Microwave Imaging using M-Sequences with a Particular Focus on Breast Cancer Detection

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by Barry Paul Cheeseman.

August 2008

Abstract

Detecting a target of interest such as a breast tumour is the primary focus of microwave imaging but once detected an additional useful property to have is some measure of the target shape. A combination of a pseudo-random noise (PN) sequence and filter has been employed to pick out the edges of a target that when used to create an image shows the target's shape clearly. A variety of target shapes are investigated with each image giving a good detection and showing the target's shape. Imaging through an outer shell to an inner target and showing both the shell and target on the image is successful with the PN sequence and filter but fails when using a wideband pulse and integrating along its length, picking up only the inner target.

A H-matrix transform previously used for communication channel capacity calculations has been investigated for use as an imaging tool. Transforming the H-matrix produces an image that shows the coupling between angles of an antenna array. The image is an angular map showing the direction of any targets present. Further expansion produced an altered version that allowed focussing down to the antenna element limit instead of the antenna array limit. In the process of converting from an angular to a Cartesian coordinate system segmenting of the antenna array data into smaller groups before the transform and then combining afterwards narrows the angular location of the target and gives a limited range measure.

Acknowledgements

I'd like to thank my supervisor Dr Y. Huang for the initial idea for the research and his help and support along the way, other members of the research group for an outlet to bounce ideas off and my family.

Table of Contents

Chapter 1

Introduction, Objectives and Summary

1.1 Introduction

The definition of an image is "a visible impression obtained by a camera, displayed on a video screen, or produced by reflection or refraction" [1]. Imaging is the process of making an image and microwave imaging uses the microwave band of the electromagnetic (EM) spectrum (wavelengths of 1-300mm) instead of the visible spectrum (wavelength of 400-700nm) to create images. So the essence of this thesis is to create images using microwave signals. Microwaves have an advantage over visible wavelengths in that they can penetrate through a range of materials including human tissue making them useful for breast cancer detection as they can penetrate and see inside the breast.

The thesis covers the stages to generate images from microwave signals using computer models and simulations for the purpose of detecting breast cancer. The microwave signals used consist of a range of pulses and pseudo-random noise (PN) sequences at frequencies up to 10GHz. These are generated using the software package Matlab(tm) from Mathworks. Computer models are built in Matlab to represent real situations which are then simulated using the Finite Difference Time Domain (FDTD) method. FDTD is a method that simulates the propagation of electromagnetic waves, including reflection and refraction. Using the computer models and FDTD the propagation of microwave signals transmitted from an antenna

through an environment (such as the inside of the breast) can be simulated and the signals at receiving antennas calculated.

The received signals are calibrated and processed to detect reflected signals, calibration and processing are considered later in the chapter. Images are created from the calibrated and processed data by migration techniques. The migration used is geometric and involves shifting and summing individual data points from each of the received signals based on signal propagation speed and antenna locations. Two related fields have been drawn on for the signal processing method. Firstly, analysis of the indoor radio propagation channel [2-3] indicates that in many cases they can be modelled by a Finite Impulse Response (FIR) filter approximation containing complex attenuation factors and delays. There is also the use of equalisation methods and encoded signals in antenna measurements [4-5] to eliminate echoes from close objects to simulate open space. Those ideas were adapted into a signal processing method that detects rather than suppresses echoes leading to signal processing that takes a different form than normal. It focusses on preserving a targets shape through the use of PN-sequences and a filter. This method detects the edges of targets so when used to create an image through migration shows a targets shape better than conventional methods.

An alternative method to create images is also shown using the H-matrix. The H-matrix gives the coupling between antenna elements in an array at a single frequency. A full investigation into a transform of the H-matrix is carried out that produces images showing the angular direction of any targets present. This is an extension of work designed to calculate the capacity of communication channels but focussed on the images produced by an intermediate step [6].

1.1.1 Calibration

Calibration for microwave imaging [7-8] is the process by which any unwanted elements are removed from the system being looked at whether it is interference from the surrounding structure holding the antenna or skin reflections. This leads to the best possible signal data. In transmit and receive systems (with reference to radar) the various antenna combinations can be summed up into three different groups. Mono-static systems use a single antenna used for both transmission and reception. The main problem is missing some of the early part of the received signal because the antenna is still transmitting when part of the signal to be received arrives, but it is easy to set up and calibrate. Bi-static systems use a single transmitter and a single receiver and it is also relatively easy to calibrate. A multi-static system is the approach used, they have at least three antenna elements in any combination of transmitter and receiver but are more difficult to calibrate.

With reference to breast cancer detection and a multi-static system there are several methods commonly used for calibration [9]. The first method is imported from ground penetrating radar (GPR) and involves windowing the signal to remove unwanted parts. For this technique to work the pulse length and transmitter and receiver locations need to be carefully chosen because wanted signals can be windowed out. A second method is the use of a breast phantom (or no target) to generate calibration signals. The calibration signals are then subtracted from the real signals to remove skin reflections and direct signals between antennas. In theory this is good but in practice is highly patient dependent due to wide variation in tissue properties [10].

The method used here is to obtain calibration for a limited set of data signals operating in a multi-static way. The method involves setting up of a grid of transmitting and receiving antennas but processing the data in pairs to remove skin reflections or direct signals. Calibration using a phantom is acceptable for controlled simulations but the inherent variability of patients makes it unlikely that a universally effective phantom could be developed. In addition due to the very small signals that are generated as reflections from a target it is vitally important to remove any large signals that could interfere. This method provides a way to remove some of the strongest signals that cause problems and will also allow the use of much wider propagation angle antennas.

1.1.2 Signal Detection

Wideband pulses used in ranging applications through to breast cancer detection have a common problem in that the pulse itself is not necessarily the same shape when it is received and can have widely varying strengths if multiple targets are present. Travelling through a medium or when the signal is reflected from objects means that the received signal is difficult to match up to what was transmitted. Common techniques to detect when the pulse is received include integration over the pulse length to detect the pulse as a region of higher than normal activity or correlation methods [5] (including both frequency and phase modulated pulses). One of the problems is signal masking where a stronger received pulse makes a weaker pulse difficult to detect.

Wideband pulses when applied to imaging applications such as ground penetrating radar (GPR) or breast cancer detection give good detection of a target of interest but the target's shape is generally not fully distinguishable. Shape information

is a useful property to have for classification purposes. Use of uniquely identifiable signals such as maximal length pseudo random noise sequences taken from channel analysis methods should help to narrow down a signal detection to a smaller time window representing a targets edge. If this can be retained through the entire process then target surfaces can be picked out and shapes can be distinguished.

1.2 Literature Review

Breast cancer is a serious disease that affects many women each year. Approximately 1 in 9 women will be diagnosed with breast cancer over their lifetime and the incidence of breast cancer in the UK in 2001 ran to 15% of cancers detected (excluding non-malignant melanoma). Looking only at women breast cancer accounted for 30% [10]. Early detection is a key part of survivability so different approaches have been tried to detect breast cancer each having their own particular advantages and disadvantages.

1.2.1 Current Methods

The current benchmark in breast cancer detection is x-ray mammography [11] . X-rays readily penetrate human tissue so can be used for imaging systems. X-ray imaging detects the contrast in densities between different tissues, denser tissues attenuate the x-rays more so appear as under exposed areas on the film. A breast tumour presents a contrast to surrounding tissues using x-ray mammography so can be detected. Mammography does have its problems however. Women with dense breasts are difficult to x-ray [12] due to a smaller contrast in densities between tumour and normal tissue and the higher level of attenuation through the tissue. Mammography also requires compression of the breast for best results, this is

uncomfortable and sometimes painful [13-14]. X-ray radiation is ionizing so may cause damage to human cells; there are limits on the amount of radiation allowed so limiting the number of x-rays over a given period of time. Microwave radiation is non-ionising causing only thermal effects [15]. Digital X-ray technology has been introduced to mammography that gives improved detection over previous analogue methods [16]. Still with these improvements other technologies are being investigated with the aim of better detecting breast cancer. Ultrasound imaging is used as an additional tool to differentiate between fluid filled and solid cysts for instance [17]. MRI (Magnetic Resonance Imaging) is a possible screening tool and is available now but is expensive and used mainly for high risk younger women where conventional methods are not as effective [18-19]. Microwave imaging is a technique that shows promise as a possible replacement or complimentary method of detecting breast cancer [20].

1.2.2 Tissue Properties

There have been several attempts to characterise the dielectric properties of human female breast tissue, most looking at excised tissue. Figure 1.1 shows an image of excised breast tissue ready for dielectric property measurement [21].

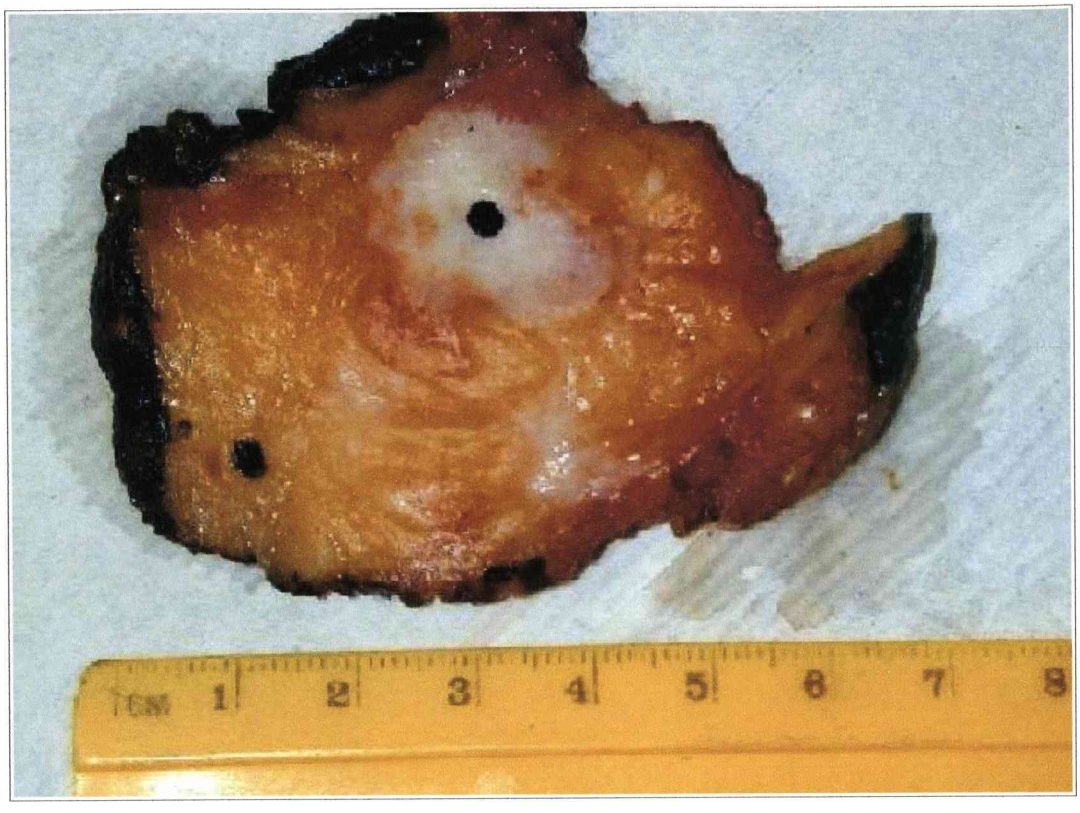

Figure 1.1 - Photograph of a tissue specimen showing malignant tissue (white region, upper right) and normal fat tissue (yellow region, lower left). The black ink spots mark the measurement sites. The black ink around the perimeter of the specimen denotes the margins of the excised tissue [21].

MRI data can be used to produce a map of the locations of the different tissue types in the breast but the dielectric properties cannot be determined and need to be measured separately. The breast is made up of different tissue types and all need to be taken into account. The main tissue types of interest are fat, normal tissue, benign tumour and malignant tumour. The contrast between malignant tumour and fat and normal tissue is required. A data set gathered at 3.2 GHz [22] found that over the samples of fat the dielectric constant varied from 2.8 to 7.6 and conductivity from 0.54 to 2.9mS cm^{-1}. A correlation between increasing dielectric constant and conductivity was found. A similar occurrence was found with normal tissue but with the dielectric constant varying from 9.8 to 46 and conductivity from 3.7 to 34mS cm^{-1}. Water content for fat was found to be around 20% by weight while normal tissue around 50%. Similar

correlations were found in benign and malignant tumours. Benign tumours showed a dielectric constant range from 15 to 67 and conductivity from 7 to 49mS cm^{-1} and malignant tumours showed a dielectric constant range from 9 to 59 and conductivity from 2 to 34mS cm^{-1}. There are a wide range of dielectric property measurements that overlap including normal tissue properties overlapping malignant tumour properties and sometimes malignant tissue has shown distinct regions of differing properties in a single patient [22]. As the frequency changes, the properties of the tissue change also, this data is from a single frequency but other data sets exist for a frequency range up to 3GHz. As tissue properties change with frequency and better resolution can be obtained with higher frequencies these data sets have been used to extrapolate values for increased frequencies and are being verified experimentally [23]. The values from [22] are the basis for any simulated results but more recent investigations into this area have new findings and over a greatly increased and much more relevant frequency range. The findings are outlined in [21] and [23] and would be the basis on which future work in this area should be based in terms of dielectric properties but have come too late for this investigation. The main findings are summarised as the very large spread of values for normal breast tissues compared to a relatively narrow spread for malignant cancers leading to much smaller differences between them in some cases and hence a much harder to detect target in microwave imaging terms.

1.2.3 Localisation Techniques

The two main forms of microwave imaging for breast cancer detection are to identify the dielectric properties of the breast and hence show a tumour as a region of higher dielectric properties or to directly detect and localise scattered waves. The first technique requires solving a complex inverse scattering problem while the second is

much simpler. A transmitter transmits an ultra-wideband pulse into the breast and the reflections from any discontinuities in the breast are detected using one or more antennas. An example antenna array from [24] is shown in Figure 1.2.

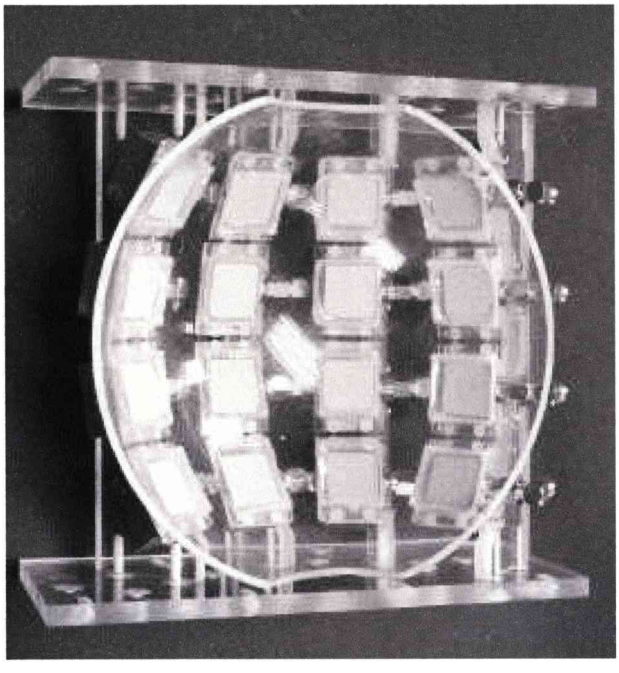

Figure 1.2 - Photograph of the symmetrical curved antenna array with 16 elements for breast cancer detection. [24]

Signal processing algorithms then localise the tumour. Techniques using shift and sum algorithms are the most common [7-8, 25-34] and are similar to geometric migration techniques used in ground penetrating radar. Versions using more complex migration techniques such as Kirchoff and wavefront migration are a possible area to study. A promising fairly recent technique is contrast source inversion (CSI) [35-36] for detecting objects where the contrast between the target and medium is large. As it is an inversion technique it gives the dielectric properties of the target directly and has led to some good theoretical and experimental results [35-38]. There are key points that all techniques should cope well with. The first is detecting more than one target, this is tricky as one of a multi-target group will usually be much stronger and mask

the others. The second is coping with the effects of varying tissue permittivity as real breast and tumour tissue is not homogeneous. Investigations into these two issues [33-34] show that they do need to be accounted for in any algorithm.

1.2.4 Signal Processing

The algorithm used to extract a specific reflected signal from the background of other reflected signals is difficult. There are several different methods that aim to extract or otherwise enhance the reflected signal that is required while degrading the background signals. GPR methods are well established and have been used as a starting point for signal processing. In GPR systems a large proportion of signal processing is applied to the received waveform, the A-scan. Applied to the A-scan data is a variety of signal processing methods designed to improve the data [25]. Noise presents a problem in all GPR systems so the first objective of a signal processing algorithm is to suppress noise. A noise reduction filter such as local averaging over the data or multiple sets of data is a common technique.

Clutter is a term referring to other reflections that are not wanted when looking for a specific targets reflection. A common technique to reduce this is to subtract the average of data sets in the area of interest. Attenuation and spreading loss is a problem with all antenna based systems. Compensation for this natural process can be performed using a time varying gain method where the further the signal has travelled the greater the gain that is applied to that data point. It does however cause signal distortion if it is not applied correctly. Low and high pass filtering is a useful procedure when dealing with real antenna systems to remove interactions between the antenna and ground and high frequency interference [25]. All of these methods could be adapted to a microwave based system to detect breast cancer.

There are of course differences between GPR and microwave imaging that have led to different algorithms being implemented; filtering, calibration and the use of specific waveforms are common differences. A radar technique based on mono-static operation has been tested [26]. It uses an antenna array where each antenna element is placed on the breast surface. Each antenna element is in turn excited by a short pulse. The same antenna element receives the reflected signal which contains the wanted target response, transmitted signal and skin reflections. As the skin reflections and transmitted signals are not wanted, a calibration procedure is carried out to remove them. The calibration step involves subtracting the average of the waveforms from all the antenna elements from each individual waveform and then low pass filtering the resultant signal using a moving average. Due to the positioning of the antenna elements (touching the skin) the average signal waveform is comprised mostly of the transmitted signal and skin reflected signals so subtracting removes them from the processed signal. The localization technique is an adaptation from synthetic aperture radar and consists of time delaying and summing points from the waveforms so that a synthetic focus is achieved at an arbitrary point. Performing this for all points in a search domain leads to the appearance of the scattering object or objects. The waveform itself is chosen so that its centre point is zero, integrating the waveform then transforms that point to a maximum which helps with signal registration. In addition compensation for radial spreading of the signal is applied and attenuation compensation can be used [7, 27].

Multi-static systems using techniques requiring subtraction-less calibration show good results [31-32]. However without calibration there is additional clutter present. Feature extraction techniques are another interesting area with two potential

uses. Firstly they can be used to extract relevant information out of the data itself such as the hyperbola observed in GPR data [39]. Extracting relevant features after a migration technique would also be useful as it allows quicker identification of relevant targets for further study.

1.2.5 Computer Models and Breast Phantoms

There are several different variations used in the construction of a computer model to generate data to test an algorithm. The simplest of which is a static model with fixed dielectric properties for all key components. It is the quickest to set up and the data it produces is the easiest to obtain acceptable results with. Upgrading this model to have frequency dependent dielectric properties and variation of ±10% around a set value is the next step [7, 27]. Taking this model and individually modelling different component parts of the breast like separate fatty and normal tissues and including glandular tissues and the chest wall leads to a very complex model [29] that is quite realistic.

However there is a current trend to use MRI derived models from a particular human [26,28,40]. The MRI data is used to map the distribution of tissue types in the breast before each tissue type is assigned it's own dielectric properties based on dielectric property measurement results [10,21,23]. The data is then inputted to a modelling and simulation algorithm such as Finite Difference Time Domain (FDTD) and simulated as per any other model. The result is a model that should be as close to an anatomically correct model as you can get but there is still the limitations of the modelling and simulation algorithm itself. Moving to a physical breast phantom is the next logical step but nothing quite matches up to real breast tissue, so simpler designs that match some properties at a limited frequency range are used. In its simplest form

only the contrast between tumour, tissue, and skin is implemented [29]. More complex forms try to approximately match the real dielectric properties with a corresponding material such as soy bean oil used to approximate fatty tissue [41].

1.3 Objectives

The objectives of the research are to develop a system to detect breast cancer using microwave imaging and provide improvements to imaging in general. Breast cancer detection requires an antenna, a test subject and a range of signal processing algorithms. The test subject consists of a computer model with fixed dielectric properties. The details of each model are given within the results section for the test in question. Sets of data have been generated using the computer model to help with the algorithm design and to test it. The antenna in the FDTD software is a hard source, Microwave Studio uses a dipole antenna and both use electric field probes as receivers.

The algorithm has to enhance the reflected signal from a tumour while suppressing other reflections. This is the signal processing part of the algorithm. It also needs to locate in three dimensions any tumours present which is the localisation part of the algorithm. The two will be separate with the ability to alter and test different signal processing and localisation parts independently but as a whole the algorithm is the main focal point. Data for testing and refining the algorithm will be obtained using computer models to represent the breast. In short the main focus is to develop an algorithm that processes a data set and localises a target, that of a breast tumour. The main complimentary part to that is a simulation tool based on FDTD in which models are simulated to generate data that is used to test the algorithm.

1.4 Summary

1.4.1 Chapter 2

This chapter discusses the development of FDTD code and makes comparisons between the FDTD code and a commercial package to ensure comparable results. An unresolved problem that was encountered with the 3D version of the code is also looked at in the time and frequency domains with commercial package comparisons that show distortion of the signals that is not present when using 2D FDTD simulations.

1.4.2 Chapter 3

This chapter covers the theory behind M-sequences including numerical and analytical results of their use and how to generate them. A look at missing symbols in a sequence shows the robustness of the sequence and the minimal amount of problems that this causes for sufficiently long sequences. Noise and simultaneous transmissions are separate sections which both produce similar effects that show an increase in background noise after processing but the target signal remains detectable. It continues into the development of a filter to match a received signal to a transmitted signal with comparisons to correlation that favour the filter. As part of the filter development the filter parameters are also investigated to determine suitable values. Finally range resolution is looked at and shows that the filter has reduced sidelobes compared to the other methods looked at.

1.4.3 Chapter 4

This covers a direct transform of a H-matrix for imaging purposes. The original method is described with example images showing the limitations of the method with regard to close targets. An altered method is presented to allow imaging of close targets that brings the minimum distance down to the antenna element limit instead of the array limit.

1.4.4 Chapter 5

This chapter looks at conventional methods of imaging starting with a phantom free calibration technique based on averaging that is used in subsequent image generation. Comparisons are made between pulse integration, correlation and filter methods as described in Chapter 3 but now using FDTD simulated results. These results use a flat array and a full circular array and show that pulse integration is good for overall detection but additional shape information is present in the other two methods. Of these the filter shows the better detection that is good enough to distinguish a variety of shapes. An investigation into averaging to trade off some fineness of detection for detection strength shows an increase in the filters detection performance at a small fineness of detection cost. Finally a small section looks at using the filter to identify relative permittivity of a homogeneous block of material in a known medium. The theoretical location of the blocks rear reflection is shown to match up to the simulated results location over a variety of relative permittivities down to a relative permittivity difference of 1.

1.4.5 Chapter 6

The last chapter gives overall conclusions and a number of suggestions for further work from general FDTD usability improvements through signal symbol investigations and improved calibration.

1.5 Publications

1. B. Cheeseman and Y. Huang, "Limited Multi-Static Calibration Technique without a Phantom for the Detection of Breast Cancer", *Proceedings of the 3rd European Radar Conference*, pp. 225-228, September 2006
2. B. Cheeseman and Y. Huang, "Delay Time Detection for Ranging Applications using a Tap Delay Line Equalizer", *Proceedings of The European Conference on Antennas and Propagation: EuCAP 2006*, Nice, UK, October 2006
3. B. Cheeseman and Y. Huang, 'Comparison of Shape Imaging Between Pulse Integration, Correlation and Adaptive Filter', *Proceedings of The European Conference on Antennas and Propagation: EuCAP 2007*, Edinburgh, UK, November 2007

Chapter 2

Finite Difference Time Domain Method

2.1 Introduction

Data for testing is an important aspect to consider and due to the versatility offered using simulated data rather than real data was the method chosen. Finite Difference Time Domain (FDTD) [42] is a well tested and widely used method of electromagnetic simulation. A FDTD code was written in Mathworks Matlab(tm) software that allowed direct input of excitation signals (such as Pseudo random Noise (PN) sequences) and also output of signals into Matlab's workspace for further processing. This offered more efficient work flow from signal generation through simulation to data processing and plotting using functions and scripts in a single package. To ensure correct operation a commercial electromagnetic simulation package was used to validate the results, CST Microwave Studio [43].

2.2 FDTD Algorithm

FDTD is a method used to simulate the propagation of electromagnetic waves. The method used is based on the three-dimensional Yee algorithm [42]. The boundary condition implemented is the Mur absorbing boundary condition [44] which is computationally efficient and would be sufficient for testing. It also uses the Courant [42] stability condition to determine the maximum time step allowed (which in this case for a given maximum frequency value would be fixed). This ensures that no interpolation is required for using data across multiple separate simulations with the same maximum frequency. Also where relevant the code would also allow the

implementation of randomised deviation of dielectric properties throughout a medium. Validation of the code is through comparison with CST Microwave studio.

The FDTD code is designed to run well on Matlab so it is coded to take advantage of Matlab's strengths. The standard code is run in loops with each loop performing a single time iteration. In each loop the electric and magnetic fields are calculated for that time iteration based on previous iterations. Normally the electric and magnetic field calculations are performed in a loop also but for Matlab loops tend to be slow unless written in a particular way and containing only functions that can be accelerated. A matrix calculation form for the electric and magnetic fields was implemented that is much quicker than running loops but has a higher memory requirement. Furthermore due to some behaviour and results considered unusual, 2D simulations were primarily used with the problems with 3D simulations considered later in the chapter.

2.2.1 FDTD Validation Results

The FDTD code and CST Microwave Studio were set up similarly and simulated. FDTD is 2D and used a hard source (single cell) for the excitation. A hard source is where at each time step the electric field value corresponding to the excitation signal is assigned directly to a cell. CST Microwave Studio is 3D and requires an antenna for simulation so a dipole antenna was used. A differentiated Gaussian pulse was used as an excitation signal. The antenna present in CST Microwave Studio changes the excitation when transmitted from the antenna so it was exported and used in the FDTD code as the hard source excitation. For both simulations an ideal probe at a distance of 0.05m from the source received the signal which was the measured

electric field. The simulation took place in a medium with dielectric properties of ε_r=9 and σ=0. Figure 2.1 shows a comparison between the two simulations, the differences would be due to using a hard source in the FDTD code compared to an antenna in Microwave Studio. There is also a slight delay present when using an antenna as a source that shows up as a constant delay over all distances.

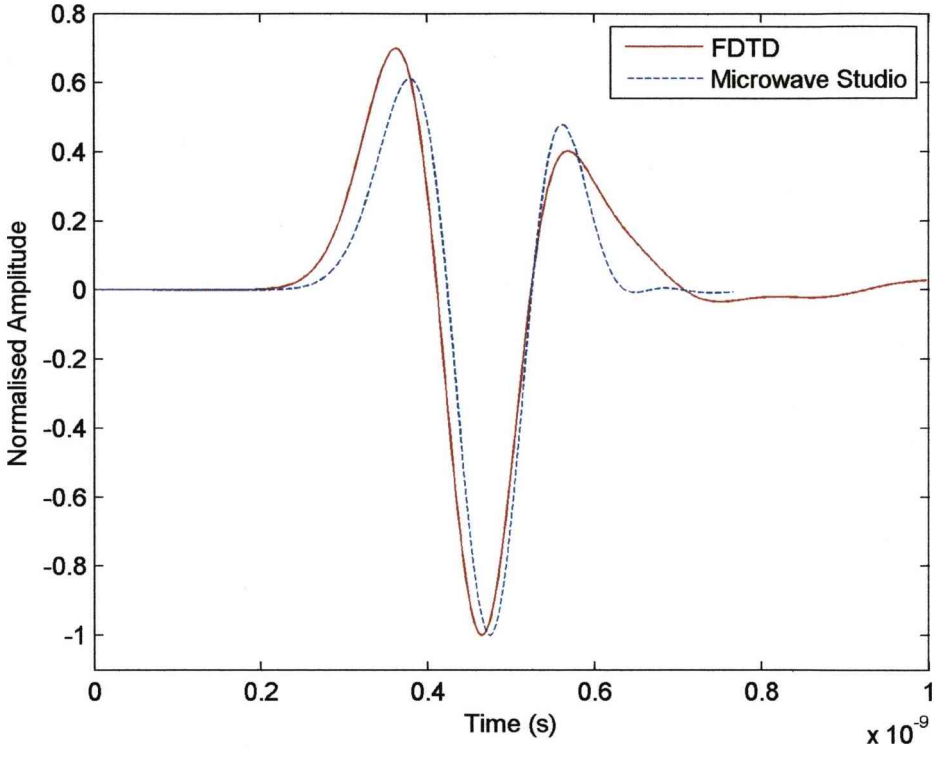

Figure 2.1 - Comparison between CST Microwave Studio and FDTD simulation of excitation signal in lossless material

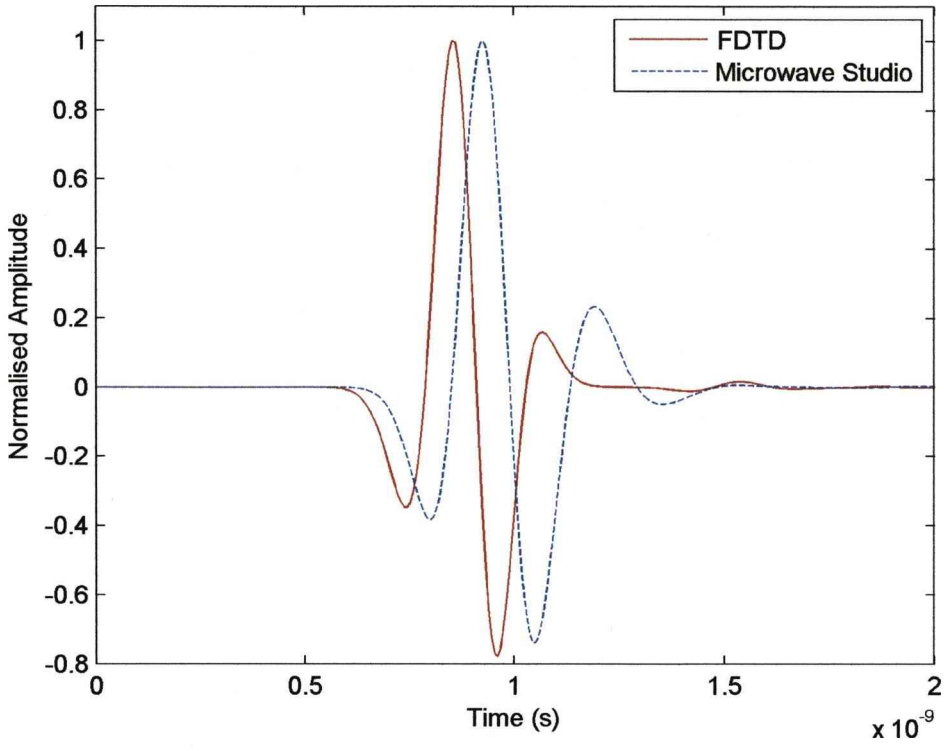

Figure 2.2 - Comparison between CST Microwave Studio and FDTD simulation of excitation signal in lossy material

Using the same excitation signal in both CST Microwave Studio and FDTD code but simulating in a lossy material (background medium of ε_r=9, σ=0.4) gives a close signal match as shown in Figure 2.2. However the signal arrives slightly earlier with the FDTD code than with CST Microwave Studio probably due to a delay in the signal being transmitted by the antenna compared to a hard source. The FDTD code simulates too slowly so eliminates the possibility of using it for an iterative matching algorithm.

2.3 Comparison between 2D and 3D FDTD Simulations

This section compares the outputs from 2D and 3D FDTD simulations. The extra dimension in the 3D simulations requires a change in the step size to remain stable

[42]. The time step is chosen to be at the Courant limit for each simulation given by (2.1)

$$\text{time step (3D)} = \frac{1}{v\left(\sqrt{\left(\frac{1}{\delta_{x^2}}\right)+\left(\frac{1}{\delta_{y^2}}\right)+\left(\frac{1}{\delta_{z^2}}\right)}\right)}$$
$$\text{time step (2D)} = \frac{1}{v\left(\sqrt{\left(\frac{1}{\delta_{x^2}}\right)+\left(\frac{1}{\delta_{y^2}}\right)}\right)} \quad (2.1)$$

In this case the cells are square so $\delta_x=\delta_y=\delta_z=\delta$ and v is the velocity of propagation for the signal given by (2.2)

$$v = \frac{1}{\sqrt{\varepsilon_r \varepsilon_0 \mu_r \mu_0}} \quad (2.2)$$

The corresponding Courant numbers are 0.707 for 2D and 0.577 for 3D. For the simulation set up given in Figure 2.3 the transmitter is a single cell hard source with the E_z field directly excited. The E_z field is read as the output at every cell over a specified distance for every time step. The simulation area is a 0.12m square for the 2D simulation and 0.12m cube for the 3D simulation. The source and output cells for the 3D simulation take place at a z-axis height of 0.06m.

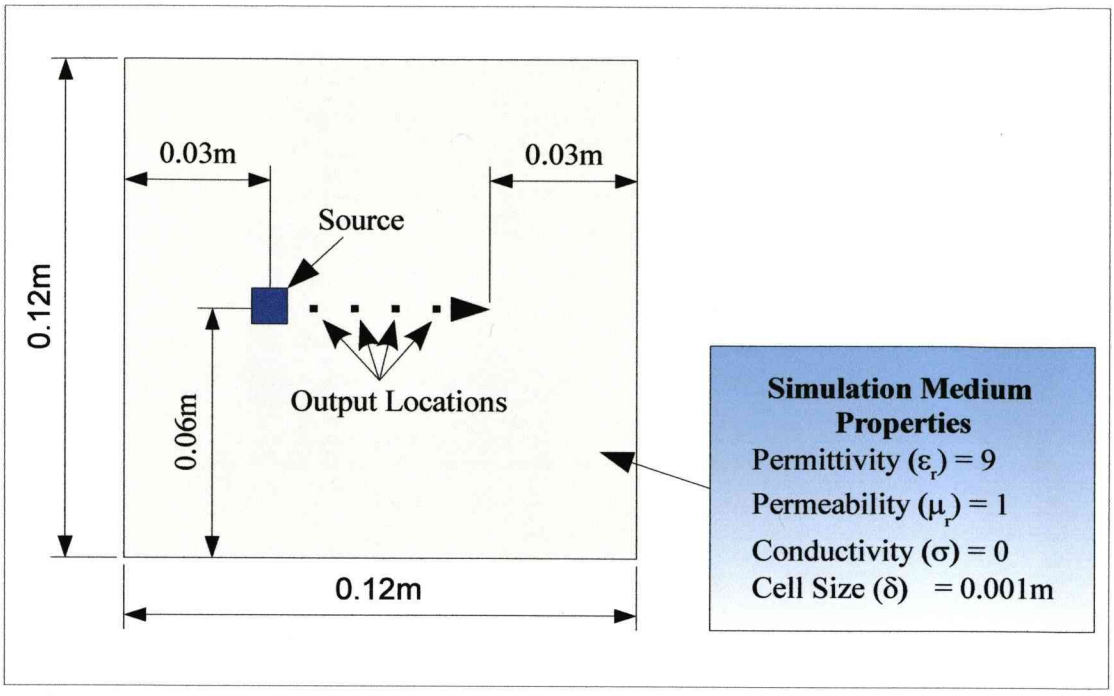

Figure 2.3 - Simulation set up

Two main excitation signals have been tested on both the 2D and 3D simulations, shown in Figure 2.4.

