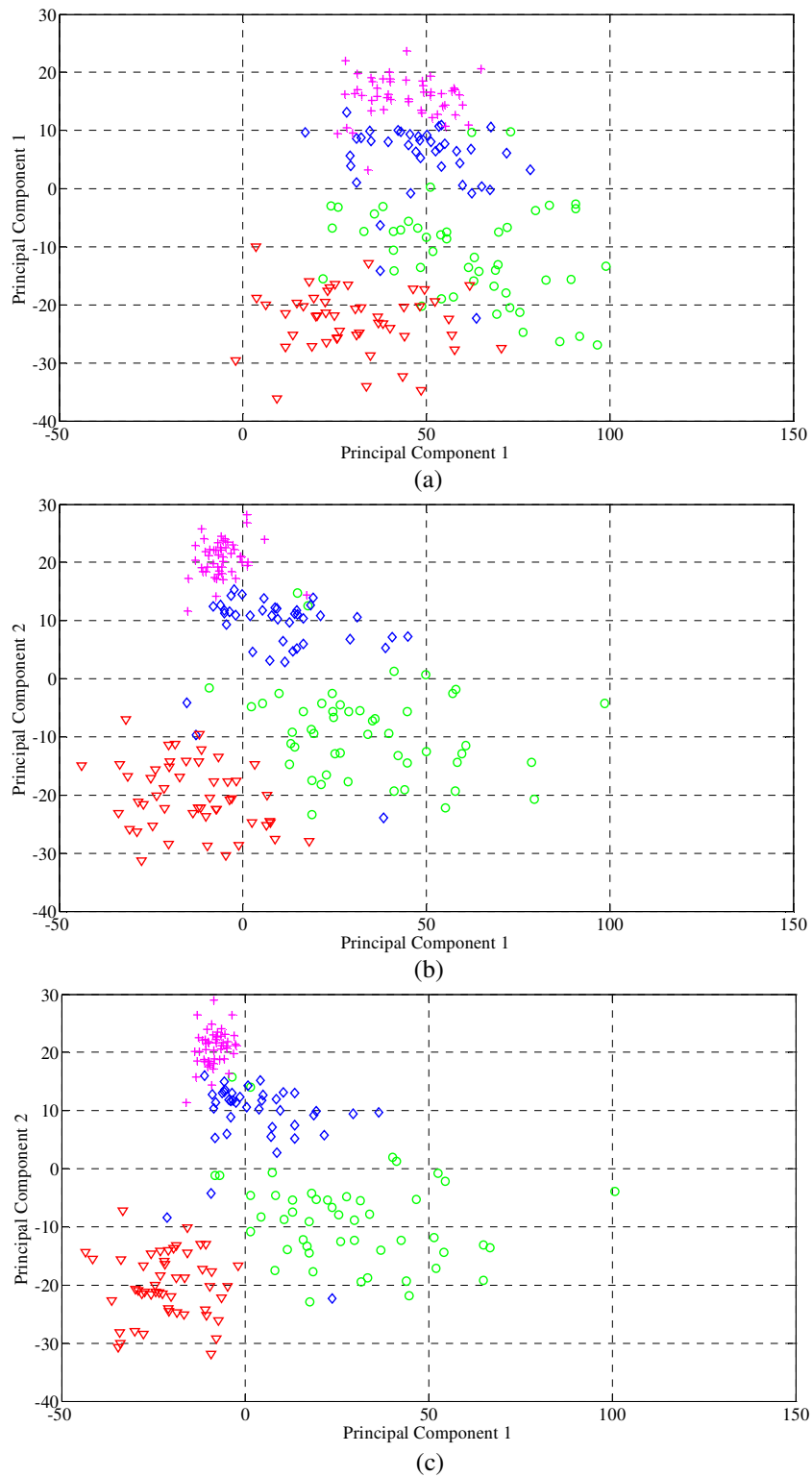


Figure 6.4 PCA plots for different level of SCR when only shape deformation error presence at frequency 151 MHz when SCR (a) 5dB (b) 15dB and (c) 30 dB
 Notation: Landrover (∇), Nissan (\diamond), BMW (\circ) and Fordfocus (+)

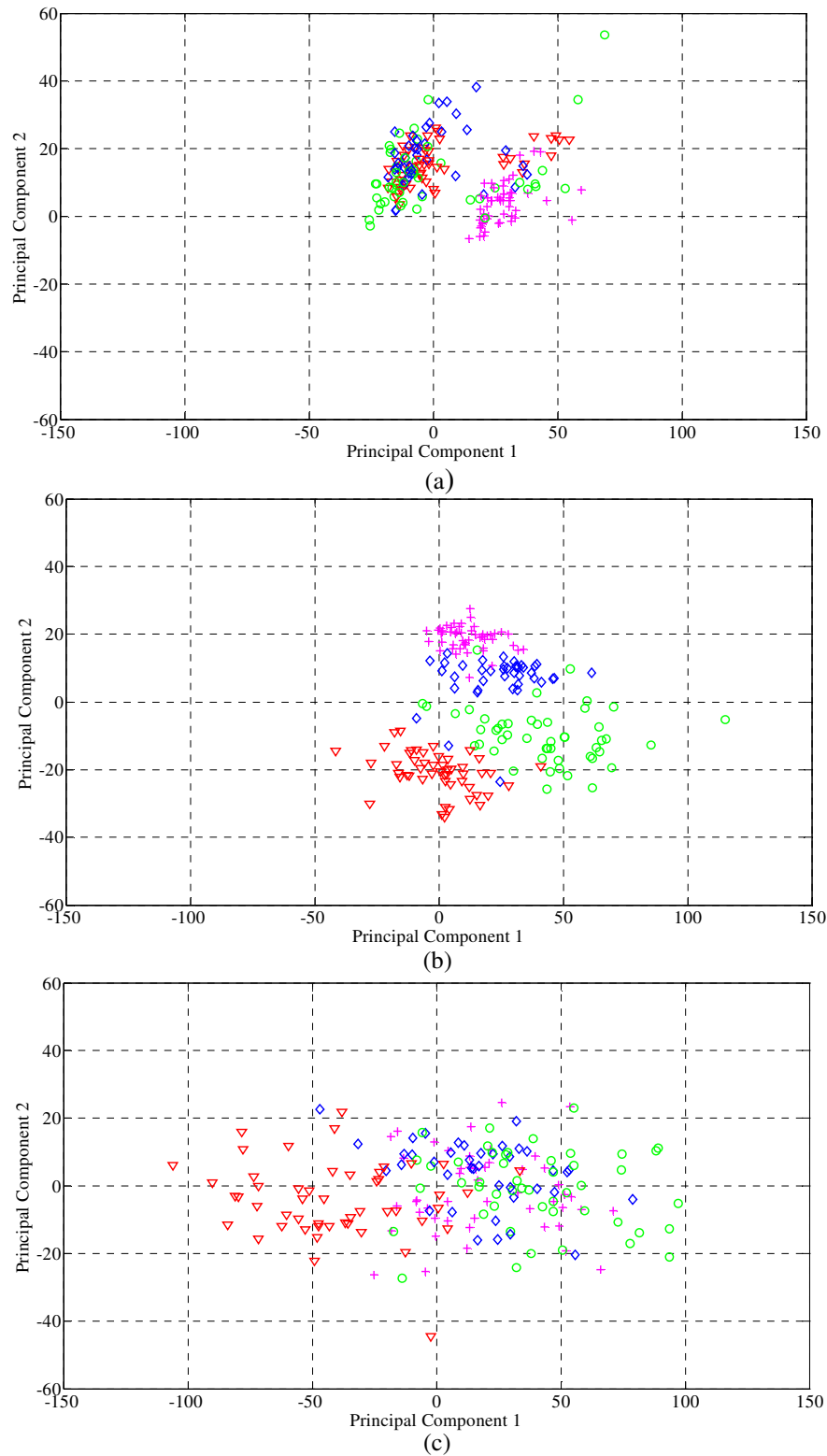
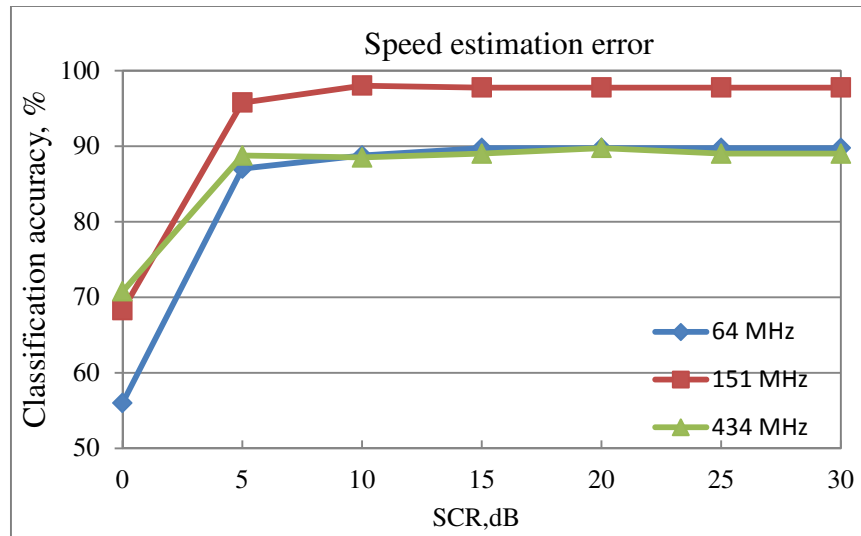
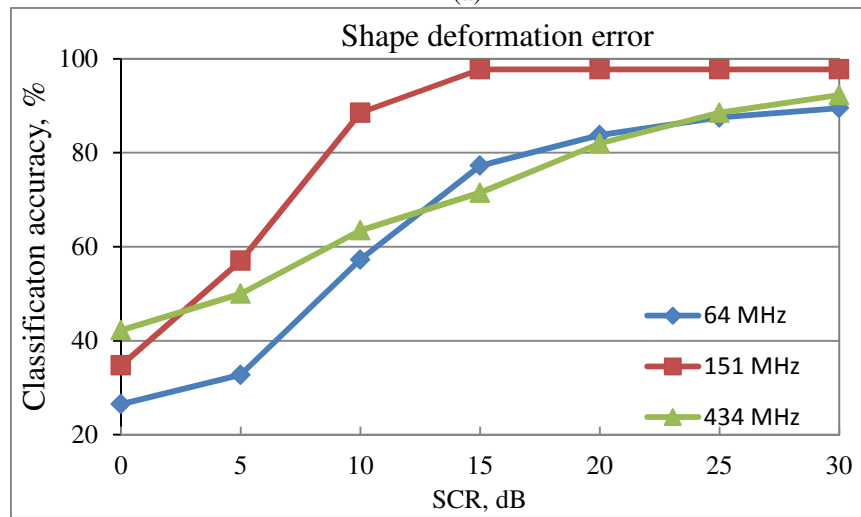


Figure 6.5 PCA plots for SCR=10 dB when only shape deformation error presence at various frequencies (a) 64 MHz, (b) 151 MHz and (c) 434 MHz

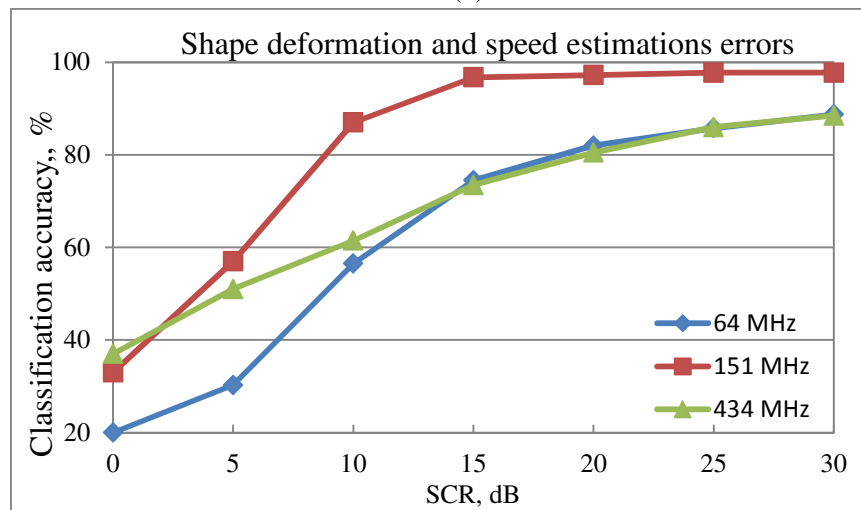
Notation: Landrover (∇), Nissan (\diamond), BMW (\circ) and Fordfocus ($+$)



(a)



(b)



(c)

Figure 6.6 Target classification accuracy

6.4 Clutter-compensated ATC system

The performance in any recognition system degrades rapidly as the mismatch between training and test conditions increases. A lot of efforts have been put to compensate the effect of noise in various areas of pattern recognition, for example, in the area of automatic speech recognition [67-72]. Varieties of methods have been proposed to overcome the effect of noise. They can be categorized into three categories:

i. Filtering the noise prior to classification

This approach is also known as signal enhancement. The purpose is to attenuate the noise in the signal by passing the signal through a filter, whose characteristics are adapted over time based on an estimate of the noise. The enhancement is typically performed in a transform domain. There has been a variety of techniques proposed, for instance, Wiener filtering [73, 74], signal subspace-based enhancement [75], Sparse Code Shrinkage [76, 77]. Wiener filtering was among the first enhancement techniques introduced and remain popular today due to its reasonable performance and low computational complexity. It performs the enhancement of the noisy signal in the DFT-based transform domain by using a filter $H(\omega)$, derived based on the minimum mean-square error criterion,

$$H(\omega) = \left[1 + \frac{P_n(\omega)}{P_s(\omega)} \right]^{-1} \quad (6.1)$$

where $P_s(\omega)$ and $P_n(\omega)$ are the spectral densities of the clean signal and noise signal, respectively.

The signal enhancement approach has not been used in this work to compensate for the clutter.

ii. Model-based noise compensation

The effect of noise can be compensated within the pattern recognition system if we know how the signal features are affected by the noise. The compensation can be performed by modifying the values of the features or the parameters of the trained models. For instance, if the noise would affect the signal features as an additive constant, then its effect could be compensated for by subtracting this constant value from the model parameters.

A possible way of performing the compensation of the model parameters is to use training database corrupted by noise to estimate the model parameters. This approach is widely used in automatic speech recognition field, for instance [72, 78]. In practical application there may be a variety of noisy conditions and levels of noise. In such a case, the training database corrupted separately by each noise type/level should be used to estimate the model parameters – this will result in a set of model parameters, each corresponding to particular noisy conditions. This is referred to as multi-environment noise-compensated model and its block diagram is illustrated in Figure 6.7. During the recognition process, an estimate of the noisy conditions needs to be obtained from the received signal. Then, the trained noise-compensated model that is closest to the current noisy condition is used. Alternatively, the noise condition information is used in the decision block to arrive at the recognition result.

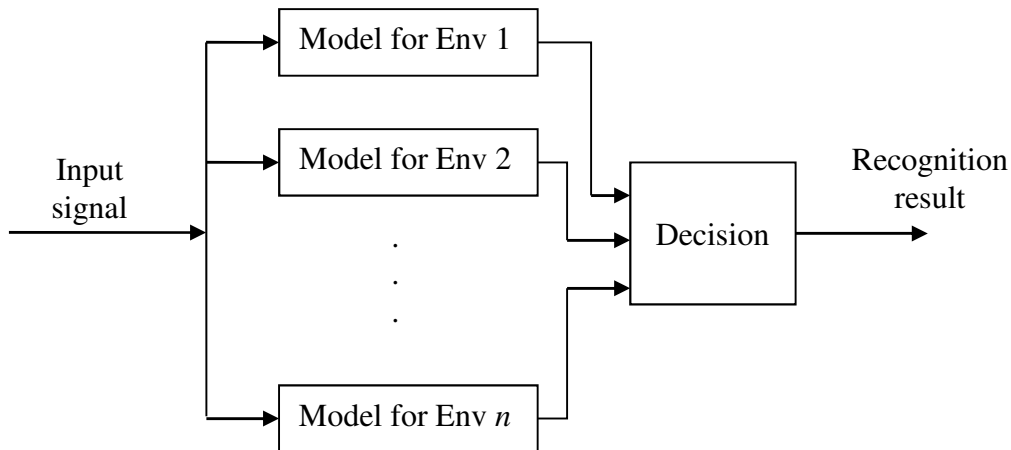


Figure 6.7 Block diagram of the multi-environment model[79]

This noise-compensation approach was used in this work to compensate for the clutter.

iii. Noise-robust feature extraction

In pattern recognition system, the feature extraction block determines the processing applied on the signal and the type of features extracted from the signal. The features should provide information to discriminate between individual classes and should be as much as possible robust to noise.