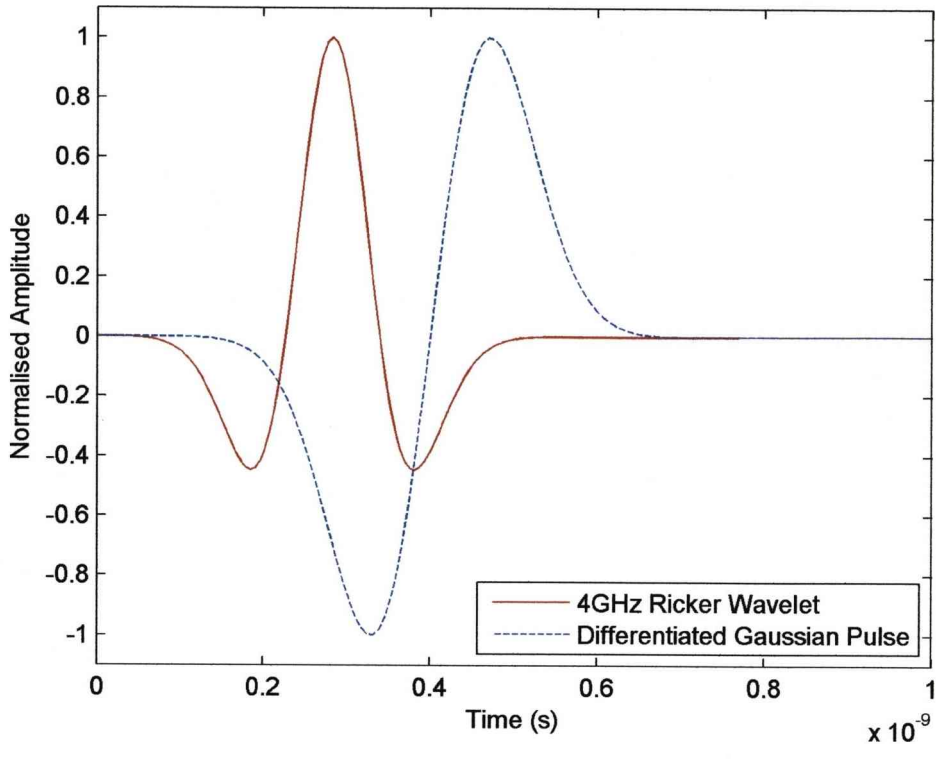

Figure 2.4 - Excitation signals

The differentiated Gaussian pulse and the Ricker wavelet are given by (2.3)

$$\text{Gauss} = \left((t-d)-4\times10^{-10}\right)e^{-\left(\frac{\left((t-d)-4\times10^{-10}\right)^2}{\left(1\times10^{-10}\right)^2}\right)}$$
$$\text{Ricker} = \frac{1-2\left(4\pi\times10^{9}(t-d)\right)^2}{e^{\left(4\pi\times10^{9}(t-d)\right)^2}} \tag{2.3}$$

Where d is the time delay applied to the waveform.

2.3.1 Simulation Results – Differentiated Gaussian Pulse

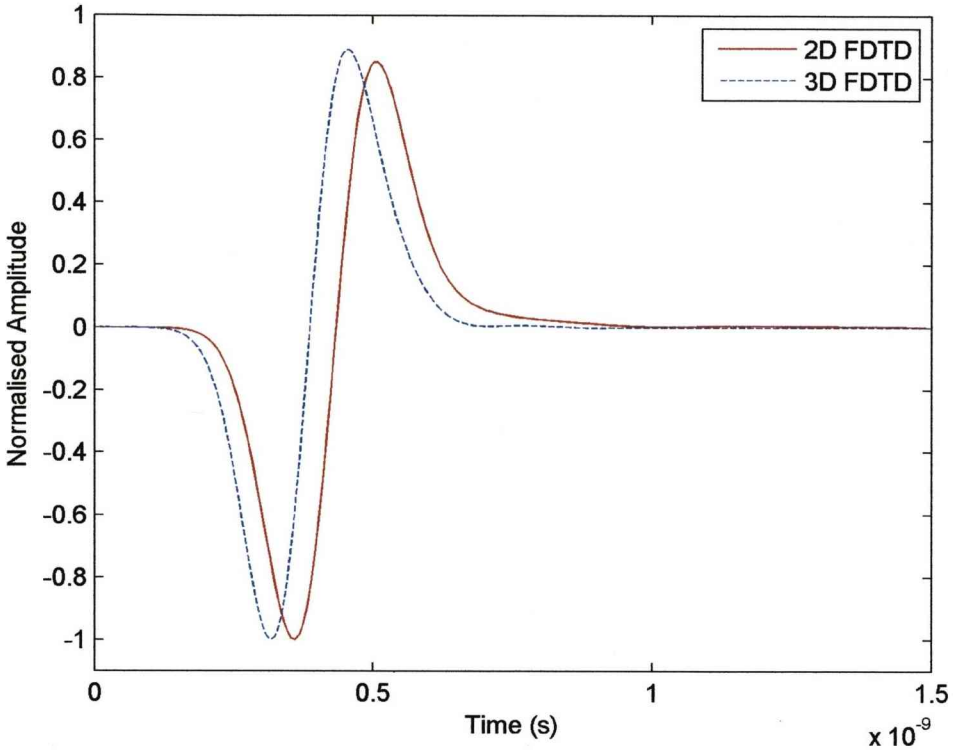

Figure 2.5 - Received differentiated Gaussian pulse at 0.003m distance from source

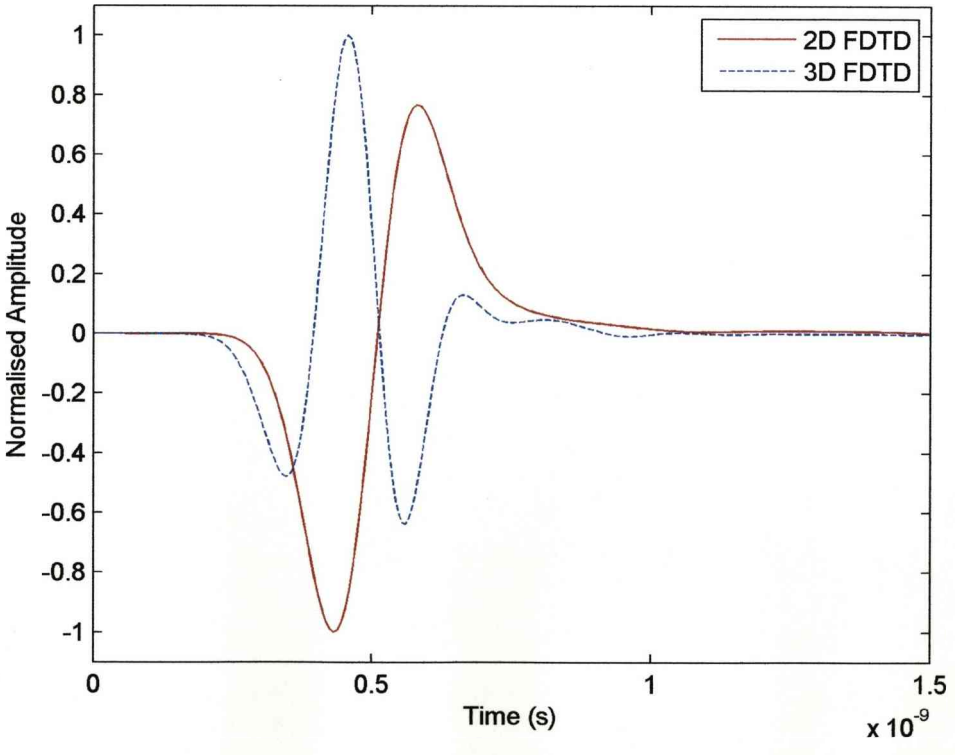

Figure 2.6 - Received differentiated Gaussian pulse at 0.01m distance from source

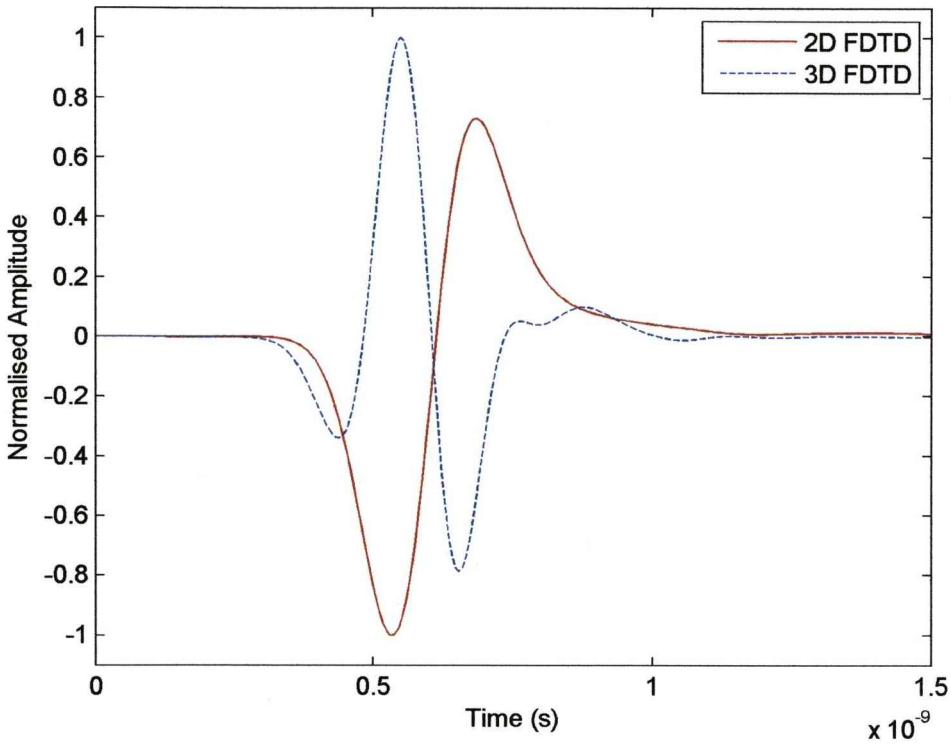

Figure 2.7 - Received differentiated Gaussian pulse 0.02m distance from source

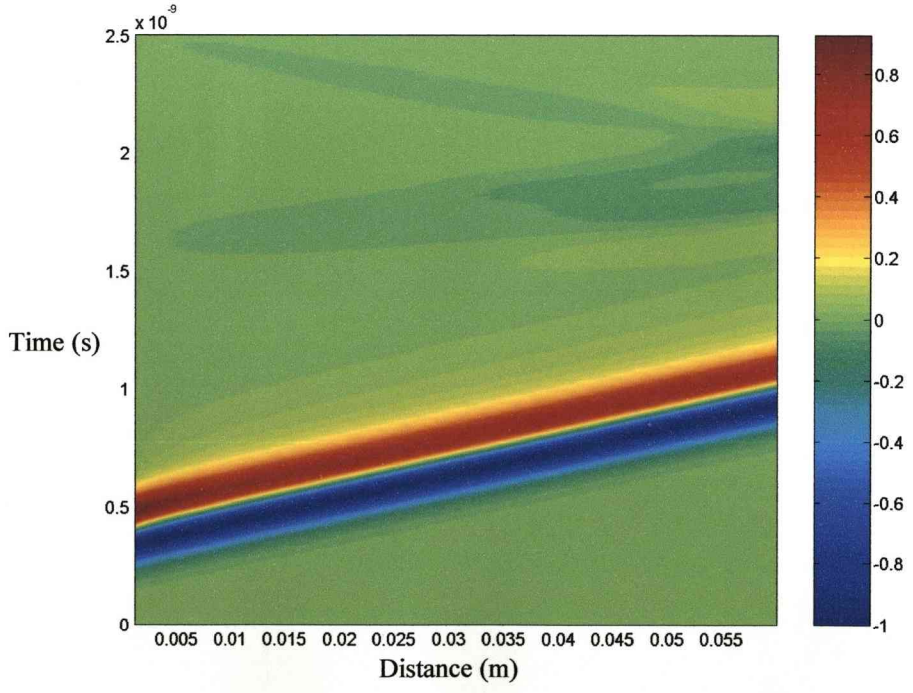

Figure 2.8 - 2D – Received differentiated Gaussian pulse plotted against distance from source

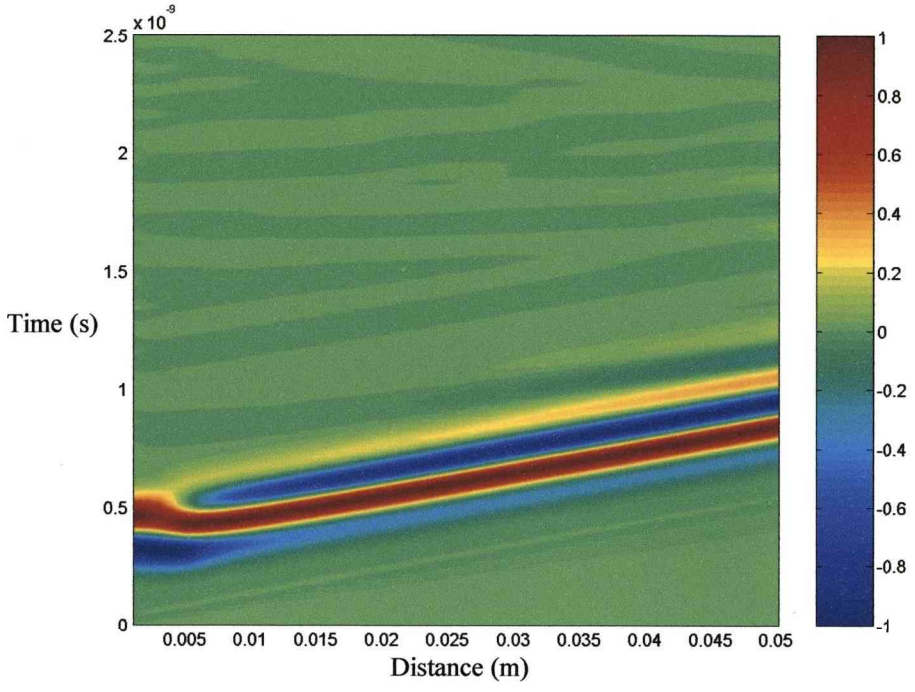

Figure 2.9 - 3D – Received differentiated Gaussian pulse plotted against distance from source

The signals match up quite well at short distances like in Figure 2.5 but quickly become very different as shown in Figures 2.6 and 2.7. Looking at the full distance range shows that the 2D results change very little in Figure 2.8 compared to a large change at a relatively small distance before becoming relatively constant over the rest of the distance for the 3D results in Figure 2.9. Characteristic dispersion patterns are also visible in the 3D results towards the larger distances but do not appear in the 2D results.

The frequency domain shows a correlation between the changes in signal shape to a change in dominant frequency for the 3D results in Figure 2.11. As the 2D signal doesn't change shape the 2D frequency domain remains constant in Figure 2.10.

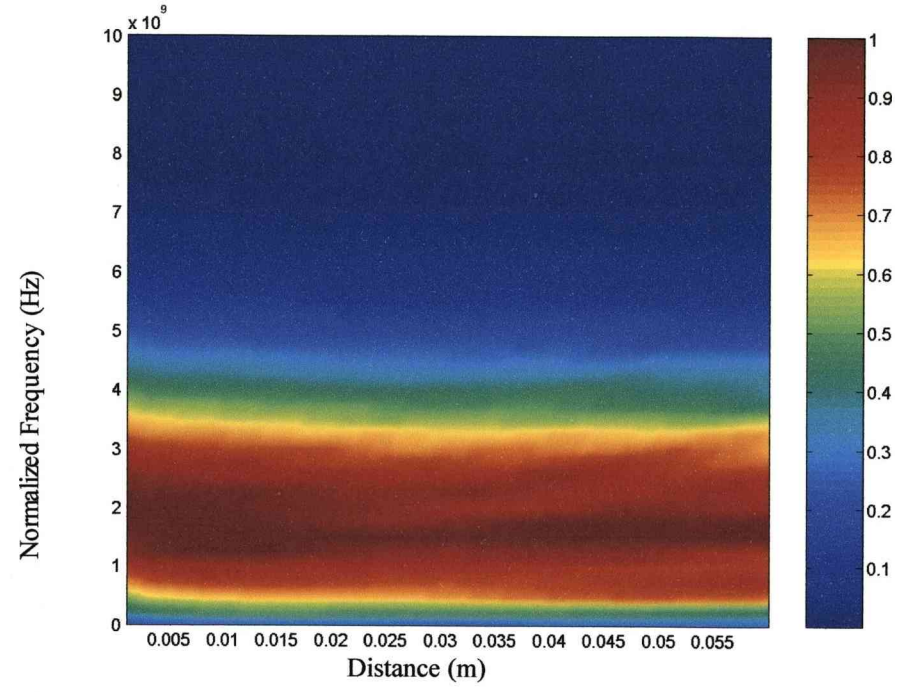

Figure 2.10 - 2D – Normalised frequency of received differentiated Gaussian pulse against receiver distance

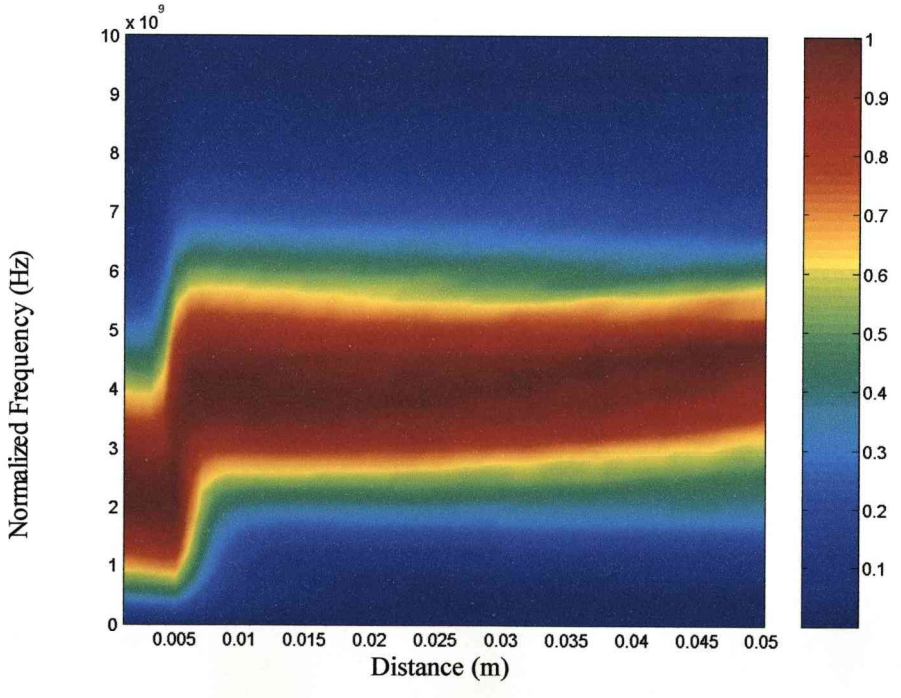

Figure 2.11 - 3D – Normalised frequency of received differentiated Gaussian pulse against receiver distance

2.3.2 Simulation Results – Ricker Wavelet

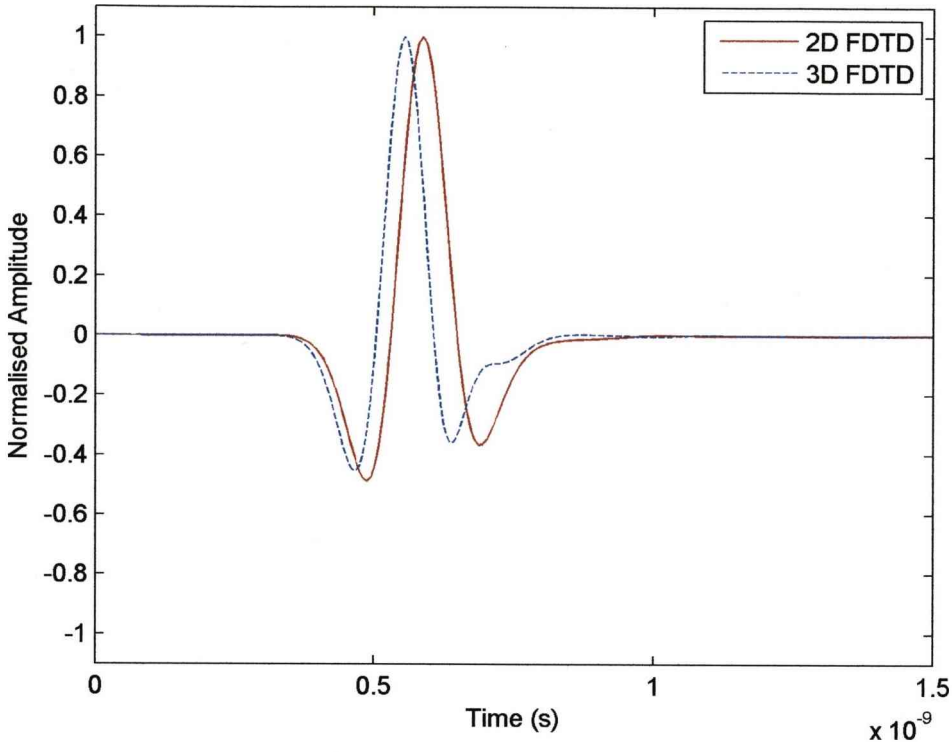

Figure 2.12 - Received 4GHz centre frequency Ricker wavelet at 0.003m distance from source

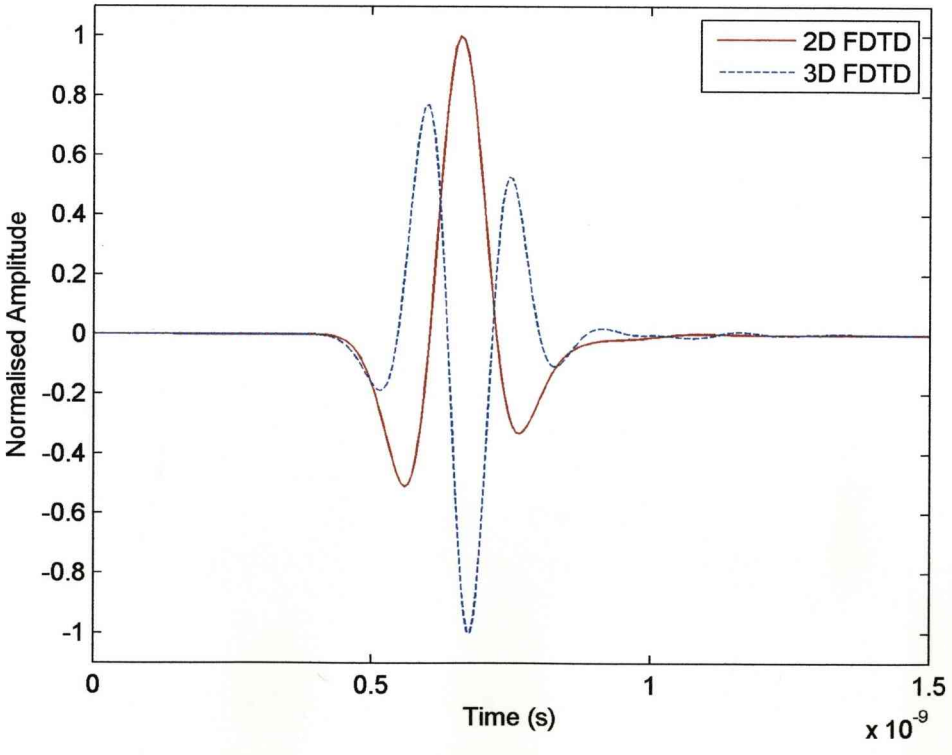

Figure 2.13 - Received 4GHz centre frequency Ricker wavelet at 0.01m distance from source

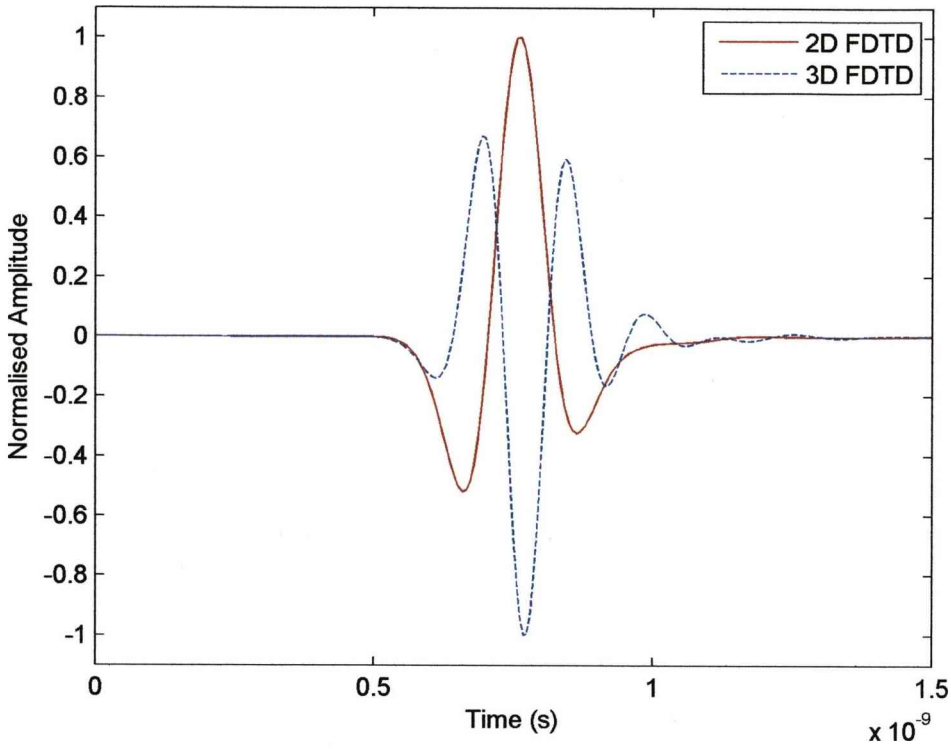

Figure 2.14 - Received 4GHz centre frequency Ricker wavelet at 0.02m distance from source

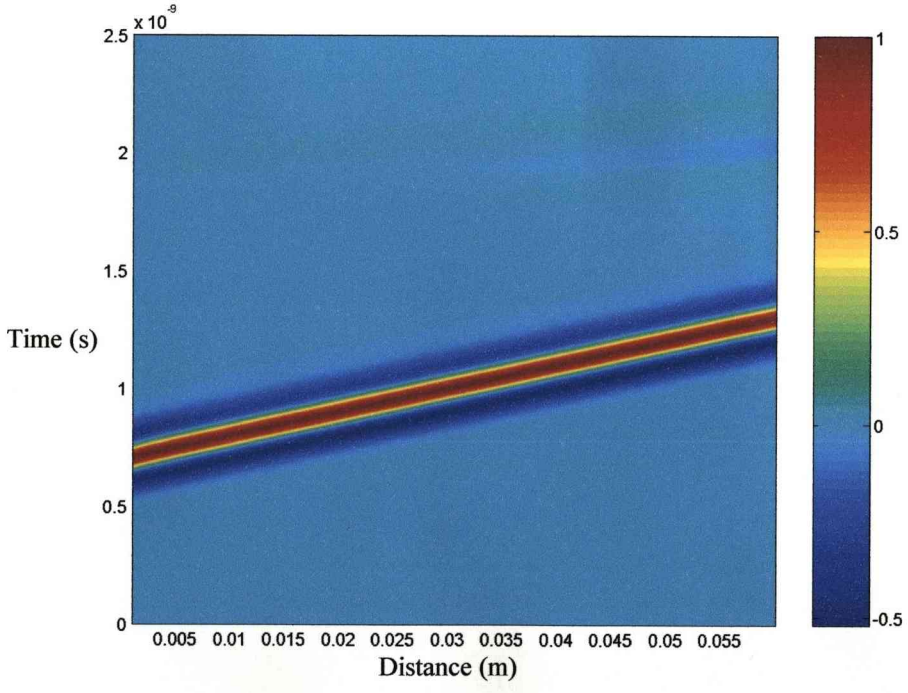

Figure 2.15 - 2D – Received 4GHz centre frequency Ricker wavelet plotted against distance from source

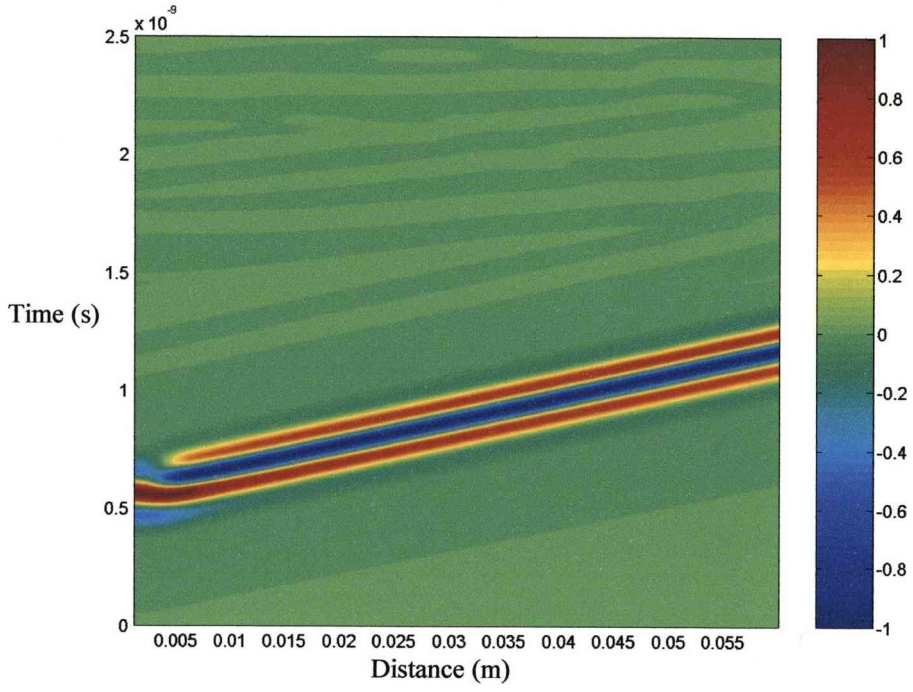

Figure 2.16 - 3D – Received 4GHz centre frequency Ricker wavelet plotted against distance from source

A similar thing occurs with the Ricker wavelet as for the differentiated Gaussian pulse. Short distances match well like in Figure 2.12 but quickly become different as in Figures 2.13-14. The 2D plot shows a near consistent signal over the distance in Figure 2.15 but again the 3D results in Figure 2.16 show a quick change in signal before settling.

Also like before, the signal change in the 3D simulation is mirrored by a change in the dominant frequency as shown in Figure 2.18 while showing no changes in Figure 2.17 for the 2D simulation.

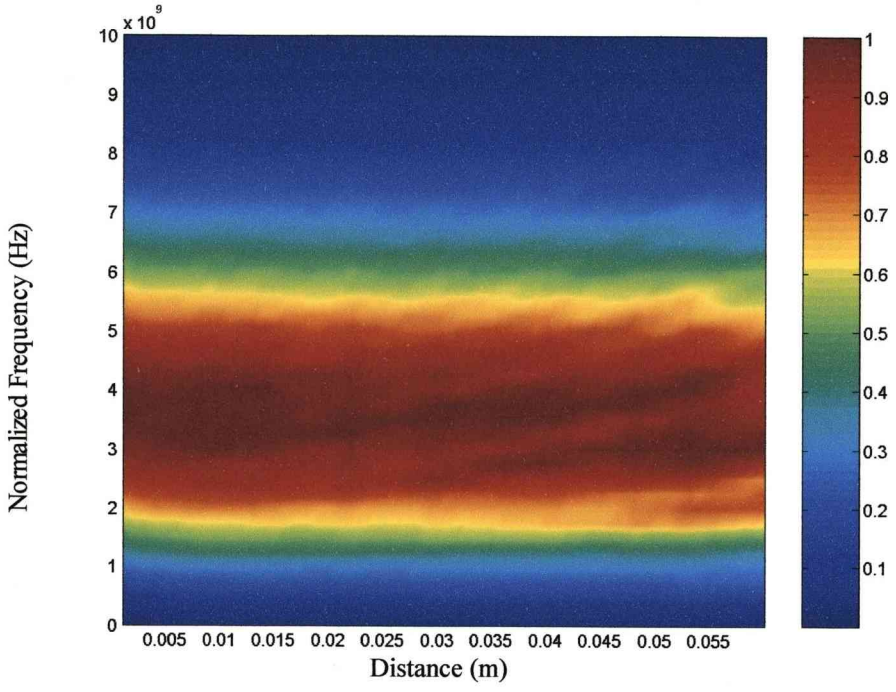

Figure 2.17 - 2D – Normalised frequency of received 4GHz centre frequency Ricker wavelet against receiver distance

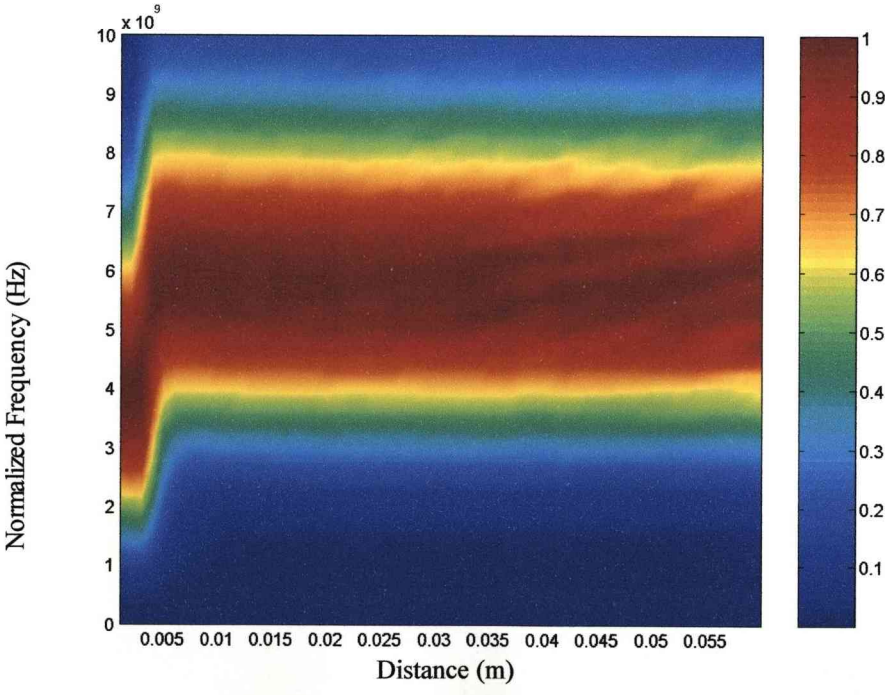

Figure 2.18 - 3D – Normalised frequency of received 4GHz centre frequency Ricker wavelet against receiver distance

Reducing the centre frequency of the Ricker wavelet from 4GHz to 2GHz in Figures 2.19-20 produces the same results but the short distance after which the signal changes shape becomes larger to approximately 0.008m, double that of the higher frequency.