This approach was used in this work to reduce the clutter. As explained in section 2.10, in our case, we use only a part of the signal spectrum to extract the features used for recognition i.e in Figure 6.8. The frequency range of the spectrum is set based on the knowledge of the properties of the signal and clutter.

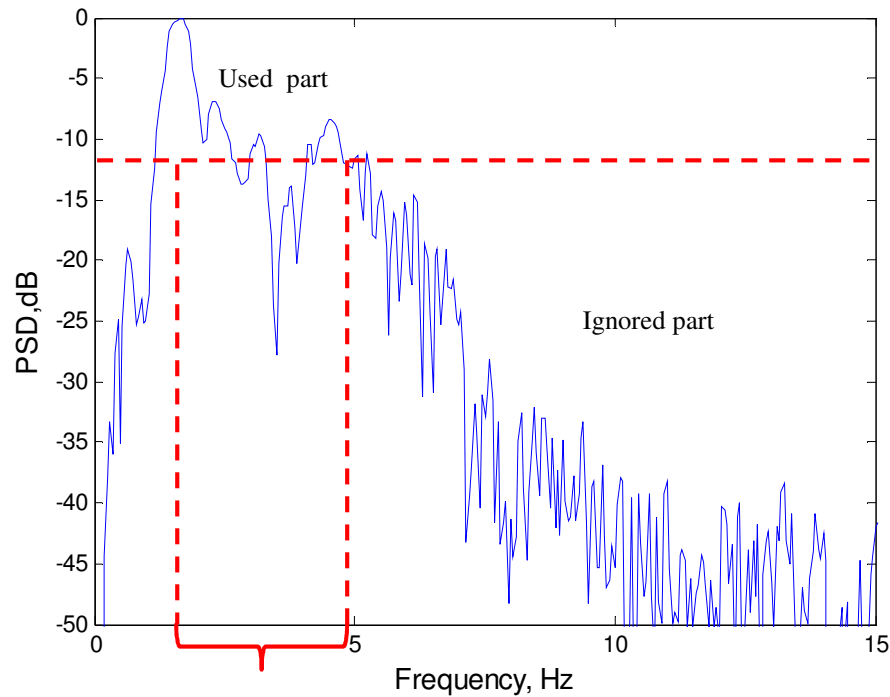


Figure 6.8 Example of feature extraction

It is known that clutter concentrates only at the low frequency part of the spectrum [45] around 0.5-1 Hz of Doppler band and the target signature may occupy wide Doppler band, for example car moving with the speed of 20 miles per hour creates signature with Doppler bandwidth of around 4 Hz for 64 MHz carrier frequency.

For the ATC, we do not have to analyse the entire target's spectrum. This is because the difference between different types of cars is observed only at higher part of the power spectrum which is within 20 dB from its maximum of 0 dB [1]; practically from 0.5 Hz till 3 Hz for 64 MHz, from 1 Hz till 6 Hz for 151 MHz and from 1.5 Hz till 7 Hz for 434 Hz. Based on this information, the very low and very high frequency parts of the spectrum are not used in the further steps of the feature extraction.

A 2-dimension PCA plots are depicted in Figure 6.9 for SCR of 5 dB, 15 dB and 30 dB at 151 MHz. Comparing the distributions of the features across various SCRs, it can be seen that, as the SCR decreases, the compactness of the features within the target type decreases and the overlap of the features between different target types increases. However, even at the SCR of 5dB, there is still a reasonable separation between different target types. Importantly, we can also notice that as the SCR varies, the location of the features varies, e.g., features corresponding to Ford-Focus, indicated by red circle, move to completely different location when the SCR changes from 30dB to 5dB. This indicates that having models trained on data with no-clutter conditions only may not work well for recognition of targets in cluttered conditions. However, it also indicates that using models trained at similar SCR conditions as appear in testing signal during the recognition should provide the best possible performance as the mismatch between the testing signal and trained models is minimized.

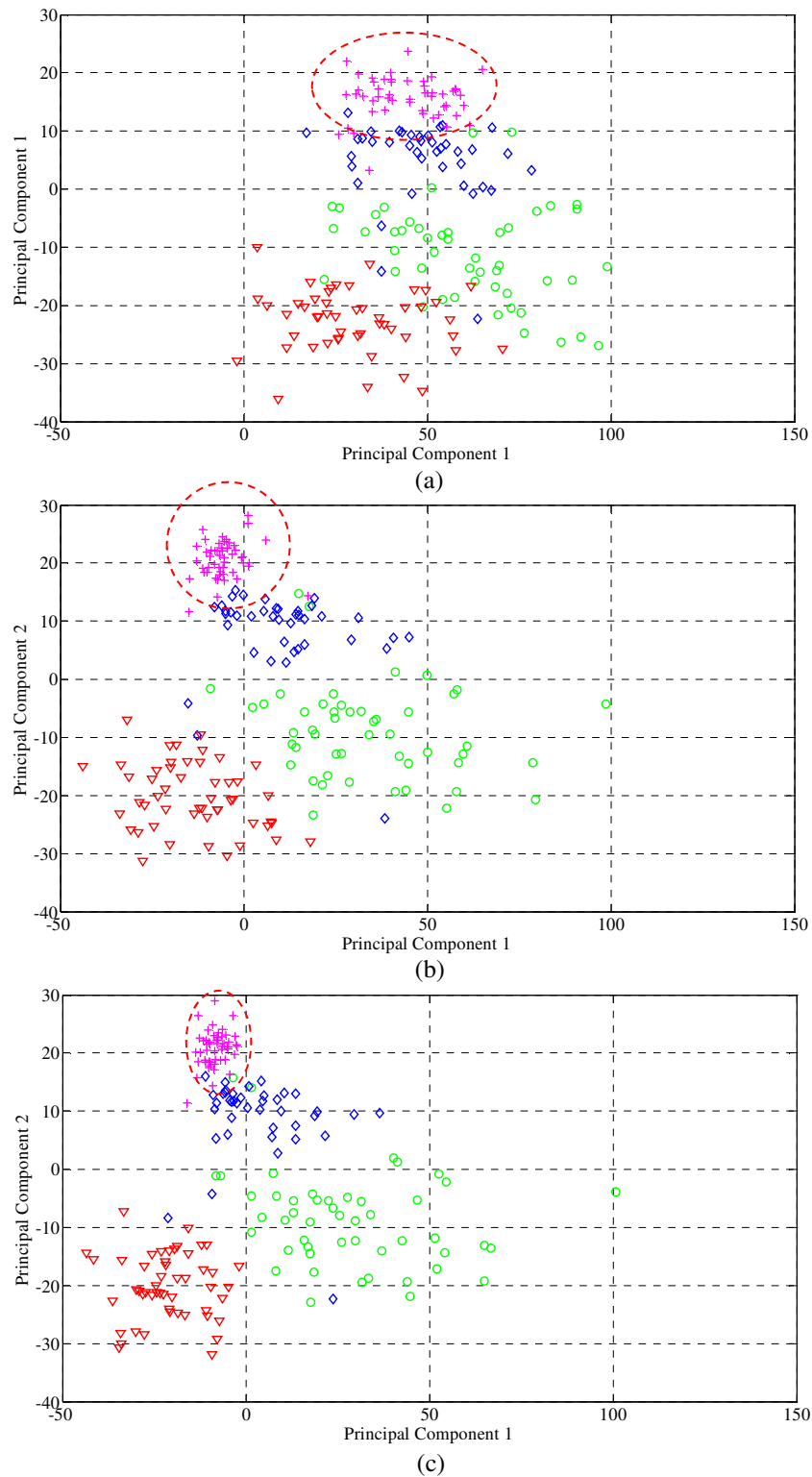


Figure 6.9 PCA plots for different level of SCR when only shape deformation error presence at frequency 151 MHz when SCR (a) 5dB (b) 15dB and (c) 30 dB
 Notation: Landrover (∇), Nissan (\diamond), BMW (\circ) and Fordfocus ($+$)

Evaluations presented in this section are taking into account the effect of clutter to both estimated speed and shape of the target's signature.

First, we will present evaluation results when a matched clutter-compensated model is used. In this case, the SCR at which the clutter is present in the unknown testing target signal is assumed to be known a-priori.

The comparison of the classification obtained using the clutter-uncompensated and clutter-compensated ATC system is shown in Figure 6.10. It can be seen that the use of the matched clutter-compensated models improves the classification accuracy significantly at low SCRs when using the 64 MHz and 151 MHz frequencies; for instance, the accuracy improves from 57% to 84.5% at 5 dB SCR using 151 MHz frequency. From the result we can see that there is only significant effect that can be observed at 434 MHz. The result also suggests that at higher SCRs, the performance is marginally lower than when using the clean trained models.

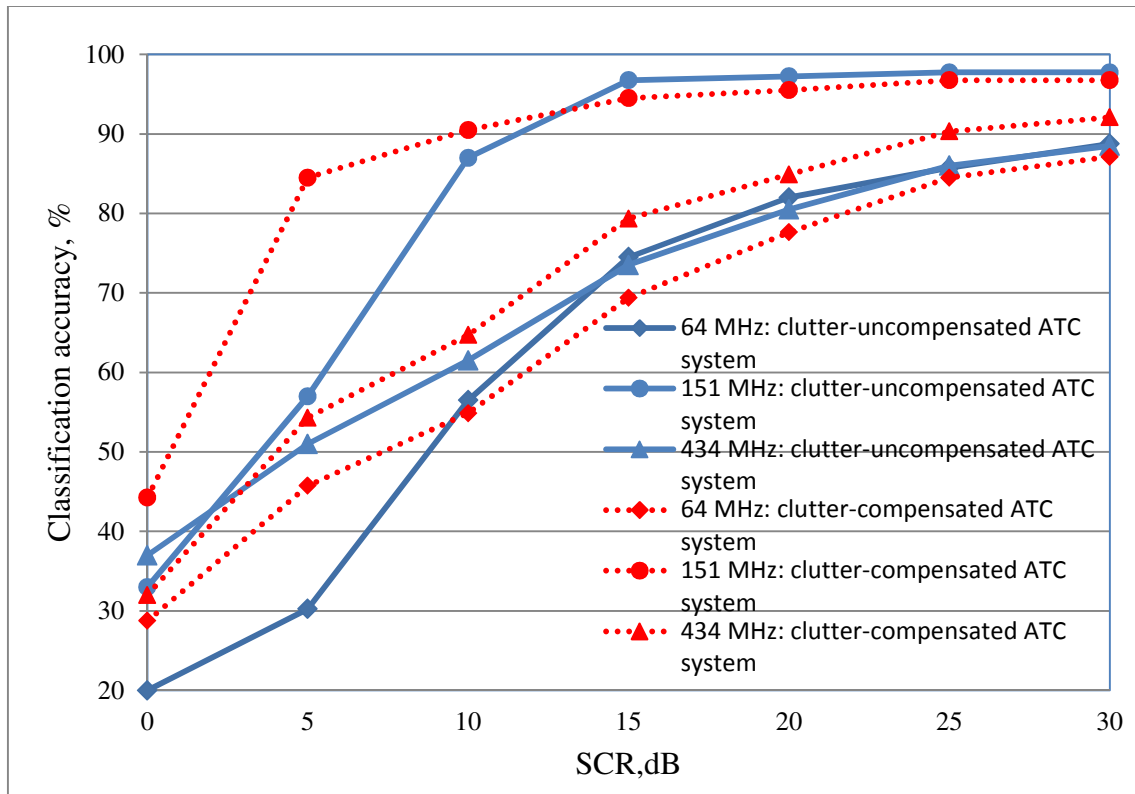


Figure 6.10 Comparison of the classification obtained using the clutter-uncompensated and clutter-compensated ATC system

The above results obtained using the matched clutter-compensated models demonstrate potential for significant classification accuracy improvements at low SCRs; however, these are only idealised since the SCR was assumed to be known a-priori. In real-world scenario, we do not know the SCR during the recognition process. However, we can calculate the SCR as the ratio between the signal and clutter power, where the clutter power can be estimated based on signal samples before and after the target signal is detected. An example of how to estimate the clutter power (and SCR) is shown in Figure 6.11. In our case, the simplest method has been performed where the first and the last 300 samples of the signal are used. The standard deviations of the clutter samples and the signal are calculated and the SCR is calculated using the same equations as given in Section 5.3.2.

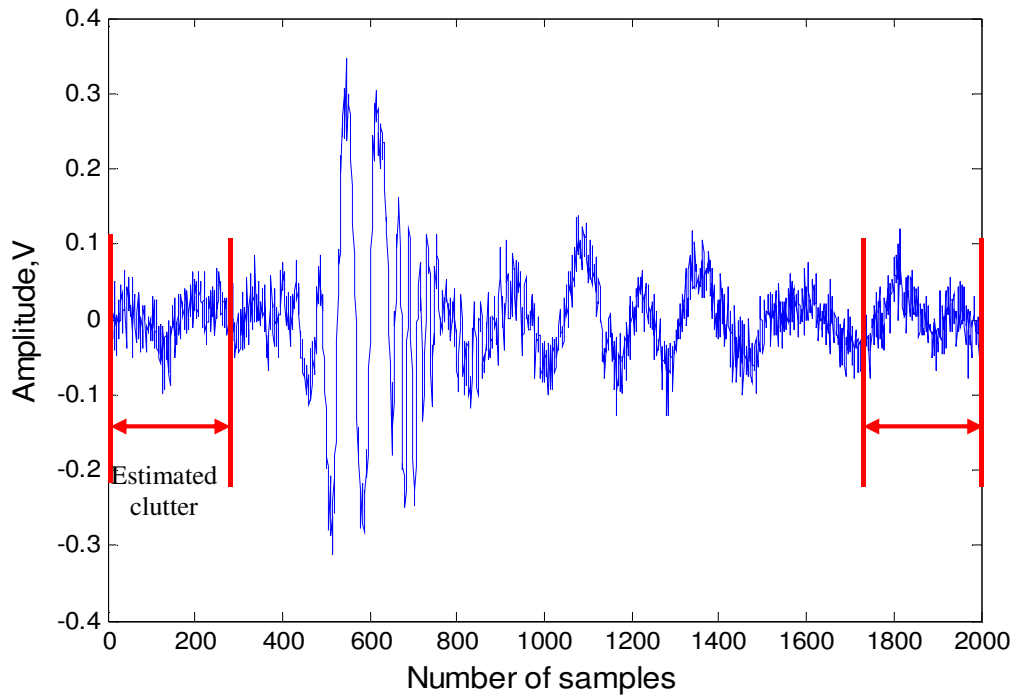


Figure 6.11 An example of the target signal with indicated part of signal used for estimation of the clutter energy

The same trained model data as in the idealised case (multiple clutter-compensated models) is used for this analysis. The SCR of the testing signal is calculated, and only then the trained model of corresponding closest SCR is used for classification process.

The obtained results are presented in Table 6.1 where the ‘ideal’ and ‘real’ case denotes the use of matched clutter-compensated models with the SCR known a-priori, multiple clutter-compensated models with the estimated SCR and clutter-uncompensated model, respectively. It is apparent from this table that the performance between the ideal and real case situations is very similar. For example at all frequencies, when the SCR=0dB, the classification accuracy for both cases has difference by only 1% whereas for SCR=30dB, both cases have the same performance.

Table 6.1 Target classification accuracy achieved by the clutter-compensated ATC system for the ideal and real case scenarios when using frequency 64, 151 and 434 MHz

| Frequency | 64 MHz | | | 151 MHz | | | 434 MHz | | |
|--------------|------------|-----------|--------|------------|-----------|--------|------------|-----------|--------|
| SCR | ATC-CC | | ATC-CU | ATC-CC | | ATC-CU | ATC-CC | | ATC-CU |
| | Ideal case | Real case | | Ideal case | Real case | | Ideal case | Real case | |
| 0 dB | 55 | 54 | 20 | 44 | 45 | 33 | 41 | 42 | 37 |
| 5 dB | 63 | 62 | 30 | 85 | 84 | 57 | 54 | 54 | 51 |
| 10 dB | 62 | 61 | 57 | 91 | 91 | 87 | 60 | 60 | 62 |
| 15 dB | 73 | 73 | 75 | 95 | 95 | 97 | 68 | 68 | 74 |
| 20 dB | 85 | 80 | 82 | 96 | 96 | 97 | 80 | 81 | 81 |
| 25 dB | 87 | 87 | 86 | 97 | 96 | 98 | 85 | 85 | 86 |
| 30 dB | 89 | 89 | 89 | 97 | 97 | 98 | 87 | 87 | 89 |

** ATC-CC –Clutter-compensated model ATC-CU–Clutter-uncompensated model

This result gives good indications that the ‘clutter-compensated’ model can improve the performance of the system when high amount of clutter is present in the signal, especially when SCR is below 10dB. However, more research on clutter compensation topic should be undertaken for example by using different compensation method or by employing some of the signal enhancement techniques or selecting part of signal which is not affected by clutter.

6.5 Conclusion

This chapter presented the influence of vegetation clutter on the Automatic Target Classification (ATC) accuracy in Forward Scattering Radar (FSR). The simulated vegetation clutter was artificially added to the recorded signals with multilevel of SCRs.

The analysis of the effect of various levels of SCRs on the classification accuracy is presented for a set of frequencies (64, 151 and 434 MHz). This analysis is performed separately for the cases of:

- the speed estimation error only,
- the spectrum's shape deformation error only, and
- both speed estimation and spectrum's shape deformation errors presented in the ATC system.

The speed estimation error only showed nearly no effect on the classification accuracy (except for the very low SCR of 0 dB), which demonstrates good robustness of the speed estimation algorithm to clutter.

The evaluation also shown that better ATC is achieved at the carrier frequency in VHF band (151 MHz), where the clutter influence is less than at higher frequency mostly affected by clutter, and the more information about the target can be obtained in comparison with low (64 MHz) frequency.

The last section of the thesis presents the experimental evaluation of the clutter-compensation technique for ATC system. The system uses cluttered signal as training data in order to reduce the mismatch between testing and training data and improve the classification. The performance using clutter-compensation technique has been evaluated in two situations: 1) ideal case when SCR of testing signal is known and 2) real case when SCR of testing signal is

unknown. In an ideal case, the clutter-compensation ATC system achieved a good improvement especially when signal is very low (when SCR is below 10dB). For a real case, the same performance can be achieved as in ideal case by estimating the SCR of the signal.

CHAPTER 7

CONCLUSIONS AND FUTURE WORKS

In this thesis, investigation of ATC performance has been performed for FSR operating in VHF/UHF frequency bands for the detection and classification of slow moving targets in clutter-free and cluttered environment. The ATC system was built using Fourier transform, Principal Component analysis (PCA) and K-Nearest Neighbour classifier as the methods to extract and transform target's features and to use them for classifying the target. The system is integrated with a coherent signal processing algorithm in order to estimate and to extract target motion parameters such as speed, crossing point and crossing angle prior to spectra normalisation process.

The performance of ATC system is evaluated in idealised conditions using experimentally obtained signals. By 'idealised' condition it is meant the specific FSR system topology, where the target is moving only perpendicularly (at 90^0) to the baseline in the middle with different velocities. Measured signals were collected in virtually "clutter free" environment assuming

that only the least possible amount of clutter is present at sites such as concrete runaway and open field. It has been shown that coherent processing delivers accurate target motion parameters which lead to high accuracy of classification.

In FSR ground system where the sensors are positioned on the ground with the baseline of hundred(s) metres, vegetation may be present all around the baseline region. This contributes into the limitations of FSR system because the absence of range resolution will cause the clutter picking up from the large area around the sensors. As a result, FSR system performance is affected by the clutter and the clutter presence potentially may cause the some errors in the target classification. Thus the task of the thesis is to suggest the candidate for the optimal classifier and to analyze its performance for FSR target classification.

The effect of clutter on the performance of ATC system is investigated by analysis of signals where simulated clutter signal is superimposed with the received target signal in the time-domain at various levels of SCRs. The evaluations were performed for three different cases:

- due to error of estimated speed by coherent processing only
- due to spectrum shape deformation because of overall signal distortion by clutter
- due to both error of estimated speed and spectrum shape deformation

The results demonstrate that clutter affects the spectrum shape the most while demonstrating the relative robustness of the speed estimation accuracy related to clutter. The results have been experimentally confirmed and it has been concluded that ATC performance degraded dramatically only when high level of clutter is present.

It has been found that the mismatch between the measured data with any SCR and "clutter-free" training data could produce worse classification results. Hence, a so called 'clutter compensation method' was proposed to compensate this effect. Instead of having clean training signals for database, the trained model is designed by using measured signals which have been contaminated with different levels of clutter in a view that clutter affects both estimated speed and shape of the target's signature. This proposed method suggests that the classification accuracy can be improved especially for low SCR.

The aim of the thesis is to improve the ATC system and it has been achieved. However, further investigation and study should be carried out. Several aspects could be investigated to explore the potential of FSR for further improvement to ground target classification. These include:

Further investigation on spectra normalization improvement for different target trajectories: In the earlier part of the thesis, we have presented the preliminary analysis on signatures for different target's trajectories. We have shown how different target trajectory parameters, such as baseline lengths, crossing points and crossing angles affect the spectra of the Doppler signature of the same target. Further investigation is required to take into account these effects for proper spectra normalisation; hence the solutions and improvements could be drawn. For example, baseline normalisation procedure is used in order to rescale Doppler signatures related to different baseline lengths. However, there are still some discrepancies in the results, which probably related to different scattering mechanisms corresponding either to Mie or optical regions. Therefore, if we can understand the cause of the problem, the baseline normalisation procedure could be improved. Furthermore, non-uniform target trajectories (for

example moving with zig-zag direction) should also be investigated. Once we know the effect of trajectory on spectra normalization, we can decide on necessary completeness of the training database.

Classification and clutter model compensation algorithm: The classifier used in this thesis is relatively simple. More sophisticated pattern recognition methods, which have demonstrated improved performance in other fields, could be employed. There are a number of other alternative classification methods which can be used like Gaussian Mixture Modelling (GMM), Neural Networks (NN) and, Hidden-Markov Model (HMM). GMM can be employed in our system since it simply models the distribution of the data by estimating its parameters such as mean and standard deviation. Whilst it has been proven in [51] that NN can be used in ATC system; however different characteristic might be chosen for the input of the network. Other than that, since our signals are time-varying (the frequency and amplitude varies over time), we can also employed HMM as it is a suitable method for modelling a time-varying signals. The chosen classification method, if possible, should also be able to reduce the effect of clutter and, hence, can reduce the size of database due to different level of clutter for each training target resulting in lesser processing time.

Bigger database: Currently, our database is relatively small – it contains signal recordings of four different types of vehicle with about 200 recorded signatures in total. It is enough for analysis of performance of coherent processing algorithm and classification algorithm. However, such small amount of data does not allow for employment of more sophisticated pattern recognition methods as it would not be sufficient to reliably estimate the parameters of such models. Thus, it is important to extend essentially the existing database in both

directions: larger number of recordings for statistical analysis and larger number of types of targets.

References

- [1] R. S. A. Raja Abdullah, "Forward Scattering Radar for Vehicle Classification," PhD, School of Electronic, Electrical and Computer engineering, University of Birmingham, Birmingham, 2005.
 - [2] N. J. Willis, "Bistatic Radar", *Technology Service Corporation*, 1995.
 - [3] R. J. Boyle and W. Wasyliwskyj, "Comparison of monostatic and bistatic bearing estimation performance for low RCS targets," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 30, pp. 962-968, 1994.
 - [4] M. I. Skolnik, "Introduction to Radar Systems", New York: *Mcgraw-Hill Book Company*, 1980.
 - [5] H. J. Siegel, "Bistatic Radar and Forward Scattering," in *National Conference Proceeding in Aeronautical Electronics*, Ohio, 1958, pp. 286-290.
 - [6] J. I. Glaser, "Some results in the bistatic radar cross section (RCS) of complex objects," *Proceedings of the IEEE*, vol. 77, pp. 639-648, 1989.
 - [7] Y. S. Chesnokov and M. V. Krutikov, "Bistatic RCS of aircrafts at the forward scattering," in *Radar, 1996. Proceedings., CIE International Conference of*, 1996, pp. 156-159.
 - [8] V. V. Chapurskiy and V. N. Sablin, "SISAR: shadow inverse synthetic aperture radiolocation," in *Radar Conference, 2000. The Record of the IEEE 2000 International*, 2000, pp. 322-328.
 - [9] N. E. A. Rashid, *et al.*, "Automatic target classification in a low frequency FSR network," in *Radar Conference, 2008. EuRAD 2008. European*, 2008, pp. 68-71.
 - [10] R. S. A. R. Abdullah, *et al.*, "Target prediction in Forward Scattering Radar," in *Applied Electromagnetics, 2007. APACE 2007. Asia-Pacific Conference on*, 2007, pp. 1-5.
 - [11] H. Cheng, *et al.*, "Forward scattering micro radars for situation awareness," in *Synthetic Aperture Radar, 2009. APSAR 2009. 2nd Asian-Pacific Conference on*, 2009, pp. 231-234.
 - [12] M. Cherniakov, Solous M., Jancovic, P., Raja Abdullah, R. S. A. and Kostylev V., "Forward Scattering Radar for Ground Targets Detection and Recognition," in *2nd Annual Technical DTC Conference*, Edinburgh, U.K., June 2005.
 - [13] M. Cherniakov, *et al.*, "Forward scattering micro sensor for vehicle classification," in *Radar Conference, 2005 IEEE International*, 2005, pp. 184-189.
-

- [14] M. Cherniakov, *et al.*, "Automatic ground target classification using forward scattering radar," *Radar, Sonar and Navigation, IEE Proceedings*, vol. 153, pp. 427-437, 2006.
 - [15] M. Cherniakov, *et al.*, "Analysis of forward scattering radar for ground target detection," in *Radar Conference, 2005. EURAD 2005. European*, 2005, pp. 145-148.
 - [16] M. Gashinova, *et al.*, "Signal detection in multi-frequency Forward Scatter Radar," in *Radar Conference (EuRAD), 2010 European*, pp. 276-279.
 - [17] V. I. Sizov, *et al.*, "Forward scatter RCS estimation for ground targets," in *Microwave Conference, 2007. European*, 2007, pp. 1700-1703.
 - [18] A. B. Blyakhman, "Multistatic Forward Scattering Radar," in *PIERS Workshop on Advances in Radar Methods*, Italy, Baveno, July 1998, pp. 107-113.
 - [19] A. B. Blyakhman, *et al.*, "Forward scattering radar moving object coordinate measurement," in *Radar Conference, 2000. The Record of the IEEE 2000 International*, 2000, pp. 678-682.
 - [20] A. B. Blyakhman and I. A. Runova, "Forward scattering radiolocation bistatic RCS and target detection," in *Radar Conference, 1999. The Record of the 1999 IEEE*, 1999, pp. 203-208.
 - [21] D. M. Gould, *et al.*, "Forward scatter radar detection," in *RADAR 2002*, 2002, pp. 36-40.
 - [22] A. B. blyakhman, Runuvo, I.A, "Bistatic Cross Section and Target Detection in Forward Scattering Radar," *Radiotechnica and Electronica*, 2001.
 - [23] M. Antoniou, *et al.*, "The concept of a forward scattering micro-sensors radar network for situational awareness," in *Radar, 2008 International Conference on*, 2008, pp. 171-176.
 - [24] V. Sizov, *et al.*, "Forward scattering radar power budget analysis for ground targets," *Radar, Sonar & Navigation, IET*, vol. 1, pp. 437-446, 2007.
 - [25] M. Cherniakov, Sizov, V. , Antoniou, M., Emileen Rashid, Peter Jankovic, and A. K. Alexander Myakinkov, "Easily deployable, multi-functional radar network," *Journal Radar Network*, vol. 5, 2008.
 - [26] G. W. Stimson, "Introduction to Airborne Radar" El Segundo, California, *Hughes Aircraft Company*, 1983.
 - [27] V. Chernyak, "Fundamentals of Multisite Radar Systems", *Gordon and Breach Science Publishers*, 1998.
 - [28] M. Cherniakov, "Bistatic Radar Principles and Practice", Chichester, *John Wiley & Sons*, 2007.
-

- [29] R. E. Hiatt, *et al.*, "Forward Scattering by Coated Objects Illuminated by Short Wavelength Radar," *Proceedings of the IRE*, vol. 48, pp. 1630-1635, 1960.
- [30] R. S. A. Raja Abdullah, Cherniakov, M., Jancovic, P., "Automatic Vehicle Classification In Forward Scattering Radar," in *1st International Workshop in Intelligent Transportation, WIT2004*, Germany, 2004, pp. 7-12.
- [31] R. S. A. Raja Abdullah, Cherniakov, M., "Forward Scattering Radar For Vehicles Classification," in *VehCom International Conference, VehCom2003*, 2003, pp. 73-78.
- [32] T. Rappaport, "Wireless communications: Principles and Practice", 2nd ed., *Prentice Hall*, 1996.
- [33] J. D. Parsons, "The mobile propagation channel", 2nd ed., *John Wiley & Sons*, 2000.
- [34] D. K. Barton and S. A. Leonov, Eds., "Radar Technology Encyclopedia", *Artech House*, 1998,
- [35] C. A. Balanis, "Antenna theory Analysis and Design", *John Wiley & Sons Inc.*, 1997.
- [36] J. J. Bowman, Senior T.B.A., Uslenghi, P.L.E., "Electromagnetic and acoustic scattering by simple shapes", Amsterdam, *North-Holland Publishing Company*, 1969.
- [37] "Doppler Effect", http://en.wikipedia.org/wiki/Doppler_effect
- [38] V. Sizov, *et al.*, "Vegetation clutter spectral properties in VHF/UHF bistatic doppler radar," in *Radar Conference, 2008. RADAR '08. IEEE*, 2008, pp. 1-6.
- [39] A. B. Blyakhman, *et al.*, "Tracking algorithm for three-dimensional bistatic forward scattering radar with weighting of primary measurements," in *Radar Conference, 2005. EURAD 2005. European*, 2005, pp. 153-156.
- [40] R. S. A. Raja Abdullah, Cherniakov, M., "Experimental Investigation of Forward Sattering Radar for Vehicle Classification," in *Postgraduate Research Conference in Electronic, Photonics and Communications, PREP2003*, 2003.
- [41] R. S. A. Raja Abdullah, Cherniakov, M., Jancovic, P., Solous M., "Progress on Using Principle Component Analysis in FSR for Vehicle Classification," in *2nd International Workshop in Intelligent Transportation, WIT2005*, Germany, 2005, pp. 7-12.
- [42] R. S. A. Raja Abdullah, *et al.*, "Neural network based for automatic vehicle classification in forward scattering radar," in *Radar Systems, 2007 IET International Conference on*, 2007, pp. 1-5.
- [43] V. Sizov, *et al.*, "Forward Scattering Mirco Radar efficiency analysis for different landscapes," in *6th EMRS DTC Tehcnical Comference*, Edinburgh, 2009.