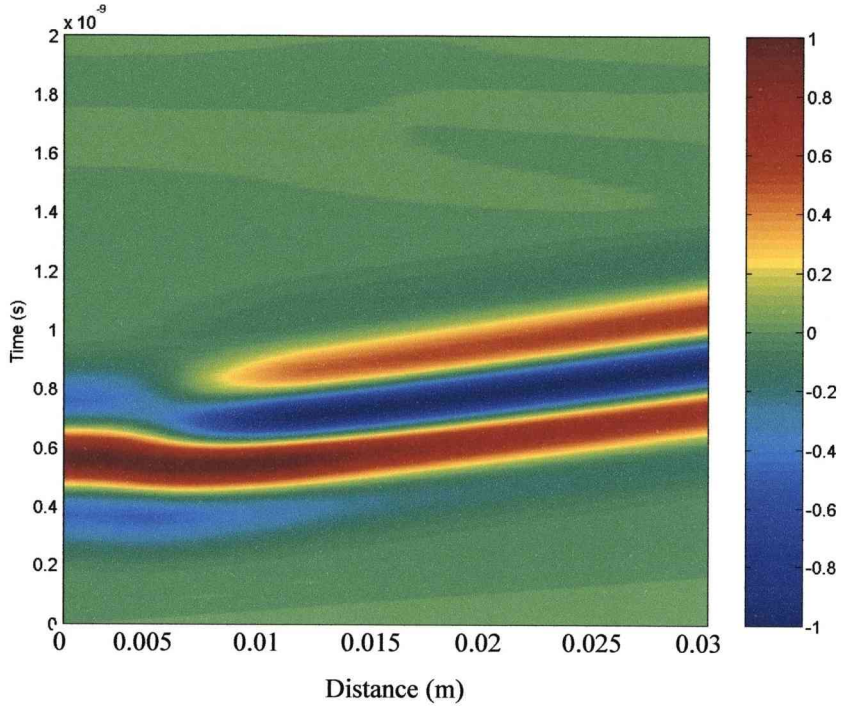

Figure 2.19 - 3D – Received 2GHz centre frequency Ricker wavelet plotted against distance from source

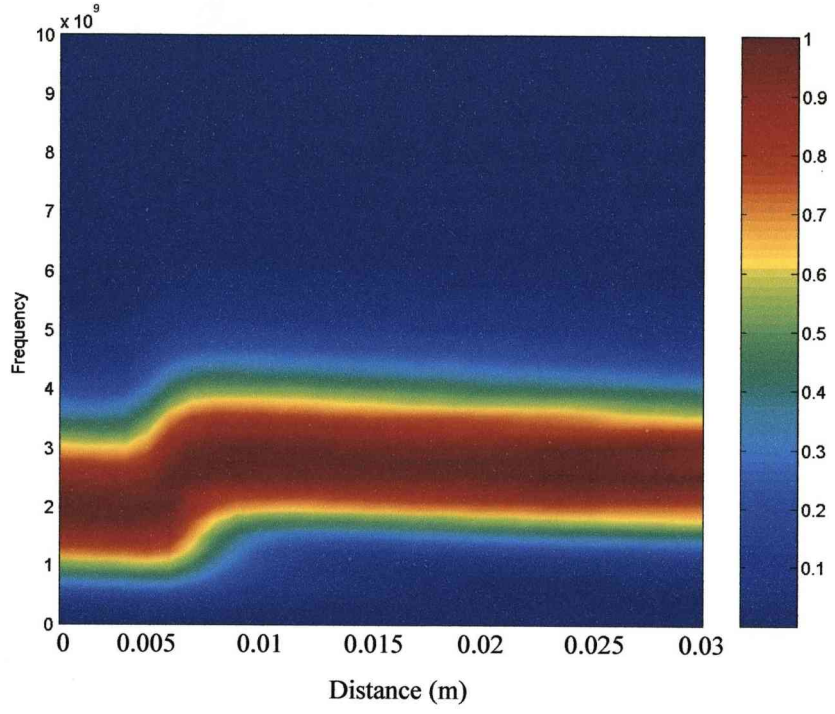

Figure 2.20 - 3D – Normalised frequency of received 2GHz centre frequency Ricker wavelet against receiver distance

2.3.3 CST Simulation Results – Differentiated Gaussian Pulse

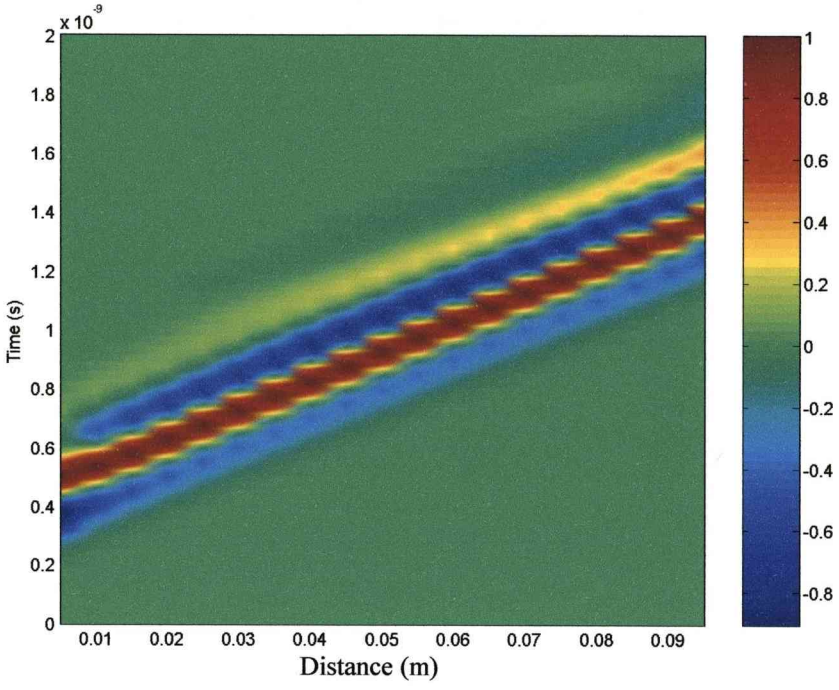

Figure 2.21 - CST – Received differentiated Gaussian pulse plotted against distance from source

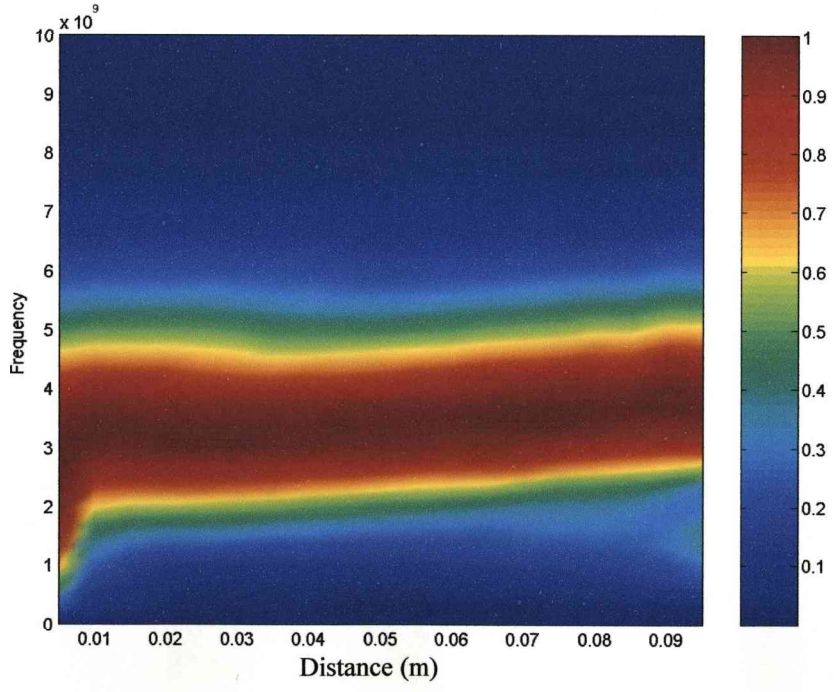

Figure 2.22 - CST – Normalised frequency of received differentiated Gaussian pulse against receiver distance

CST Microwave Studio shows similar results to the 3D FDTD code. It shows a change in the signal shape as before in Figure 2.21 incorporating the dispersion patterns at longer distances. The frequency spectrum in Figure 2.22 also shows the same change at a similar distance from the transmitter (0.005m). In this case the transmitted signal goes through a dipole antenna.

2.3.4 Section Conclusions

Dispersion effects are visible in the 3D simulations using the FDTD code and CST Microwave Studio. They are not visible in the 2D simulations. The change in signal shape happens very soon after transmission in the 3D simulations only. This occurs using a single celled hard source for FDTD and a dipole antenna in CST Microwave Studio. The change in signal is also frequency dependent and occurs further away for lower frequencies (with a fixed simulation environment). Other simulations not shown indicate that halving the simulation time step (and hence halving the Courant number) does not have any noticeable effect on either the 2D or 3D results with relation to causing or altering the change in signal. One noticeable finding is this behaviour occurs on the reactive near field boundary as given for electrically small antennas by Eq. (2.4)

$$r = \frac{\lambda}{2\pi} \tag{2.4}$$

2.4 Chapter Conclusions

Comparisons between the commercial package CST Microwave Studio and FDTD code has shown that the FDTD gives comparable results usable for simulation purposes to generate data. Even though the 3D FDTD version also gives comparable

results to CST Microwave Studio there are question marks over its results so only 2D simulations will be used. On the FDTD code itself there are limitations.

The first limitation is memory requirements, it can only simulate up to 2.1 million cells in the problem domain (with a 1.7GHz Pentium 4 class processor and 1GB of memory this gives a simulation time of approximately 1 hour). There is a design trade off in this as the memory requirements are just under twice that of FDTD code written to utilise loops, the extra memory is used to calculate large sections of these loops directly in a matrix form that runs quicker under Matlab. This means with a problem domain of 0.2m x 0.1m x 0.2m the smallest cell size is 0.00125m which corresponds to $1/10\lambda$ of 8GHz in the dielectric medium used. Ideally a smaller cell size down to $1/20\lambda$ would be better but it cannot be simulated. This also limits the maximum frequency of the excitation signal to 8GHz.

2D simulations on the other hand do not have these problems for the size of the problem domain looked at. The boundary condition also has significant reflections under certain conditions making it less than ideal, as a greater spacing between the simulation objects and boundary is required leading to a larger simulation (more time and memory). In summary all simulations carried out to generate data will be using FDTD code with Mur boundary conditions in a 2D simulation.

Chapter 3

PN-Sequences, Correlation and Filters

Pseudo random noise sequences, usually referred to as PN-sequences, are binary sequences that have properties linking them to random noise [45]. They have a number of useful properties that make them very important in communication systems. The goal in using PN-sequences is to create a uniquely identifiable signal that can be distinguished from background noise and to allow detection at very low signal levels. Wherever possible a subset of PN-sequences called M-sequences will be used. Some of the benefits of PN-sequences include noise resilience, multiple signal transmission and reception capabilities, encryption and its uniquely identifiable properties.

3.1 M-Sequences

This is a Pseudo-random Noise (PN) sequence of maximal length from a given length (n) of linear shift register. An M-sequence has a set of properties that meet randomness criteria as laid out in [46] and given below.

1. Balance
   (a) Sequence length is given by 2^n-1
   (b) There are 2^{n-1} ones and $2^{n-1}-1$ zeroes
2. Run
   (a) 2^{n-2} runs of ones alternating with 2^{n-2} runs of zeroes
   (b) 2^{n-k-2} runs of each length for $1 \leq k \leq n-2$
   (c) 1 run of $n-1$ zeroes and 1 run of n ones

3. Two Level Correlation

   (a) In a cyclic shift there are $2^{n-1}-1$ agreements and 2^{n-1} disagreements

   (b) Using ± 1 the normalized auto-correlation equals 1 for $\tau = qp$ and $-1/p$ for all others (where q is any integer number and $p = 2^n-1$)

As an example a shift register of length $n = 4$ has the following properties

1. Balance

   (a) Sequence length is 15

   (b) It has 8 ones and 7 zeroes

2. Run

   (a) There are 4 runs of ones alternating with 4 runs of zeroes

   (b) There are 2 runs of length 1 and 1 run of length 2

   (c) There is also a single run of zeroes with length 3 and of ones with length 4

3. Two Level Correlation

   (a) In a cyclic shift there are 7 agreements and 8 disagreements

   (b) The normalised auto-correlation is 1 for $\tau = 15q$ or $-1/15$ for all others

By logical extension of the alternating runs structure the last number in the sequence will always be different from the first.

3.2 Generation of a PN-Sequence

A common way to generate PN-sequences is to use a linear shift register and exclusive-or (XOR) some of the output bits to create the next input bit [46]. The PN-sequence is the actual output read from the last bit. Choosing which bits to take the XOR from determines the length of the PN-sequence. For a given Linear Shift Register of length n bits, the longest PN-sequence is 2^n-1 and any PN-sequence of

this length is an M-sequence. An example of a 7 bit (n=7) PN sequence generator that outputs an M-sequence is given in Figure 3.1. For any given n there are usually several different configurations to generate an M-sequence. This version XOR's the last two positions in the linear shift register (bits 6 and 7) for the next input bit. The 7 bits already in the shift register are a random seed value to ensure consistent operation and hence become the first 7 bits of the sequence. The only limitation on the seed value is that it cannot be all 0s.

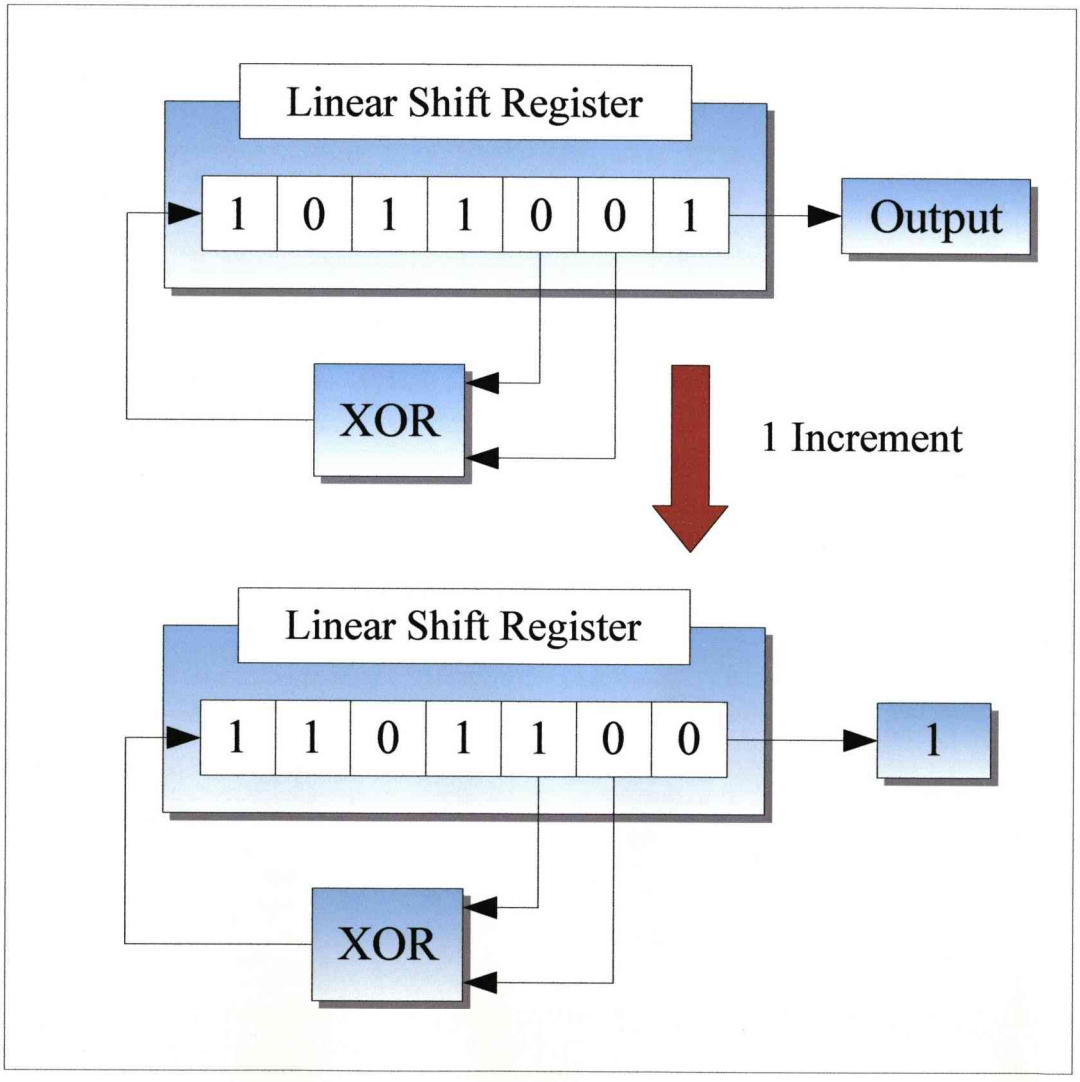

Figure 3.1 - 7 bit Linear Shift Register for M-sequence generation

Longer or shorter sequences can be generated by altering the number of bits in the linear shift register and looking up published lists of which bits to take the XOR from. Longer sequences generally have more options and can generate many unique sequences of the same length.

3.3 Experimental Results looking at M-Sequence Properties

Using the generation method from the previous section a selection of M-Sequence lengths from 15 to 1023 were generated and then autocorrelated in two ways, cyclic autocorrelation where the sequence wraps around (ideal case) and standard autocorrelation using the single sequence only (Most likely case for a singularly transmitted signal). The most noticeable thing is the difference between the two cases away from the zero delay correlation point. The cyclic correlation does reduce down to the predicted theoretical value of $-1/(2^n-1)$ [46] but the standard autocorrelation with a single sequence shows some variation. It's also noticeable that as the sequence length increases this variation reduces also.

Example

A short M-sequence consisting of 7 bits is correlated against itself as a single sequence. The M-sequence being used is [-1, 1, -1, 1, 1, 1, -1] padded with as many zeros as necessary at the start and end. Calculation for a single bit of the correlation consists of a bit by bit multiplication of the two overlapping sequences and then all the bits of the multiplication result are summed. A single correlation bit example is given in Table 3.1 and the complete correlation result is given in Table 3.2.

Table 3.1 – Calculation of single bit for correlation

	Bit Value												Sum
Reference Sequence	0	0	0	0	-1	1	-1	1	1	1	-1	0	
Correlating Sequence	0	-1	1	-1	1	1	1	-1	0	0	0	0	
Bit Multiplication	0	0	0	0	-1	1	-1	-1	0	0	0	0	-2

Table 3.2 –Full correlation result

	Bit Value																				
Reference Sequence	0	0	0	0	0	0	0	-1	1	-1	1	1	1	-1	0	0	0	0	0	0	0
Correlation	0	1	-2	-1	-2	1	-2	7	-2	1	-2	1	-2	1	0	0	0	0	0	0	0

The worked correlation example shows no trailing off towards the edges when only a few bits of the two signals are overlapping and contributing to the correlation. Although there would be an upper limit on the values they could take (equal to the number of bits overlapping) it doesn't appear to have a significant visible effect on this example or the following numerical examples.

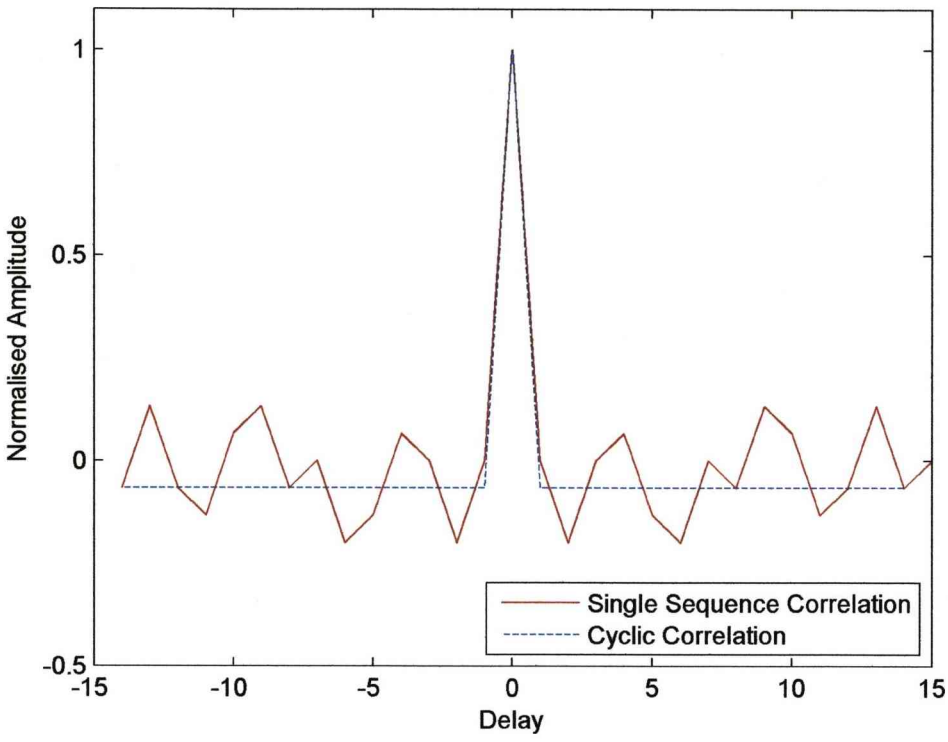

Figure 3.2 - Autocorrelation for a sequence length of 15 (n=4)

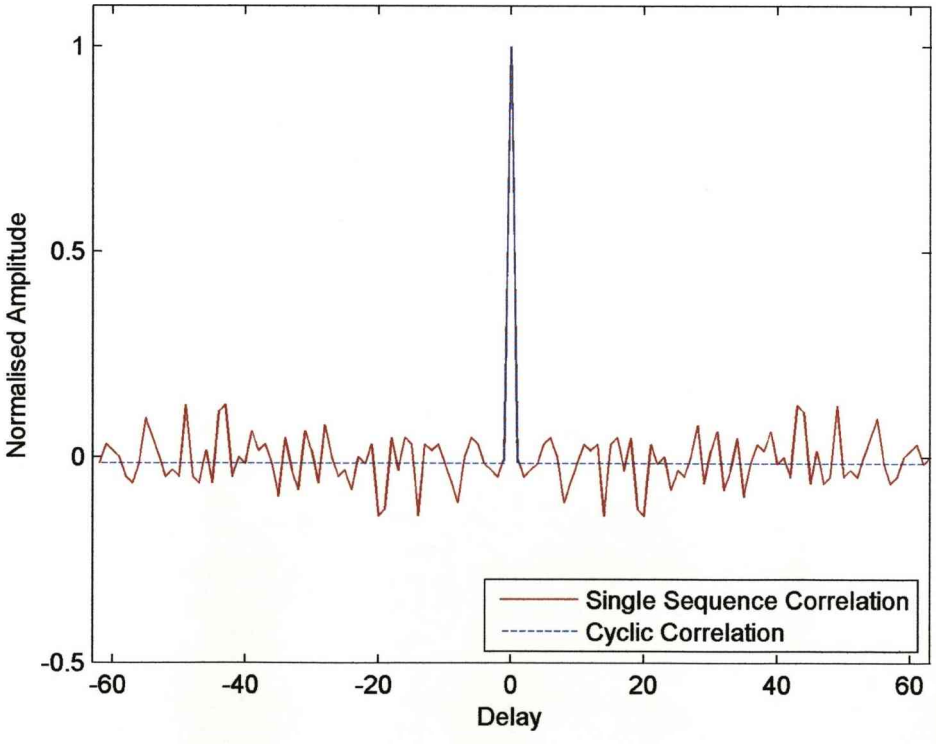

Figure 3.3 - Autocorrelation for a sequence length of 63 (n=6)

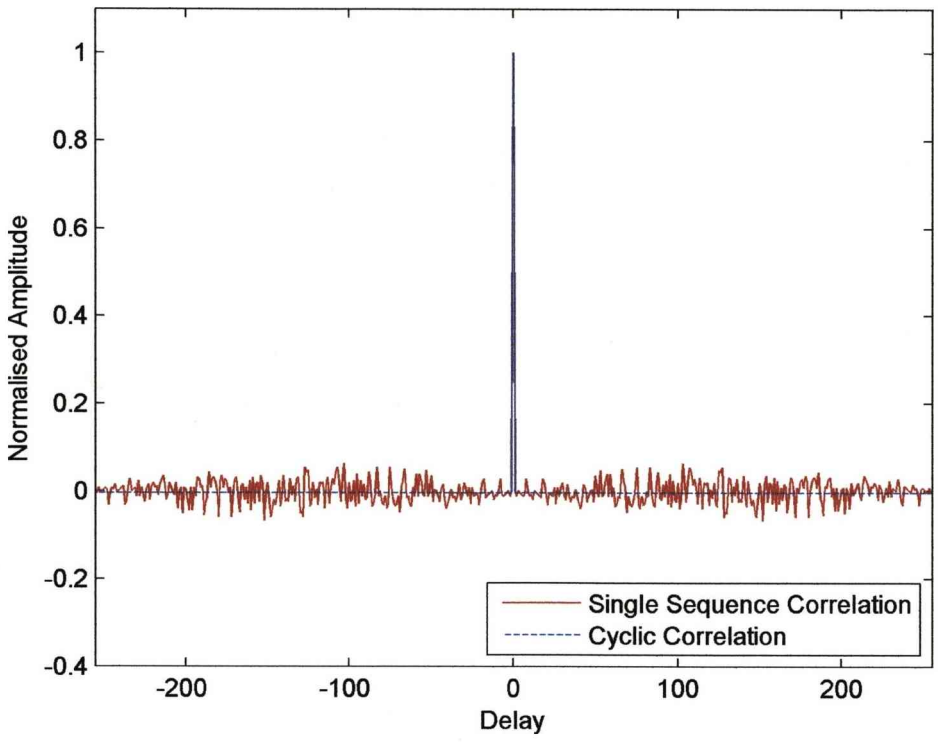

Figure 3.4 - Autocorrelation for a sequence length of 255 (n=8)

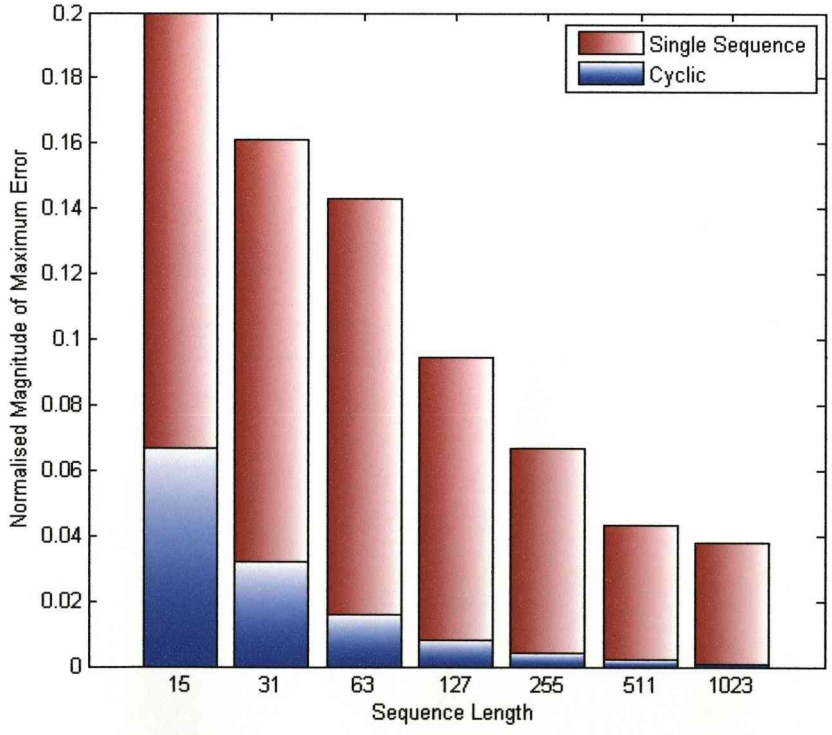

Figure 3.5 - Normalized magnitude of maximum error compared to sequence length

The general trend is that the normalized maximum error is reducing with sequence length but it's clear that the cyclical autocorrelation will yield better results with a lower error as shown in Figure 3.5 and that error will be of a fixed value as shown in Figures 3.2 to 3.4. It's clear that cyclical correlation is the better choice but it does require extra work to use. Firstly it will behave like single sequence correlation prior to the signal being received and only behave like the cyclical correlation if the sequence is repeated. Effectively this means to achieve the ideal result would require correlation against the centre signal in a transmitted signal consisting of at least 3 identical sequences. There will also always be a small negative no correlation value but with longer sequences it's insignificant.

3.4 Pulse Correlation

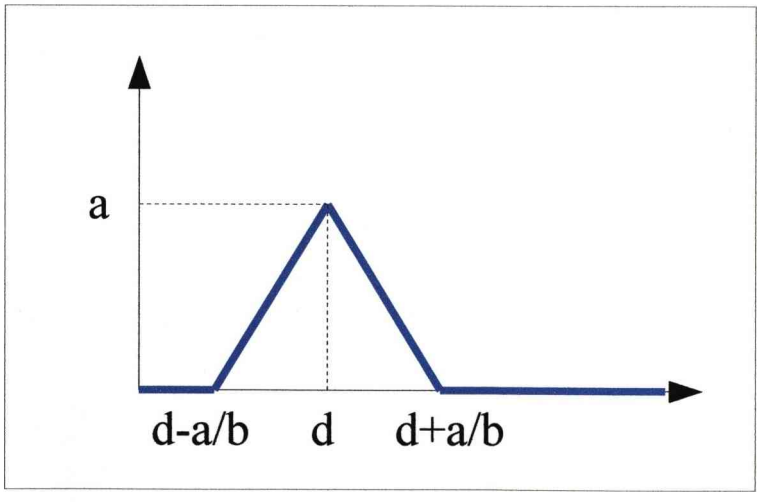

Figure 3.6 - Pulse shape

The pulse shape shown in Figure 3.6 is given by Eq. (3.1). Note that the signal limits run from $d-a/b$ to $d+a/b$ for a total pulse width of $2a/b$.

$$f(x)=\begin{cases} 0 & \text{for } x\le d-\frac{a}{b} \\ a+b(x-d) & \text{for } d-\frac{a}{b}<x\le d \\ a-b(x-d) & \text{for } d<x\le d+\frac{a}{b} \\ 0 & \text{for } d+\frac{a}{b}<x \end{cases} \tag{3.1}$$

The correlation integral (3.2) needs to be solved in several parts for this pulse.

$$R(\Delta)=\lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} f(x)f(x-\Delta)dx \tag{3.2}$$

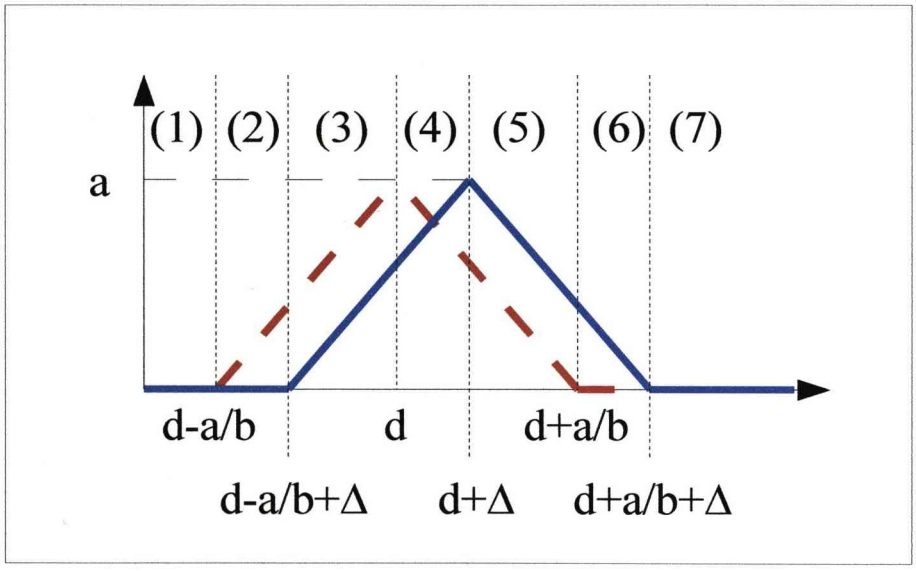

Figure 3.7 - Correlation Integral Combinations

Due to the various overlap segments, the integral is split into 7 parts, some of which are zero and the normalisation factor is also removed. Part 4 is a special case with two variations depending on the delay value Δ, different limits are used in the integration accordingly. Solutions for the integrals of each section is given in the appendix section A.1.1.

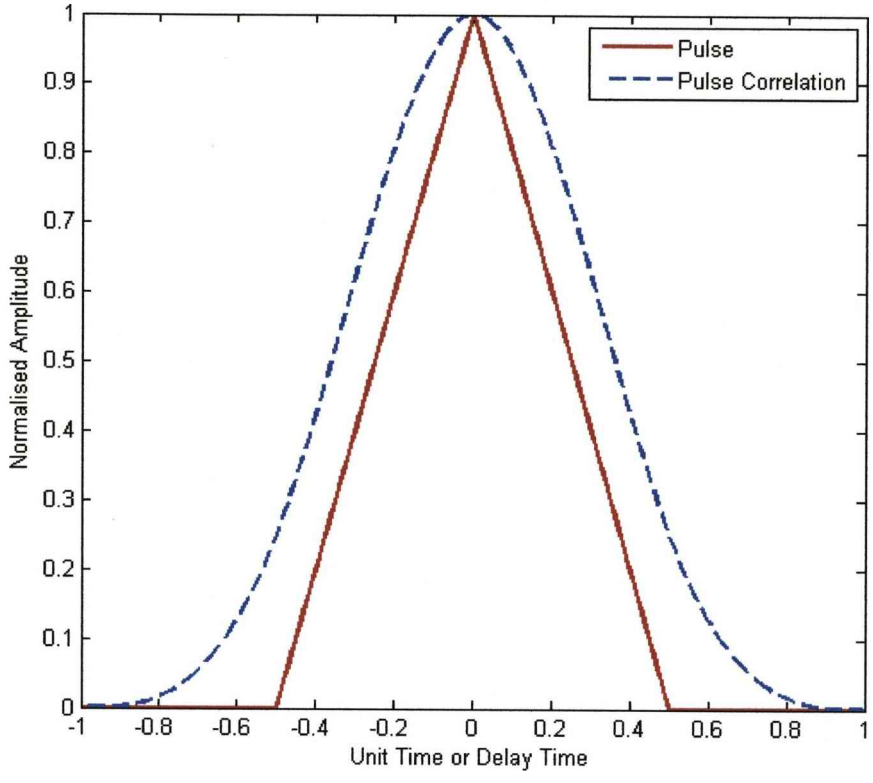

Figure 3.8 - Single pulse compared to correlation

$$R(\Delta)=\begin{cases} 0 & \text{for } \frac{2a}{b}<|\Delta| \\ \frac{b^2|\Delta|^3}{2}-ab|\Delta|^2+\frac{2a^3}{3b} & \text{for } |\Delta|\le\frac{a}{b} \\ -\frac{b^2|\Delta|^3}{6}+ab|\Delta|^2-2a^2|\Delta|+\frac{4a^3}{3b} & \text{for } \frac{a}{b}\le|\Delta|<\frac{2a}{b} \end{cases} \tag{3.3}$$

The correlation result should be symmetric about Δ so Eq. (3.3) uses delay magnitude $|\Delta|$ and is split due to section 4 having 2 parts. Its application to cross-correlation of a signal and a delayed copy of itself is simply $R(\Delta-d)$ where d is the appropriate delay. Note that the correlation result for a pulse of width $2a/b$ has a width of $4a/b$. Figure 3.8 shows the result.

3.4.1 PN Sequence Correlation

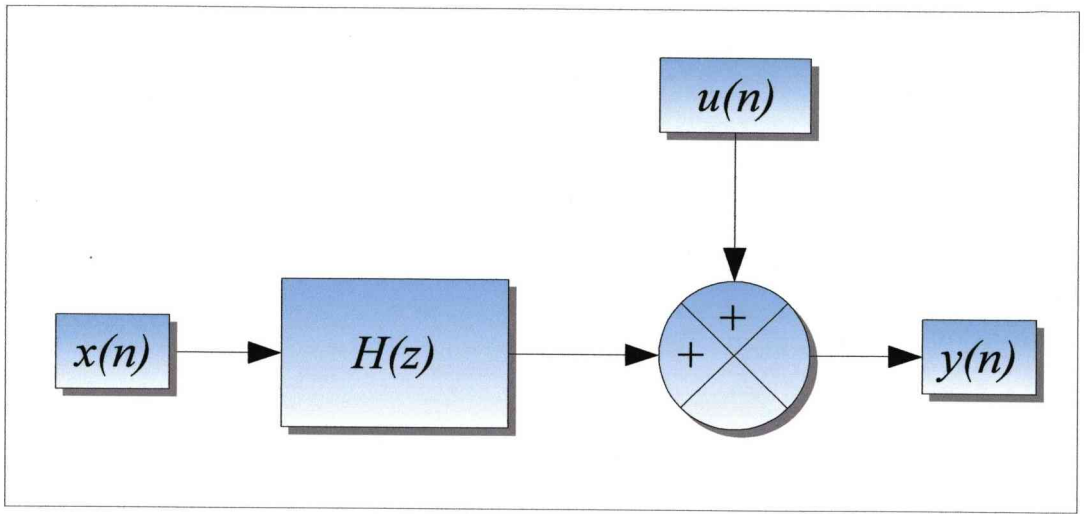

Figure 3.9 - Block Diagram

This section is based on [47]. A signal *x(n)* is passed through a channel *H(z)* and has a noise signal *u(n)* added with the result being *y(n)*.

$$y(n)=h(n)*x(n)+u(n) \tag{3.4}$$

The correlation between the input signal *x(n)* and the output *y(n)* is given by Eq. (3.4)

$$\begin{aligned} r_{xy}&=\varepsilon\{(h(n)*x(n)+u(n))(\bar{x}(n-k))\} \\ r_{xy}&=\varepsilon\{(h(n)*x(n))(\bar{x}(n-k))\}+\varepsilon\{u(n)(\bar{x}(n-k))\} \\ r_{xy}&=h(n)*\varepsilon\{x(n)(\bar{x}(n-k))\}+\varepsilon\{u(n)(\bar{x}(n-k))\} \\ r_{xy}&=h(n)*r_{xx}+r_{xu} \end{aligned} \tag{3.5}$$

If *x(n)* is a sufficiently long PN sequence encoded as +1 and −1, r_{xx} tends to a unit impulse and the result is

$$r_{xy}=h(n)+r_{xu} \tag{3.6}$$

If the noise term (r_{xu}, the correlation between signal and noise) is small then r_{xy} becomes the impulse response of the channel *h(n)*.