- [44] M. Gashinova, *et al.*, "Signature modelling and coherent target detection for Forward Scattering Radar (FSR) sensors," in *International Radar Symposium*, Hamburgh, Germany, 2009.
- [45] M. Gashinova, *et al.*, "Empirical model of vegetation clutter in forward scatter radar micro-sensors," in *Radar Conference, 2010 IEEE*, pp. 899-904.
- [46] Mutaz Salah, *et al.*, "Speed Estimation in Forward Scattering Radar by Using Standard Deviation Method," *Modern Applied Science*, vol. 3, March 2009 2009.
- [47] R. O. Duda, Hart P.E, Stork D.G., "Pattern Classification", *John Wiley & Sons*, 2000.
- [48] B. G. Batchelor, "Pattern Recognition", New York, *Plenum Press*, 1978.
- [49] M. Jincheng, Wanlin, Y., "Nearest Neighbor Classifier Based On Riemannian Metric In Radar Target Recognition," presented at the 2005 IEEE International Radar Conference 2005.
- [50] Y. Yang, Qui, Y., Lu, C., "Automatic Target Classification Experiments on the MSTAR SAR Images," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2005 and First ACIS International Workshop on Self-Assembling Wireless Network*, 2005.
- [51] N.K. Ibrahim, *et al.*, "Artificial Neural Network Approach in Radar Target Classification," *Journal of Computer Science* 5, vol. 1, p. 9, 23-32, 2009 2009.
- [52] A. Webb, "Statistical Pattern Recognition", 2nd Edition ed., *John Wiley & Sons Ltd.*, 2002.
- [53] H. Hotelling, "Analysis of a Complex of Statistical Variables into Principal Components," *Journal of Educational Psychology*, pp. 498-520, 1933.
- [54] M. Jaruszewicz and J. Mandziuk, "Application of PCA method to weather prediction task," in *Neural Information Processing, 2002. ICONIP '02. Proceedings of the 9th International Conference on*, 2002, pp. 2359-2363 vol.5.
- [55] T. F. Karim, *et al.*, "Face recognition using PCA-based method," in *Advanced Management Science (ICAMS), 2010 IEEE International Conference on*, 2010, pp. 158-162.
- [56] J. O. Kim, Mueller, C.W, "Introduction to Factor Analysis", *Sage University Paper*, 1984.
- [57] L. I. Smith. (2002). A Tutorial on Principal Components Analysis. http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

- [58] N. E. A. Rashid, Jancovic, P., M. Gashinova, M. Cherniakov, and V. Sizov, "The effect of speed on the automatic target classification accuracy in FSR," in *IRS2009*, Hamburg, 2009.
 - [59] V. Sizov, *et al.*, "FSR sensors network: performance and parameters," in *7th EMRS DTC Technical Conference – Edinburgh 2010*, Edinburgh, 2010.
 - [60] N.A. Zakaria, *et al.*, "Synthetic Environment for Forward Scattering Radar Detection," in *Asia Pacific Conference on Defence & Security Technology 2009 (DSTC 2009)*, Kuala Lumpur, Malaysia, 2009.
 - [61] J. B. Billingsley, "Low-angle radar land clutter: measurements and empirical models" *Scitech Publishing Inc.*, 2001.
 - [62] D. K. Barton, "Land clutter models for radar design and analysis," *Proceedings of the IEEE*, vol. 73, pp. 198-204, 1985.
 - [63] M. Sekine and Y. Mao, "Weibull Radar Clutter," *Institution of Engineering and Technology*.
 - [64] "Complexity of real signal", <http://www.katjaas.nl/hilbert/hilbert.html>
 - [65] H. Cheng, *et al.*, "Quasi-optimal signal processing in ground Forward Scattering Radar," in *Radar Conference, 2008. RADAR '08. IEEE*, 2008, pp. 1-6.
 - [66] Eathnet Online. " Pulse Compression", <http://envisat.esa.int/handbooks/asar/CNTR2-6.htm>
 - [67] B. P. Milner and S. V. Vaseghi, "Comparison of some noise-compensation methods for speech recognition in adverse environments," *Vision, Image and Signal Processing, IEE Proceedings -*, vol. 141, pp. 280-288, 1994.
 - [68] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 27, pp. 113-120, 1979.
 - [69] J. Holmes and N. Sedgwick, "Noise compensation for speech recognition using probabilistic models," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86.*, 1986, pp. 741-744.
 - [70] L. Karray and A. Martin, "Towards improving speech detection robustness for speech recognition in adverse conditions," *Speech Commun.*, vol. 40, pp. 261-276, 2003.
 - [71] S. V. Vaseghi and B. P. Milner, "Noise compensation methods for hidden Markov model speech recognition in adverse environments," *Speech and Audio Processing, IEEE Transactions on*, vol. 5, pp. 11-21, 1997.
-

- [72] P. Jancovic and M. Köküer, "Incorporating the voicing information into HMM-based automatic speech recognition in noisy environments," *Speech Communication*, vol. 51, pp. 438-451, 2009.
- [73] R. McAulay and M. Malpass, "Speech enhancement using a soft-decision noise suppression filter," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28 pp. 137 - 145 April 1980 1980.
- [74] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, pp. 1586-1604, 1979.
- [75] Y. Ephraim and H. L. Van Trees, "A signal subspace approach for speech enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 251 - 266 Jul 1995 1995.
- [76] A. Hyvärinen, "Sparse Code Shrinkage: Denoising of Nongaussian Data by Maximum Likelihood Estimation," *Neural Computation*, vol. 11, pp. 1739-1768, 1999.
- [77] P. Jancovic, *et al.*, "Speech enhancement based on Sparse Code Shrinkage employing multiple speech models," *Speech Communication*, vol. In Press, Uncorrected Proof.
- [78] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Communication*, vol. 16, pp. 261-291, 1995.
- [79] Ji Ming, *et al.*, "Modeling the Mixtures of Known Noise and Unknown Unexpected Noise for Robust Speech Recognition " in *7th European Conference on Speech Communication and Technology*, Aalborg, Denmark, 2001.

Appendix A: Antenna specification

PANORAMA ANTENNAS

P13

Private Mobile Radio

| Band | Frequency Range | Colour Coding |
|------|-----------------|---------------|
| E3 | 67-74 MHz | Red |
| E4 | 74-81 MHz | Yellow |
| E5 | 81-88 MHz | Green |
| H4 | 141-151 MHz | Yellow |
| H5 | 149-159 MHz | Grey |
| H6 | 156-162 MHz | Orange |
| H7 | 162-174 MHz | White |
| JRC | 139-157 MHz | Red |
| K5 | 174-192 MHz | Red |
| K6 | 192-208 MHz | Green |
| K7 | 208-225 MHz | Orange |
| L | 220-250 MHz | Grey |
| M | 245-275 MHz | Orange |
| N | 270-300 MHz | Green |
| P | 300-336 MHz | White |
| R | 330-366 MHz | Blue |
| S | 350-392 MHz | Yellow |
| S1 | 380-400 MHz | Grey |
| S2 | 410-430 MHz | White |
| T | 390-432 MHz | Red |
| TET | 380-430 MHz | Orange |
| U | 430-472 MHz | Green |
| UT | 406-472 MHz | Blue |
| W | 470-512 MHz | White |



How to order
MFX with M6 @ 168MHz
Order
MFX-M6-H7

















| Type | MXK | PXX | MVQ | MFX | PUG | MQ | MFXU |
|----------------------|---------------|---------------|----------------|----------------|----------------|----------------|---------------|
| Frequency (MHz) | 67-88 | 141-225 | 141-336 | 141-366 | 350-512 | 330-512 | 330-512 |
| Pre-tuned bands | E3-E5 | H4-K7 | H4-P | H4-R | S-W | R-W | R-W |
| Bandwidth @ 2:1 VSWR | 5% | 5% | 10% | 7% | 8% | 10% | 8% |
| Construction | Helical | Helical | ¼ wave | Helical | ½ wave | ¼ wave | Helical |
| Length (mm) | 195 @68MHz | 87 @146MHz | 500 @146MHz | 170 @146MHz | 340 @451MHz | 156 @451MHz | 72 @451MHz |
| Max diameter (mm) | 14 | 10 | 14 | 10 | 16 | 14 | 10 |

Portables

Panorama Antennas Ltd, Frogmore, London, SW18 1HF, United Kingdom • www.panorama.co.uk

Tel +44 (0)20 8877 4444 • Fax +44 (0)20 8877 4477 • sales@panorama.co.uk

| | Connector | Panorama Type | As used by: |
|---|-----------|-----------------------------------|---|
|  | BNC | MFXU, MQ, MFX, MVQ, MXK, PXX | CLEARSTONE, ICOM, MAXON, NABISHI, PHILIPS, TRT, YAESU |
|  | CSP | MFXU, MQ, MFX, MXK | SAGEM, SECURICOR, TALCO, KEY (KP05) |
|  | ICF4 | MFXU, MQ, MFX, PXX | ICOM (later models) |
|  | M6 | MFXU, MQ, M MFX, MXK | ALCATEL |
|  | MX | MFXU, MQ, MFX, MVQ, MXK, PXX | KENWOOD, MAXON, MOTOROLA, TAIT |
|  | MG | MFXU, MQ, MFX, MVQ, MXK, PXX | MOTOROLA JEDI (GP900 / GP1200 / HT / MT Series) + VISAR |
|  | MV | MFXU, MQ, MFX, MVQ, MXK, PXX | SIMOCO (SRP 1000/8000) |
|  | PRP73 | MFXU, MFX, MXK | SIMOCO, PRP73 / 76 |
|  | SL100 | MFXU, MQ, MFX, PXX | MAXON (SL100) |
|  | SMAFR | MFXU, MQ, MFX, PXX | KENWOOD (SMA) |
|  | SMAMO | MFXU, MQ, MFX, MXK | TAIT (ORCA + 5000 Series) |
|  | TNC | MFXU, MQ, PUG, MFX, MVQ, MXK, PXX | ALCATEL, ASCOM, BOSCH, ICOM, KENWOOD, NABISHI, SIMOCO |
|  | TNK | MFXU, MQ, MFX, MVQ, MXK, PXX | GRUNDIG, KEY, KYODO |
|  | TNM | MFXU, MQ, MFX, MVQ, MXK, PXX | KEY, KYODO, NIROS |
| | VX410 | MQ, MFXU, PXX, MFX | Yaesu Vertex VX Range |
| | SL80 | MQ, MFXU, PXX, MFX | Maxon SL80 |
| | SL90 | MQ, MFXU, PXX, MFX | Maxon SL90 |

Panorama Antennas Ltd, Frogmore, London, SW18 1HF, United Kingdom • www.panorama.co.uk

Tel +44 (0)20 8877 4444 • Fax +44 (0)20 8877 4477 • sales@panorama.co.uk



433MHz Yagi Antenna

- Frequency: 433MHz
- VSWR: ≤ 1.5
- Gain: 10dBi
- Front to Back Ratio: 14dB
- Polarization: Vertical or Horizontal
- Max Power: 150w
- Input Impedance: 50Ω
- Connector Type: SMA
- Lead Length: 750mm
- Support pole: 38-55mm

Appendix B : Samples of Records from test sessions at Tilesford Airfield

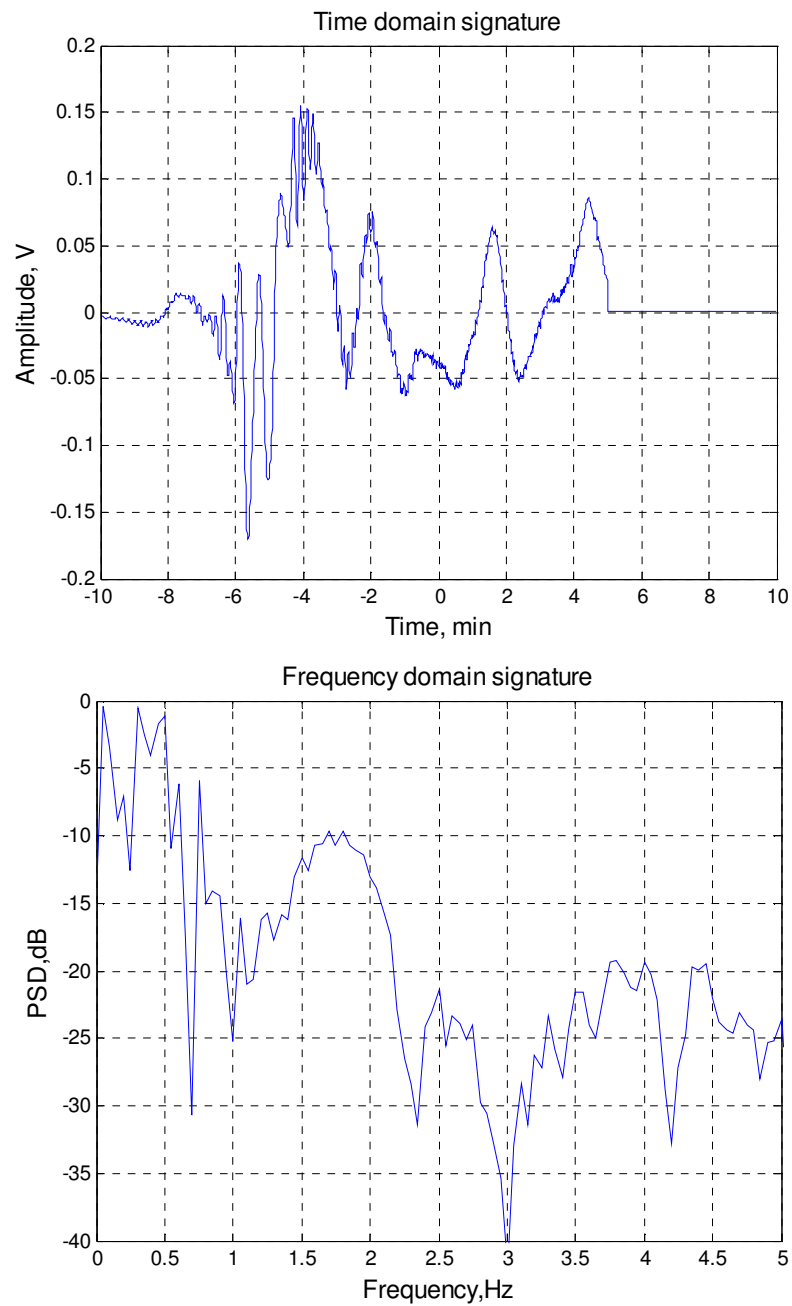
| File Name | File link | Descriptions | | BL Range (m) | Speed m/s | Attenuators (dB) | | | | | |
|--|---|--------------------|-----------------------|--------------|-----------|------------------|-------------------|------------------|----------|----------|------------|
| | | | | | | 3 Channels (MHz) | | 4 Channels (MHz) | | | |
| | | Target | Comments | | | 64.125 | 434.60 (D) | 434.45 (O) | 135.05 | 144.05 | 151 173.28 |
| 2) The experiment with moving of FSR position without changing of geometry and on concrete ground surface conditions. Car speed ~20m/h, middle, 90°. | | | | | | | | | | | |
| 1 | Fordfocus1_files\position1_1.htm | Car (LR) move ← | Position 1 (Figure 3) | 100 | 9 | 20 RX | 0 and (10+10 ant) | 0 | Not used | Not used | Not used |
| 2 | Fordfocus1_files\position1_2.htm | → | Position 1 (Figure 3) | 100 | 8 | 20 RX | 0 and (10+10 ant) | 0 | Not used | Not used | Not used |
| 3 | Fordfocus1_files\position1_3.htm | ← | | | 9.7 | | | 0 | | | |
| 4 | Fordfocus1_files\position1_4.htm | → | | | 8.5 | | | 0 | | | |
| 5 | Fordfocus1_files\position1_5.htm | ← | | | 8.2 | | | 0 | | | |
| 6 | Fordfocus1_files\position1_6.htm | → | | | 8 | | | 0 | | | |
| 7 | Fordfocus1_files\position2_7.htm | → | Position 2 (Figure 3) | 100 | 8.5 | 20 RX | 0 and (10+10 ant) | 0 | Not used | Not used | Not used |
| 8 | Fordfocus1_files\position2_8.htm | ← | Position 2 (Figure 3) | 100 | 8.5 | 20 RX | 0 and (10+10 ant) | 0 | | | |
| 9 | Fordfocus1_files\position2_9.htm | → | | | 8.5 | | | 0 | | | |
| 10 | Fordfocus1_files\position2_10.htm | ← | | | 8 | | | 0 | | | |
| 11 | Fordfocus1_files\position2_11.htm | → | | | 8.5 | | | 0 | | | |
| 12 | Fordfocus1_files\position2_12.htm | ← | | | 8.2 | | | 0 | | | |
| 13 | Fordfocus1_files\position3_13.htm | → | Position 3 (Figure 3) | 100 | 8.5 | 20 RX | 0 and (10+10 ant) | 0 | Not used | Not used | Not used |
| 14 | Fordfocus1_files\position3_14.htm | ← | Position 3 (Figure 3) | 100 | 8.8 | 20 RX | 0 and (10+10 ant) | 0 | | | |
| 15 | Fordfocus1_files\position3_15.htm | → | | | 8.3 | | | 0 | | | |
| 16 | Fordfocus1_files\position3_16.htm | ← | | | 8.7 | | | 0 | | | |
| 17 | Fordfocus1_files\position3_17.htm | → | | | 8.6 | | | 0 | | | |
| 18 | Fordfocus1_files\position3_18.htm | ← | | | 9.4 | | | 0 | | | |

Appendix C : Example of an Html File

Car: VolvoV40

Filename:47R.bin

Example of time and frequency domain signature for 151MHz



Appendix D: Example of manual on how to perform target classification

1. To extract features from the signals, click on the 'run_vehRec_featExtract_psd_training' MATLAB code. At this stage, a PCA-based feature vector is extracted from the vehicle signature. The type of vehicle can be selected from the database as the input to the algorithm. Before executing the program, a few inputs should be inserted:

- a. freqUsed

- The system uses three different frequencies (In this case : 64 MHz, 151 MHz and 434 MHz)
- Choose 1, 2 or 3 for selected frequency as shown.
 1. 1 represents 64 MHz
 2. 2 represents 151 MHz
 3. 3 represents 434 MHz

```

1  %%
2  %% Feature extraction for vehicle recognition:
3  %% Calculation of psd for all given data - includes the speed
4  %% normalization. PSDs of vehicles in each category is saved into a
5  %% .mat file with the category name (psdCar1, etc) .
6  %% created by: peter
7  %% modified by: emileen, 04 March 2009
8  %%
9
10 - clear;
11 - close all;
12 - randn('state', 0);
13  %% Parameter (choose 1, or 3)
14
15 - Nfrequency1 = 2;    % Frequency used:1 = 64 MHz, 2= 151 MHz, 4= 434 MHz
16
17 - freq=[64 151 434];
18 - freqUsed =freq(Nfrequency1);
19 - index=[20:60 20:100 30:100];
20 - indFreqUse =index(Nfrequency1);
21 - num_signal=2000;

```

Layout Feature Extraction: Inserting frequency

b. Vehicles

- The type of vehicle can be selected from the database folder as the input to the algorithm.

```

22
23 %% Read the list of files to be processed
24
25 - [listFileFull{1}, listFileShort{1}] = readListFile('..\fileList_fordfocus.txt');
26 - [listFileFull{2}, listFileShort{2}] = readListFile('..\fileList_Landrover_19april08.txt');
27 - [listFileFull{3}, listFileShort{3}] = readListFile('..\fileList_nissan.txt');
28 - [listFileFull{4}, listFileShort{4}] = readListFile('..\fileList_bmw308i.txt');
29

```

Layout Feature Extraction: Inserting list of vehicles

Once the frequency is set, run the MATLAB program by pressing F5. The code starts by reading the vehicle data from the database. After loading the data, the rest of the processing will be automatically executed. All spectra (for all vehicle types) are saved and stored according to vehicle type (example: psdCar1) and frequency. Later, the stored spectra will be used in the classification process.

```

37 %% saving feature extraction
38
39 - switch (freqUsed)
40 -     case 64
41 -         save psdCar1_64 psdCar1;
42 -         save psdCar2_64 psdCar2;
43 -         save psdCar3_64 psdCar3;
44 -         save psdCar4_64 psdCar4;
45
46 -     case 151
47 -         save psdCar1_151 psdCar1;
48 -         save psdCar2_151 psdCar2;
49 -         save psdCar3_151 psdCar3;
50 -         save psdCar4_151 psdCar4;
51
52 -     case 434
53 -         save psdCar1_434 psdCar1;
54 -         save psdCar2_434 psdCar2;
55 -         save psdCar3_434 psdCar3;
56 -         save psdCar4_434 psdCar4;
57
58 - end

```

Layout Feature Extraction: Saving data

2. To perform the classification process, click on the 'run_vehRec_testLeave1Out'. The user needs to decide at which frequency that he/she wishes to analyse the data. Once the frequency is selected, press F5 and the processing will be executed.

```

15  %% Set the parameters to be used
16  - freqUsed =151;           %64; 151; %434;
17  - nPCA = 3;                %% number of PCA components
18  - nKNN = 3;                %% number of nearest neighbours considered

```

Layout for Classification: Inserting frequency

```

1  %% Vehicle classification - Training/Testing procedure using leave-one-out
2  %% method for the kNN classification
3  %% Loads the PSDs of all given vehicles. Leaves one out for testing
4  %% and use the remaining data for training. With the training data,
5  %% perform estimation of the PCA matrix. Keeps the PCA-components of
6  %% the training data and use these in k-NN classifier.
7  %% database based on average psd
8  %% Peter Jancovic, 04 March 2008
9  %% edited by emileon 27 august 2008
10 %% updated 13 february 2010
11
12 close all;
13 clear all;
14
15 %% Set the parameters to be used
16 freqUsed =151;           %64; 151; %434;
17 nPCA = 3;                %% number of PCA components
18 nKNN = 3;                %% number of nearest neighbours considered
19
20 %% Load PSDs for each vehicle type and set frequency-region of the PSD to
21 %% be used for PCA calculation (indFreqUse=1:60 for 64MHz;
22 %% indFreqUse=20:100 for 151MHz; indFreqUse=40:100 for 434MHz;)
23
24 switch (freqUsed)
25 case 64
26     %% load psd
27     indFreqUse=20:70; load psdCar1SNRO_64; load psdCar2SNRO_64; load psdCar3SNRO_64;load psdCar4SNRO_64;
28     %% load database

```

Layout of Classification program

Appendix E : Explanation of software

This appendix aims to give an overview of the operation for Matlab software for experiment on vehicle classification. An explanation of will be given of the code in files:

1. SpeedEstiMarinaSim_MeasuredSignal.m
2. run_vehRec_featExtract_psd_database.m
3. run_vehRecNoisy_testLeave1Out.m
4. SigMod_forEmileen.m
5. ClutterSim.m

A CD named 'Matlab code' with a running version of these codes is attached to the thesis

D1. Automatic Speed estimation

- This code is used to estimate the speed of vehicles
- This code starts by reading the vehicle from the database. It loads and process the data separately
- User needs to specify which frequency is to be used for estimating the speed
- After loading the data, reference signals of each target is estimated based on the combination of expected speed, crossing point and crossing angles.
- By correlating target signal with reference signals, the maximum correlation can be found
- The estimated speed will be saved as 'velocityEstMar.mat'. The saved speed will be used in feature extraction code.

% Matlab version 7.11.10 (R2010b)

% File name: SpeedEstiMarinaSim_MeasuredSignal.m

% Purpose: to estimate speed of the target automatically

% Created by: Marina Gashinova

% Edited by: Emileen Rashid

clear all

% function [] = coherent(frequency, baseline, sRate, tMulti, winAv, signal)

tic;

```
randn('state', 0);  
close all;  
clear all;  
clc;
```

%% Call and assign data file

```
dirCara='C:\Emileen\Emileen\m_files\thesis\database\nissan';  
filelista=dir([dirCara '*.bin']);  
nFilesa = size(filelista,1);
```

% .mat file directory to save estimated speed

```
velocityEst_FNmat = [dirCara 'velocityEstMar.mat'];  
frequency=64*10^6;      % insert frequency to be used  
baseline=50;           % insert baseline length
```

%% For embedded noise in speed

```
for j=1:nFilesa;
```

```
    j;  
    fileName = [dirCara filelista(j).name];
```

% for 3 channels equipment use 6; for 4 channels equipment use 8

```
[data] = readTargetSignalBin(fileName, 6);
```

% to choose frequency: 64Mhz use 1, for 155Mhz use 3 and for 434Mhz use 5

```
signal= data(1,:);  
SigLength=length(signal);  
td=signal;
```

%% Parameters

```
eps0 = 8.85412e-12;  
mu0 = 4*pi*1e-7;
```

```
vlight = 1/sqrt(eps0*mu0);
c=vlight;
BL=baseline;           % in m
sRate=100;             % sampling rate , Hz
tMulti=0;              % threshold multiplier (10.3)
winAv=1;
f_s=sRate;
Fs=sRate;
f_LPF=20-0.1;          % cut-off frequency for low pass filter
f_HPF=0.5;             % cut-off frequency for low pass filter: 64MHz=0.5,151=1.5MHz and
434MHz=2.5
signal=signal1-mean(signal1)*1;
```

%% Crossing points

```
CrossPoint=[0 5 10 15 20 24]; % defining crossing point from the centre of the
baseline
nCROSSPoint=length(CrossPoint);
nYc=nCROSSPoint;
```

%% Velocity

```
minV = 0;              %Expected minimum absolute velocity of target
maxV = 15;             %Expected maximum absolute velocity of target
dAV =0.1;              %Step in absolute velocity between target signatures
Ve=[minV:dAV:maxV];
angleVector = [pi/2, pi/3, pi/4];
t2=1/f_s*(1:SigLength);
tmax=(1/f_s)*SigLength;
figure (1);plot(t2,signal);
xlabel('Time, min');ylabel('Amplitude,V')
```

% % winAv- number of cells for background estimation

% % Create a multi-dimensional array of target signatures based on different values of velocity, crossing point and angle. Values are independent of one another (no optimisation) and uniformly spaced.

f_s=sRate;

% % Create Vector of crossing points

nCrossPoint = length(CrossPoint); %Total number of crossing points

% % Create vectors of velocity components Vx and Vy determined by crossing angle of target

```

nAV = floor((maxV-minV)/dAV) + 1;      %Number of absolute velocities
nAngles = length(angleVector);          %Number of inputted angles
TotV = nAV*nAngles;                    %Length of vectors to store Vx and Vy
xVel = zeros(1,TotV);                  %Create vector to store Vx
yVel = zeros(1,TotV);                  %Create vector to store Vy
count = 1;                             %Start count of indices of vectors for Vx and
Vy
for iVel = 1:nAV                        %Loop over absolute velocity
    absVel = minV + (iVel-1)*dAV;       %Calculate absolute velocity for this iteration
    for iAngle = 1:nAngles              %Loop over trajectory angles
        xVel(count) = absVel*cos(angleVector(iAngle)); %Calculate Vx
        yVel(count) = absVel*sin(angleVector(iAngle)); %Calculate Vy
        count = count + 1;              %Find a way to reduce this to not counting one
    end
end
end

```

%% Filtering process (hardware filter)

```

nyquist = sRate/2; %Nyquist frequency

[z1, p1, k1] = butter(5,f_LPF*2/f_s,'low'); %Hardware LPF of 5 order - filter coefficients
[SOS1, gain1] = zp2sos(z1, p1, k1); %Convert coefficients to SOS form
lpf = dfilt.df2tsos(SOS1, gain1); %Create filter object

[z, p, k] = butter(1,f_HPF*2/f_s,'high'); %Hardware HPF of 1 order - filter coefficients
[SOS, gain] = zp2sos(z, p, k); %Convert coefficients to SOS form
hpf = dfilt.df2tsos(SOS, gain); % Create filter object

signal1=signal;

```

%% Calculate position of target as it moves on it trajectory, for a given transit time and sample rate.

```

TTar=floor(tmax-(tmax/5)); %Total time of target motion
nSamples = TTar*f_s + 0; %Total number of samples in signature records
TTar = nSamples/f_s; %Actual total time of target motion,due to
rounding
dT = 1/f_s; %Time step
Ref = zeros(1,nSamples); %reservation for one ref function
RefArr = zeros(nSamples,TotV,nCrossPoint);%reservation for array of reference functions

```

%% loop over crossing point

```

for iYc = 1:nCrossPoint %Loop over crossing points
for iV = 1:TotV %Loop over velocity Vx and Vy
time = dT*(1:nSamples); %Time at which sample will occur
factor = time - TTar/2; %Multiplicative constant
xPos = CrossPoint(iYc) + xVel(iV)*factor;%Calculate x position
yPos = yVel(iV)*factor; %Calculate Y position
txRange = sqrt((xPos + baseline/2).^2 + yPos.^2); %Calculate range of target from Tx
rxRange = sqrt((xPos - baseline/2).^2 + yPos.^2); %Calculate range of target from Rx

```

```

    refSig = -sin(2*pi*frequency/c*(txRange+rxRange+(TargetLength)-BL*1)); %Generate
reference target signature
    refSig1=gausswin(nSamples,2)'.*refSig;
    fRefSig(:,iV,iYc) = filter(hpf, refSig1); %Filter the signature - LPF then HPF
end
end

refTimeV = (0:nSamples-1)/f_s; %Time vector for reference signal
toc;
tic;

% % filter for integration of the product

f_LPF_integr=0.8;
[z5,p5,k5] = butter(5,f_LPF_integr*2/f_s,'low'); %Hardware LPF of 5 order

[sos5,g5] = zp2sos(z5,p5,k5); %Convert to SOS form
Filt_integr = dfilt.df2tsos(sos5,g5); %Create a dfilt object

sigLength = length(signal); %Total length of measured signal record
windowSamples = nSamples; %Number of samples in a window - to match
reference signals
overlap = 0.75; %Set overlap as a fraction of the window length
overlapSamples = ceil(overlap*windowSamples); %Number of samples in window overlap
nFullWindows = floor((sigLength - overlapSamples)/(windowSamples - overlapSamples));
%Total number of full windows in signal
sampleStep = windowSamples - overlapSamples; %Window step size in samples
corLength = 2*windowSamples-1; %Length of correlation product
corA = zeros(corLength,TotV,nCrossPoint);
velInd = zeros(1,nFullWindows);
cPInd = zeros(1,nFullWindows);

% % Potentially not needed results - maybe remove from code to speed up as could just
use results

```

```

maxCorA = zeros(1,nFullWindows);
bestCorA = zeros(corLength,nFullWindows);
fBestCorA= zeros(corLength,nFullWindows);
maxFBestCorA = zeros(1,nFullWindows);

```

%% Correlations

```

for iWindow = 1:nFullWindows                                %Loop over number of windows
    NumbWind(iWindow)=iWindow;
    startSample = sampleStep*(iWindow - 1) + 1;             %Start sample of window
    endSample = startSample + (windowSamples - 1); %End sample of window
    for iCrossPt = 1:nCrossPoint                             % Loop over crossing points
        for iXYVel = 1:TotV                                  %Loop over velocity Vx and Vy
            corA(:,iXYVel,iCrossPt) =
xcorr(fRefSig(:,iXYVel,iCrossPt),signal(startSample:endSample),'none'); %Correlate
windowed signal with all refernce signatures
        end
    end
    [maxCor,maxCorInd] = max(corA(:));                       %Find maximum of correlation
products(best correlation) for window and its index in the array
    [s vel cP] = ind2sub([corLength,TotV,nYc],maxCorInd); %Find subscripts of maximum of
correlation products
    maxCorA(iWindow) = maxCor;                               %Record maximum of
correlation products for signal window
    bestCorA(:,iWindow) = corA(:,vel,cP);                   %Save correlation product with
maximum for signal window (best reference signal match)
    velInd(iWindow) = vel;                                   %Index of velocity component
for best correlation product
    cPInd(iWindow) = cP;                                     %Index of crossing point for best correlation product
    fBestCorA(:,iWindow) = filter(Filt_integr,corA(:,vel,cP)); %Filtered best correlation
product
    maxFBestCorA(iWindow) = max(fBestCorA(:,iWindow)); %Maximum of filtered best
correlation product

```

```

if iWindow == nFullWindows
    subSample = endSample;           % Number of samples unique to each separate
window
else
    subSample = endSample - overlapSamples;
end

nSig(startSample:subSample) = maxFBestCorA(iWindow); % Create new waveform from
maximums of filtered correlation products
end

clear corA;
nSigTimeV = (0:subSample-1)/f_s;    % Create time vector for new signal
toc