3.4.2 Correlation for a PN Sequence Constructed from Pulses

Using the defined pulse as a PN sequence symbol and constructing a PN sequence with no spacing between symbols leads to the cyclic correlation output for strong correlation values in Eq. (3.7) and no correlation values in Eq. (3.8).

$$R(\Delta)\begin{cases}\left(\frac{b^2}{2}|\Delta|^3-ab|\Delta|^2+\frac{2a^3}{3b}\right)(2^n-1)+\frac{b^2}{6}\left(\frac{2a}{b}-|\Delta|\right)^3 \\ -ab\left(\frac{2a}{b}-|\Delta|\right)^2+2a^2\left(\frac{2a}{b}-|\Delta|\right)-\frac{4a^3}{3b}\end{cases} \text{for } -\frac{a}{b}\le\Delta<\frac{a}{b}$$
$$\begin{cases}\left(-\frac{b^2}{6}|\Delta|^3+ab|\Delta|^2-2a^2|\Delta|+\frac{4a^3}{3b}\right)(2^n-1) \\ -\frac{b^2}{2}\left(\frac{2a}{b}-|\Delta|\right)^3+ab\left(\frac{2a}{b}-|\Delta|\right)^2-\frac{2a^3}{3b}\end{cases} \text{for } \frac{a}{b}\le\Delta<\frac{2a}{b} \text{ and } -\frac{2a}{b}<\Delta\le-\frac{a}{b} \tag{3.7}$$

The strong correlation values repeat at the sequence length so are used every qp symbols where p is the sequence length ($p=2^n-1$) and q is any integer value.

$$R(\Delta)\begin{cases}-\frac{b^2}{2}|\Delta|^3+ab|\Delta|^2-\frac{2a^3}{3b}+\frac{b^2}{6}\left(\frac{2a}{b}-|\Delta|\right)^3 \\ -ab\left(\frac{2a}{b}-|\Delta|\right)^2+2a^2\left(\frac{2a}{b}-|\Delta|\right)-\frac{4a^3}{3b}\end{cases} \text{for } -\frac{a}{b}\le\Delta<\frac{a}{b}$$
$$\begin{cases}\frac{b^2}{6}|\Delta|^3-ab|\Delta|^2+2a^2|\Delta|-\frac{4a^3}{3b} \\ -\frac{b^2}{2}\left(\frac{2a}{b}-|\Delta|\right)^3+ab\left(\frac{2a}{b}-|\Delta|\right)^2-\frac{2a^3}{3b}\end{cases} \text{for } \frac{a}{b}\le\Delta<\frac{2a}{b} \text{ and } -\frac{2a}{b}<\Delta\le-\frac{a}{b} \tag{3.8}$$

The no correlation values repeat at the symbol length ($2a/b$) but the strong correlation values take priority over the no correlation values if the limits are met. Figure 3.10 shows the correlation over several symbol lengths compared to the single symbol correlation. Note the periodic feature and its fluctuations between -0.033 and -0.066,

the lower limit of -0.066 being the *1/p* value (where p=15) for correlation of a maximal length PN sequence.

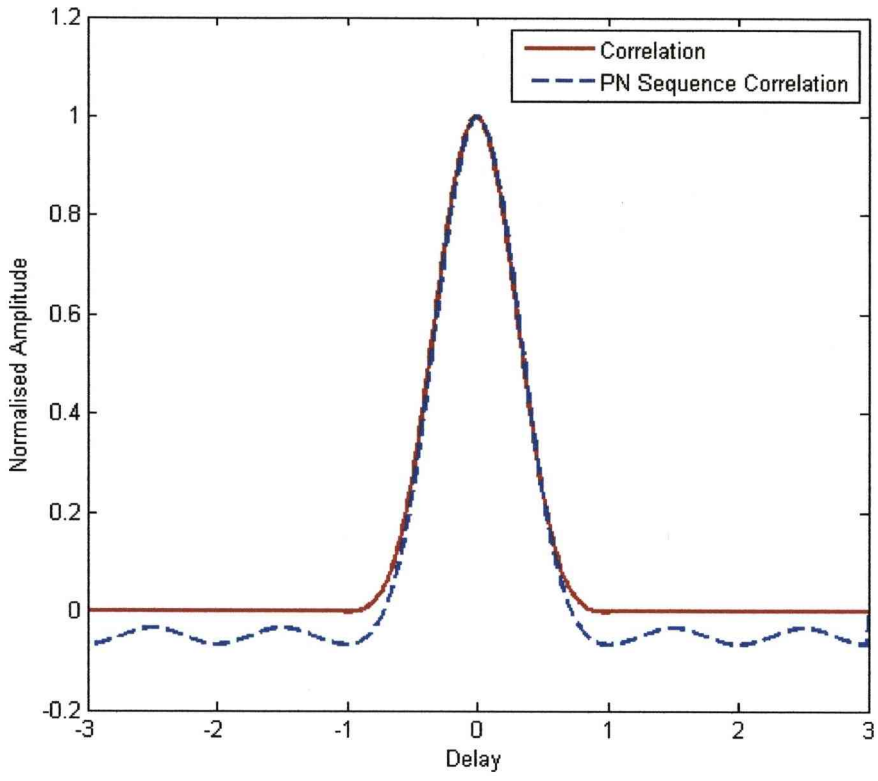

Figure 3.10 - Arbitrary pulse PN sequence correlation (p=15, a=1, b=2)

The use of a maximal length PN sequence of pulses shows minor deterioration over a single pulse with regards to correlation but will provide significant other benefits.

3.5 Filtering

Using a PN sequence of pulses has shown minor deterioration in the output after correlation compared to a single pulse. The use of a filter and appropriate algorithm should provide a benefit in detecting a PN sequence without the small spreading associated with correlation and the small amount of deterioration. A Wiener filter and least mean square (LMS) algorithm will be used for this purpose. Figure 3.11 shows

the block diagram where the goal is to make the filter an approximation of the channel the signal is sent through by matching an input (transmitted) signal to a desired (received signal). The working is based on [47].

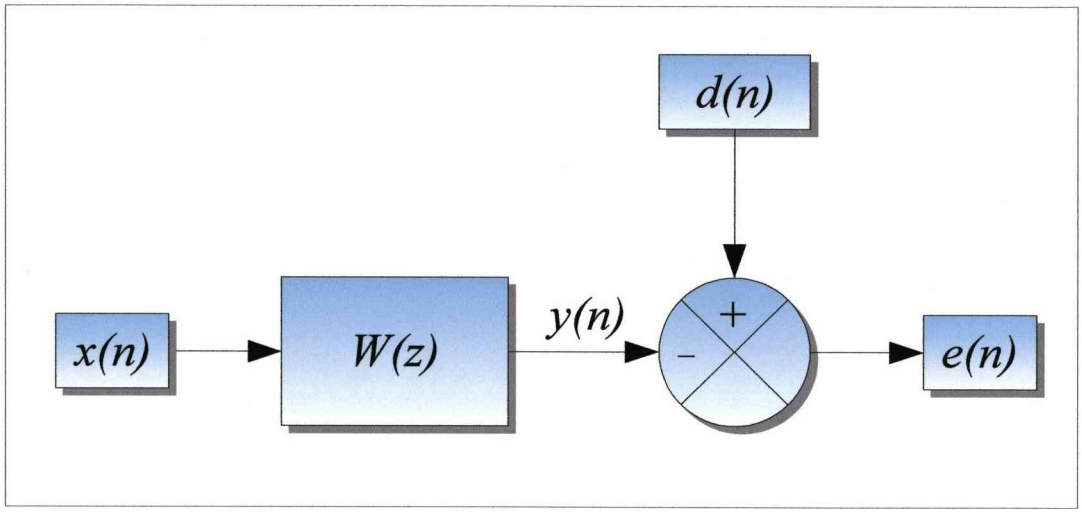

Figure 3.11 - Filter Block Diagram

Table 3.3 – Filter variables

Name	Description
x(n)	Input signal
W(z)	Wiener filter
y(n)	Estimated desired signal (wiener filter output)
d(n)	Desired signal
e(n)	Error signal [*d(n)−y(n)*]

For a $(p-1)$th order filter the Wiener filter is given as

$$W(z)=\sum_{n=0}^{p-1}\omega(n)z^{-n} \tag{3.9}$$

From Figure 3.11 we get

$$y(n)=\sum_{l=0}^{p-1}\omega(l)x(n-l) \tag{3.10}$$

To find the filter coefficients the mean-squared error (MSE) needs to be minimised

$$\xi=\varepsilon\{|e(n)|^2\}=\varepsilon\{|d(n)-y(n)|^2\} \tag{3.11}$$

To minimise MSE

$$\begin{aligned}\frac{\partial \xi}{\partial \omega^*(k)} &= \frac{\partial}{\partial \omega^*(k)} \varepsilon\left\{e(n)e^*(n)\right\} \\ &= \varepsilon\left\{e(n)\frac{\partial e^*(n)}{\partial \omega^*(k)}\right\} = 0\end{aligned} \tag{3.12}$$

Noting that

$$\frac{\partial e^*(n)}{\partial \omega^*(k)} = -x^*(n-k) \tag{3.13}$$

$$\varepsilon\left\{e(n)x^*(n-k)\right\} = 0 \quad \text{for } k=0, 1, ..., (p-1) \tag{3.14}$$

Substituting for *e(n)* from Eq (3.10) and Eq. (3.11) gives

$$\varepsilon\left\{\left(d(n) - \sum_{l=0}^{p-1} \omega(l)x(n-l)\right)x^*(n-k)\right\} = 0 \tag{3.15}$$

$$\begin{aligned}&\varepsilon\left\{d(n)x^*(n-k)\right\} \\ &\quad - \sum_{l=0}^{p-1} \omega(l)\varepsilon\left\{x(n-l)x^*(n-k)\right\} = 0\end{aligned} \tag{3.16}$$

x(n) and *d(n)* are jointly wide-sense stationary (WSS) so

$$\varepsilon\left\{d(n)x^*(n-k)\right\} = r_{dx}(k) \tag{3.17}$$

$$\varepsilon\left\{x(n-l)x^*(n-k)\right\} = r_{xx}(k-l) \tag{3.18}$$

This leads to

$$\sum_{l=0}^{p-1} \omega(l)r_{xx}(k-l) = r_{dx}(k) \tag{3.19}$$

$$\begin{bmatrix} r_{xx}(0) & r_{xx}^*(1) & \cdots & r_{xx}^*(p-1) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}^*(p-2) \\ \vdots & \vdots & & \vdots \\ r_{xx}(p-1) & r_{xx}(p-2) & \cdots & r_{xx}(0) \end{bmatrix} \begin{bmatrix} \omega(0) \\ \omega(1) \\ \vdots \\ \omega(p-1) \end{bmatrix} = \begin{bmatrix} r_{dx}(0) \\ r_{dx}(1) \\ \vdots \\ r_{dx}(p-1) \end{bmatrix} \tag{3.20}$$

Autocorrelation sequences are conjugate symmetric:

$$r_{xx}(k)=r^{*}_{xx}(-k) \tag{3.21}$$

So eventually leading to the Wiener-Hopf equation

$$R_{xx}\omega=r_{dx} \tag{3.22}$$

Table 3.4 – Wiener-Hopf equation variables and sizes

Name	Size	Description
p	-	Filter order (number of filter coefficients)
R_{xx}	$p \times p$	Hermitian Toeplitz matrix of auto-correlations
ω	$p \times 1$	Vector of filter coefficients
$\underline{r}_{dx}$	$p \times 1$	Vector of cross-correlations between the desired signal *d(n)* and the observed signal *x(n)*

3.5.1 Example Solutions

Some example solutions are given in this section that can also show the difference between the final filter solution (ω) and the correlation solution (r_{dx})

3.5.1.1 Example 1 - x=[1, 0, 0] and d=[0, 1, 0]

This gives r_{xx}=[1, 0, 0] and r_{dx}=[0, 1, 0]. Inputting to the Wiener-Hopf equation gives

$$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\omega(0)\\ \omega(1)\\ \omega(2)\end{bmatrix}=\begin{bmatrix}0\\ 1\\ 0\end{bmatrix}$$

$$\begin{bmatrix}\omega(0)\\ \omega(1)\\ \omega(2)\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}^{-1}\begin{bmatrix}0\\ 1\\ 0\end{bmatrix}$$

$$\begin{bmatrix}\omega(0)\\ \omega(1)\\ \omega(2)\end{bmatrix}=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\ 1\\ 0\end{bmatrix}$$

$$\begin{bmatrix}\omega(0)\\ \omega(1)\\ \omega(2)\end{bmatrix}=\begin{bmatrix}0\\ 1\\ 0\end{bmatrix}$$

The solution for the filter is as expected with a delay of one unit indicated by $\omega(1)=1$.

Now take a more complex example.

3.5.1.2 Example 2 - x=[0, 0.5, 1, 0.5, 0] and d=[0, 0.5, 1, 0.5, 0]

The first two numbers in x and the first and last numbers in d are used to calculate r_{xx} and r_{dx} but are not included in the matrices.

r_{xx}=[1.5, 1, 0.25] and r_{dx}=[1, 1.5, 1]

$$\begin{bmatrix} 1.5 & 1.0 & 0.25 \\ 1.0 & 1.5 & 1.0 \\ 0.25 & 1.0 & 1.5 \end{bmatrix} \begin{bmatrix} \omega(0) \\ \omega(1) \\ \omega(2) \end{bmatrix} = \begin{bmatrix} 1.0 \\ 1.5 \\ 1.0 \end{bmatrix}$$

$$\begin{bmatrix} \omega(0) \\ \omega(1) \\ \omega(2) \end{bmatrix} = \begin{bmatrix} 1.5 & 1.0 & 0.25 \\ 1.0 & 1.5 & 1.0 \\ 0.25 & 1.0 & 1.5 \end{bmatrix}^{-1} \begin{bmatrix} 1.0 \\ 1.5 \\ 1.0 \end{bmatrix}$$

$$\begin{bmatrix} \omega(0) \\ \omega(1) \\ \omega(2) \end{bmatrix} = \begin{bmatrix} 1.6 & -1.6 & 0.8 \\ -1.6 & 2.8 & -1.6 \\ 0.8 & -1.6 & 1.6 \end{bmatrix} \begin{bmatrix} 1.0 \\ 1.5 \\ 1.0 \end{bmatrix}$$

$$\begin{bmatrix} \omega(0) \\ \omega(1) \\ \omega(2) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Again the solution is as expected. Note that the correlation process r_{dx} gives a wide detection [0, 0.25, 1, 1.5, 1, 0.25, 0], nominally over the 3 units that the signal occupies. The optimum filter output though is a single point. This is a potential advantage that a filter can have over correlation.

3.6 Adaptive Filter

The adaptive filter is a linear tap delay line version updated using the LMS algorithm with working based on [47-48]. It takes in the transmitted signal ($x(n)$) and tries to

match it to the received signal, in this case the desired signal (*d(n)*). The coefficients (*ω(n)*) are used directly to indicate approximate strength and delay of any received signal either direct or reflected. The updating of the coefficients requires calculation of the error first using Eq. (3.23) before calculating the update using Eq. (3.24). The first set of coefficients *ω(n)* are generated using random numbers.

$$e(n)=d(n)-\left[\omega_n(1) \quad \cdots \quad \omega_n(p)\right]\begin{bmatrix} x(n) \\ \vdots \\ x(n-p) \end{bmatrix} \tag{3.23}$$

$$\omega_{n+1}=\omega_n+\mu e^*(n)x(n) \tag{3.24}$$

The parameters that make up the filter control its performance. Along with the signals, the controlling parameters are p, the number of filter coefficients and μ, the step size of the update algorithm. Also because the signals are of limited length in order to get stable performance the filter coefficient update needs to be used multiple times with the same signals (*x(n)* and *d(n)*). The number of times the update is used is also a controlling parameter called the number of iterations. If required every filter coefficient need not be used leading to a filter coefficient every 2 or more time steps as opposed to every time step, the coefficient spacing.

3.6.1 Influence of the M-Sequence Length

This is to test the performance of the filter with a variable length of sequence and to compare it's performance to correlation. The filter was set up to use 1000 coefficients with single time unit precision and 100 signal iterations. A set of sequence lengths ranging from 15 to 1023 ($p=2^n-1$ for n from 4 to 10) was generated before being run through the filter and correlation procedures. Increasing the number of symbols from 15 in Figure 3.12 to 255 in Figure 3.14 reduces the variation of the minimum value

around the ideal value of 0 for the filter method. The filter though gives a smaller variation, not by much with short sequence lengths such as 15 but by a sequence length of 255 is rivalling the cyclic correlation. One problem with the filter is that single sequence correlation becomes zero when outside the area of interest while the filter still shows some variation that lessens with increased sequence length.

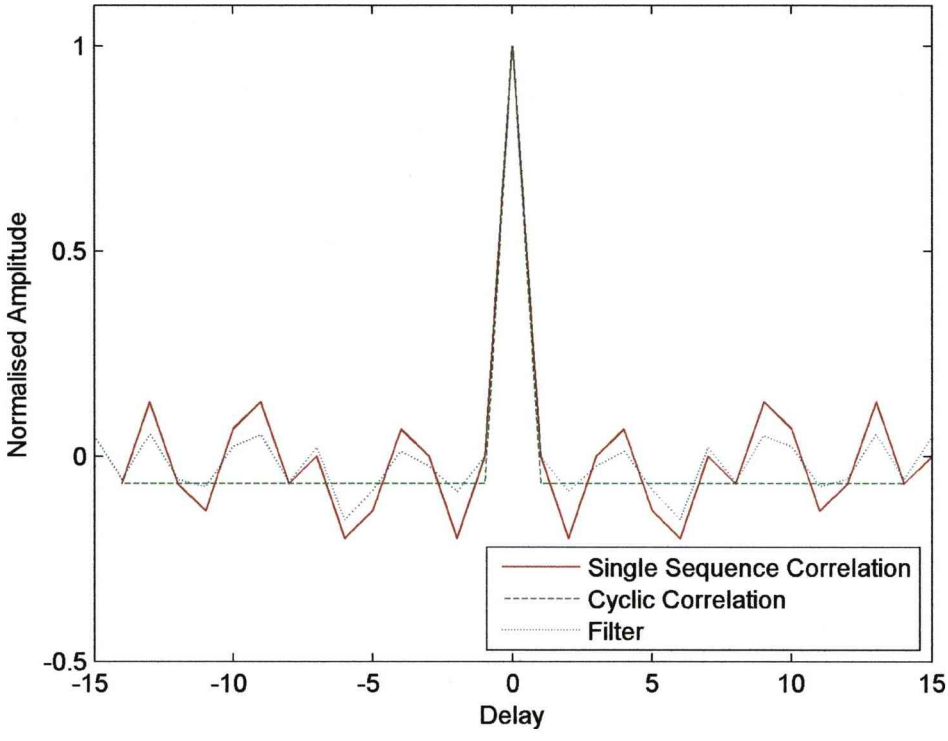

Figure 3.12 - Filter vs. Autocorrelation for a sequence length of 15 (n=4)

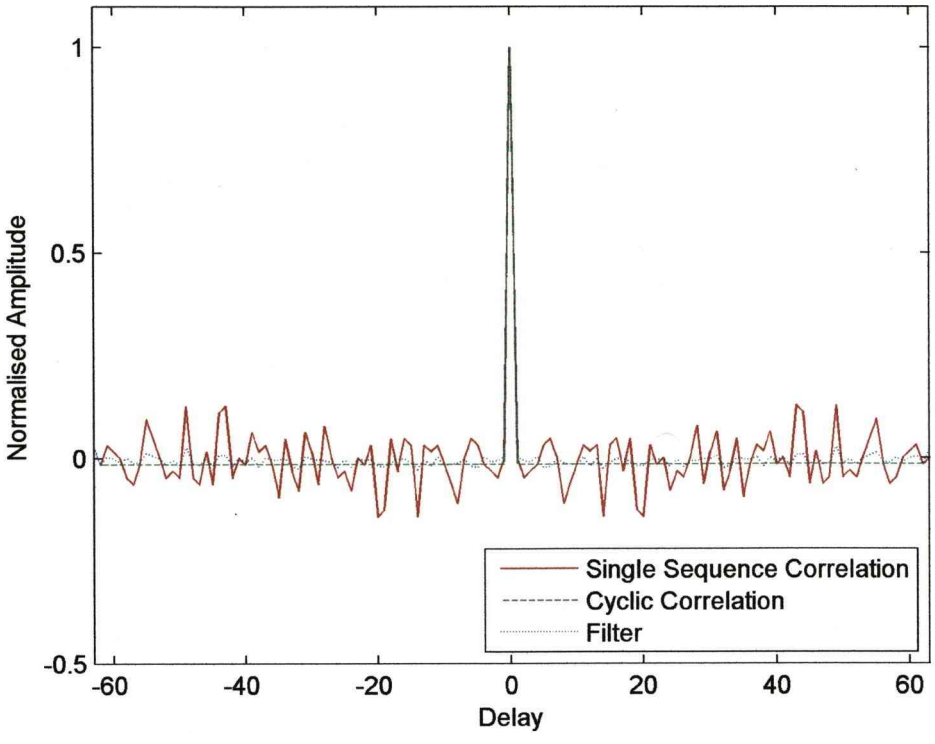

Figure 3.13 - Filter vs. Autocorrelation for a sequence length of 63 (n=6)

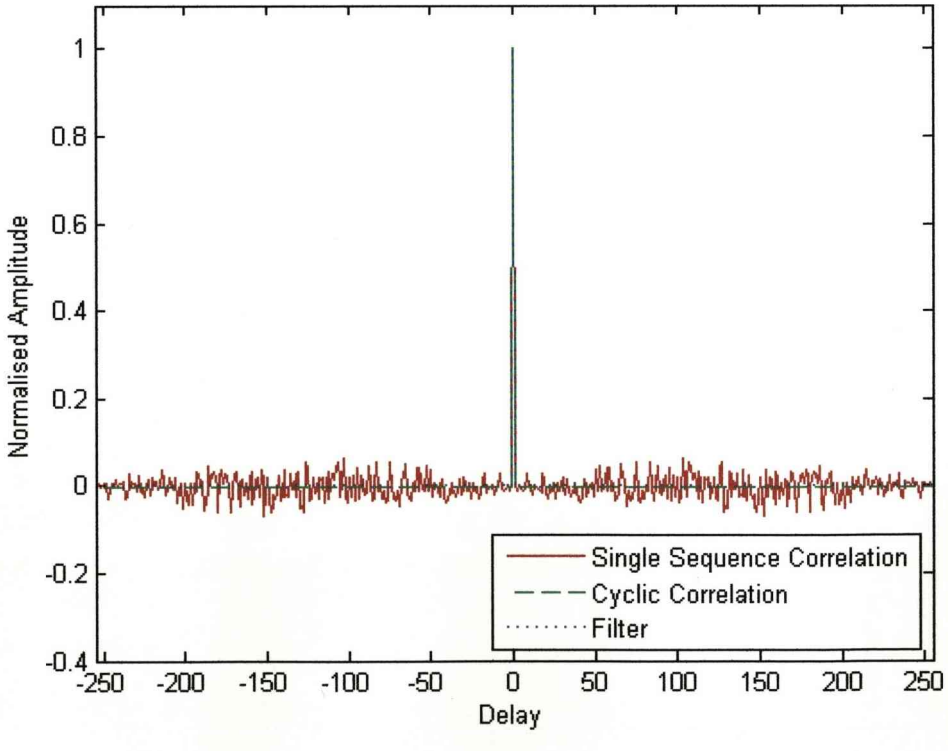

Figure 3.14 - Filter vs. Autocorrelation for a sequence length of 255 (n=8)

Note that in Figures 3.12-3.14 there is a slight difference between the cyclic correlation and filter where the two settle to $1/p$ and 0 respectively.

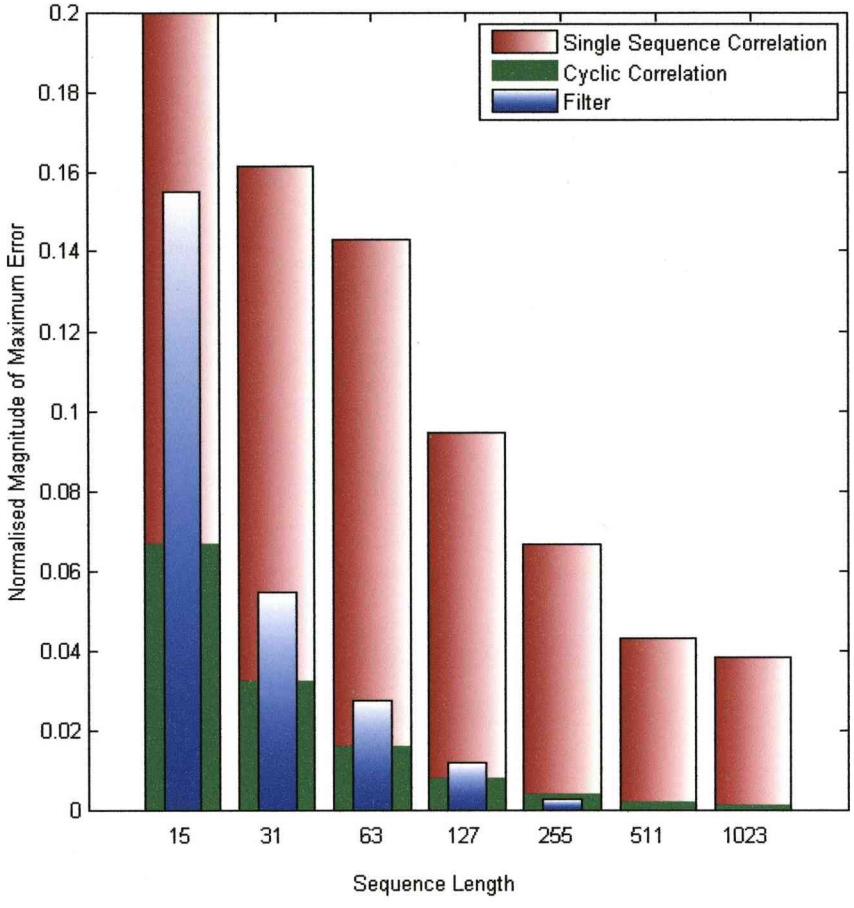

Figure 3.15 - Normalized magnitude of maximum error compared to sequence length

The error plot shown in Figure 3.15 is lower for the filter than for the single sequence correlation regardless of the sequence length. The filter and correlation show larger errors than the cyclic correlation due to the filter process itself and using only parts of the sequence. The filter shows much better management of this leading to the lower error. The filter error also drops below the cyclic correlation from a sequence length of 255 upwards (due to the cyclic correlation having a permanent error of $1/p$ compared to the ideal 0 value from the filter). This should lead to better performance at detecting a received signal for any sufficiently long sequence length when using the

filter instead of a correlation procedure. In the end the filter method is the best option in almost all cases. In order to use cyclic correlation the sequence must be transmitted at least three times, so it's clear that the effort to use cyclic correlation will always be worth it. This is because for three times the transmission time you get better performance than for a longer PN sequence using single sequence correlation. However if a longer PN sequence is used then the filter method offers better performance. In summary it is better to increase the PN sequence length and use the filter method than to use multiple transmissions of a shorter sequence to use cyclic correlation.

3.6.2 Filter Parameter Comparisons

The filter has a number of different parameters that affect performance. The key ones to investigate are the number of coefficients, coefficient spacing, update step size and number of iterations. The number of coefficients is directly related to the range of delay to detect a signal and also has an impact on computation time. Coefficient spacing is an additional factor that generates a reduced data solution where appropriate (so 1 coefficient for every X data points), one to one mapping will be used to match the one data point per symbol signal being used. Update step size and number of iterations relate to convergence of the filter to the optimum solution. To test these factors a 255 symbol long M-sequence is used with a delay of 500 units for the target signal.

3.6.2.1 Number of Coefficients

The number of coefficients determines the range of delay that can be detected, it is important that the number is larger than the expected delay so that the delayed signal

is detected correctly. The detection peak value, no detection minimum, maximum and mean are given in Table 3.5 for a range of coefficient values (100 iterations and 0.001 step size).

Table 3.5 – Effects of Number of Coefficients

Number of Coefficients	**Detection Peak**	**No Detection**		
		Minimum	**Maximum**	**Mean**
400	-	-0.0885	0.133	-0.0015
600	0.9996	-0.0005	-0.0001	-0.0003
800	0.9980	-0.0015	0.0004	-0.0005
1000	0.9959	-0.0024	0.0015	-0.0006
1200	0.9945	-0.0036	0.0024	-0.0006

Increasing the number of coefficients shows a slight variation in detection peak and the no detection minimum, maximum and mean. This is a slight degradation due to the larger number of coefficients and the coverage of the data range that it uses.

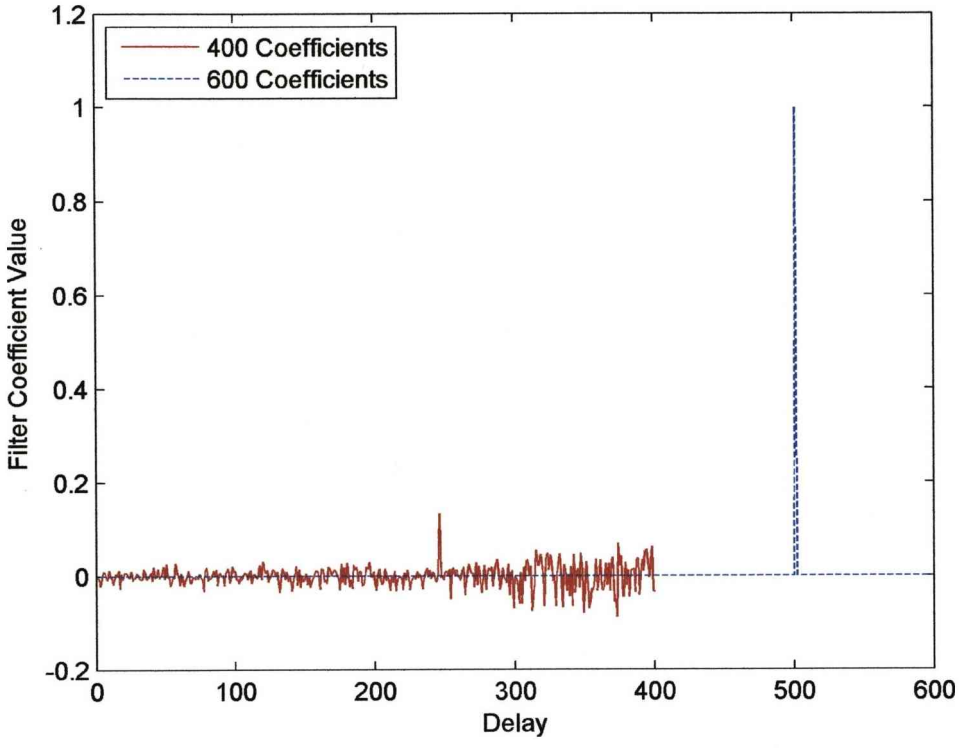

Figure 3.16 - Too few coefficients comparison

However when the number of coefficients drops below the delay value of 500, as highlighted in Figure 3.16, the detection is lost and the no detection results become much worse, the filter tries to find the best detection match but it actually lies outside of the accessible range. The detection is failed due to too few coefficients and produces a poor output.

3.6.2.2 Number of Iterations and Step Size

The updating of the filter's coefficients is performed using the LMS algorithm. The number of iterations it takes to converge depends on how accurate the values need to be and the choice of update step size chosen. Using an M-sequence with a length of 255 with several different step sizes and plotting overall error versus number of iterations gives a good idea of how much iteration is required.

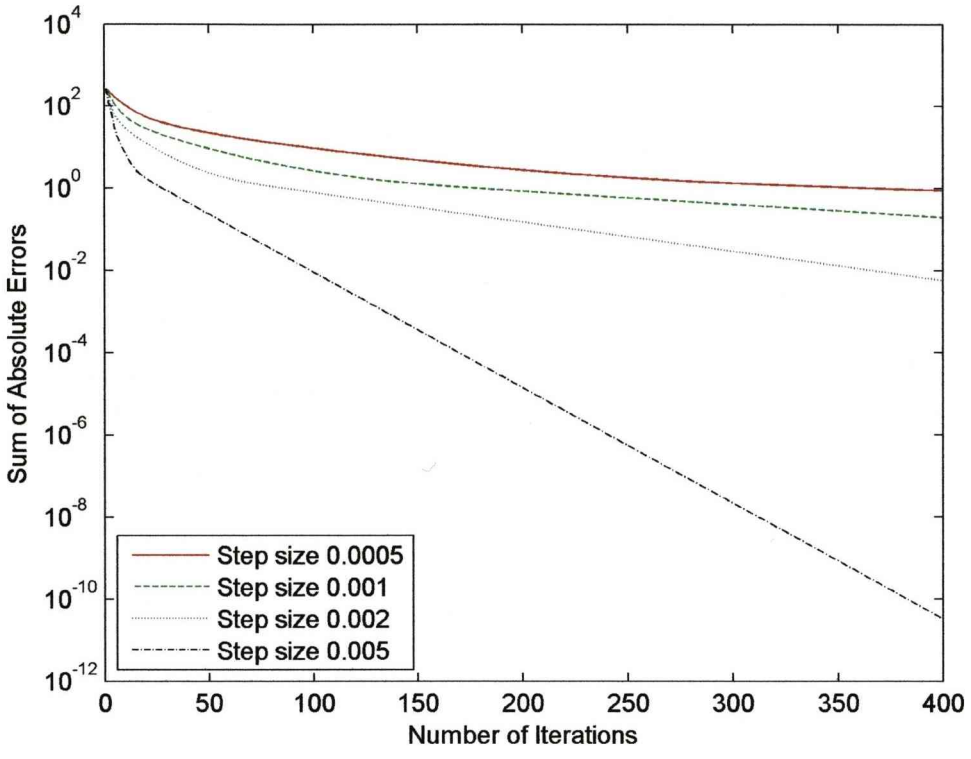

Figure 3.17 - Iterations and Step Size Error

Regardless of the step size a certain number of iterations are required before the error becomes sufficiently small. In this case the larger step sizes show faster convergence and a smaller overall error however there is a limit to how large it can be as one of the tested values, 0.01, was unstable. The convergence rate is also dependent on the signal so although a value of 0.005 looks the best a more conservative value of 0.001 would help ensure stability for different signals and provide a good convergence. Tailoring to the expected signals used would be a good idea in practice.

3.7 Simultaneous Transmission

One of the useful properties of M-sequences is the ability to transmit multiple different sequences of the same length simultaneously and using correlation identify each individual sequence at the receiving end. Correlation of a sequence with a

different sequence ideally results in zero correlation but in general will behave like noise and have an effect. In the worst case all the sequences could be received at the same time. Six 255 bit length sequences were generated and summed to give 2, 4 and 6 sequence combinations. The processes were run on these to give a comparison between filter, correlation and cyclic correlation to examine the effects of extracting a single M-sequence from a group. The data represents the background noise generated by the process away from the detection, is normalised to the peak value of the detection and where a number of sequences are present the process is run for each individual sequence and the result averaged (So for six sequences each individual sequence is correlated against the sum of all six, statistics calculated and then averaged to get a single value). The statistics are given in Table 3.6.

Table 3.6 – Effects of Number of Simultaneous Sequences

Number of Sequences	Correlation	Cyclic Correlation	Filter
Normalised Mean Value			
1	-0.0020	-0.0039	-0.0008
2	-0.0020	-0.0039	-0.0012
4	-0.0019	-0.0039	-0.0014
6	-0.0019	-0.0039	0.0000
Normalised Maximum Value			
1	0.0627	-0.0039	0.0004
2	0.2111	0.2296	0.2106
4	0.2957	0.3410	0.2932
6	0.3277	0.3582	0.3432
Normalised Minimum Value			
1	-0.0667	-0.0039	-0.0024
2	-0.1741	-0.1259	-0.1434
4	-0.2499	-0.2539	-0.2772
6	-0.3217	-0.3365	-0.3302

The overall effect is that the individual sequences are detectable but with more sequences combined there is an increase in background noise after processing. It is also process independent as all show similar results with increasing numbers of

simultaneous sequences. Taking a look at examples for a single sequence detected out of combined 2, 4 and 6 sequences shows some variations in the distributions of background noise.