```

% % Create vectors of pre-threshold, threshold and detections

```

detection = zeros(nFullWindows,1);
winStart = zeros(nFullWindows,1);
avVect = zeros(winAv,1);           % Reserved array of stdev of win Av consecutive
windows, so if winAv=3 - then this is 3 element vector
for iWindow = 1:nFullWindows       %Loop over number of windows
    startSample = sampleStep*(iWindow - 1) + 1;    %Start sample of window
    endSample = startSample + (windowSamples - 1); %End sample of window
    winStart(iWindow) = (startSample - 1)/f_s;      %Start time of each window
    if iWindow == nFullWindows
        preThreshSample = endSample;               %Number of samples unique to each
seperate window
    else
                                                %Accounts for end of signal
        preThreshSample = endSample - overlapSamples;
    end
    if iWindow <= winAv                     %Condition to produce initial set of standard deviation
averages from first winAv windows
        winSD = std(bestCorA(:,iWindow));          %Standard deviation of this windows
correlation product
        avVect(iWindow) = winSD;                  %Fill averaging vector with standard deviations
    end
end

```

```
preThresh = mean(avVect(1:iWindow));%Perform averaging of standard deviations
avVect(iWindow) = preThresh;      %Assign pre-threshold value
threshold = preThresh*tMulti;      %Assign threshold value
preThreshVec(startSample:preThreshSample) = preThresh;  %Store pre-threshold
value
threshVec(startSample:preThreshSample) = threshold;      %Store threshold value
else
winMax = max(nSig(startSample:endSample));
if winMax > threshold
detection(iWindow) = 1;
preThreshVec(startSample:preThreshSample) = preThresh;
threshVec(startSample:preThreshSample) = threshold;
else
winSD = std(bestCorA(:,iWindow));
avVect(1:(winAv-1)) = avVect(2:winAv);%Sort for the case of winAv = 1, this works
but could be better
avVect(winAv) = winSD;
preThresh = mean(avVect);
avVect(winAv) = preThresh;
threshold = preThresh*tMulti;
preThreshVec(startSample:preThreshSample) = preThresh;
threshVec(startSample:preThreshSample) = threshold;
end
end
end
```

% % Count number of detections and corresponding time durations

```
offsetDetection = [detection(2:end); NaN];      %Offset detection vector by one
sample
risingEdge = find(detection == 0 & offsetDetection == 1); %Find rising edges
startWindow = risingEdge + 1;                  %Actual start window
fallingEdge = find(detection == 1 & offsetDetection == 0); %Find falling edges which are
actual end windows
```

```
nDetections = length(risingEdge); %Count rising edges
fprintf('Number of detections is %i.\n', nDetections);
if (nDetections == length(fallingEdge))
    nDuration = nDetections;
    riseOnly = 0;
else
    nDuration = nDetections - 1;
    riseOnly = 1;
end
for iDetect = 1:nDuration
    ordinal = iptnum2ordinal(iDetect); %translate the integer number into a
    string
    tDetect = winStart(startWindow(iDetect));
    eTDetect = winStart(fallingEdge(iDetect));
    dDetect = eTDetect - tDetect;

    fprintf('The %s detection began at %fs,ended at %fs and lasted %fs min.\n', ordinal,
    tDetect/60, eTDetect/60 , dDetect/60);
end
if riseOnly
    ordinal = iptnum2ordinal(nDetections);
    tDetect = winStart(startWindow(nDetections));
    fallingEdge(nDetections) = nFullWindows;
    fprintf('The %s detection was at %fs and lasted till the end of the record.\n', ordinal,
    tDetect/60);
end

% % Find average window in each range of detection windows and locate appropriate
values of velocity and angle

for iDD=1:nDetections
    [max_corr1(iDD),ind1(iDD)]=max(maxCorA(startWindow(iDD):fallingEdge(iDD)));
end
```

```

avDetWin = startWindow+ind1'-1           %Average window in detection!!!!
TTot_adapt= winStart(avDetWin)/60;

figure(112)

for iDetk=1:nDetections
subplot(nDetections,1,iDetk), plot(abs(bestCorA(:,avDetWin(iDetk)))), grid on;
strD=num2str(iDetk);
legend(strcat('Det event',strD));
end

    tgtCPoint = CrossPoint(cPInd(avDetWin));
    tgtXVel = xVel(velInd(avDetWin));
    tgtYVel = yVel(velInd(avDetWin));
    tgtVa = sqrt(tgtXVel.^2+tgtYVel.^2);
    tgtVakmph= tgtVa %.*1e-3.*3600;
    angleT=atand(tgtYVel./tgtXVel);
    MaxRefSig=fRefSig(:,velInd(avDetWin(iDetk)));
    tgtVa=tgtVa';
    angleT=angleT';
    tgtCPoint=tgtCPoint';

    velocityEst(j).fileName = filelista(j).name;
    velocityEst(j).velValue =tgtVa;
    velocityEst(j).CrossPoint =tgtCPoint;
    velocityEst(j).AngleT =angleT;
    velocityEst(j).RefSig=MaxRefSig;
    velocityEst(j).Baseline=baseline;
for iDetect = 1:nDetections
    ordinal = iptnum2ordinal(iDetect);
    fprintf("\n \n The %s detection crossed at \n %f m \n',ordinal,tgtCPoint(iDetect));
    fprintf('with velocity components \n %fm/s, %f m/s.\n',tgtXVel(iDetect),tgtYVel(iDetect));
    fprintf('or with abs velocity components \n %f m/s, %f
Degree\n',tgtVakmph(iDetect),angleT(iDetect));

```

end

```
sigTimeV = (0:length(signal)-1)/sRate;           %Create original signal time  
vector
```

% % Plots

figure (333)

```
for iDetk=1:nDetections  
    startSample1 = sampleStep*(avDetWin(iDetk) - 1) + 1; %Start sample of window  
    endSample1= startSample1 + (windowSamples - 1);      %End sample of window  
    time1= dT*(startSample1:endSample1);  
    subplot(nDetections,2,iDetk*2-1), plot(time1,signal(startSample1:endSample1)), grid on;  
    subplot(nDetections,2,iDetk*2),  
    plot(time1,fRefSig(:,velInd(avDetWin(iDetk)),cPInd(avDetWin(iDetk)))), grid on;  
    strD=num2str(iDetk);  
    legend(strcat('Detected section ',strD));  
end  
end;  
eval(['save ' velocityEst_FNmat ' velocityEst']);
```

D2. Experiment on vehicle recognition

- The code extracts features of vehicles. The type of vehicles is selected from the database as the input of the algorithm
- Few parameters need to be set as the input of the code
 - Types of vehicle
 - Frequency of the signal

MAIN PROGRAM:

% Matlab version 7.11.10 (R2010b)

% File name: run_vehRec_featExtraxt_psd_database.m

% Purpose: Calculation of psd for all given data - includes the speed normalization. PSDs of vehicles in each category are saved into a .mat file with the category name (psdCar1, etc).

% Created by: Emileen Rashid

clear;

close all;

randn('state', 0);

%% Parameter (choose 1, 2 or 3)

Nfrequency1 = 2; % Frequency used: 1 = 64 MHz, 2= 151 MHz, 4= 434 MHz

freq=[64 151 434];

freqUsed =freq(Nfrequency1);

index=[20:60 20:100 30:100];

indFreqUse =index(Nfrequency1);

num_signal=2000;

%% Read the list of files to be processed

[listFileFull{1}, listFileShort{1}] = readListFile('.\fileList_nissan.txt');

[listFileFull{2}, listFileShort{2}] = readListFile('.\fileList_Landrover_19april08.txt');

[listFileFull{3}, listFileShort{3}] = readListFile('.\fileList_bmw308i.txt');

```
[listFileFull{4}, listFileShort{4}] = readListFile('.\fileList_fordfocus.txt');
```

% % Feature Extraction (PSD) using velocity estimation algorithm

```
[psdCar1,f1] = featExtract_psd_velEst(listFileFull{1}, 'velocityEst.mat', freqUsed,  
num_signal);  
[psdCar2,f2] = featExtract_psd_velEst(listFileFull{2}, 'velocityEst.mat', freqUsed,  
num_signal);  
[psdCar3,f3] = featExtract_psd_velEst(listFileFull{3}, 'velocityEst.mat', freqUsed,  
num_signal);  
[psdCar4,f4] = featExtract_psd_velEst(listFileFull{4}, 'velocityEst.mat', freqUsed,  
num_signal);
```

% % Saving feature extraction

```
switch (freqUsed)  
    case 64  
        save psdCar1_64 psdCar1;  
        save psdCar2_64 psdCar2;  
        save psdCar3_64 psdCar3;  
        save psdCar4_64 psdCar4;  
  
    case 151  
        save psdCar1_151 psdCar1;  
        save psdCar2_151 psdCar2;  
        save psdCar3_151 psdCar3;  
        save psdCar4_151 psdCar4;  
  
    case 434  
        save psdCar1_434 psdCar1;  
        save psdCar2_434 psdCar2;  
        save psdCar3_434 psdCar3;  
        save psdCar4_434 psdCar4;  
end
```

%% Calculation of the PCA matrix using the function princomp.m.

```
nFilesCar(1)=size(psdCar1,1);nFilesCar(2)=size(psdCar2,1);
nFilesCar(3)=size(psdCar3,1);nFilesCar(4)=size(psdCar4,1);
psdCarAll =
[psdCar1(:,indFreqUse);psdCar2(:,indFreqUse);psdCar3(:,indFreqUse);psdCar4(:,indFreqUse
)];
[pc,score,latent,tsquare] = princomp(psdCarAll);
```

%% Plotting the data in the PC-space 2D and 3D

```
plotDataPCAspace(score(:,1:2), [1 2], [ nFilesCar(1) nFilesCar(2) nFilesCar(3) nFilesCar(4)
], ['rv';'bd';'go'; 'm+'])
```

```
plotDataPCAspace(score(:,1:3), [1 2 3], [ nFilesCar(1) nFilesCar(2) nFilesCar(3)
nFilesCar(4)], ['rv';'bd';'go'; 'm+'])
```

%% Vehicle recognition - Training procedure for kNN classification**%% For testing procedure, please disable this function**

```
nPCA=5; % no of PCA to use
nKNN=4; % no of K nearest neighbour
nameCar{2}='nissan';
nameCar{1}='landrover';nameCar{3}='bmw308i';nameCar{4}='fordfocus';
nFilesCarAll = sum(nFilesCar(1:4));
trainModelKNN=vehRec_train(psdCarAll,nPCA);
trainModelKNN.nFilesCar = nFilesCar;
trainModelKNN.nameCar = nameCar;
save trainModelKNN trainModelKNN % saving database
```

%% Plotting the variance retain by each of the PC

```
percent_explained=100*latent/sum(latent);
figure;
```



```
pareto(percent_explained);  
xlabel('Principle Component');  
ylabel('variance Explained(%');
```

%% Plotting target's PSD

```
figure  
plot(f1(1:600),psdCar1(1,1:600),'r')  
hold  
plot(f2(1:600),psdCar2(1,1:600),'b')  
plot(f3(1:600),psdCar3(1,1:600),'g')  
plot(f4(1:600),psdCar4(1,1:600),'m')  
xlim([0 5])  
ylim([-40 0])  
xlabel('Frequency,Hz');  
ylabel('PSD,dB');
```

SUBFUNCTION: readListFile.m

% Matlab version 7.11.10 (R2010b)

% File name: readListFile.m

% Purpose: Reads the list of files to be processed and stores in the variable

% Created by: Peter Jancovic

function [listFileFull, listFileShort] = readListFile(fileName_listFile)

fid = fopen(fileName_listFile, 'rt');

i=1;

while feof(fid)==0

 fn_inp = fscanf(fid, '%s', 1);

 fn_inp = strrep(fn_inp, '/', '\');

 listFileFull{i} = fn_inp;

% % Extract only the name of the file (no directory, no extension); Assuming the extension is 3 letters long

 fn_inp_short = fn_inp(max(findstr(fn_inp, '\'))+1:end-4);

 listFileShort{i} = fn_inp_short;

 i=i+1;

end

fclose(fid);

SUBFUNCTION: featExtract_psd_velEst.m

% Matlab version 7.11.10 (R2010b)

% File name: runFeatExtract_psd_velEst.m

% Purpose: Calculate the PSD for a given list of files and return. Using the velocity estimates obtained by an automatic algorithm.

% Created by: Emileen Rashid

```
function [powSpec,f] = featExtract_psd_velEst(listFileFull, fileName_velEst, freqUsed,  
num_signal)
```

```
dataDir = listFileFull{1}(1:max(findstr(listFileFull{1},'\')));  
nFiles = length(listFileFull);
```

%% Read the estimated velocity from .mat file

```
velocityEst_FNmat = [dataDir fileName_velEst];  
eval(['load ' velocityEst_FNmat]);
```

```
switch (freqUsed)
```

```
    case 64,
```

```
        freqUsedInd = 1;
```

```
    case 151,
```

```
        freqUsedInd = 3;
```

```
    case 434,
```

```
        freqUsedInd = 5;
```

```
end
```

```
fileNameAll = {velocityEst(1:nFiles).fileName};
```

%% Process the target signals for the current directory

```
for j=1:nFiles;
```

%% Read the binary file containing 6 target signals

```
    fileName = listFileFull{j};
```

```
fileNameShort = fileName(max(findstr(fileName,'\')+1:end);
[data] = readTargetSignalBin(fileName, 6);
    td = data(freqUsedInd,:);
if length(td)<num_signal
    td(end+1:num_signal)=0;
else
    td=td(1:num_signal);
end
```

```
fs = 100;           % Sampling frequency (6*100 Hz)
```

% % Time array

```
dt = 1/(fs);
n = length(td);    % number of samples
t = [-n/2*dt:dt:(n-1)/2*dt];
```

% % Adding DC filter

```
n=length(td);
avg = sum(td)/n;
td = td-avg;
```

% % Adding Hamming window

```
w = hamming(n);
W = diag(sparse(w));
td = w' .* td;
figure(1)
plot(t,td),grid on    % plotting signal
title('Time domain signature','FontSize',12);
xlabel('Time, min','FontSize',12);
ylabel('Amplitude, V','FontSize',12);
```

%% Find the velocity corresponding to the current filelist(j).name

```
ind = find(strcmp(fileNameAll(1:nFiles),fileNameShort),1);  
velEst(j) = velocityEst(ind).velValue;
```

%% Calculate PSD (use signal with padded zeros)

```
nFFT = length(td);  
[psgw,f] = calcPSD_v1(td, fs, nFFT);      %% calculating PSD  
storepsgw(j,1:length(psgw))=psgw;        %% store psgw
```

%% Normalise the calculated PSD by the max power level

```
normalised=10*log10(psgw/max(psgw));  
storeNormalised(j,1:length(normalised)) = normalised;
```

%% Normalize the power spectrum based on the estimated velocity**%% Considers that the velocity effects the spectrum in a form of widening/narrowing**

```
SpeedNormalised = resample(normalised,100,round(10*velEst(j)));  
powSpec(j,(1:length(SpeedNormalised))) = (SpeedNormalised)';
```

```
end
```

SUBFUNCTION: vehRec_train.m

% Matlab version 7.11.10 (R2010b)

% File name: vehRec_train.m

% Purpose: Training procedure for kNN classification. Returns the variable trainModelKNN which contains the PCA matrix and the vehicles in the PCA-space.