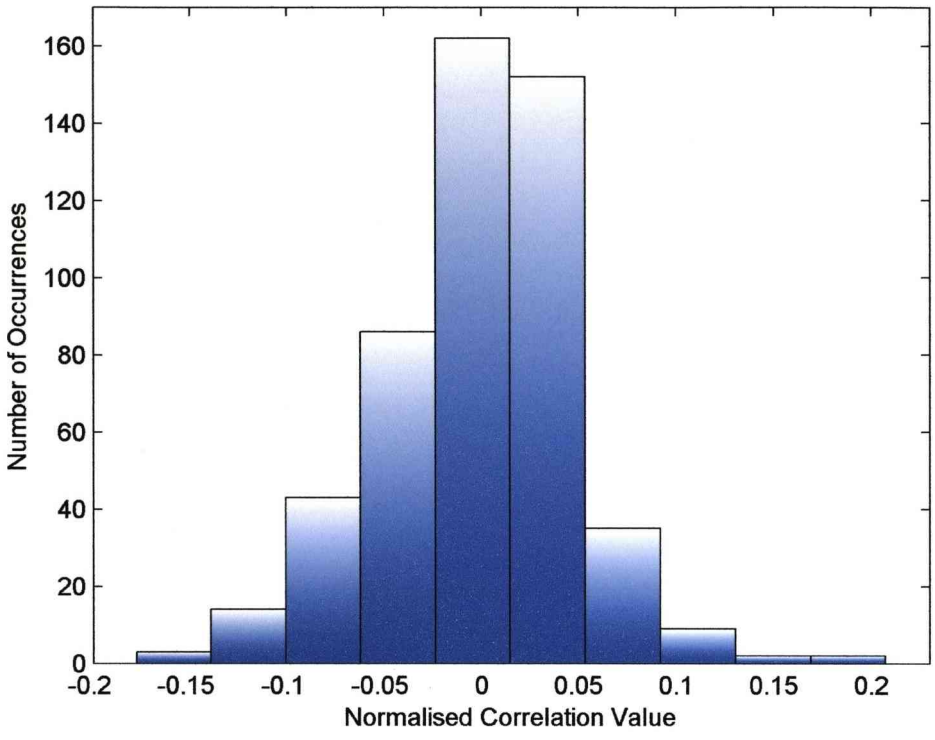

Figure 3.18 - Background value distribution from 2 Sequences using correlation

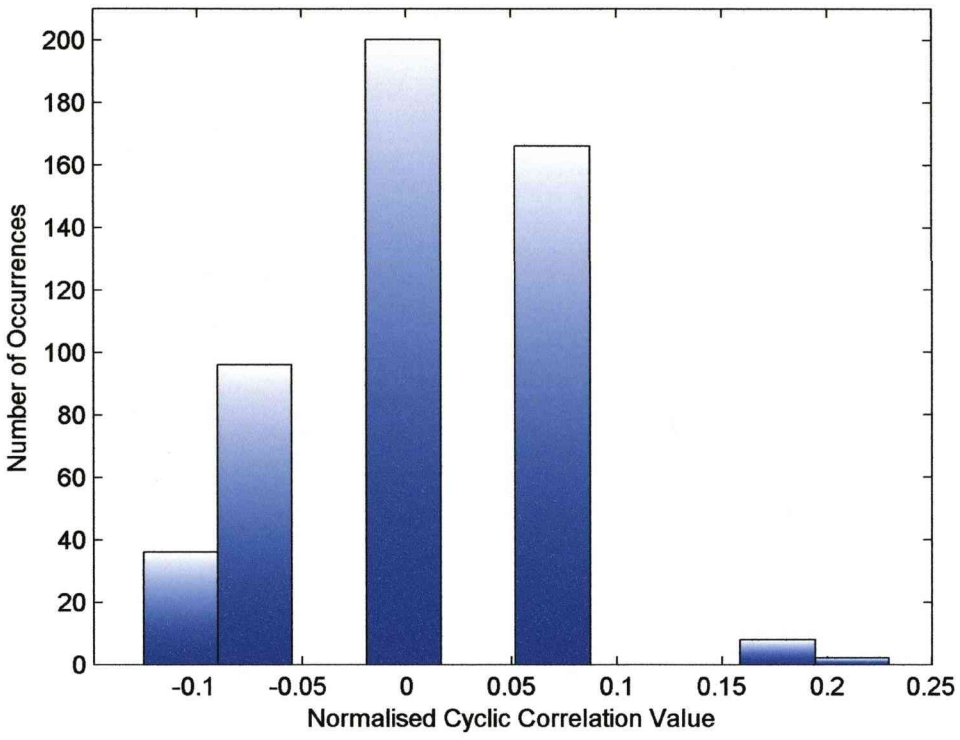

Figure 3.19 - Background value distribution from 2 Sequences using cyclic correlation

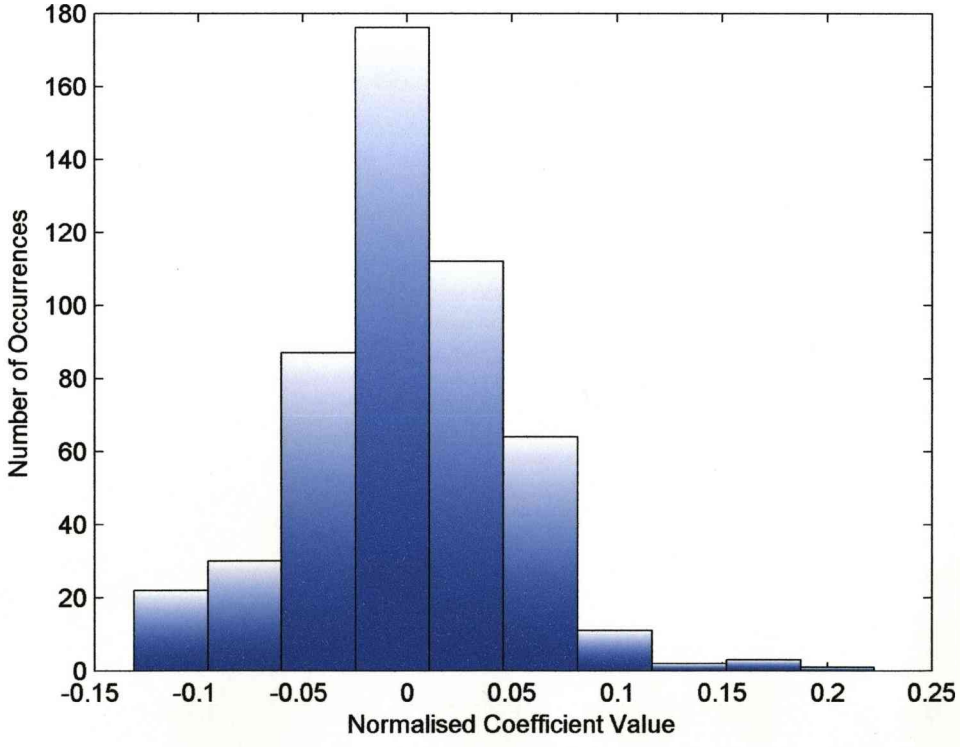

Figure 3.20 - Background value distribution from 2 Sequences using the filter

Correlation in Figure 3.18 gives a strong central near zero mean with an approximate normal distribution spreading to ±0.2. Cyclic correlation in Figure 3.19 has a similar central mean value but within the distribution only specific values occur leading to the isolated distribution graph. The filter in Figure 3.20 is similar in shape to the correlation distribution with a slightly narrower range but a longer small positive tail is present. This shows that there are occasional larger background values with the filter but they are still consistent with the ranges of the other two methods.

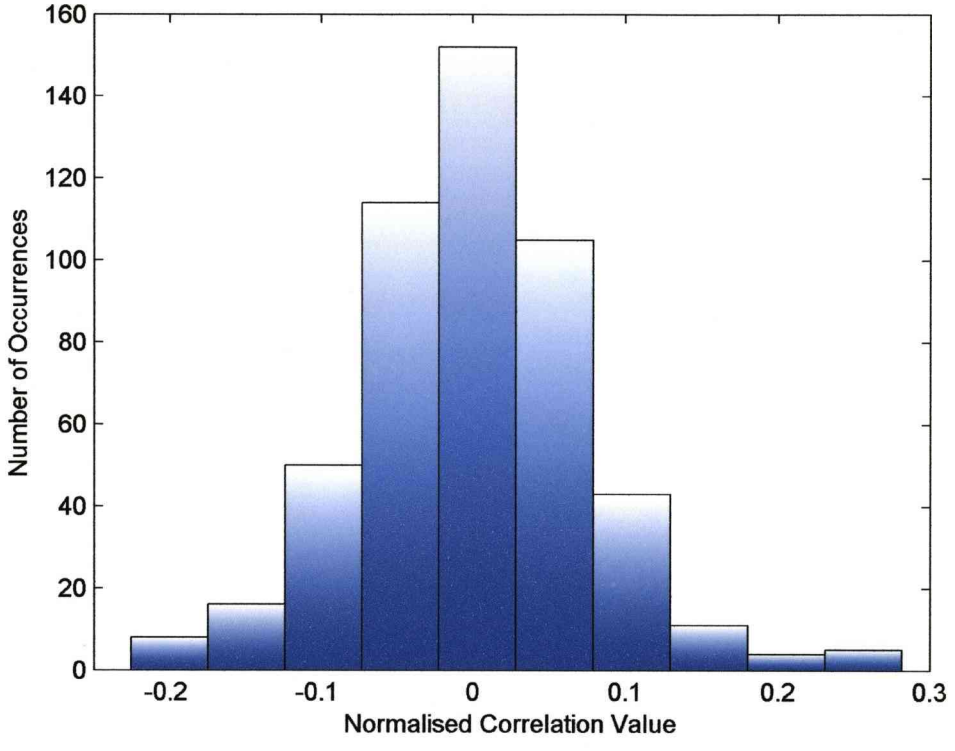

Figure 3.21 - Background value distribution from 4 Sequences using correlation

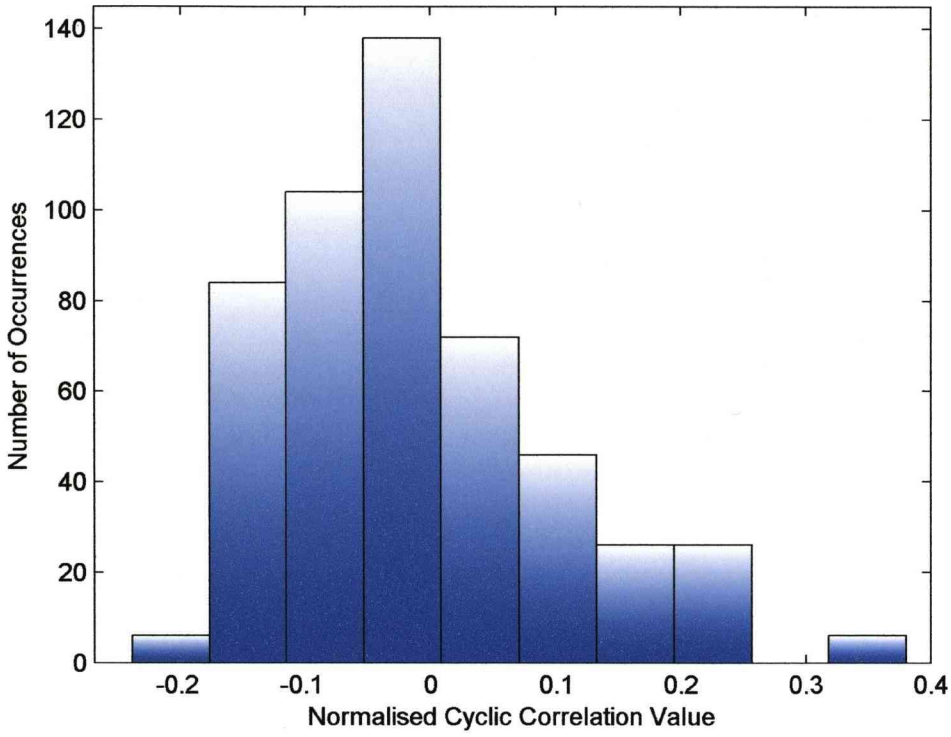

Figure 3.22 - Background value distribution from 4 Sequences using cyclic correlation

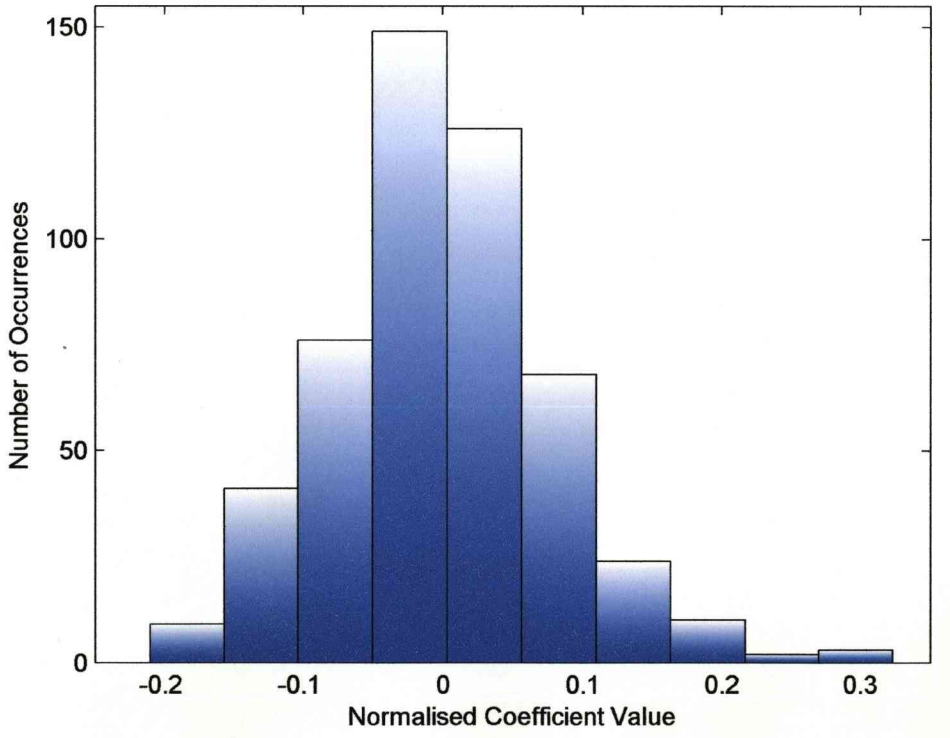

Figure 3.23 - Background value distribution from 4 Sequences using the filter

Both the correlation and filter distributions in Figure 3.21 and 3.23 show similar normal distribution shapes that are slightly wider than for 2 sequences and have a clearly visible small positive tail. Cyclic correlation in Figure 3.22 is skewed towards the negative side showing the differences in distributions with very similar statistical values.

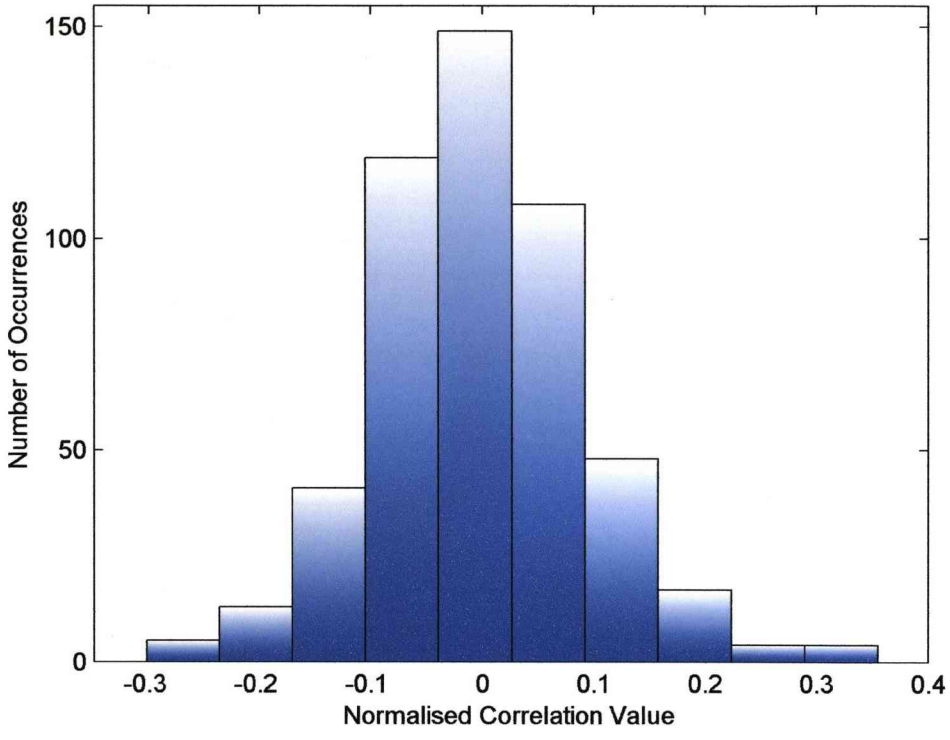

Figure 3.24 - Background value distribution from 6 Sequences using correlation

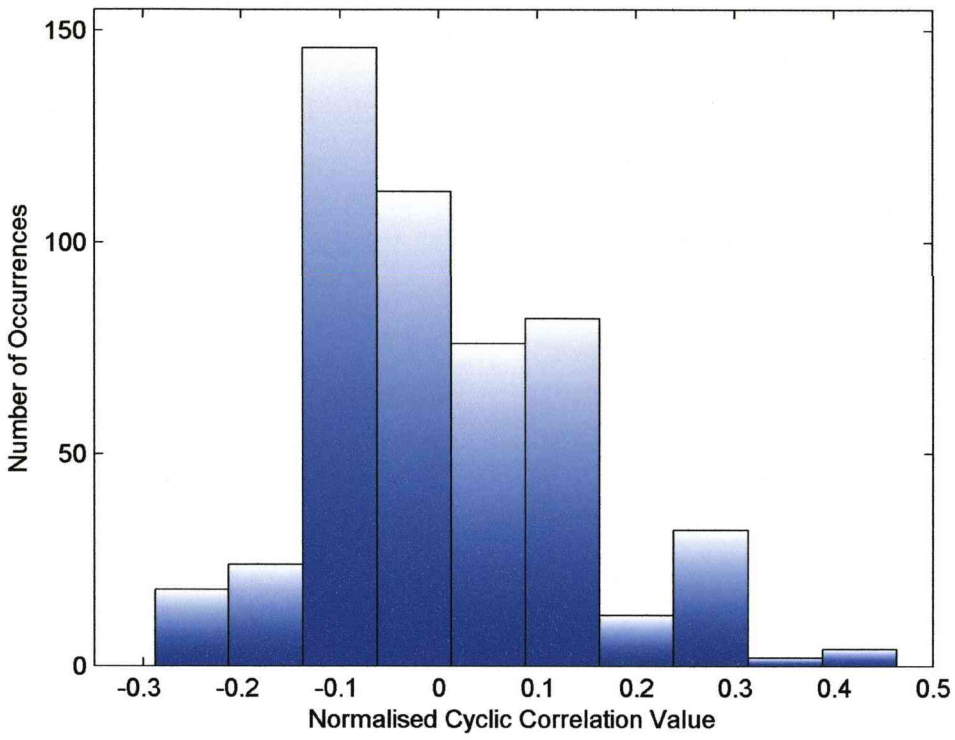

Figure 3.25 - Background value distribution from 6 Sequences using cyclic correlation

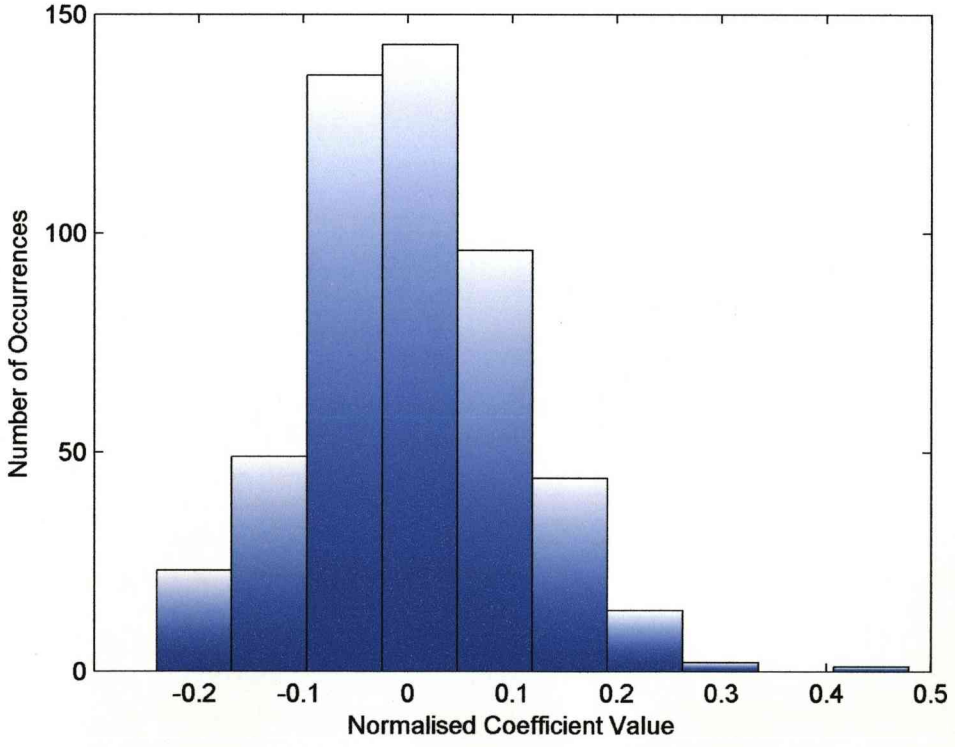

Figure 3.26 - Background value distribution from 6 Sequences using the filter

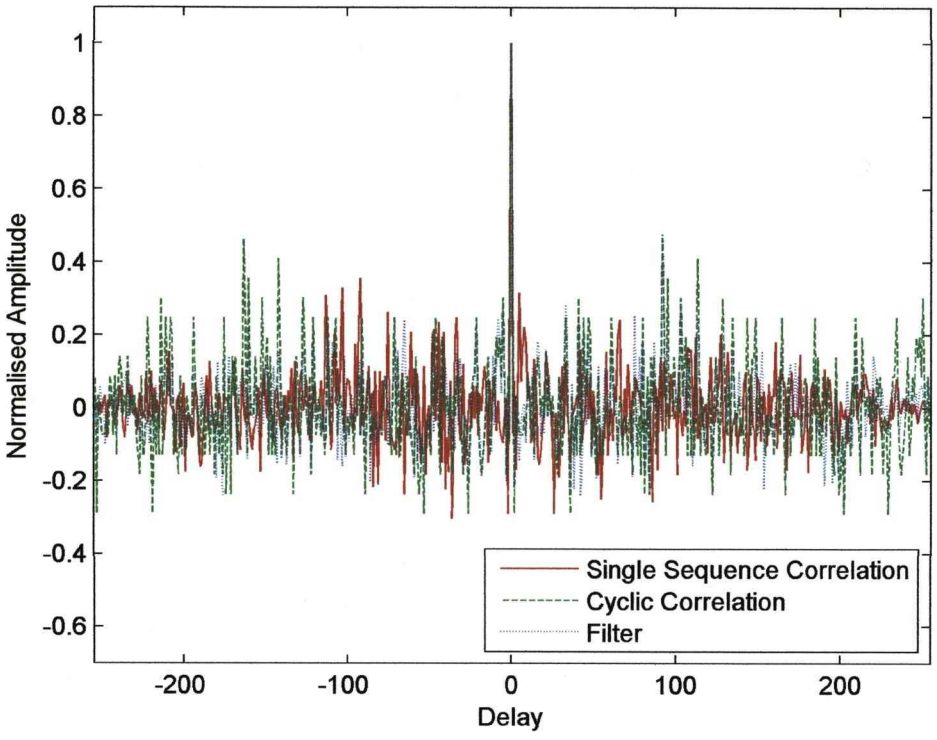

Figure 3.27 - Comparison of single sequence detections from 6 Sequences

The three 6 sequence distributions in Figures 3.24 to 3.26 match quite well with those using 4 sequences showing only an increased spreading. From Figure 3.27, the distributions and statistical values it is clear that single sequences can be quite easily detected from a group of sequences causing only an increase in background values that looks in most cases similar to noise (noise effects are covered later), even in a worst case scenario when all the signals are received simultaneously.

3.8 Missing Symbols

Using a system based purely on the M-sequence (without encoding it in a signal or after a sequence has been recovered) it is useful to consider what happens if a symbol is missing or misidentified. A missing symbol is one in which the correlation between that symbol and any other is zero and a misidentified symbol is when it is actually the

opposite symbol to what it should be. In a cyclic autocorrelation it is easy to work out the effects a single missing or misidentified symbol can cause using the properties of M-sequences described in Chapter 3.1 [46]. Considering these properties a number of simple error relations have been analytically derived and then numerically validated.

3.8.1 Correlation Error at Peak

A correct correlation leads to 2^n-1 agreements over 2^n-1 symbols giving a normalised result of 1. If a number of symbols (b) are missing then the normalised output for this is given by Eq. (3.25) and similarly for misidentified symbols in Eq. (3.26)

$$\frac{2^n-1-b}{2^n-1}=1-\frac{b}{2^n-1} \tag{3.25}$$

$$\frac{2^n-1-2b}{2^n-1}=1-\frac{2b}{2^n-1} \tag{3.26}$$

It can be seen that it translates to one minus the proportion of missing symbols to total symbols or twice the proportion of misidentified symbols to total symbols. This leads to the conclusion that no correct detection is possible if around half the symbols are misidentified.

3.8.2 General Correlation Error Maximums

Looking at the correlation away from the peak gives an idea of how robust the system is so incorrect target detections don't become too large and interfere with the correct detection. Due to the structure of the M-sequence leading to agreements and disagreements during correlation the following worst case values are valid only when b is limited to a maximum of $2^{n-1}-1$ (the total number of agreements). Note that if b

achieves this (unlikely) value then the misidentification will give a false correlation peak. Normalising to the correct detections peak gives worst case maximum and minimum for missing symbols

$$\frac{\pm b-1}{2^n-1-b} \tag{3.27}$$

and misidentified symbols

$$\frac{\pm 2b-1}{2^n-1-2b} \tag{3.28}$$

The maximum is achieved when all the symbol errors are contained in the disagreement symbols and conversely the minimum is when the symbol errors are contained in the agreement symbols. For a number of errors in the received signal it is quite unlikely that they would all simultaneously contribute to an agreement or disagreement and so produce a maximum or minimum although it is of course possible.

3.8.2.1 Single Error

The average correlation error for a single symbol error is as follows. The chances of the error falling on an agreement are given in Eq. (3.29) and disagreement in Eq. (3.30).

$$\frac{2^{n-1}-1}{2^n-1} \tag{3.29}$$

$$\frac{2^{n-1}}{2^n-1} \tag{3.30}$$

Taking the average as the sum of all outcome probabilities multiplied by the value for that outcome gives the average correlation error in Eq. (3.31) and for a single missing

symbol (b=1) in Eq. (3.32) (Where a and d represent agreement and disagreement respectively with P and O standing for probability and outcome result).

$$\text{avg error} = P(error(a))O(a) + P(error(d))O(d) \tag{3.31}$$

$$\begin{aligned}\text{avg error} &= \left(\frac{2^{n-1}-1}{2^n-1}\right)\left(\frac{-\mathrm{b}-1}{2^n-1-\mathrm{b}}\right)+\left(\frac{2^{n-1}}{2^n-1}\right)\left(\frac{\mathrm{b}-1}{2^n-1-b}\right)\\ &= \left(\frac{2^{n-1}-1}{2^n-1}\right)\left(\frac{-2}{2^n-2}\right)+\left(\frac{2^{n-1}}{2^n-1}\right)\left(\frac{0}{2^n-2}\right)\\ &= -\frac{1}{2^n-1}\end{aligned} \tag{3.32}$$

The average result is the same for a misidentified symbol

3.8.2.2 Two Errors

Taking the previous notation but introducing a prefix number to a and d to represent which symbol error it refers to gives the average error for two symbols.

$$\begin{aligned}\text{avg error} = {} & P(error(1\mathrm{a}))P(error(2\mathrm{a}))O(2\mathrm{a})\\ & + P(error(1\mathrm{a}))P(error(2\mathrm{d}))O(1\mathrm{a}2\mathrm{d})\\ & + P(error(1\mathrm{d}))P(error(2\mathrm{a}))O(1\mathrm{d}2\mathrm{a})\\ & + P(error(1\mathrm{d}))P(error(2\mathrm{d}))O(2\mathrm{d})\end{aligned} \tag{3.33}$$

When there are two errors the minimum and maximum correlation values remain as in Eq (3.27) and (3.28) (with b=2), the average is expected to vary because there are now 4 ways to get two errors, two on agreement, two on disagreement and 2 ways to get one of each but as shown in Eq. (3.34) the solution can be simplified down to the same average solution found in Eq. (3.35).

$$
\begin{aligned}
&\left(\frac{2^{n-1}-1}{2^n-1}\right)\left(\frac{2^{n-1}-2}{2^n-2}\right)\left(\frac{-b-1}{2^n-1-b}\right)\\
&\quad+\left(\frac{2^{n-1}-1}{2^n-1}\right)\left(\frac{2^{n-1}}{2^n-2}\right)\left(\frac{-1}{2^n-1-b}\right)\\
&\quad+\left(\frac{2^{n-1}}{2^n-1}\right)\left(\frac{2^{n-1}-1}{2^n-2}\right)\left(\frac{-1}{2^n-1-b}\right)\\
&\quad+\left(\frac{2^{n-1}}{2^n-1}\right)\left(\frac{2^{n-1}-1}{2^n-2}\right)\left(\frac{b-1}{2^n-1-b}\right)\\
&\frac{\left(-2^{2n-2}+2^{n-1}+2^n-2-b2^{2n-2}+b2^{n-1}+b2^n-b2\right)}{\left(2^n-1\right)\left(2^n-2\right)\left(2^n-1-b\right)}\\
&\quad+\frac{2\left(-2^{2n-2}+2^{n-1}\right)}{\left(2^n-1\right)\left(2^n-2\right)\left(2^n-1-b\right)}\\
&\quad+\frac{\left(b2^{2n-2}-b2^{n-1}-2^{2n-2}+2^{n-1}\right)}{\left(2^n-1\right)\left(2^n-2\right)\left(2^n-1-b\right)}\\
&\qquad=\frac{\left(-2^{2n}+2^{n+1}+2^n-2+b2^n-b2\right)}{\left(2^n-1\right)\left(2^n-2\right)\left(2^n-1-b\right)}\\
&=-\frac{1}{2^n-1}
\end{aligned}
\tag{3.34}
$$

From this it is highly probable that in the usable range (small value of *b*) the average error for the cyclic correlation case when symbols are missing or misidentified will always be the solution given by Eq. (3.32) and (3.34). This also indicates that the maximum and minimum errors would be the limiting factor on the total number of permissible missing or misidentified symbols while retaining useful operation. In all cases a misidentified symbol is worse than a missing symbol due to the larger minimum and maximum errors that it would produce.

3.8.3 Numerical Results

A short M-sequence of 15 symbols with symbol errors (missing) in the received signal is compared across the cyclic correlation, correlation and filter methods. Up to four symbol errors are looked at with every unique combination of location within the M-sequence used. From these the normalised mean, maximum and minimum values are calculated for the no detection results (with reference and normalisation to the peak detection) and distribution of values examined. The mean value for 15 symbols as predicted by theory would be -1/15 and this is the value for the cyclic correlation for up to the four symbol errors tested. The correlation and filter methods varied slightly depending on the number of missing symbols and also showed a smaller mean that tended towards the same value also over the range of missing symbols.

Table 3.7 – Comparison of mean values for missing symbols

	Number of Symbols Missing			
	1	**2**	**3**	**4**
Cyclic Correlation	-0.0667	-0.0667	-0.0667	-0.0667
Correlation	-0.0333	-0.0333	-0.0333	-0.0333
Filter	-0.0302	-0.0296	-0.0290	-0.0285

The minimum and maximum values are shown in Figure 3.28 and 3.29. They show that with regard to these values the cyclic correlation shows the lowest values with the filter slightly better than the correlation. It does show however that a number of missing symbols does degrade the system significantly. At only four missing symbols out of a total of 15 the background clutter away from a detection peak value of 1 reaches -0.45 and 0.275 in the best case scenario with cyclic correlation with the filter and correlation performing worse. Again these worst case values match the theoretical values from the previous sections for the cyclic correlation case.

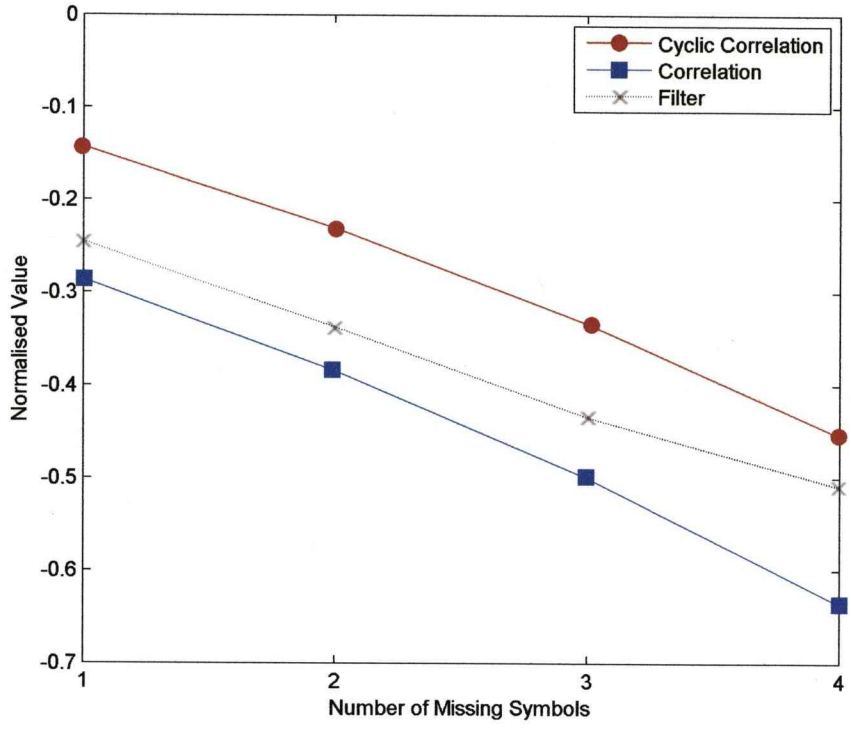

Figure 3.28 - Missing symbols minimum normalised value

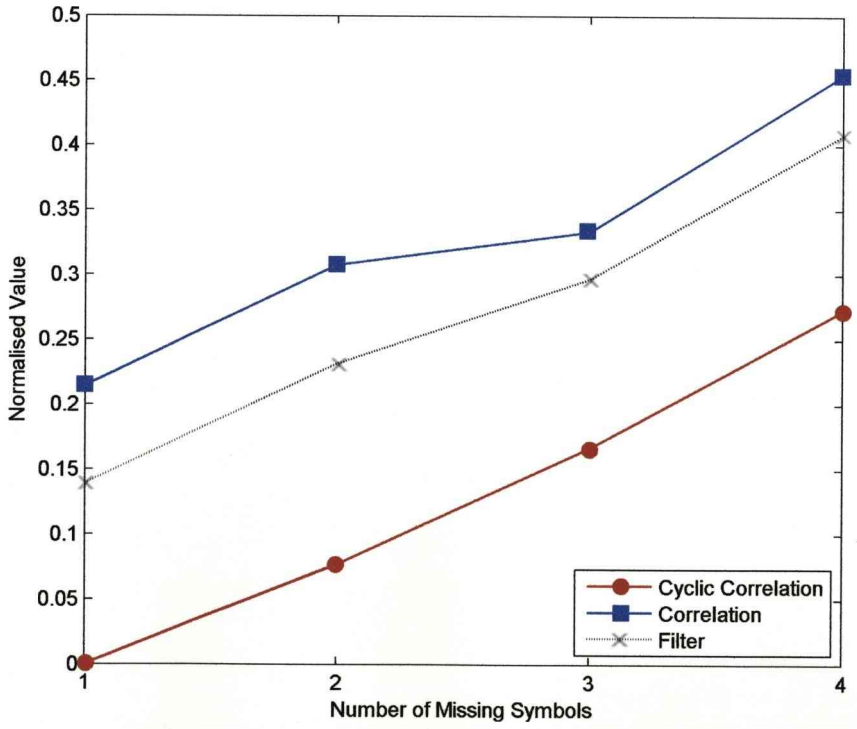

Figure 3.29 - Missing symbols maximum normalised value

A further note of interest is the distribution of values (excluding detection peak), the worst case maximum and minimum values are rare within a single cyclic correlation and they may not appear at all. A large number of missing symbols is generally bad but within a long enough M-sequence the chances of a worst case maximum or minimum appearing become much smaller. The distributions show a decreased proportion of number of maximum and minimum values when moving from 3 missing symbols in Figure 3.30 to 4 missing symbols in Figure 3.31.