% Created by: Peter Jancovic

function [trainModelKNN] = vehRec_train(psdCarAll, nPCA);

[pc,score,latent,tsquare] = princomp(psdCarAll);

trainModelKNN.psdCarAllMean = mean(psdCarAll);

trainModelKNN.transPCA = pc(:,1:nPCA);

trainModelKNN.dataPCA = score(:,1:nPCA);

save trainModelKNN trainModelKNN

SUBFUNCTION: calcPSD v1.m

% Matlab version 7.11.10 (R2010b)

% File name: calcPSD_v1.m

% Purpose: to calculate power spectral density

% Created by: Emileen Rashid

function [inpSigPSD, freqAxisLab] = calcPSD_v1(inpSig, Fs, nFFT);

if (nFFT<length(inpSig))

 disp('WarningMine: nFFT is less than the signal length');

end

if isempty(nFFT)

 nFFT = 2*length(inpSig);

end

freqRes = Fs/nFFT;

freqAxisLab = freqRes*[0:nFFT-1]; % calculate the labels for the frequency

axis

inpSigFFT = fft(inpSig,nFFT); % FFT calculation

inpSigPSD = inpSigFFT.*conj(inpSigFFT)/nFFT; % power spectral density

inpSigPSD = abs(inpSigPSD(1:floor(nFFT/2)));

D3. Experiment on vehicle classification

- Since we have small database, for testing and training, the same database is used.
- The code implements a leave-one-out procedure where the code leaves one out for testing and uses the remaining data for training.
- User needs to specify which frequency is to be used for estimating the speed, number of PCA components and KNN to be used
- This code loads the PSDs of all given vehicles.
- The testing data is transform testing PSD to the PCA-space
- The Euclidean distance between testing and training data in PC space was then calculated. The distance is calculated based on specified number of PCs.
- The testing data is classified by finding the smallest distances.
- The code is given below with description of the function used.

% Matlab version 7.11.10 (R2010b)

% File name: run_vehRecNoisy_testLeave1Out.m

% Purpose: to calculate power spectral density

% Created by: Peter Jancovic

% Modified by: Emileen Rashid

clear all;

% % Set the parameters to be used

freqUsed = 64; % 64 MHz, 151MHz or 434 MHz;

nPCA = 3; % number of PCA components

nKNN = 3; % number of nearest neighbours considered

% % Load PSDs for each vehicle type and set frequency-region of the PSD to be used for PCA calculation

switch (freqUsed)

 case 64


```
indFreqUse=20:70; load psdCar1_64; load psdCar2_64; load psdCar3_64;load
psdCar4_64;
case 151
indFreqUse=30:100; load psdCar1_151; load psdCar2_151; load psdCar3_151;load
psdCar4_151;
case 434
indFreqUse=40:100; load psdCar1_434; load psdCar2_434; load psdCar3_434;load
psdCar4_434;
end
```

%% Take only a part of the PSD

```
nFilesCar(1)=size(psdCar1,1);nFilesCar(2)=size(psdCar2,1);
nFilesCar(3)=size(psdCar3,1);nFilesCar(4)=size(psdCar4,1);
psdCarAll = [psdCar1(:,indFreqUse); psdCar2(:,indFreqUse); psdCar3(:,indFreqUse);
psdCar4(:,indFreqUse)];
nFilesCarAll = sum(nFilesCar(1:4));

nCarTypes = 4;
indCarTest = 0;
AddNoiseIndCarTest=0;
recConfMatrix = zeros(nCarTypes,nCarTypes);
```

%% Perform a Leave-1-Out training/testing procedure

```
for j=1:nCarTypes,
if (j>1)
indCarTest = sum(nFilesCar(1:j-1));
end

nFilesCarCurr(1)=nFilesCar(1); nFilesCarCurr(2)=nFilesCar(2);
nFilesCarCurr(3)=nFilesCar(3); nFilesCarCurr(4)=nFilesCar(4);
nFilesCarCurr(j) = nFilesCarCurr(j)-1;
```

```
for i=1:nFilesCar(j),
```

```
    indCarTest = indCarTest + 1;
```

```
    %% Form the training and testing data
```

```
    psdCarAll_trainL1O = [psdCarAll(1:indCarTest-1,:);
```

```
    psdCarAll(indCarTest+1:end,:)];
```

```
    %% Perform training
```

```
    trainModelKNN = vehRec_train(psdCarAll_trainL1O,nPCA);
```

```
    %% Transform the testing PSD to the PCA-space
```

```
    psdCar_test=psdCarAll(indCarTest,:);
```

```
    pcaCar_test=(psdCar_test-
```

```
    trainModelKNN.psdCarAllMean)*trainModelKNN.transPCA;
```

```
    %% Perform testing
```

```
    pcaCar_test_Matrix = repmat(pcaCar_test,nFilesCarAll-1,1);
```

```
    distMatrix = sqrt(sum((pcaCar_test_Matrix - trainModelKNN.dataPCA).^2,2));
```

```
    %% Calculating distance between testing signal to training data
```

```
    distCar1Ave_tmp=sort(distMatrix(1:nFilesCarCurr(1)));
```

```
    distCar2Ave_tmp=sort(distMatrix(nFilesCarCurr(1)+1:nFilesCarCurr(1)+nFilesCarC  
urr(2)));
```

```
    distCar3Ave_tmp=ort(distMatrix(nFilesCarCurr(1)+nFilesCarCurr(2)+1:nFilesCarCu  
rr(1)+nFilesCarCurr(2)+nFilesCarCurr(3)));
```

```
    distCar4Ave_tmp=sort(distMatrix(nFilesCarCurr(1)+nFilesCarCurr(2)+1:nFilesCarC  
urr(1)+nFilesCarCurr(2)+nFilesCarCurr(3)+nFilesCarCurr(4)));
```

```
    % finding mean for each distance
```

```
    distCar1Ave = mean(distCar1Ave_tmp(1:nKNN));
```

```
    distCar2Ave = mean(distCar2Ave_tmp(1:nKNN));
```

```
    distCar3Ave = mean(distCar3Ave_tmp(1:nKNN));
```

```
    distCar4Ave = mean(distCar4Ave_tmp(1:nKNN));
```

```
[m, mI] = min([distCar1Ave distCar2Ave distCar3Ave distCar4Ave]);  
recConfMatrix(j,mI) = recConfMatrix(j,mI)+1;  
end  
  
%% Calculating percentage of classification  
  
recogAcc(j) = recConfMatrix(j,j)/sum(recConfMatrix(j,:));  
end  
  
disp(['Recognition Accuracy for each car type: ' num2str(round(recogAcc*100)) '  
[%]');
```

D4. Simulation signal using two-ray path propagation model

- Simulated signal is simulated using two-ray path propagation model.
- This is in order to control analysis environment for example speed, clutter free

% Matlab version 7.11.10 (R2010b)

% File name: SigMod_forEmileen.m

% Purpose: to simulated target signal using two-ray path propagation model

% Created by: Marina Gashinova

% Modified by: Emileen Rashid

close all

clear all

% % Parameters setting

Nfrequency=3;

BL=100; % baseline length

a=15; %m/s

b=a*3.6; %conversion to km/h

crossing=0; %Crossing point starting from the middle

TrajPar1=[b; 90; crossing]; %speed in km/h, second - angle in deg,

NLength=1;% %for Landrover=1, fordd=2,bmw=3, skoda=4

NHeight=NLength;

height=[2.2 1.503 1.421 1.498 1]; %LR=2.2,ford=1.503,bmw=1.421,skoda=1.498

length=[4.8 4.468 4.531 4 1]; %LR=4.8, ford=4.468, bmw=4.531, skoda=4

tHeight=height(NHeight);

tLength=length(NLength);

FLAG=1; %FLAG = 0 if there is no pi/2 phase shift (cos approximation). 1 - otherwise (sin approximation)

sRate=100; % sampling rate ,Hz

%% Constant: Global eps0 mu0 vlight freq lpf h hpf h BL

```

eps0 = 8.85412e-12;
mu0 = 4*pi*1e-7;
vlight = 1/sqrt(eps0*mu0);
c=vlight;

flag=4;
if flag==4
    freq=[64.125 135.05 151.00 434.45]*1e6 ;
else
    freq=[64.125 434.45 135.05]*1e6;
end

```

%% Antenna's height

```

hT1=[0.25 0.25 0.2 0.19];           % transmitter height
hR1=[0.25 0.25 0.2 0.19];           % receiver height
f_s=sRate;
f_LPF=5-0.1;                         % cut-off frequency for LPF
f_HPF=0.1;                           % cut-off frequency for HPF
f_LPF_s=f_LPF/2;
f_HPF_s=0.25;

Nfreq=Nfrequency;
FreqT=freq(Nfreq);
BL_t=BL;                             % baseline
dT=BL_t/2;
Speed_t=TrajPar1(1);                 % Speed in kmph
Angle_t=TrajPar1(2)-0.0001*0;        % angle in degree
CrossingPoint_x0=0;                  % crossing point from the middle (Normal to BL)
CrossingPoint_y0=TrajPar1(3);

Speed_MpS=Speed_t*1e3/3600;AngleRad=Angle_t*pi/180;    % Calculating speed

```

```
V_x=Speed_MpS*sin(AngleRad);V_y=Speed_MpS*cos(AngleRad);  
hT=hT1(Nfreq);  
hR=hR1(Nfreq);
```

```
Tsig=20; % Signal long in time  
Nsig=Tsig*f_s; % number of samples
```

```
TimeSig=(-Nsig/2:Nsig/2)*1/f_s;
```

%% Trajectory by x(t) and y(t)

```
x_n=V_x*TimeSig+CrossingPoint_x0; % target trajectory in x-axis  
y_n=V_y*TimeSig+CrossingPoint_y0; % target trajectory in y-axis
```

%% Reflection Coefficient

```
Polar='V';  
EpsR=15; % epsilon of ground  
SigmaR=0.02; % conductivity of ground
```

```
%(eps,sigma) (7,0.002) - concrete dry; (15,0.05) - average ground;  
%(25,0.02) -wet ground
```

%% Forward Scatter RCS of target

```
ApertEff=tLength*tHeight;  
Lambda=vlight/FreqT;  
AlfaH=atan(x_n./(dT+y_n));  
BetaH=atan(x_n./(dT-y_n));  
Fi=atan(V_x./V_y);  
dR1=sqrt((dT-y_n).^2+x_n.^2);  
dT1=sqrt((dT+y_n).^2+x_n.^2);  
aNu1=atan(hT./dT1);
```

```

aNu2=atan((tHeight-hT)./dT1);
ANu=aNu1+aNu1;

bNu1=atan(hR./dR1);
bNu2=atan((tHeight-hR)./dR1);
BNu=bNu1+bNu1;
RCS_lin=4*pi*((sin(-
AlfaH+Fi)).^2.*(cos(ANu)).^2*(ApertEff/Lambda)^2.*sinc((pi*tLength*sin(AlfaH+Fi).*sin(
AlfaH+BetaH))./(pi*Lambda)).^2.*sinc((pi*tHeight.*cos(ANu).*sin(ANu+BNu))./(pi*Lamb
da)).^2);

RCS_dB=10*log10(RCS_lin);

st2=sqrt(RCS_lin);
% st2=1;
% plot(TimeSig,st2)

%% DIRECT PATH SIGNAL Tx - Rx (LEAKAGE SIGNAL)

%% Direct signal

w1=sqrt(BL_t.^2+(hT-hR).^2); Fi1=2*pi*w1*FreqT./vlight; A1=1./(2*Fi1);
E1=A1.*exp(i*Fi1);

%% Multipath signal

w2=sqrt(BL_t.^2+(hT+hR).^2); ThetaL2=atan((hT+hR)/BL_t); Fi2=2*pi*w2*FreqT./vlight;
KRef=RefCoeff( EpsR,SigmaR,FreqT,ThetaL2,Polar);
A2=KRef./(2*Fi1);
E2=A2.*exp(i*Fi2);
clear w1 w2 Fi1 Fi2 A1 A2 KRef

%% Transmitter-to-target

z=0.88;

```

%% Direct signal

```
w3=sqrt((dT-y_n).^2+(hT-z).^2+x_n.^2);  
Fi3=2*pi*w3*FreqT/vlight;  
A3=1./(2*Fi3);  
E3=A3.*exp(i*Fi3);
```

%% Multipath signal

```
w4=sqrt((dT-y_n).^2+(hR+z).^2+x_n.^2); ThetaL4=atan((hT+z)./w4);  
Fi4=2*pi*w4*FreqT/vlight;  
KRef=RefCoeff( EpsR,SigmaR,FreqT,ThetaL4,Polar);  
A4=KRef./(2*Fi4);  
E4=A4.*exp(i*Fi4);
```

```
clear w3 w4 Fi3 Fi4 A3 A4 KRef
```

%% TARGET-TO-RECEIVER**%% Direct signal**

```
w5=sqrt((dT+y_n).^2+(hT-z).^2+x_n.^2);  
Fi5=2*pi*w5*FreqT/vlight; A5=1./(2*Fi5);  
E5=A5.*exp(1i*Fi5);  
E5=A5.*exp(1i*(Fi5-pi/2*FLAG));
```

%% Multipath signal

```
w6=sqrt((dT+y_n).^2+(hR+z).^2+x_n.^2);  
ThetaL6=atan((hR+z)./w6); Fi6=2*pi*w6*FreqT/vlight;  
KRef=RefCoeff( EpsR,SigmaR,FreqT,ThetaL6,Polar);  
A6=KRef./(2*Fi6);  
E6=A6.*exp(i*Fi6);  
E6=A6.*exp(1i*(Fi6-pi/2*FLAG));
```

```
clear w5 w6 Fi5 Fi6 A5 A6 KRef
```


%% FREE SPACE APPROXIMATION

```
LeakPower=20*log10(abs(E1));  
TargSig=E3.*E5.*st2;  
TargSigAbs=abs(TargSig);  
ReceivedSig=E1-TargSig;  
ReceivedPower=20*log10(abs(ReceivedSig));
```

%% TRP APPROXIMATION

```
LeakPowerTRP=20*log10(abs(E1)+abs(E2));  
TargSigTRP=(E3+E4).*(E5+E6).*st2;  
  
% plot(TimeSig,real(TargSigTRP))  
TargSigAbsTRP=abs(TargSigTRP);  
ReceivedSigTRP=E1+E2-TargSigTRP;  
ReceivedSigTRPAbs=abs(ReceivedSigTRP);  
ReceivedPowerTRP=20*log10(ReceivedSigTRPAbs);  
ETRP=ReceivedSigTRPAbs;  
ATRP=ReceivedPowerTRP;
```

%% RECEIVER CALIBRATION

```
X6=[-140 -130 -120 -110 -100 -90 -80 -70 -60 -50];  
Y64=[0.54 0.574 0.742 1.011 1.276 1.521 1.779 2.015 2.141 2.146];  
Y135=[0.536 0.604 0.808 1.098 1.355 1.59 1.882 2.117 2.38 2.424];  
Y173=[0.469 0.528 0.732 0.982 1.248 1.502 1.752 1.998 2.285 2.388];  
Y434=[0.397 0.428 0.6 0.85 1.162 1.425 1.661 1.965 2.207 2.394];  
  
Y_all=vertcat(Y64,Y135,Y173,Y434);  
  
Ps1=polyfit(X6,Y_all(Nfreq,:),4);  
  
Fs2=polyval(Ps1,X6);
```

```
% plot(X6,Y_all(Nfreq,:), 'o', X6, Fs2, '-')
```

%% ADJUSTMENT TO AMPLITUDE OF THE DOPPLER SIGNATURE BY REAL DOPPLER RECEIVER HARDWARE

```
SigCal=polyval(Ps1,ATRP);  
SigCal1=SigCal-mean(SigCal);  
SigCal2=SigCal1*32;
```

%% FILTERING

%% Hardware filters

```
[z1, p1, k1] = butter(5,f_LPF*2/f_s, 'low'); % Hardware LPF of 5 order - filter coefficients  
[SOS1, gain1] = zp2sos(z1, p1, k1); % Convert coefficients to SOS form  
lpf_h = dfilt.df2tsos(SOS1, gain1); % Create filter object
```

```
[z2, p2, k2] = butter(1,f_HPF*2/f_s, 'high'); % Hardware HPF of 1 order - filter coefficients  
[SOS2, gain2] = zp2sos(z2, p2, k2); % Convert coefficients to SOS form  
hpf_h = dfilt.df2tsos(SOS2, gain2); % Create filter object
```

```
PL2=SigCal2;  
RSSI=SigCal;  
figure (999)  
plot(TimeSig,RSSI)
```

```
figure (555)  
plot(TimeSig,PL2)
```

%% saving Doppler signature

```
tdName = [ 'RCSBL', num2str(BL), 'm.mat']; %  
td=PL2;  
save(tdName, 'td');
```

%% saving RSSI

```
tdName2= ['RSSIsigSinCross',num2str(crossing),'m.mat'];  
td=RSSI;  
save(tdName2,'td');
```