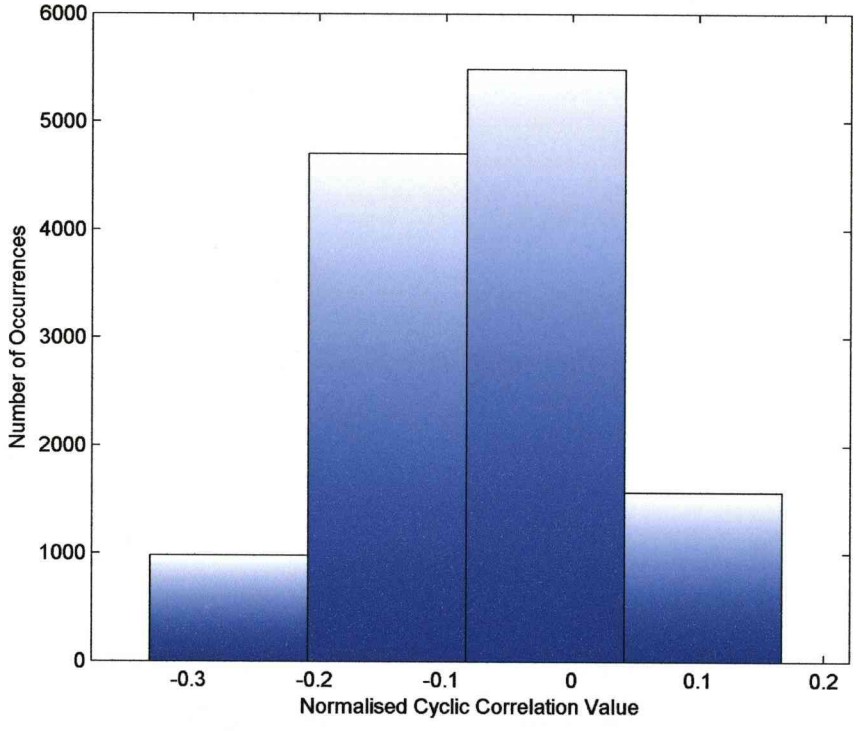

Figure 3.30 - Value distribution for 3 missing symbols with cyclic correlation

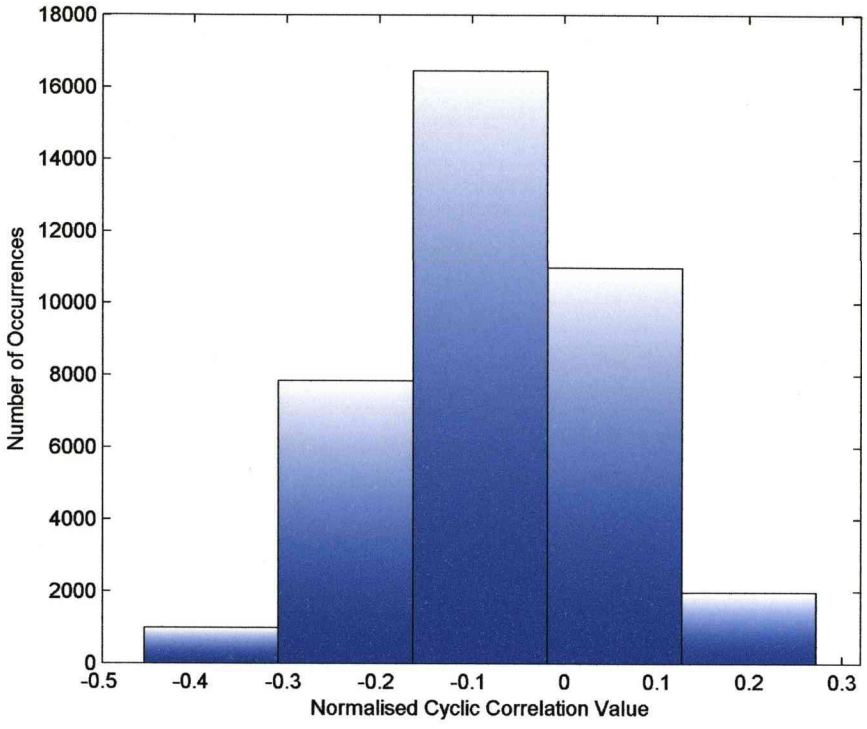

Figure 3.31 - Value distribution for 4 missing symbols with cyclic correlation

Although the correlation and filter results are strictly worse they also show a decrease in frequency of occurrence when compared to the total for their worst case maximum and minimum values when increasing the number of missing symbols.

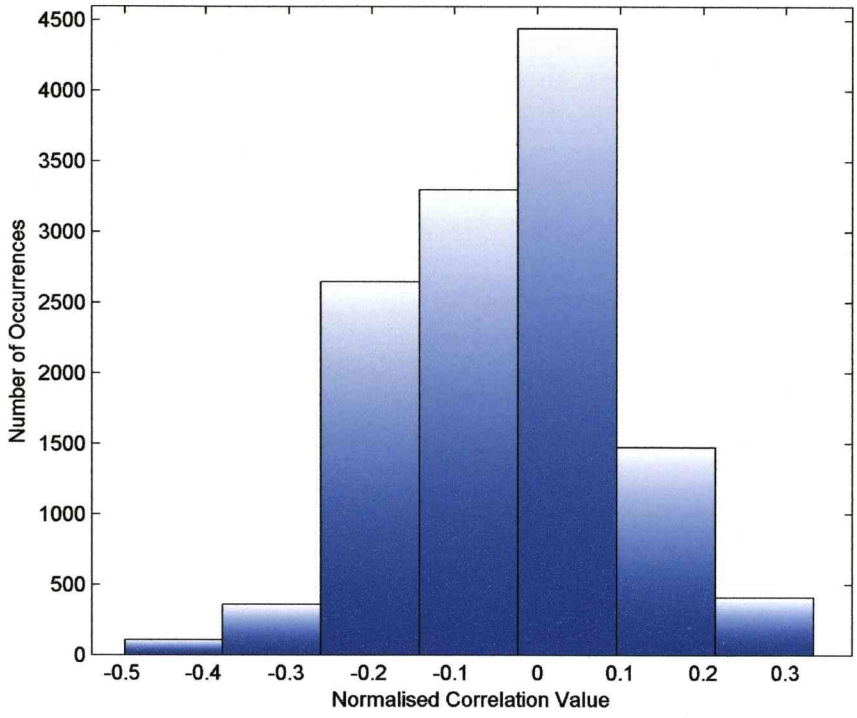

Figure 3.32 - Value distribution for 3 missing symbols with correlation

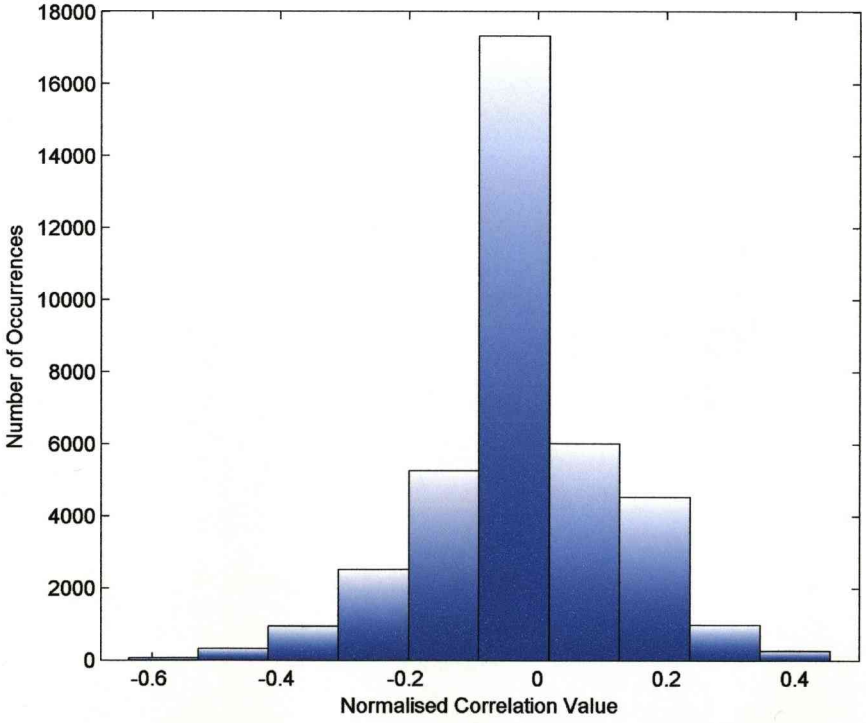

Figure 3.33 - Value distribution for 4 missing symbols with correlation

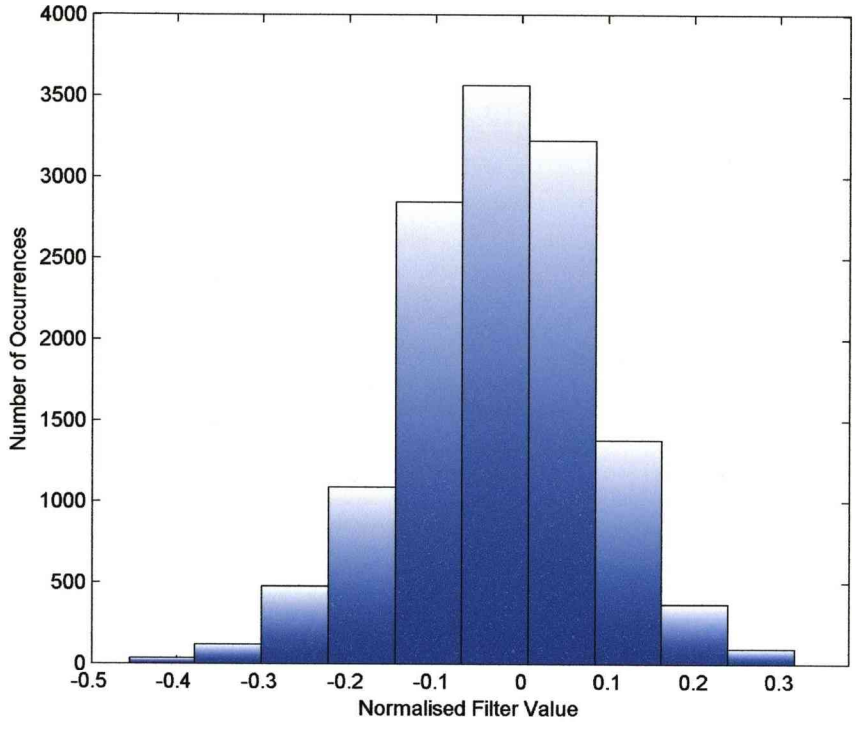

Figure 3.34 - Value distribution for 3 missing symbols with filter

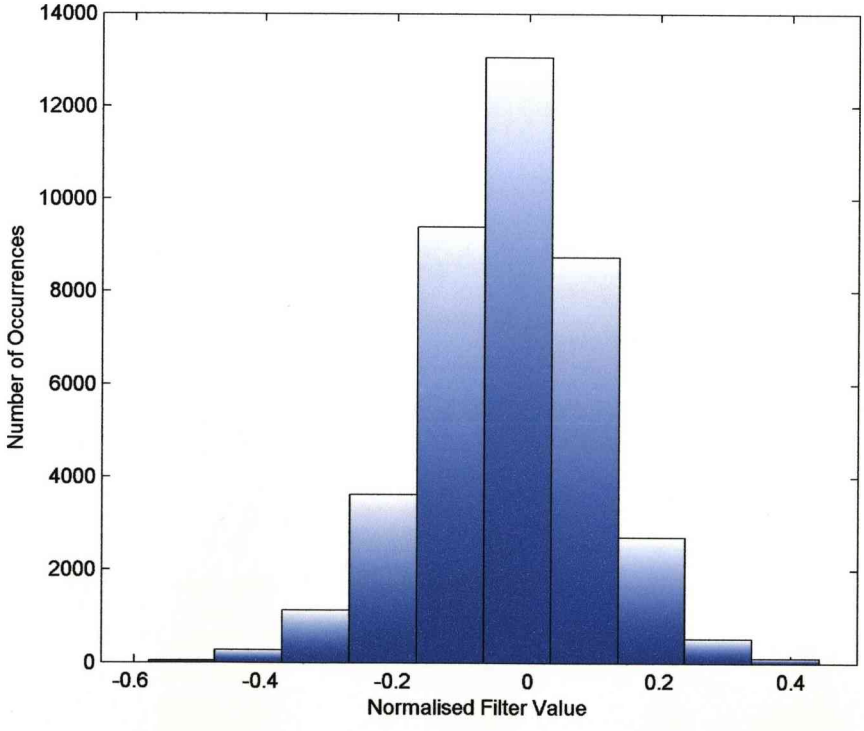

Figure 3.35 - Value distribution for 4 missing symbols with filter

For long M-sequences the distribution of values obtained for an increasing number of missing symbols looks like it will tend to a Gaussian distribution. This should also hold true within a single correlation or filter operation.

3.9 Noise and Samples per Symbol

Noise is a problem in any real system, the following tests give an idea of performance in the presence of noise for several possible signals of varying M-sequence lengths and samples (bits) per symbol. M-sequence lengths of 63, 127 and 255 are used along with 1, 2 and 4 bits per symbol. Random white Gaussian noise is generated with a mean of 0 and standard deviation of 0.05, 0.1 and 0.2.

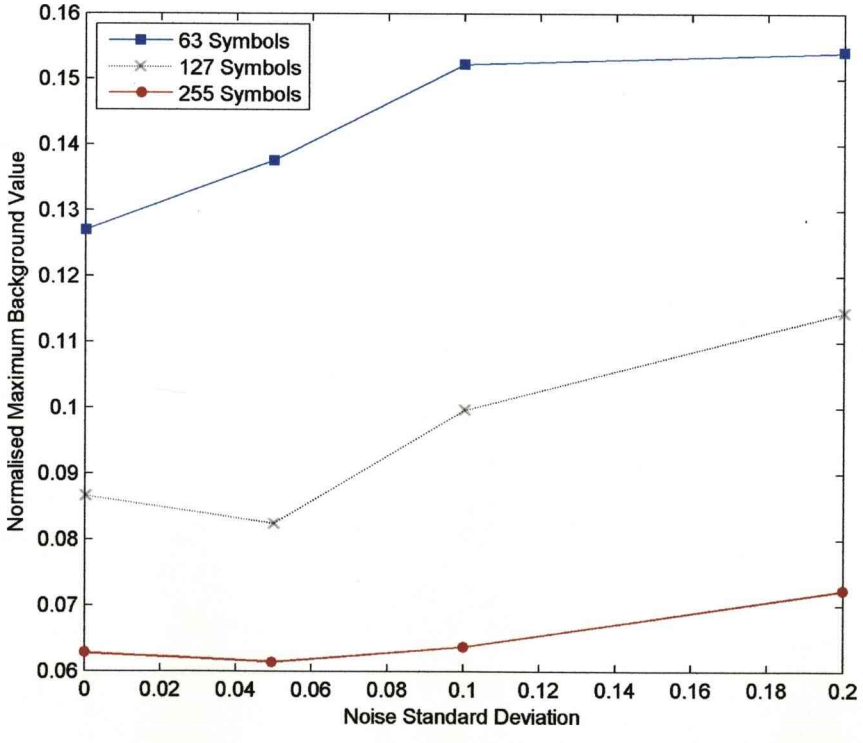

Figure 3.36 - Correlation - Maximum background value against noise

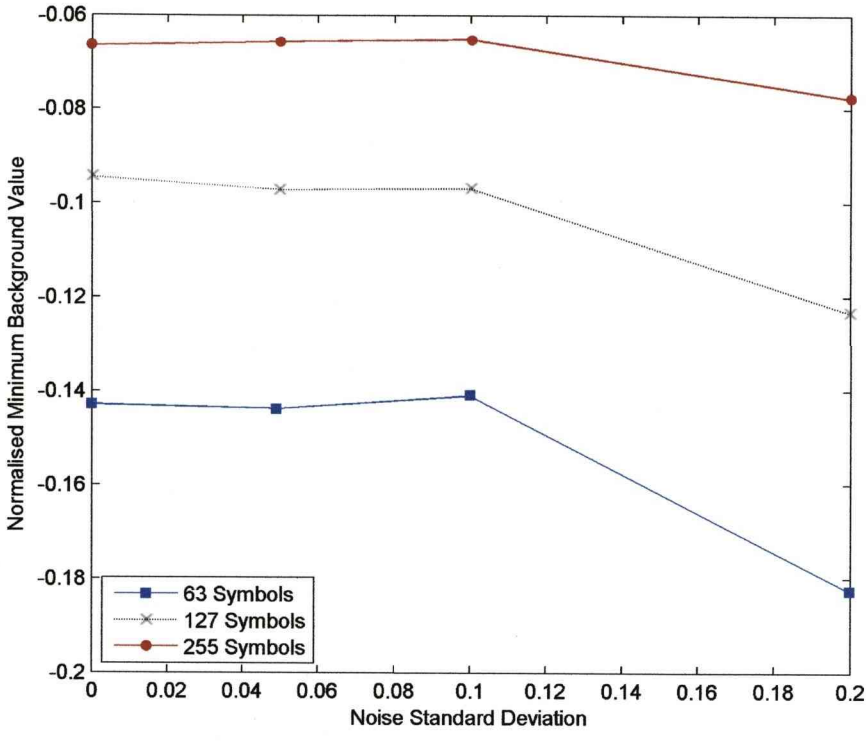

Figure 3.37 - Correlation - Minimum background value against noise

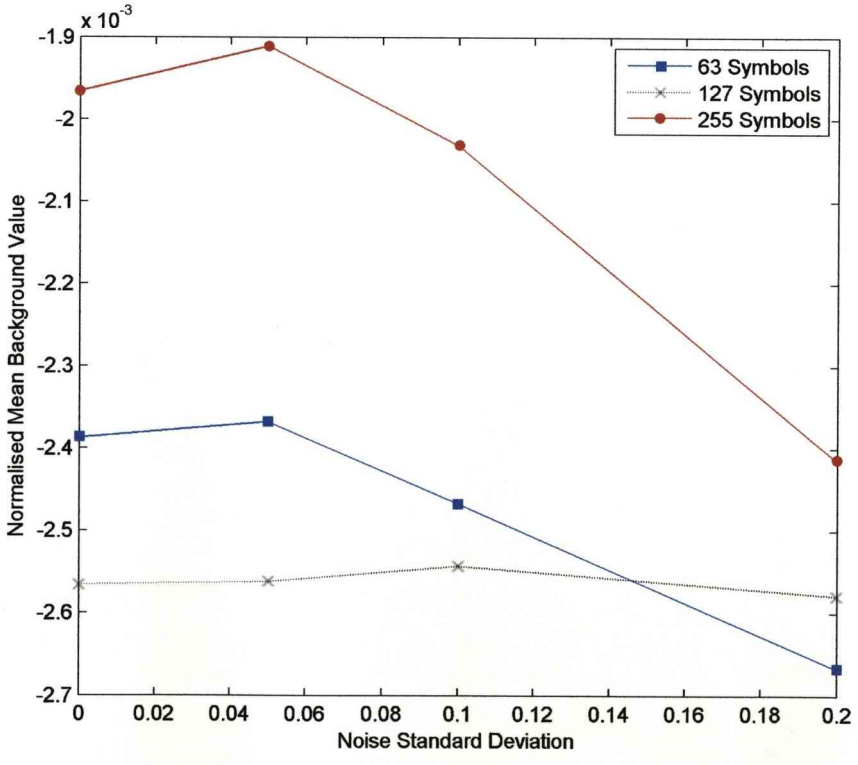

Figure 3.38 - Correlation - Mean background value against noise

The correlation versions in Figures 3.36 to 3.38 show the expected result (with one anomaly in Figure 3.38), with the longer sequences with more symbols having overall smaller minimum, maximum and mean over the shorter sequences but getting progressively worse with increasing noise levels.

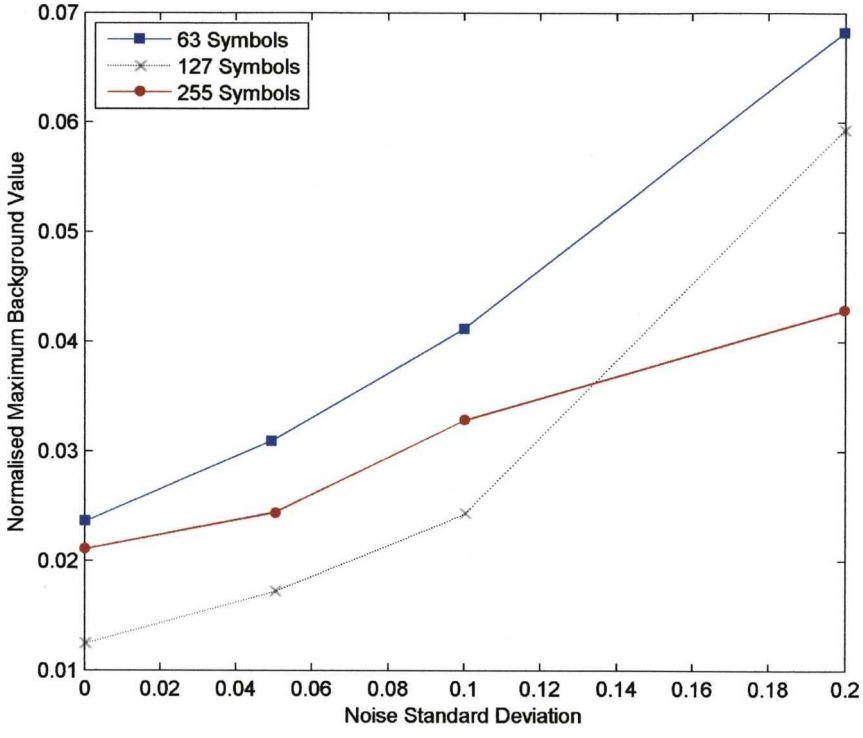

Figure 3.39 - Filter - Maximum background value against noise

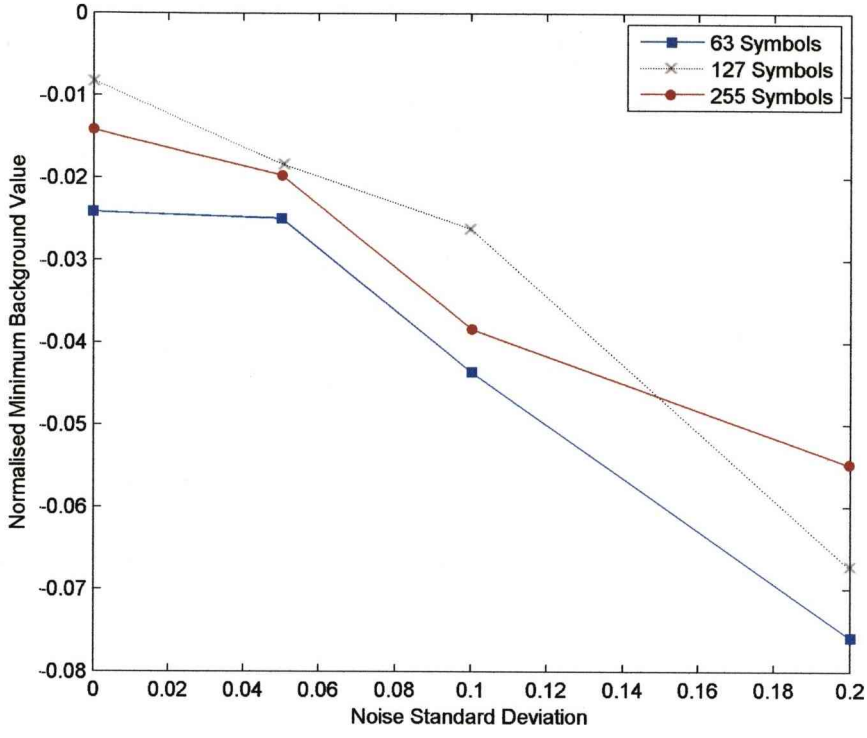

Figure 3.40 - Filter - Minimum background value against noise

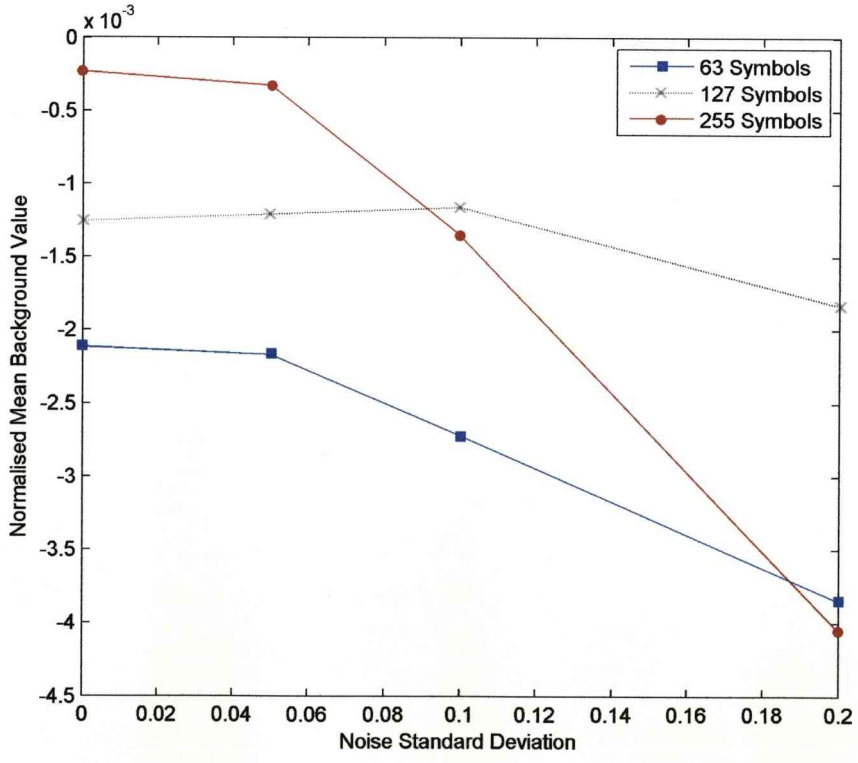

Figure 3.41 - Filter - Mean background value against noise

The filter results are very different with a much a tighter grouping and the longer sequence not necessarily being the best in all cases at any noise level. Generally the filter gives better overall results than correlation with smaller minimum, maximum and mean values for the three sequence lengths. It can be noted though that the filter values are increasing more rapidly with higher noise than the correlation so with a high enough noise value the filter will be surpassed in performance by correlation. One question that arises is whether the increased performance of both methods is due to the sequences themselves or the actual physical length in data points. Applying four and two data points per symbol for 63 and 127 symbol sequences brings the total lengths of those two sequences up to approximately that of the 255 symbol sequence. The expectation is that sequence length rather than total length is the more important parameter.

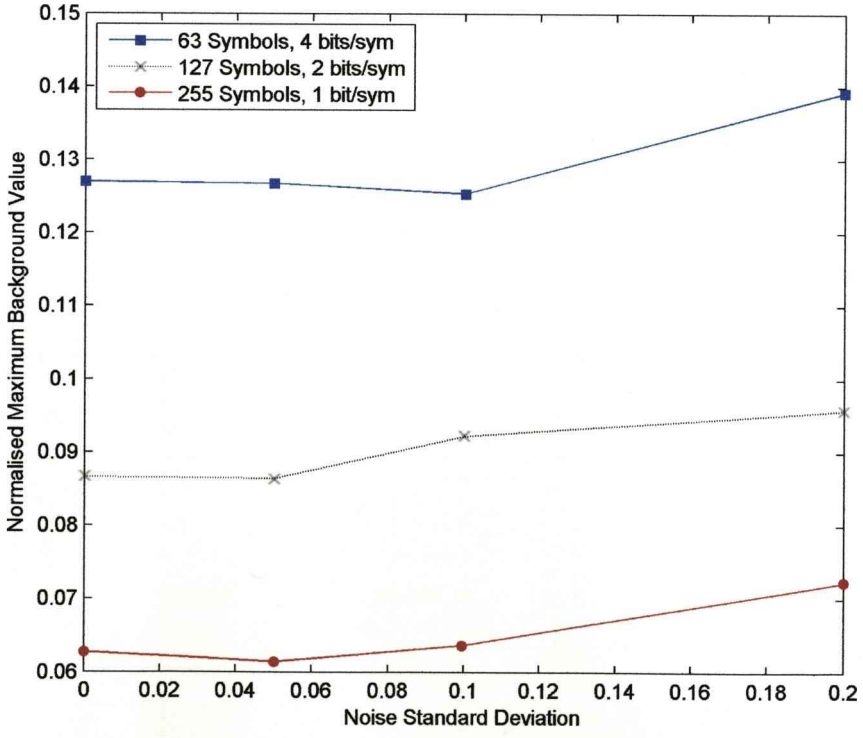

Figure 3.42 - Correlation - Maximum background value against noise

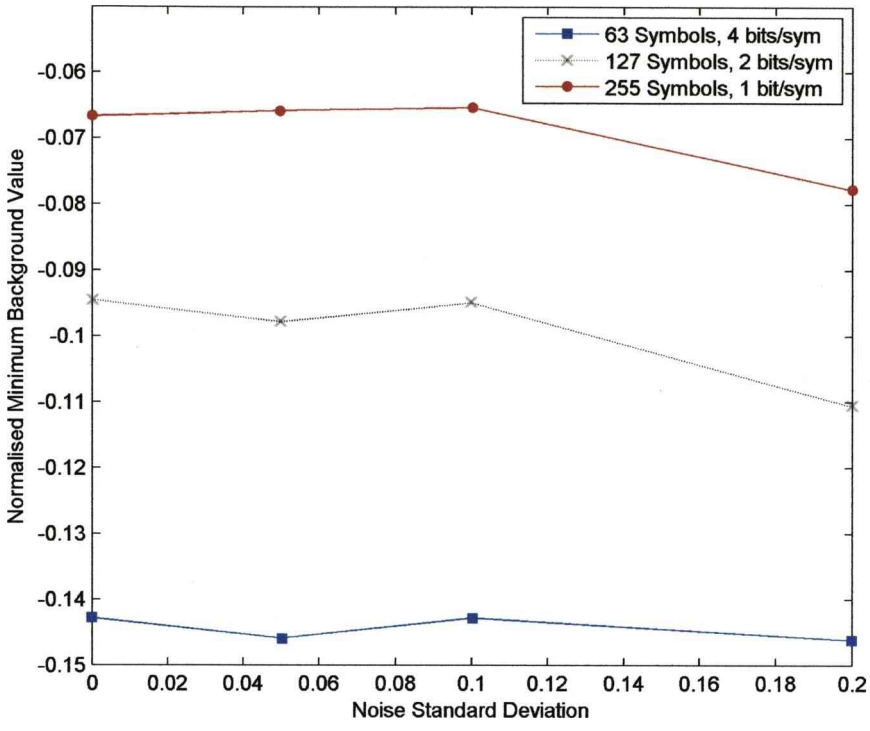

Figure 3.43 - Correlation - Minimum background value against noise

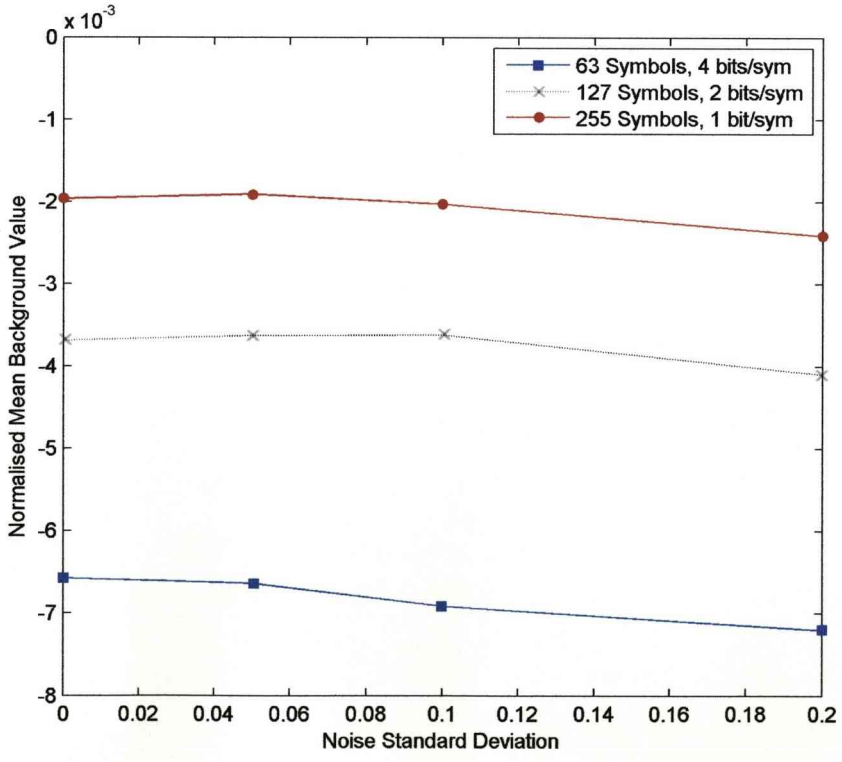

Figure 3.44 - Correlation - Mean background value against noise

The correlation results in Figures 3.42 to 3.44 are broadly similar in maximum and minimum values with a significant degradation in mean for longer individual symbols. So for correlation the longer symbols (and hence lower frequency) does not alter the maximum and minimum values too much but does cause the mean value to become worse. The properties of the M-sequence indicate that the no correlation condition has a small negative value so the degradation of the mean is expected, that small negative value multiplied by the number of bits per symbol giving an approximate value of the mean.

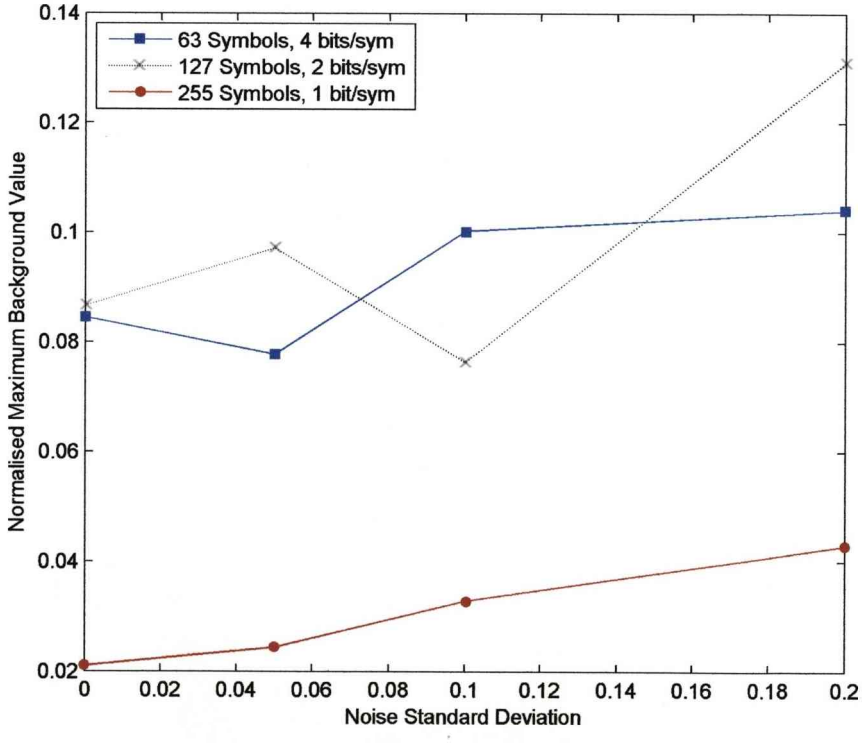

Figure 3.45 - Filter - Maximum background value against noise

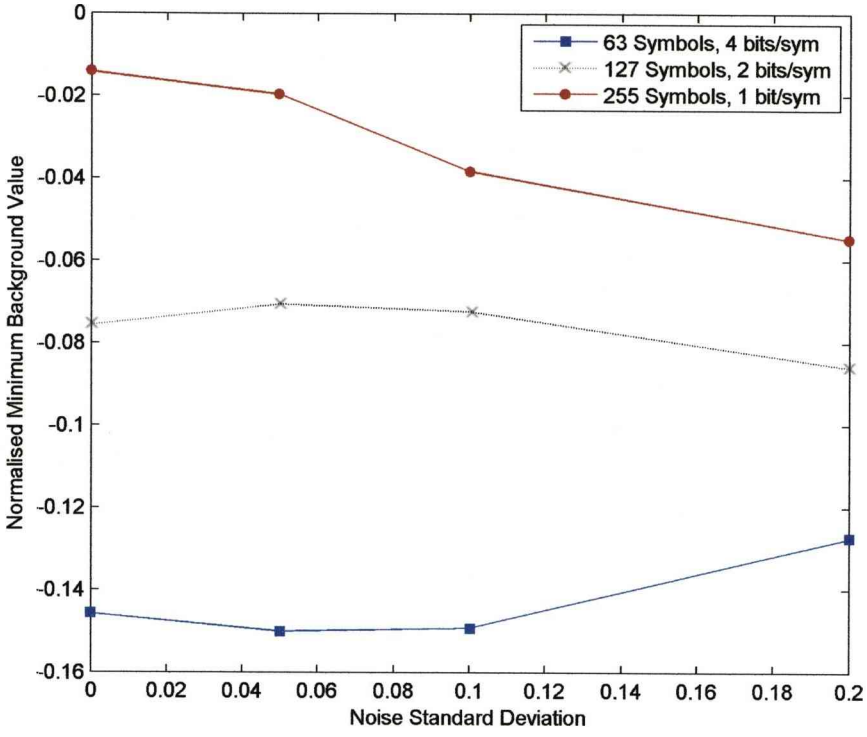

Figure 3.46 - Filter - Minimum background value against noise

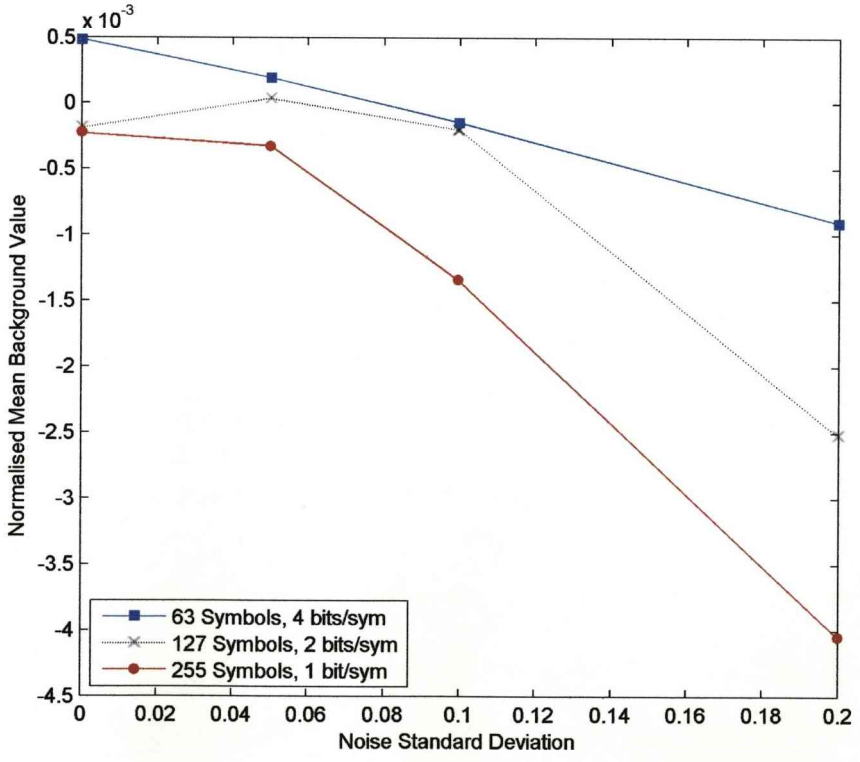

Figure 3.47 - Filter - Mean background value against noise

The filter results again are different with a degradation in maximum and minimum values for longer symbols while showing an improved lower mean value. Where the correlation results are broadly comparable between multiple bits per symbol in maximum and minimum values the filter is not, producing worse results with increasing number of bits per symbol. Another feature of note is the plotted detection results for multiple bits per symbol. Figure 3.48 shows the comparison for correlation where the detection is spread over several data points for 4 bits per symbol when compared to 1 bit per symbol. This fits in with what was observed in an earlier section of this chapter for correlation, in this case with 4 bits per symbol the detection is 7 bits wide. That is clearly not the case in Figure 3.49 which shows a single point detection for the filter but it does have significant ringing around it.

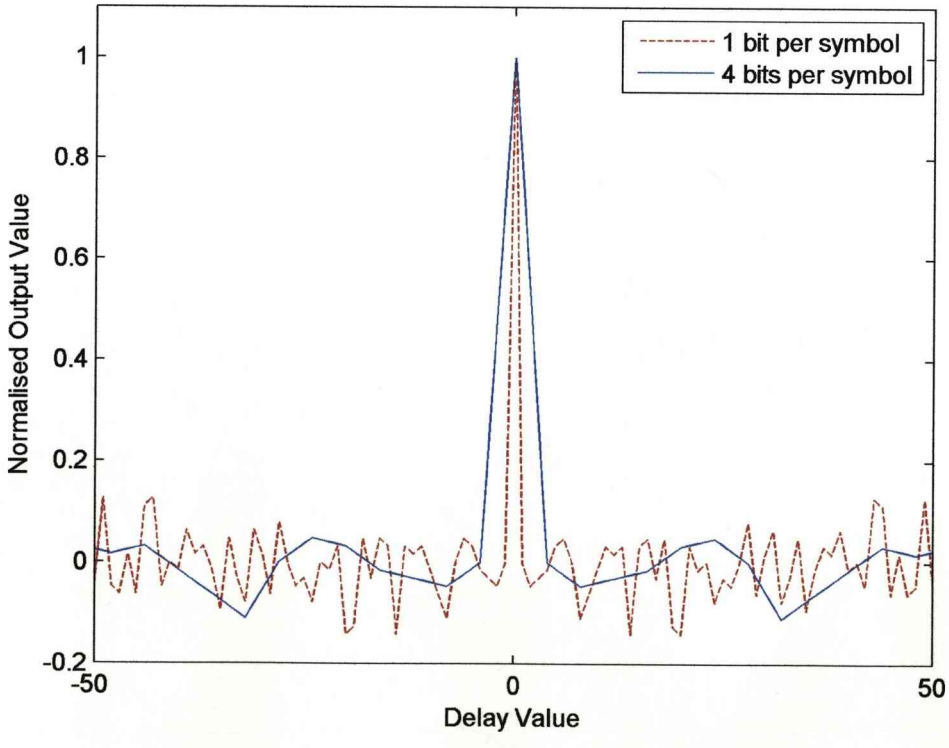

Figure 3.48 - Detection result with correlation

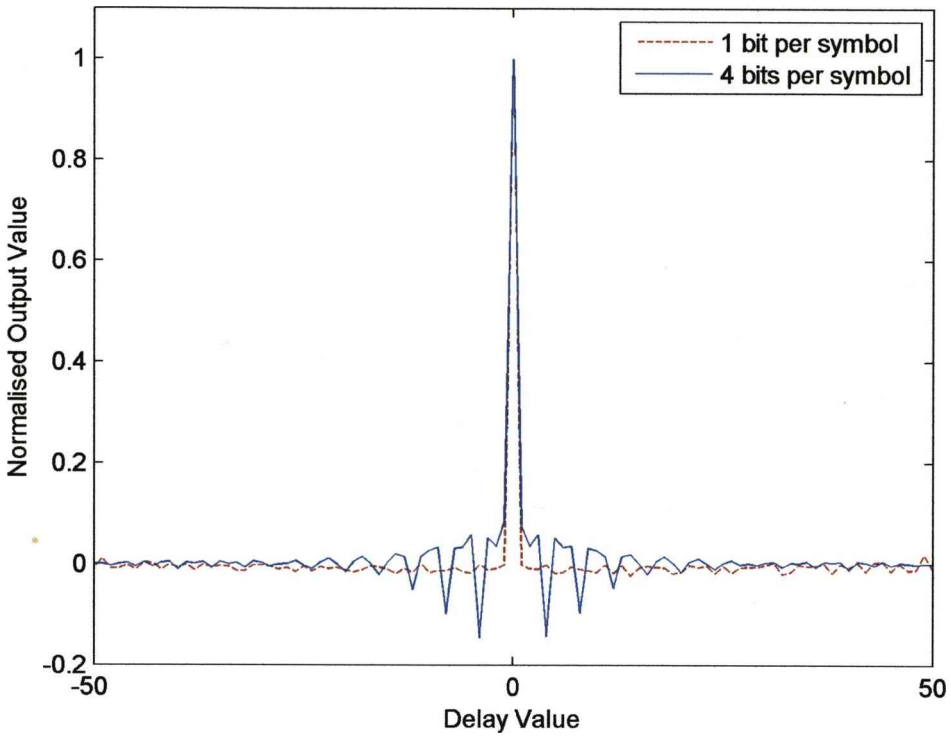

Figure 3.49 - Detection result with filter

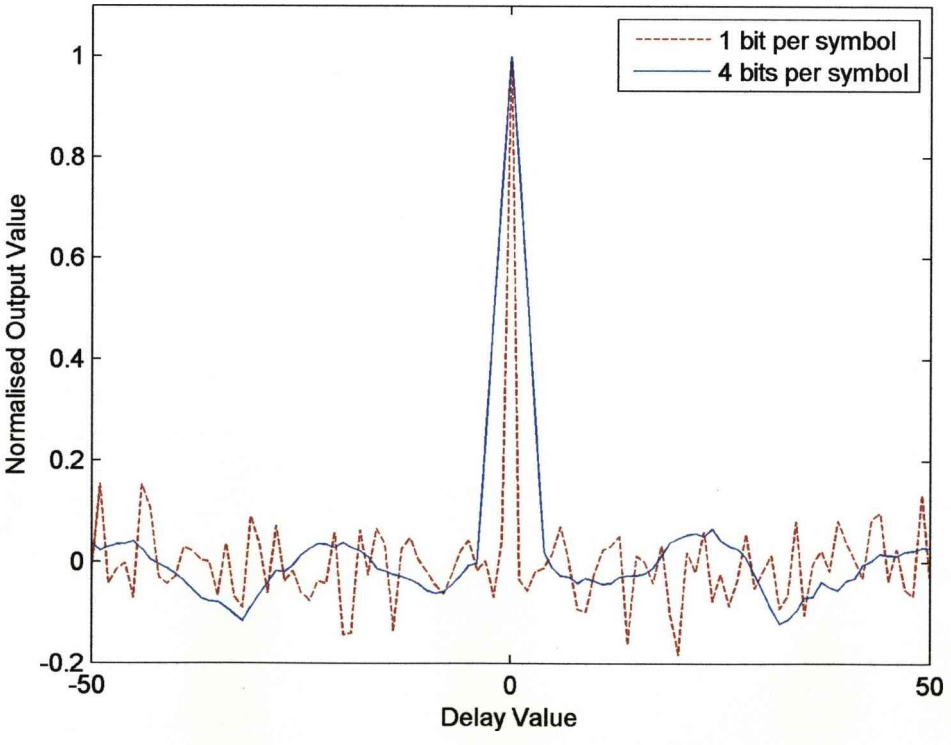

Figure 3.50 - Noisy detection result with correlation

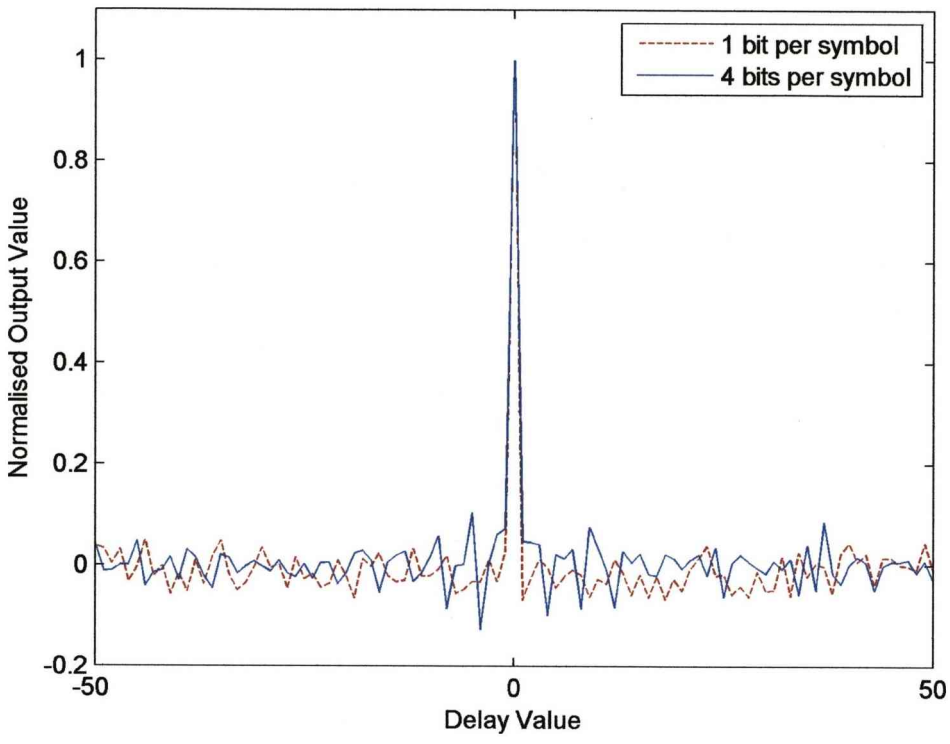

Figure 3.51 - Noisy detection result with filter

When noise (0.2 STD) is included the correlation shows little difference in Figure 3.50 but the filter in Figure 3.51 does show some change with the 1 bit per symbol case showing noticeable degradation.

3.10 Pulse Compression Comparison

Pulse compression is a technique that allows for the transmission of a low peak-power, long-duration coded pulse and attain the fine range resolution and improved detection performance of a short duration, high peak-power pulse system [49]. It is a form of signal modulation using a wide bandwidth. The most common use for the technique is in radar applications including GPR. Range resolution is a measure of the minimum separation between two or more separate targets that exist along the same axis but at different ranges that can still be distinguished. At smaller separations the

separate targets merge into one. Generally the frequency modulation method using chirp signals is used but a phase modulated method is used in some applications (weather radar [49]). In principle they should offer similar results but other considerations such as hardware availability dictate use. The range resolution of a signal using pulse compression is similar to a single pulse of the same bandwidth.

3.10.1 Comparison – Chirp vs. Phase

A pulse of fixed length is used to compare a frequency modulated signal (chirp) and a phase modulated signal. The chirp signal is a frequency modulated sine wave where the linear frequency sweep is from 2.5 to 7.5GHz (it's a down chirp so actually it starts at the higher frequency and is reduced) over a pulse length of 5ns. The phase modulated signal is encoded using binary phase shift keying (BPSK) with the PN sequence consisting of 25 symbols. The carrier and symbol rates are 5GHz leading to an approximate bandwidth of 2.5 to 7.5GHz. The total pulse lengths as well as bandwidths are approximately equal in order to give a good comparison. In addition the signals are bandpass filtered with the pass band from 2.5 to 7.5GHz. A comparison between the filtered and unfiltered signals are shown in Figure 3.52 for the chirp signal and Figure 3.53 for the PN signal.

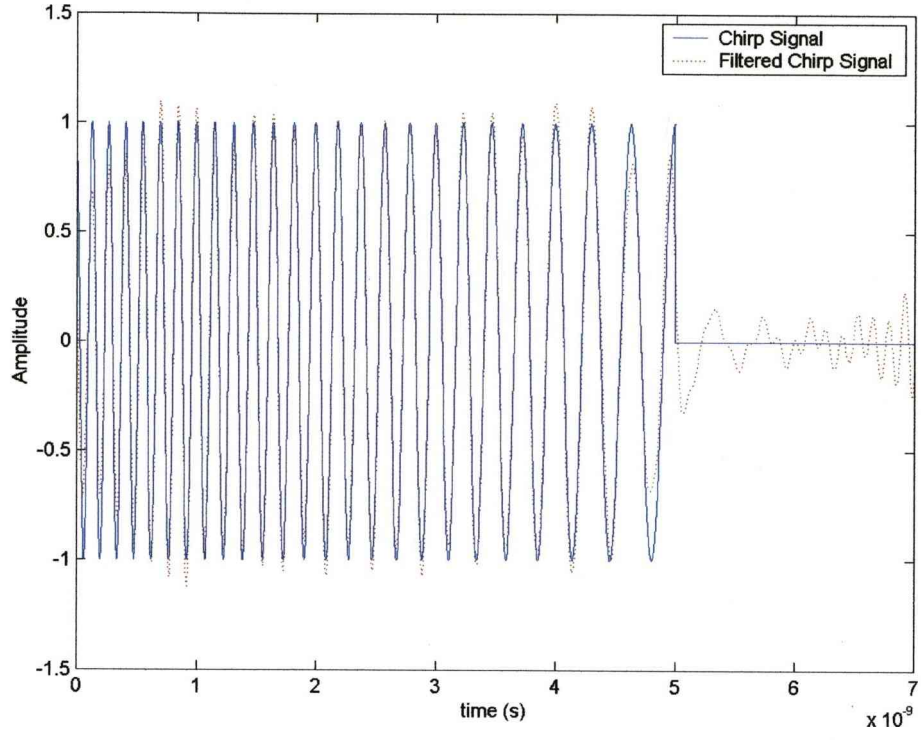

Figure 3.52 - Delayed chirp waveform

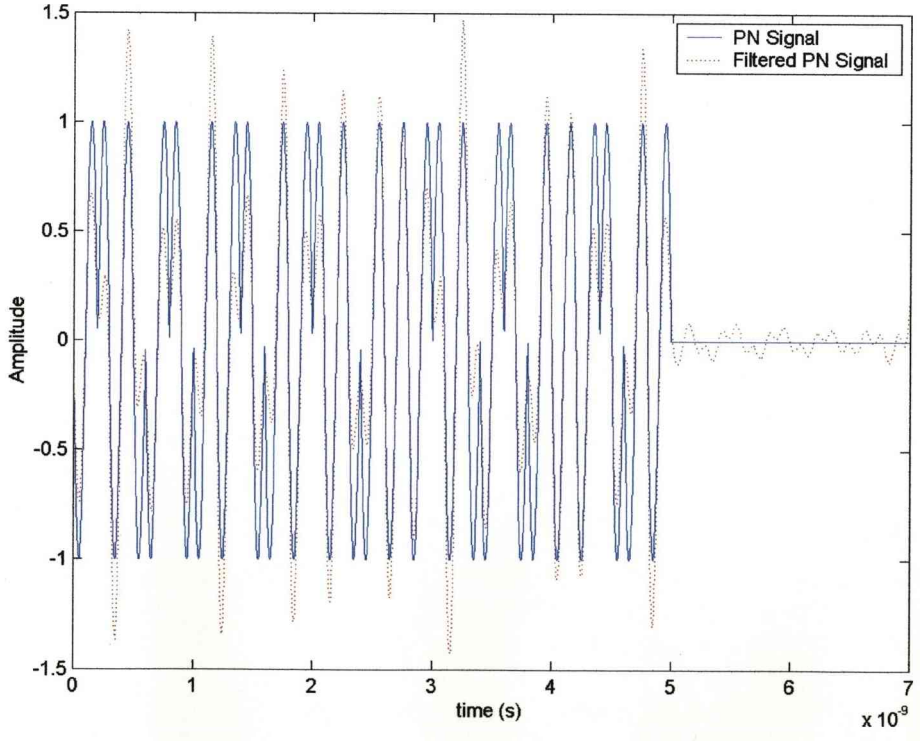

Figure 3.53 - Delayed PN waveform

Signal demodulation for the chirp signal consists of a time domain convolution of the delayed signal with a time reversed reference signal. The phase method uses a matched filter demodulation as well as a cross-correlation and the filter method (latter two use a reference signal). An additional phase modulated signal consists of a direct PN sequence (also bandpass filtered). For the chirp pulse a time reversed reference signal is convolved with it while the PN waveform is correlated with a reference signal. The delay value prediction is given by the maximum output of these processes. Firstly the pulses are subject to a fixed delay of 0.88ns.

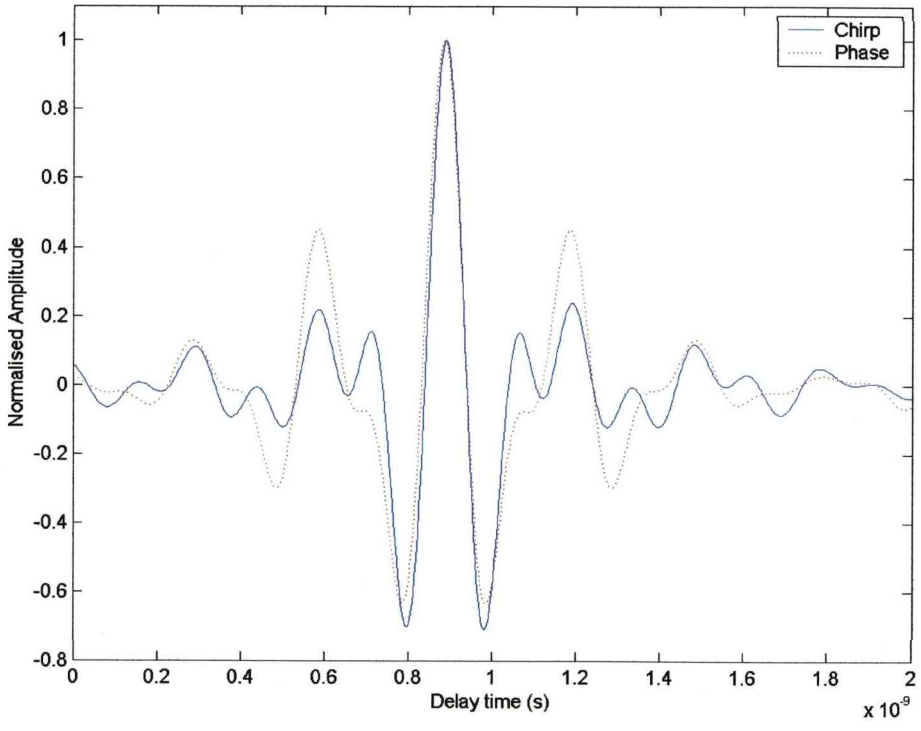

Figure 3.54 - Predicted delay value comparison

Both signals and processing methods give similar results as indicated in Figure 3.54. The chirp has slightly lower negative first side lobes but is overall slightly quicker to decay. Both give the correct delay of 0.88ns as a maximum and the delay prediction is narrow compared to the pulse length. Comparisons between the phase signals will be

examined in the next section but a quick comparison between the filter and chirp signals would be useful.

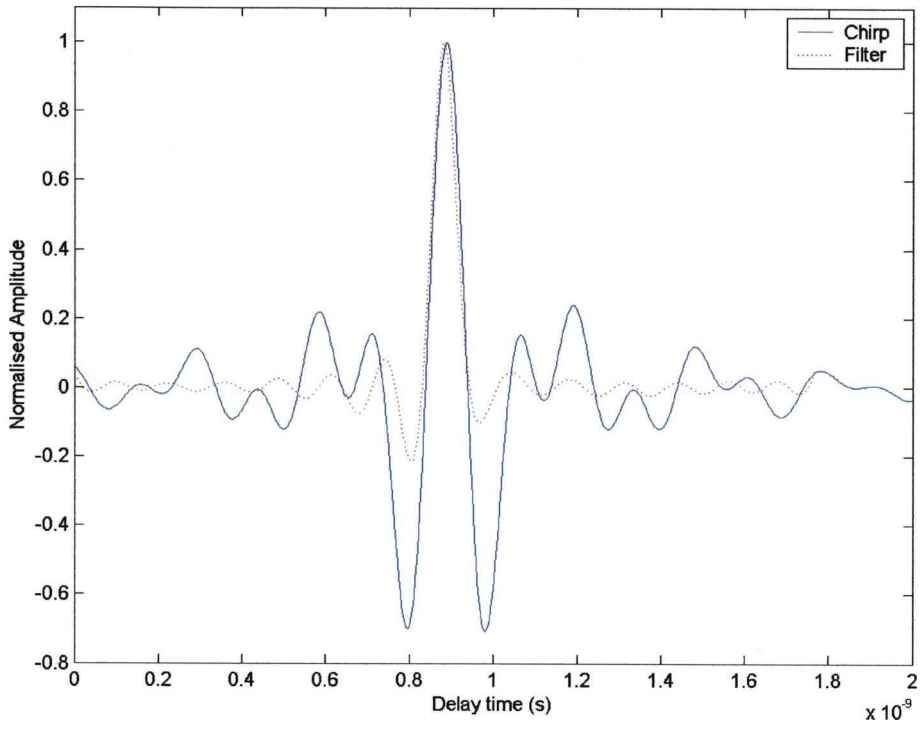

Figure 3.55 - Predicted delay value comparison

The filter gives lower ripples, particularly for the first sidelobes making it the better choice in this case as shown in Figure 3.55.

3.10.1.1 Phase Signal Variation

Different phase signals will produce differently shaped delay prediction graphs, hence the use of 3 slightly different signals. The matched filter does not give a delay output like the other methods; it uses a direct time output. So if the end of the pulse is at 1.5ns the filter produces its indicative output at that time. In order to match the graphs up, the matched filter graphs are shifted by the pulse length to give the correct delay.

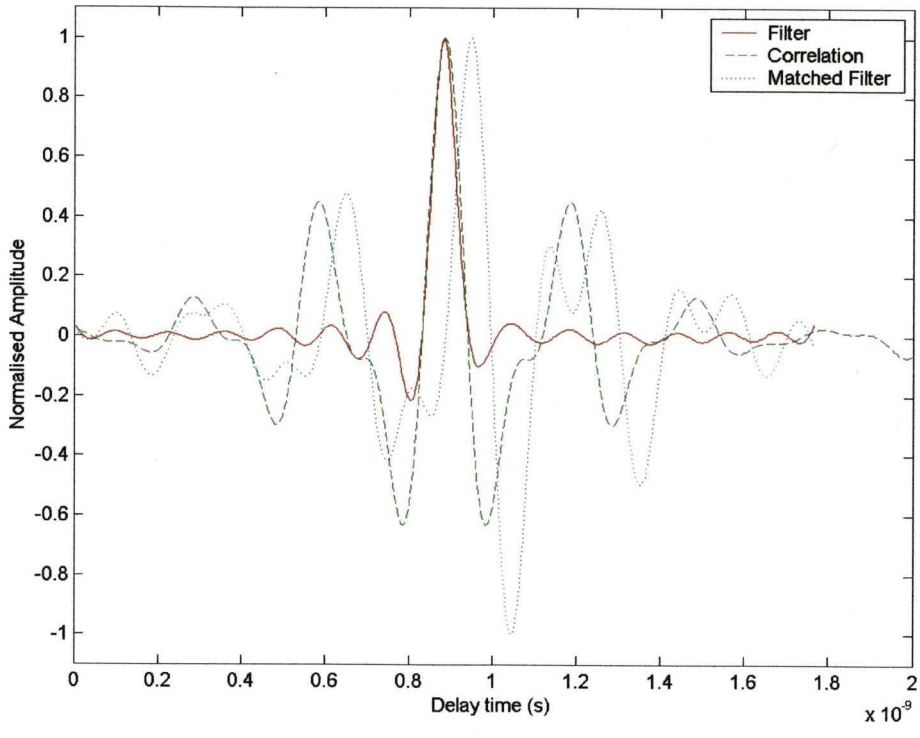

Figure 3.56 - Delay Prediction using phase modulation – 5GHz carrier

Using a 5GHz carrier in Figure 3.56 means each symbol is made up of a single period of the sine wave with appropriate phase. The matched filter produces a single copy of that symbol at the start of the predicted delay point. The correlation output gives a maximum output at the correct delay. The filter gives an output similar to the correlation at the correct delay and overall has lower sidelobes making it the best of the phase methods.

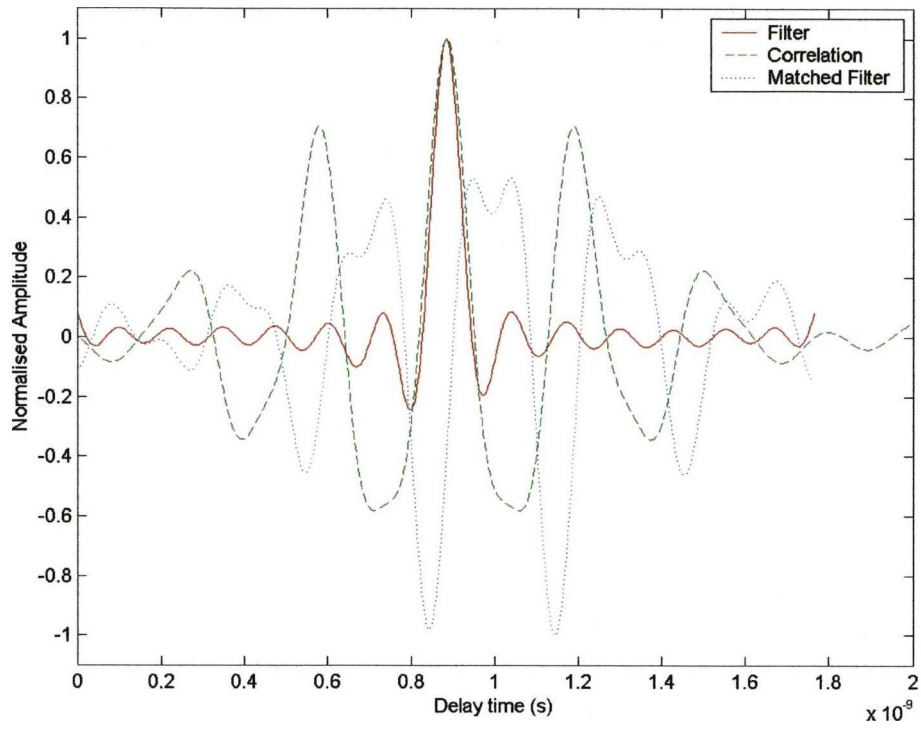

Figure 3.57 - Delay Prediction using direct PN sequence

The direct PN sequence signal is greatly impacted by the bandpass filtering of the signal. The matched filter although giving a correct detection is poor due to the signal shape after bandpass filtering. Correlation also shows a small degradation in performance with an increased size of the sidelobes. The filter method shows near identical performance despite the difference in signals, which is quite a useful ability.

3.10.1.2 Range Resolution Comparison

To compare the relative performance benefits of each method a theoretical test of range resolution will be used. A delayed pulse will be located at 0.7ns with a second delayed pulse added with a delay varying from 0.7ns to 1.05ns. The overall performance will be judged on how close the two pulses can get before they merge together. As the symbol shape affects performance the two signal types used so far

(BPSK with 5GHz carrier and direct PN sequence) will be compared along with the chirp pulse.

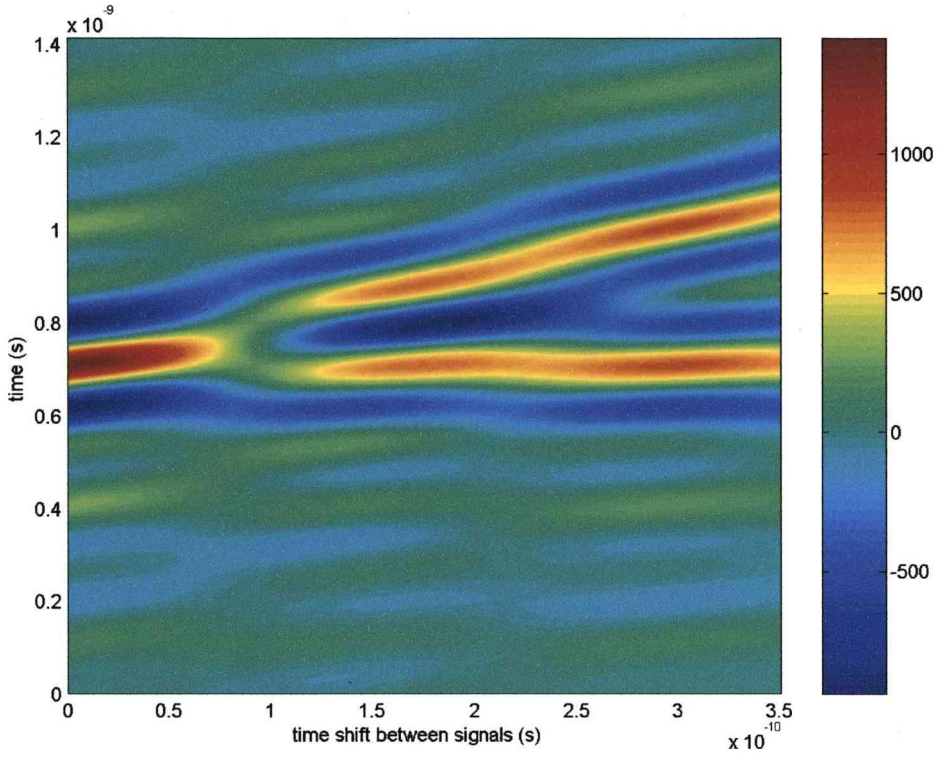

Figure 3.58 - Chirp Signal

The chirp signal shows a minimum separation of around 1.0ns. There are some problems though including strong negative troughs either side of the main detection peaks.

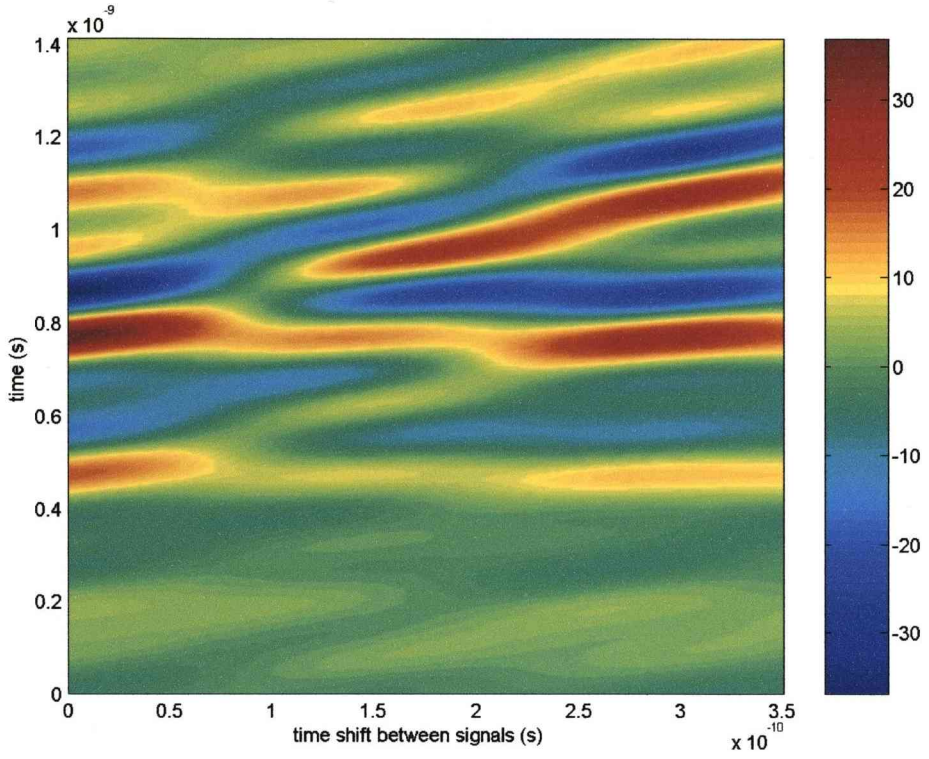

Figure 3.59 - Matched filter using phase modulation – 5GHz carrier

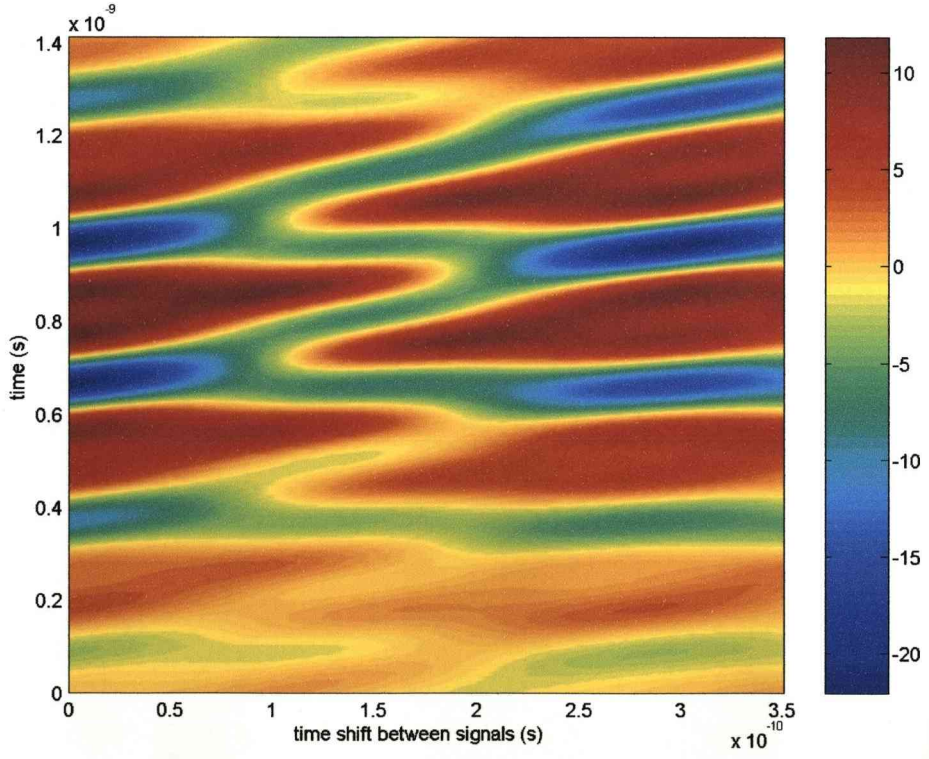

Figure 3.60 - Matched filter using phase modulation – Direct PN sequence

The matched filter shows its clear dependence on the symbol being used. The full sine wave symbol in Figure 3.59 shows both parts and ultimately makes it difficult to separate the pulses below 1.1ns. The direct PN sequence in Figure 3.60 is exceptionally difficult to separate the pulses due to the poor signal shape after bandpass filtering. Correlation in Figure 3.61 gives fairly good separation down to 1.0ns but with slightly stronger sidelobes visible. Use of the direct PN sequence in Figure 3.62 though increases that minimum separation to 1.2ns.