D5. Clutter simulation

- To simulate clutter based on the real environment clutter characteristics
- This is in order to analyse the effect of clutter to ATC system

% Matlab version 7.11.10 (R2010b)

% File name: SigMod_forEmileen.m

% Purpose: to simulated clutter signal that has similar characteristic as environment clutter

% Created by: Emileen Rashid

% function [NonStatClutter1] = ClutterSim(nSamples,clLevel,Nfrequency1,sRate)

% % parameters

close all

fs=20; % sampling frequency

fs2=fs/2; % Nyquist

dt = 1/(fs); % frequency step

time=20; % in min

totTime=time*60;

nSamples=totTime*fs;

clLevel = 4; % clutter level: 1 = weak; 2 = medium; 3 = strong; 4 = very strong;

Nfrequency1 = 3; % Frequency used:1=64MHz, 2=135MHz,

3=173MHz,4=434MHz

cutFEnv=0.08; % cutoff frequency for hilbert in 2nd stage

orderFEnv=2; % 2nd order filter for hilbert in 2nd stage

% sRate = 40; % sampling Rate

vlight= 3*10.^8; % speed of light

sRate=fs; % sampling frequency

fs2=fs/2; % nyquist

dt = 1/(fs);

```
freq=[64 135 173 434]*10^6;
frequency =freq(Nfrequency1);
CluttLevel=[1 2 3 4];
ClutterLevel =CluttLevel(clLevel);
strFreq=strcat(num2str(frequency*10^-6),blanks(1),' MHz');
strCLevel=strcat(num2str(ClutterLevel),blanks(1));
nSamples2=nSamples*2;      % number of samples for nonstationary envelope
t=1/sRate*(1:nSamples);
bin=500;                    % distribution parameter
t = [0:dt:(nSamples-1)*dt];
```

% % Stage 1: generating coloured noise

% % filter characteristics

```
CutF0=[0.4 0.5 0.5 0.5; 0.5 0.5 0.5 0.5; 0.4 0.4 0.4 0.4;0.5 0.5 0.5 0.5];
%(row - clutter strength,column - freq)

fNoise0=[0.001 0.001 0.001 0.001; 0.0015 0.002 0.0025 0.003; 0.0015 0.002 0.0025
0.003;0.006 0.003 0.003 0.003;];

envRangeMin0=[0.0024 0.0095 0.0118 0.0137; 0.0148 0.0861 0.0912 0.0773;0.0085 0.1312
0.143 0.1634; 0.01731 0.1544 0.1533 0.08341;]*1.2;

envRangeMax0=[0.0041 0.0228 0.0911 0.152; 0.0405 0.8593 1.0618 1.3007; 0.0939 1.8141
2.4551 3.2627; 0.2694 2.7789 3.0868 4.2691;]*1.2;

stdClutter0=[0.0016 0.0106 0.0252 0.0447; 0.1446 0.2518 0.3068 0.3234; 0.0247 0.4471
0.5536 0.6561; 0.03658 0.6673 0.7259 0.9493];

fOrder0=[1 1 1 2; 2 2 2 2; 2 2 2 3;3 3 3 3];

stdNoise=fNoise0(clLevel,Nfrequency1);
stdClutter=stdClutter0(clLevel,Nfrequency1);
```

```

envRangeMin=envRangeMin0(clLevel,Nfrequency1);
envRangeMax=envRangeMax0(clLevel,Nfrequency1);
cutF=CutF0(clLevel,Nfrequency1);
fOrder=fOrder0(clLevel,Nfrequency1);

```

%% clutter + noise generation

```

AWG=normrnd(0,stdClutter,1,nSamples); % generate random noise using AWGN
ScaleAWG=AWG; % clutter after scaled to real noise amplitude
*(realsqrt(2)*(stdClutter))

```

%% noise in receiver

```

coh=0.25;
[b2,a2] = butter(fOrder,coh/fs2,'high'); % Butterworth filter coefficients
clutterH = filter(b2,a2,ScaleAWG);

col=9.9;
[b1,a1] = butter(fOrder,col/fs2,'low'); % Butterworth filter coefficients
clutterL = filter(b1,a1,clutterH);

[b3,a3] = butter(fOrder,cutF/fs2,'low'); % Butterworth filter coefficients
clutter = filter(b3,a3,clutterL); % filter noise=clutter

noise=normrnd(0,stdNoise,1,nSamples); % generate random noise using AWGN
noise1=noise; % noise floor

StatClutter=clutter; % coloured noise/stationary clutter
figure (1)
plot(t/60,StatClutter),grid on % plotting stationary clutter
title('StatClutter','FontSize',12);
xlabel('Time, min','FontSize',12);
ylabel('Amplitude, V','FontSize',12);

```

```
NormStatClutter=StatClutter/max(StatClutter); % Normalised to 1
figure (2)
plot(t/60, NormStatClutter ) % plotting Normalised Stationary Clutter
title('Normalised Stationary Clutter','FontSize',12);
xlabel('Time, min','FontSize',12);
ylabel('Amplitude, V','FontSize',12);
```

% % Stage 2: modulation

% % pre-envelope

```
nSamples3=nSamples*3;
Env=normrnd(0,1,1,nSamples2); % generate random gaussian noise
env2 = abs(hilbert(Env)); % pre-envelope
[b4,a4] = butter(orderFEnv,cutFEnv/fs2,'low'); % Butterworth filter coefficients
envF = filter(b4,a4,env2); % filter noise=clutter
envF1=envF(nSamples+1:nSamples2);
```

```
figure(3)
plot(t/60,envF1) % plotting envelope
title('Envelope','FontSize',12);
xlabel('Time, min','FontSize',12);
ylabel('Amplitude, V','FontSize',12);
```

```
Env1=envF1-min(envF1); % normalised min value to 0
n1=length(envF1);
```

```
figure (4)
plot(t/60,Env1)
title('Normalised Min Envelope to 0','FontSize',20);
xlabel('Time, min','FontSize',12);
ylabel('Amplitude, V','FontSize',12);
```

```
Env2=Env1/max(Env1); % max amplitude normalised to 1
```

```
figure (5)
plot(t/60,Env2)
title('Normalised Max Envelope to 1','FontSize',20);
xlabel('Time, min','FontSize',12);
ylabel('Amplitude, V','FontSize',12);

ModEnvelope=Env2.*(envRangeMax-envRangeMin)+(envRangeMin); % modulation
process
```

```
figure(6)
plot(t/60,ModEnvelope)
title('Modulated Signal','FontSize',12);
xlabel('Time, min','FontSize',12);
ylabel('Amplitude, V','FontSize',12);
```

% % modulation process

```
NonStatClutter= (ModEnvelope.*NormStatClutter.*max(NormStatClutter));
% modulate stationary clutter to the envelope
```

% % adding noise floor

```
NonStatClutter1=NonStatClutter+noise1; % adding noise floor
```

```
% stdNonStatClutter1=std(NonStatClutter1);
% hgsave(strcat(strFreq,'Simulated Clutter'))
```

% % Plotting non-stationary clutter

```
figure (7)
plot(t/60,NonStatClutter1,'LineWidth',1)
hold on
title('Non-Stationary Clutter','FontSize',12);
axis([min(t/60) max(t/60) 0 max(NonStatClutter1) ])
```



```
xlabel('Time, min','FontSize',12);
ylabel('Amplitude, V','FontSize',12);
% hgsave(strcat(strFreq, ' NonStatClutterLevel',strCLevel))
```

%% Finding envelope for non-stationary clutter

```
env5 = abs(hilbert(NonStatClutter1));
[b4,a4] = butter(orderFEnv,cutFEnv/fs2,'low');    % Butterworth filter coefficients
envF2 = filter(b4,a4,env5);                      % envelope for non-stationary clutter
```

```
signal=NonStatClutter1(2401:length(NonStatClutter1))
envelope=envF2(2401:length(envF2));
```

```
n1=length(signal);
t1=[0:dt:(n1-1)*dt];
```

%% plotting envelope for non-stationary clutter

```
figure (8)
plot(t/60,envF2,'r','LineWidth',1)
title('Non-Stationary Clutter Envelope','FontSize',12);

axis([min(t/60) max(t/60) 0 max(envF2)*2.5 ])
xlabel('Time, min','FontSize',12);
ylabel('Amplitude, V','FontSize',12);
hgsave(strcat(strFreq, ' SignalWithEnvLevel',strCLevel))
```

%% Finding power spectrum density

%% psd for signal

```
[d f ] = pwelch(NonStatClutter1,[],[],[],fs);    % to find the PSD for the non stat clutter
        x = f;                                  % freq in x-axis
        y = d;                                  % data in y-axis
```

```
y1 = smooth(y,101);  
% smooth fxn for the data, Z = SMOOTH(Y,SPAN) smooths data Y using SPAN as the  
number of points used to compute each element of Z.
```

```
% figure (4)  
% plot (d,10*log10(f));
```

```
%% psd for envelope
```

```
[d1 f1 ] = pwelch(env5,[],[],[],fs); % to find the PSD for the signal  
xx = f; % freq in x-axis  
yy = d; % data in y-axis  
  
y2 = smooth(yy,101);  
  
figure(9)  
semilogx(f,10*log10(y1),'g-','LineWidth',2.0), grid on,  
xlim([10^-1 10^1]);  
title(strcat(strFreq, ' Power Spectral Density for the Non-stationary Clutter '))  
legend('Power spectrum')  
xlabel('Freq, Hz');  
ylabel('Power spectrum, dB');  
hgsave(strcat(strFreq, ' PSD Clutter Level',strCLevel))
```

%% Distributions

```
signal1 = StatClutter; % stationary clutter  
signal2 = NonStatClutter1; % Non Stationary clutter  
Samp_length1 = length (signal1); % calculated the length of the stationary clutter  
Samp_length2 = length (signal2); % calculated the length of the non stationary  
clutter
```

%% Weibull Distribution

```
t1=[0:dt:(n1-1)*dt];
```

```
signal3 = normrnd(0,1,1,10000);
```

```
Samp_length3 = length (signal3);
```

```
bin1=500;
```

%% Stationary

```
[FF2,int2,Q3]= weibull_dist_uni(signal3, Samp_length3,bin1); % weibull pdf  
distribution sub-function
```

```
figure (10)
```

```
plot(int2,FF2/Samp_length3,'g','linewidth',2), hold on,
```

```
% plot histogram with weibull distribution fit for stationary
```

```
% plot histogram with normrnd distribution fit
```

```
plot(int2,Q3/Samp_length3,'r');
```

```
title(strcat('Weibull Distribution for', blanks(1), strFreq, ' Stationary Clutter '))
```

```
legend ('Weibull Fit', 'Histogram')
```

```
hgsave(strcat(strFreq, ' WeibullStatLevel',strCLevel))
```

%% Non Stationary

```
[FF3,int3,Q4,fitb]= weibull_dist_uni(signal2, Samp_length2, bin);
```

```
% weibull pdf distribution sub-function
```

```
sprintf(' %5.1f ',fitb) ;
```

```
strwbl=strcat(num2str(sprintf(' %5.1f ',fitb)),blanks(1),' shape factor');
```

```
figure (11)
```

```
plot(int3,FF3/Samp_length2,'g','linewidth',2), hold on, % plot histogram with weibull  
distribution fit for non stationary
```

% % plot histogram with normal random distribution fit

```

plot(int3,Q4/Samp_length2,'r');
title(strcat( 'Weibull Distribution for ', blanks(1), strFreq, ' Non-Stationary Clutter'))
legend (strcat('Weibull Fit', strwbl), 'Histogram')
hgsave(strcat(strFreq, ' WeibullNonStatLevel',strCLevel))

```

```

temp1=NonStatClutter1(1:length(NonStatClutter1));
temp2=envF2(1:length (envF2));
temp1(temp1<max(temp1))=0;
temp2(temp2<max(temp2))=0;
NonStatClutter_p=find(temp1);
envF2_p=find(temp2);
shift_value=abs(NonStatClutter_p-envF2_p);

```

```

figure (555)
plot(t/60,NonStatClutter1)
xlabel('Non-Stationary Clutter','FontSize',12);
hold on;

plot(t(1:length (shift_value:length(envF2)))) /60
,1.2*envF2(shift_value:length(envF2)), 'r','LineWidth',2)
legend('NonStatClutter','Envelope')
xlabel('Time, min');
ylabel('Amplitude,V');

```

% % Plot all graphs

```

figure (12)

subplot(221)

plot(t/60,NonStatClutter,'b'), grid on, hold on, % plotting envelope for non-stationary
clutter
plot(t/60,envF2,'r','linewidth',2);

```

```
title(strcat(strFreq, ' Non-stationary Clutter with envelope'))
legend('NonStatClutter','Envelope')
xlabel('Time, min');
ylabel('Amplitude,V');

subplot(222)
plot(f1,10*log10(d1)), grid on,hold on,      % plotting stationary clutter
plot (f1,10*log10(y2),'r', 'linewidth',2)
title(strcat(strFreq, ' Power Spectral Density for the Non-stationary Envelope '))
legend('PSD')
xlabel('Freq, Hz');
ylabel('Power spectrum, dB');

subplot (223)
semilogx(f,10*log10(y1/max(y1)),'g-','LineWidth',2), grid on,
xlim([10^-1 10^1]);

title(strcat(strFreq, 'Power Spectral Density for the Non-stationary Clutter'))
legend('Power spectrum')
xlabel('Freq, Hz');
ylabel('Power spectrum, dB');

subplot (224)
plot(int3,FF3/Samp_length2,'g','linewidth',2), grid on, hold on,

plot(int3,Q4/Samp_length2,'r');
title(strcat( strFreq,'Weibull Distribution for Non-stationary Clutter '))
legend (strcat('Weibull Fit', strwbl), 'Histogram')
xlabel('Amplitude,V');
ylabel('PDF');

hgsave(strcat(strFreq, ' AllPlotLevel',strCLevel))
```

Appendix F: Publications

1. Antoniou, M.; Sizov, V.; Cheng Hu; Jancovic, P.; Abdullah, R.; Rashid, N.E.A.; Cherniakov, M.,' *The concept of a forward scattering micro-sensors radar network for situational awareness*, 2008 international conference on radar, Adelaide, 2-5 Sept. 2008, pp. 171 – 176
2. Rashid, N.E.A. Antoniou, M. Jancovic, P. Sizov, V. Abdullah, R. Cherniakov, M.,' *Automatic target classification in a low frequency FSR network*, Proceedings of the 5th European Radar Conference, 30-31 Oct. 2008, pp. 68 - 71
 - The ATC performance using low carrier frequency is discussed in page 80-82
3. Mikhail Cherniakov, Vladimir Sizov, Michael Antoniou, Emileen Rashid, Peter Jancovic, Alexander Myakinkov and Andrei Kuzin,' *Easily deployable, multi-functional radar network*' Codex, Issue 2 Autumn 2008
4. Rashid, N.E.A. Antoniou, M. Jancovic, P. Sizov, V. and Cherniakov, M.,' *Effects of Speed Estimation Accuracy on ATC in FSR*,' Hamburgh, 2009
 - The speed estimation algorithm is discussed in page 125-131
 - The theoretical analysis and result of speed estimation error is discussed in page 138-141

5. V. Sizov, M. Gashinova , N. E. Rashid , J. Chen and M. Cherniakov,' ***Forward Scattering Micro Radar efficiency analysis for different landscapes,***' 6th EMRS DTC Technical Conference – Edinburgh 2009

6. Rashid, N.E.A. Jančovič, P. Gashinova, M. Cherniakov, M. Sizov, V.,' ***The effect of clutter on the automatic target classification accuracy in FSR,***' 2010 IEEE Radar Conference, Washington DC, 10-14 May 2010
 - The types of errors that might occurs in ATC when clutter is present is discussed in page175-176
 - The calculation of SCR is discussed in page 142
 - The clutter model compensation used to compensate the effect of clutter is discussed in page183-186
 - Overall experimental result on the effect of clutter on automatic target classification is discussed in Chapter 6

7. V. Sizov, M. Gashinova , N.E.A. Rashid, N.A. Zakaria, P. Jancovic and M. Cherniakov, '***FSR Sensors Network: Performance and Parameters,***' 7th EMRS DTC Technical Conference – Edinburgh 2010