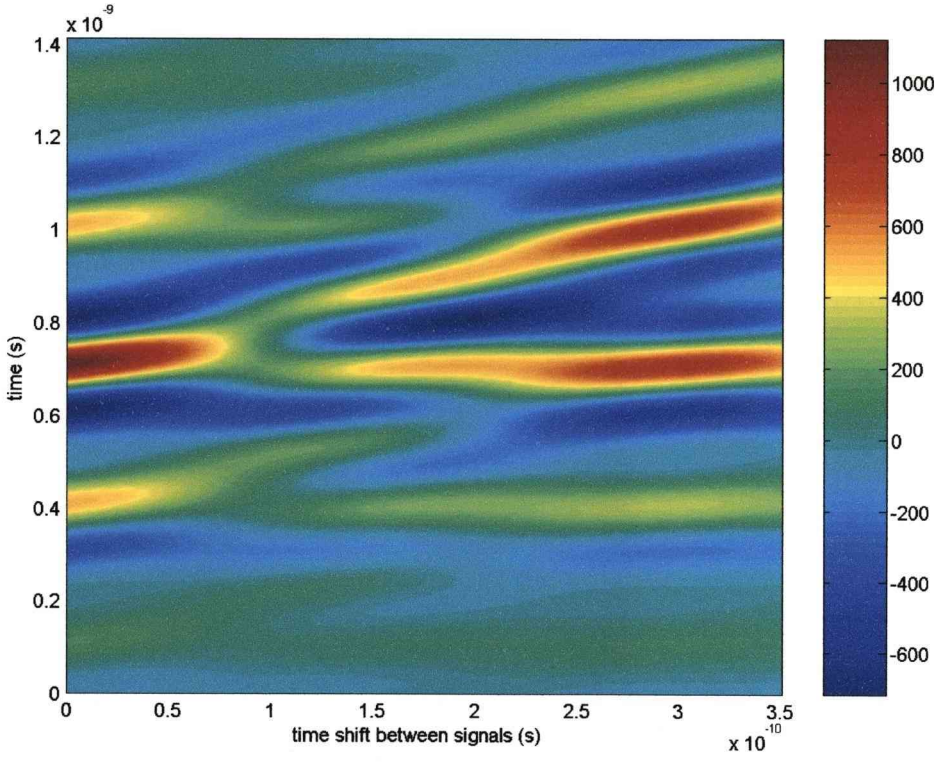

Figure 3.61 - Correlation using phase modulation – 5GHz carrier

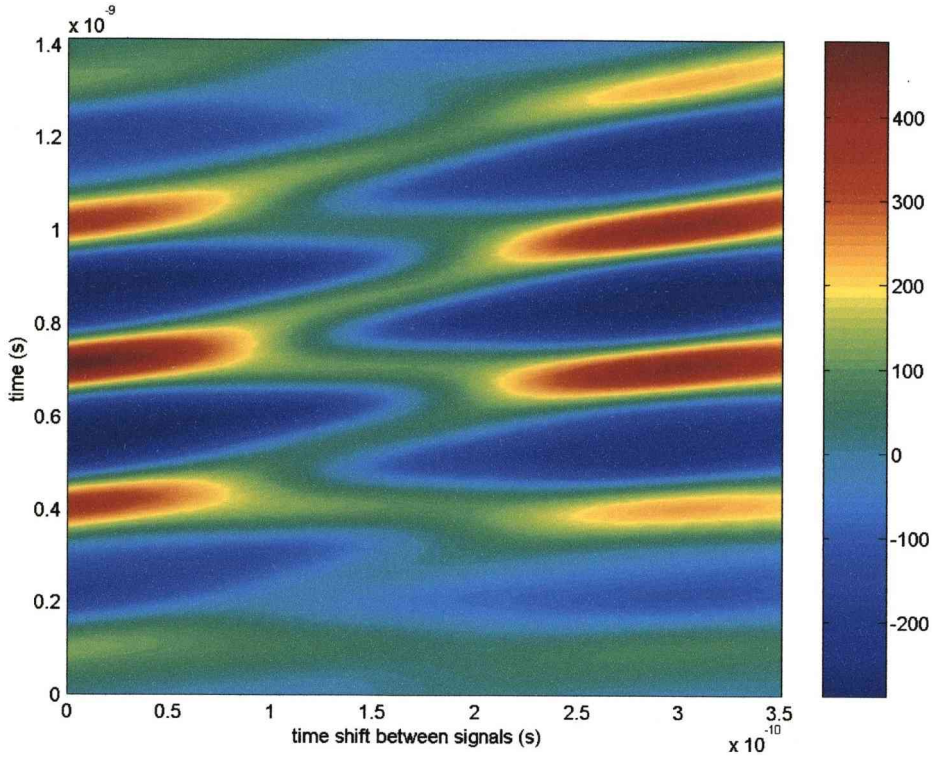

Figure 3.62 - Correlation using phase modulation – Direct PN sequence

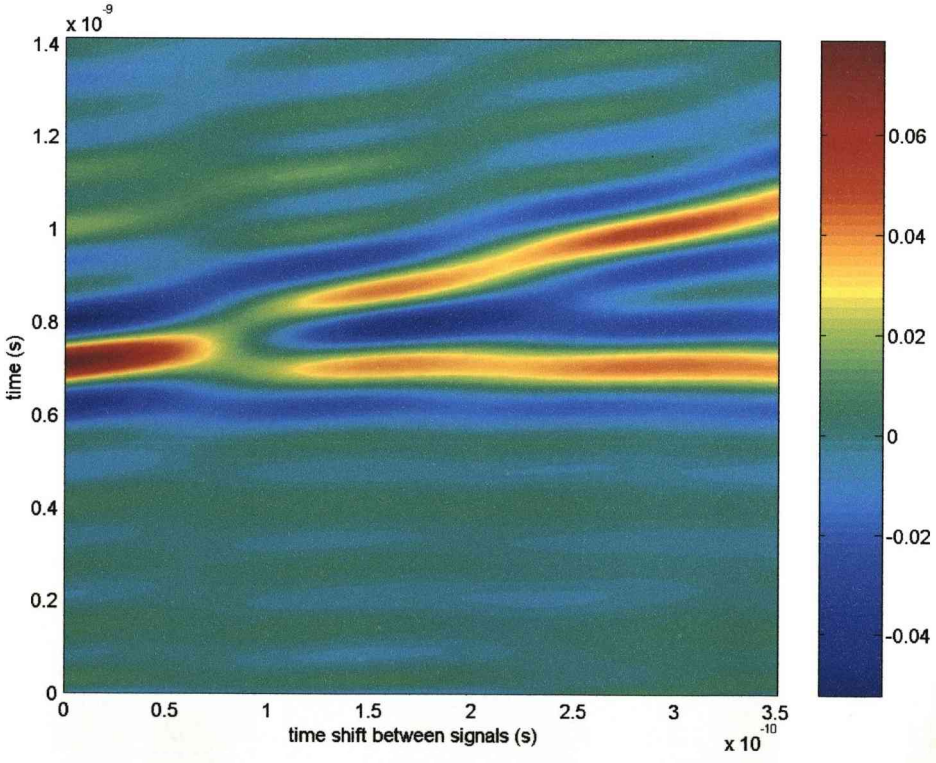

Figure 3.63 - Filter using phase modulation – 5GHz carrier

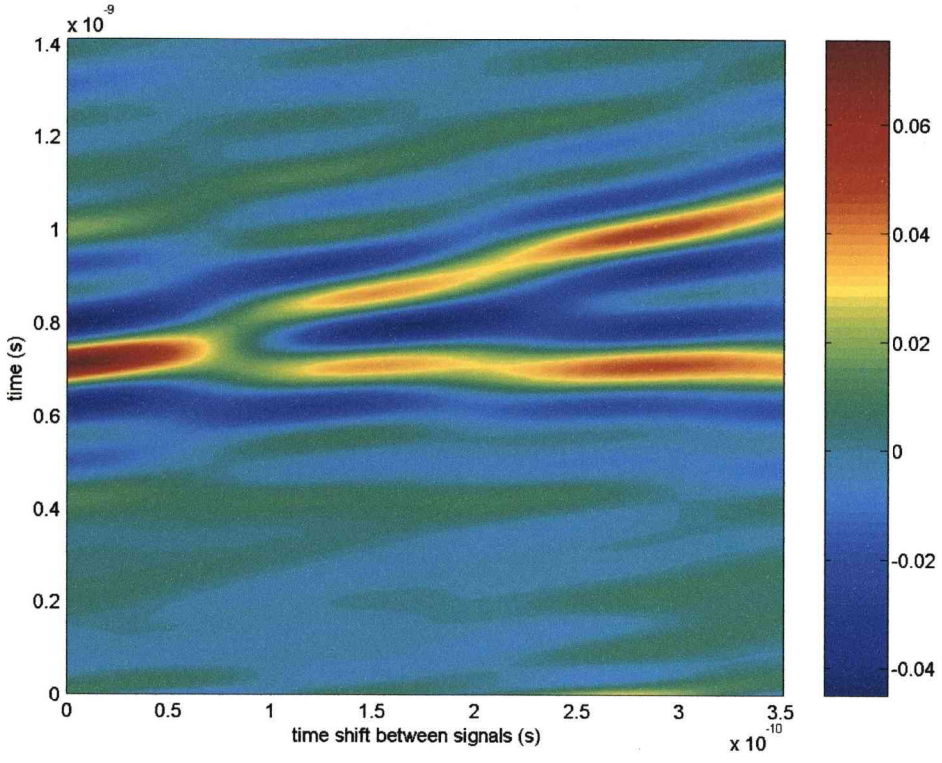

Figure 3.64 - Filter using phase modulation – Direct PN sequence

The filter performance matches that of the chirp and correlation methods but with slightly weaker sidelobes. Both Figure 3.63 and Figure 3.64 have a range resolution of 1.0ns, despite the poor signal used in Figure 3.64 that caused problems for the matched filter in Figure 3.60.

3.10.1.3 Section Conclusions

The chirp and phase modulation methods appear similar in performance leading to selection based on other factors such as hardware considerations. If non-real time processing (like for most imaging applications) is used then a phase modulated pulse combined with the filter method gives the lowest sidelobes.

3.11 Chapter Conclusions

The M-sequence has been introduced and explained with some of the potential benefits shown as well as its generation. Methods based on correlation and filtering techniques have been presented along with idealised performance in a range of situations. Various filter parameters have been investigated in order to provide an idea of what values to choose to give good performance. Multiple M-sequences received simultaneously and then extracted has been shown to be possible although the noise that it generates is significant but is essentially process independent as both correlation methods and the filter give similar results.

M-sequences with missing symbols have been examined to determine the effects that can have on signal detection and the background clutter that is created or increased. Most tests have shown the superiority of the filter method over correlation where cyclic correlation (the best) is not available. Cyclic correlation can be used when using repeated transmission of a sequence but the filter method performs better with longer sequences so it's a choice between a longer PN sequence and the benefits that provides and shorter repeated sequences to use cyclic correlation.

Missing symbols in the received signal have been found to have a degrading effect but can be managed by using longer sequences to reduce the chance of worst case maximum and minimum values appearing and producing false positive detections. Some examples including noise and longer symbols show noise degradation of the filter method and correlation but at sufficiently low noise levels the filter remains better particularly when longer symbol lengths are used and correlation causes the detection to spread out. The minimum time delay between two identical

signals (relevant to range resolution) has also been found to favour the filter method over correlation in noise free circumstances due to the lower sidelobe levels.

Chapter 4

Imaging Using a Direct Transform of the H-Matrix

This is based on a method described in [6], where it is used primarily as a way to have a visual representation of a communication channel and to better calculate channel capacity. The chapter expands on the method and takes it further, looking specifically at its performance in imaging and addressing its shortcomings in that field. It involves processing the H-matrix so that instead of coupling transmit and receive antennas it couples transmit and receive angles of the whole transmit and receive array. The transformed version is referred to as a virtual channel representation or virtual H-matrix. This can be used directly to show coupling in a given angular direction, which indicates a reflecting object to detect. The first sections cover the method as described and some shortcomings it has when related to imaging before a modified version is presented that enables the type of imaging interested in.

4.1 H-matrix Generation

The H-matrix defines the coupling between a transmitted signal and the corresponding received signal. The H-matrix is frequency domain and for T transmissions and Q receptions the H-matrix takes the form of a $T \times Q$ matrix (for a single frequency). The relationship between transmitted signal (s), received signal (x) and the H-matrix ($\boldsymbol{H}$) is given in Eq. (4.1) and expanded in Eq. (4.2).

$$x = \mathbf{H}\mathbf{s} \tag{4.1}$$

$$\begin{bmatrix} x_1 \\ \vdots \\ x_Q \end{bmatrix} = \begin{bmatrix} H_{1,1} & H_{2,1} & \cdots & H_{P,1} \\ \vdots & \vdots & \ddots & \vdots \\ H_{1,Q} & H_{2,Q} & \cdots & H_{P,Q} \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_P \end{bmatrix} \tag{4.2}$$

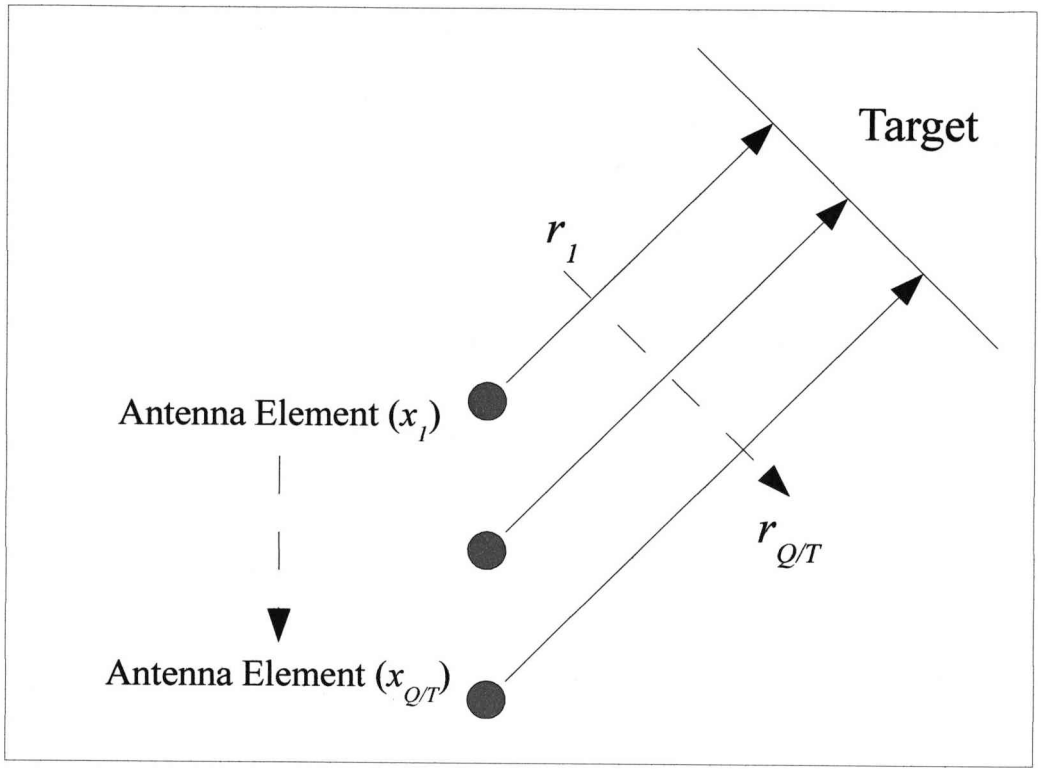

Figure 4.1 - Antenna Naming and distance relations (mono-static)

The H-matrix is generated using the radar equation for spreading loss and a phase term given in Eq. (4.3) where r is the distance from the antenna array element to the target, r_q for the receiver to target distance and r_t for the transmitter to target distance. The other variables are the targets reflection coefficient Γ and signal wavelength λ.

$$H_{(q,t)} = \frac{\Gamma\lambda^2}{(4\pi)^3 r_{q^2} r_{t^2}} e^{-j2\pi\left(\frac{r_q + r_t}{\lambda}\right)} \tag{4.3}$$

4.2 H-Matrix Imaging

In order to transmit a signal in a particular direction you alter the phase of individual array elements so the signals constructively interfere in the desired direction and negatively interfere in other directions. In this case the phase alteration will be applied in the processing not the transmission stage. The simplest antenna array is a uniform linear antenna array (1-D) given in Figure (4.2) where d is the antenna spacing and ϕ is the angle of transmission or reception.

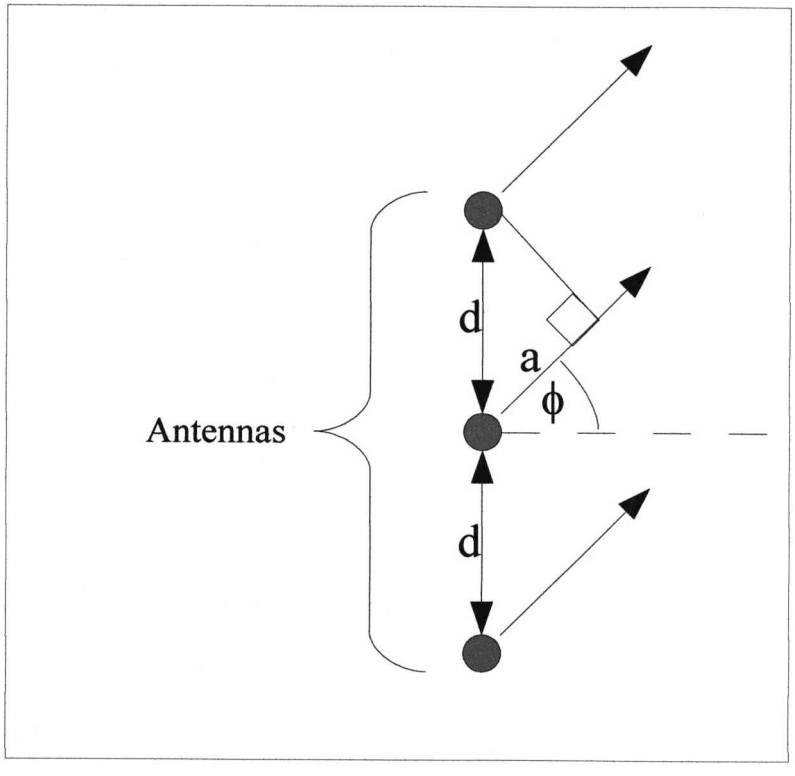

Figure 4.2 - Antenna Array

The path length difference between adjacent antenna elements is given by Eq. (4.4)

$$\begin{aligned} a &= d\cos(90-\varphi) \\ a &= d\sin(\varphi) \end{aligned} \tag{4.4}$$

This then translates into a phase difference (θ) given in Eq. (4.5) where λ is the signals wavelength

$$\frac{a}{\lambda}=\frac{d\sin(\varphi)}{\lambda}=\theta \tag{4.5}$$

The transform that focuses a transmitted signal or directs a received signal to a given angle is given in Eq. (4.6). They are usually referred to as the array steering or array response vectors respectively.

$$\mathbf{a}(\theta)=\frac{1}{\sqrt{T}}\begin{bmatrix}1\\ e^{-j2\pi\theta}\\ \vdots\\ e^{-j2\pi(T-1)\theta}\end{bmatrix} \tag{4.6}$$

θ is periodic due to the sine function and $\alpha(\theta)$ is also periodic. However due to the inverse sine function θ is limited to the range $-\beta<\theta<\beta$ in order to keep ϕ within the function bounds as highlighted in Eq. (4.7) (bounds for ϕ are then $-\pi/2<\phi<\pi/2$).

$$\theta=\frac{d\sin(\varphi)}{\lambda}=\beta\sin(\varphi),\quad \varphi=\sin^{-1}\left(\frac{\theta}{\beta}\right) \tag{4.7}$$

As a consequence of $\alpha(\theta)$ being periodic with a period of 1 there are further limitations on θ that reduce the acceptable range to $-0.5<\theta<0.5$. Values outside this range will lead to aliasing. The boundaries then are fully laid out in Eq. (4.8)

$$-0.5<\theta<0.5,\sin^{-1}\left(\frac{-0.5}{\beta}\right)<\varphi<\sin^{-1}\left(\frac{0.5}{\beta}\right) \tag{4.8}$$

This does imply that β cannot be between -0.5 and 0.5 and values less than -0.5 or greater than 0.5 reduce the angular spread of ϕ. Two values are defined in Eq. (4.9) that form the limits to the later summations.

$$\tilde{Q}=\frac{(Q-1)}{2}\text{ and }\tilde{T}=\frac{(T-1)}{2} \tag{4.9}$$

4.2.1 Initial Version

The relationship between the H-matrix ($\boldsymbol{H}$) and the virtual H-matrix ($\boldsymbol{H}_v$) is given in Eq. (4.10) with a rearranged form in Eq. (4.11) and the relevant terms are defined in Eq. (4.12) and Eq. (4.13)

$$\mathbf{H}=\sum_{q=-\tilde{Q}}^{\tilde{Q}}\sum_{t=-\tilde{T}}^{\tilde{T}}\mathbf{H}_v(q,t)\boldsymbol{\alpha}_{\boldsymbol{R}}\left(\tilde{\theta}_{R,q}\right)\boldsymbol{\alpha}_{\boldsymbol{T}}^{\boldsymbol{H}}\left(\tilde{\theta}_{T,t}\right)=\tilde{\mathbf{A}}_R\mathbf{H}_v\tilde{\mathbf{A}}_T^H \tag{4.10}$$

$$\mathbf{H}_v=\tilde{\mathbf{A}}_R^H\mathbf{H}\tilde{\mathbf{A}}_T \tag{4.11}$$

$$\begin{aligned}\tilde{\mathbf{A}}_T(t)&=\frac{1}{\sqrt{T}}\begin{bmatrix}1 & \cdots & 1\\ e^{\frac{-j2\pi t}{T}} & \ddots & e^{\frac{-j2\pi t}{T}}\\ \vdots & & \vdots\\ e^{\frac{-j2\pi(T-1)t}{T}} & \cdots & e^{\frac{-j2\pi(T-1)t}{T}}\end{bmatrix}\\ \tilde{\mathbf{A}}_R(q)&=\frac{1}{\sqrt{Q}}\begin{bmatrix}1 & \cdots & 1\\ e^{\frac{-j2\pi q}{Q}} & \ddots & e^{\frac{-j2\pi q}{Q}}\\ \vdots & & \vdots\\ e^{\frac{-j2\pi(Q-1)q}{Q}} & \cdots & e^{\frac{-j2\pi(Q-1)q}{Q}}\end{bmatrix}\end{aligned} \tag{4.12}$$

$$\begin{aligned}\tilde{\theta}_{R,q}&=\frac{q}{Q}\ \text{where}\ -\tilde{Q}\le q\le\tilde{Q}\\ \tilde{\theta}_{T,t}&=\frac{t}{T}\ \text{where}\ -\tilde{T}\le t\le\tilde{T}\end{aligned} \tag{4.13}$$

Taking the $\boldsymbol{H}_v$ version in Eq. (4.11), substituting for the relevant matrices and rearranging to get the summation version leads to Eq. (4.14). The $\boldsymbol{H}_v$ subscripts x and y represent the matrix indices and are in the range $1<x<T$ and $1<y<Q$ corresponding to the actual angles from the transmitter and receiver, given in Eq. (4.13).

$$H_{v_{(x,y)}}=\frac{1}{\sqrt{Q}\sqrt{T}}\sum_{q=1}^{Q}\sum_{t=1}^{T}\frac{\Gamma\lambda^2}{(4\pi)^3 r_{q^2}r_{t^2}}e^{-j2\pi\left(\frac{r_q+r_t}{\lambda}\right)}e^{-j2\pi\left(\frac{(t-1)\left(x-\frac{T+1}{2}\right)}{T}-\frac{(q-1)\left(y-\frac{Q+1}{2}\right)}{Q}\right)} \tag{4.14}$$

$$\varphi_T(x)=\sin^{-1}\left(\frac{\left(x-\frac{T+1}{2}\right)\lambda}{Td}\right) \text{ and } \varphi_Q(y)=\sin^{-1}\left(\frac{\left(y-\frac{Q+1}{2}\right)\lambda}{Qd}\right) \quad (4.15)$$

In Eq. (4.14) r_q and r_t vary depending on the location of the target. Keeping the assumption that the target is a large enough distance away for all the antenna element signals to be parallel (basis of the method) they can be replaced by a form of $r_{ref}+\Delta$ where Δ is the path length difference between a reference antenna (r_{ref}) and the required antenna $r_{q/t}$. Choosing the reference antenna to be r_1, the values for r_q and r_t are given in Eqs. (4.16-17).

$$r_t=r_1+(t-1)\frac{\left(f-\frac{T+1}{2}\right)}{T} \quad (4.16)$$

$$r_q=r_1+(q-1)\frac{\left(g-\frac{Q+1}{2}\right)}{Q} \quad (4.17)$$

In these cases f and g represent transmit and receive angles within which the target lies (angles given by Eq. (4.15) with f equals x and g equals y). Substituting in for r_q and r_t leaves the full version split into magnitude and phase as in Eq. (4.18-20).

$$H_{v_{(x,y)}}=\frac{1}{\sqrt{Q}\sqrt{T}}\sum_{q=1}^{Q}\sum_{t=1}^{T}\alpha_{(t,q)}e^{-j2\pi\beta_{(t,q)}} \quad (4.18)$$

$$\alpha_{(t,q)}=\frac{\Gamma\lambda^2}{(4\pi)^3\left(r_1-\left(\frac{(t-1)\left(g-\frac{T+1}{2}\right)}{T}\right)\lambda\right)^2\left(r_1-\left(\frac{(q-1)\left(f-\frac{Q+1}{2}\right)}{Q}\right)\lambda\right)^2} \quad (4.19)$$

$$\beta_{(t,q)}=\left(\frac{2r_1}{\lambda}\right)-\left(\frac{(t-1)\left(g-\frac{T+1}{2}\right)}{T}+\frac{(q-1)\left(f-\frac{Q+1}{2}\right)}{Q}\right)+\left(\frac{(t-1)\left(x-\frac{T+1}{2}\right)}{T}-\frac{(q-1)\left(y-\frac{Q+1}{2}\right)}{Q}\right) \tag{4.20}$$

Further simplification of the phase term Eq. (4.20) leads to Eq. (4.21)

$$\beta_{(t,q)}=\left(\frac{2r_1}{\lambda}\right)+\left(\frac{(t-1)(x-g)}{T}+\frac{(q-1)(y+f-Q-1)}{Q}\right) \tag{4.21}$$

The goal is for the complex terms for each received signal ($H_{(q,t)}$) to take on the same value leading to a maximum in the sum of complex variables highlighted in Eq. (4.21). If the targets transmit reference g equals the viewing reference x and receive reference f equals viewing reference $y-Q-1$ then the two terms cancel out leading to a maximum with those angles. However the effective reversal of reference y is not a useful property for imaging and is caused by the complex conjugate transpose of the receive component in the original $\boldsymbol{H}_v$ Eq. (4.22).

$$\mathbf{H}_v=\tilde{\mathbf{A}}_R^H\mathbf{H}\tilde{\mathbf{A}}_T \tag{4.22}$$

4.2.2 Equation Solution

The solution to Eq. (4.18-21) to get a maximum is when the last term is constant over the entire sum. With t and q varying independently a solution is when g equals x and f equals $y-Q-1$ making the last term equal to 1. A secondary solution is when $g-x$ and $f+y-Q-1$ both equal an integer multiple of the corresponding number of antenna elements (T and Q). Regardless of the value of q or t it would always lead to an

integer multiple of 2π. This is the periodic limit of the method leading to aliasing and can only be achieved when the target is located outside of the field of view of the antenna array. For a very large number of antenna elements, the limit on the arrays field of view is given by Eq. (4.23)

$$-\sin^{-1}\left(\frac{\lambda}{2d}\right)\leq\varphi\leq\sin^{-1}\left(\frac{\lambda}{2d}\right) \tag{4.23}$$

If the array has a front looking field of view from $-\pi/2$ to $\pi/2$ then the correct antenna spacing for the algorithm to match is $\lambda/2$. If the spacing is increased to λ, then the field of view would be reduced and become -0.523 rad to 0.523 rad. Targets that could be picked up outside that range would then alias into the image as $g-x$ and $f-y$ can now take on a value that is a multiple of T and Q respectively (due to g and f no longer lying within the standard ranges of 1 to T and 1 to Q). Going back to the main problem, making the assumption that the viewing reference (x and y–Q–1) matches the target reference (f and g) on both transmitting and receiving sides causes the last term of Eq. (4.21) to cancel out so removing the antenna location dependence (caused through q and t). This leaves the maximum value solution as Eq. (4.24)

$$H_{v_{(x,y)}} = \frac{1}{\sqrt{Q}\sqrt{T}} \sum_{q=1}^{Q} \sum_{t=1}^{T} \left| \begin{array}{l} \dfrac{1}{\left(r_1 - \left(\dfrac{(t-1)\left(g - \dfrac{T+1}{2} \right)}{T} \right) \lambda \right)^2} \\ \times \dfrac{\Gamma \lambda^2}{(4\pi)^3} e^{-j2\pi \left(\frac{2r_1}{\lambda} \right)} \\ \times \dfrac{1}{\left(r_1 - \left(\dfrac{(q-1)\left(f - \dfrac{Q+1}{2} \right)}{Q} \right) \lambda \right)^2} \end{array} \right| \tag{4.24}$$

If the assumption is carried on and r_1 is much larger than the λ term (within the brackets) it can be reduced further to an approximate maximum given by Eq. (4.25)

$$H_{v_{\max}} = \frac{\sqrt{QT}\,\Gamma \lambda^2}{(4\pi)^3 (r_1)^4} \tag{4.25}$$

This also raises the possibility that the distance to the target could be calculated using a rearranged version given in Eq. (4.26). The accuracy depends on the relative sizes of r_1 to the λ term and whether an accurate prediction of the reflection coefficient can be made.

$$r_1 = \sqrt[4]{\frac{\sqrt{QT}\,\Gamma \lambda^2}{(4\pi)^3 H_{v_{\text{target}}}}} \tag{4.26}$$

Note however that the maximum of the λ term increases with the total number of transmitters and receivers as illustrated in Figure 4.3. (Maximum occurs when both t and g equal T).

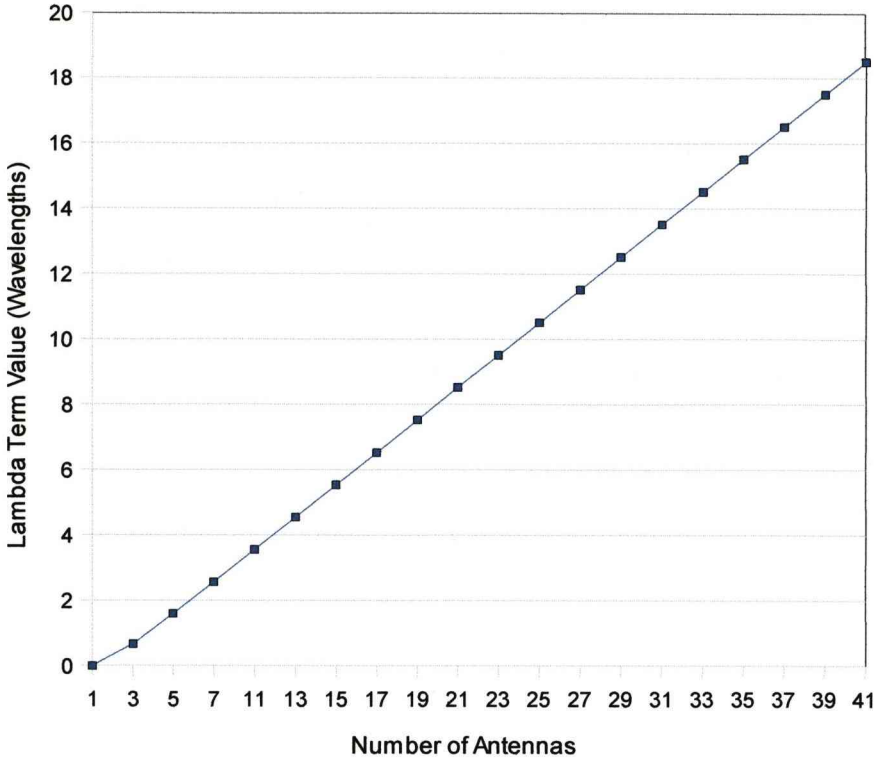

Figure 4.3 - Relation between λ term and antenna number

The thing to note here is that with increasing antenna number there is the potential for significant alteration in the maximum output value and overall variation if the target is too close.

4.2.3 Generic Form Imaging

Replacing the modification term with a generic symbol representing the alteration for the transmitting component (θ_t) and the receiving component (θ_q) gives Eq. (4.27). Then performing a transform of the delay terms r_t and r_q to a reference (r_1) plus modification (θ_a for transmitting and θ_b for receiving). Split the modification off and simplify to Eq. (4.28)

$$H_{v_{(x,y)}} = \frac{1}{\sqrt{Q}\sqrt{T}} \sum_{q=1}^{Q} \sum_{t=1}^{T} \frac{\Gamma \lambda^2}{(4\pi)^3 r_{t^2} r_{q^2}} e^{-j2\pi\left(\frac{r_t + r_q}{\lambda}\right)} e^{-j2\pi(\theta_t + \theta_q)} \qquad (4.27)$$

$$H_{v_{(x,y)}} = \frac{1}{\sqrt{Q}\sqrt{T}} \sum_{q=1}^{Q} \sum_{t=1}^{T} \frac{\Gamma \lambda^2}{(4\pi)^3 (r_1 - \theta_a)^2 (r_1 - \theta_b)^2} e^{-j2\pi\left(\frac{2r_1}{\lambda}\right)} e^{j2\pi(\theta_a - \theta_t + \theta_b - \theta_q)} \quad (4.28)$$

θ_a is a function of t and g, θ_t is a function of t and x, θ_b is a function of q and f and θ_q is a function of q and y. Due to this dependency and the use of a complex sum the generic version shows a maximum when $\theta_a + \theta_b$ equals $\theta_t + \theta_q$ (or both are integer multiples over the entire range for the periodic out of bounds solution). The solution for the non-generic parallel version has θ_a equal θ_t and θ_b equal θ_q to give the maximum. If the target is close enough to void the parallel assumption θ_a and θ_b change significantly so θ_t and θ_q must change to match.

4.2.4 DFT Matrix

There is considerable similarity between the DFT matrix and the two matrices either side of $\mathbf{H}_v$ (in Eq. (4.11) and shown fully in Eq. (4.12)). Firstly the DFT operation is shown in Eq. (4.29) and expanded to a matrix multiplication in Eq. (4.30).

$$X_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi kn}{N}} \quad \text{for } 0 \le k \le \text{N-1} \quad (4.29)$$

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_k \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} 1 & \cdots & 1 \\ e^{\frac{-j2\pi k}{N}} & \ddots & e^{\frac{-j2\pi k}{N}} \\ \vdots & & \vdots \\ e^{\frac{-j2\pi(N-1)k}{N}} & \cdots & e^{\frac{-j2\pi(N-1)k}{N}} \end{bmatrix} \quad (4.30)$$

Compare the transform matrix from the DFT shown in Eq. (4.31) to the version used by one of those used to generate $\mathbf{H}_v$ shown in Eq. (4.32).

$$\frac{1}{\sqrt{N}}\begin{bmatrix} 1 & \cdots & 1 \\ e^{\frac{-j2\pi k}{N}} & \ddots & e^{\frac{-j2\pi k}{N}} \\ \vdots & & \vdots \\ e^{\frac{-j2\pi(N-1)k}{N}} & \cdots & e^{\frac{-j2\pi(N-1)k}{N}} \end{bmatrix} \tag{4.31}$$

$$\frac{1}{\sqrt{T}}\begin{bmatrix} 1 & \cdots & 1 \\ e^{\frac{-j2\pi t}{T}} & \ddots & e^{\frac{-j2\pi t}{T}} \\ \vdots & & \vdots \\ e^{\frac{-j2\pi(T-1)t}{T}} & \cdots & e^{\frac{-j2\pi(T-1)t}{T}} \end{bmatrix} \tag{4.32}$$

The matrices are identical if $T=N$ and $t=k$. In the DFT the range for k is $0 \leq k \leq N-1$ but the range for t differs as shown in Eq. (4.33).

$$-\frac{(T-1)}{T} \leq t \leq \frac{(T-1)}{T} \tag{4.33}$$

This range corresponds to a variant form of DFT, the centred DFT (CDFT). The operation in Eq. (4.11) has both a forward and an inverse component so the resultant $\boldsymbol{H}_v$ matrix remains in the frequency domain.

4.2.5 Operation

A target is at a set distance and direction from the antenna array. In order to transmit in that direction the transform given in Eq. (4.11) can be used on the received signals. It alters the phase of each signal so that they are all in phase along the path to the target. The target reflects the transmitted signals back which arrive at the receivers with different phases. The transform can be used again so the received signals all have the same phases for a given receiving direction. If there is a target in the selected direction for both the transmitting and receiving parts the phase alteration by the transform will align all the signal phases into approximately the same direction

(complex plane) so the signals will add constructively in the sum to generate H_v. If there is no target in chosen direction then the signals will add destructively leading to a much smaller complex value for that direction in H_v. Taking the magnitude of H_v will then show the coupling between transmit and receive angles due to the constructive or destructive adding of the signals that make up that transmit/receive angle combination. Note that it is operating purely on the received signals and does not direct the beams during transmission.

4.3 Results

4.3.1 Single Target

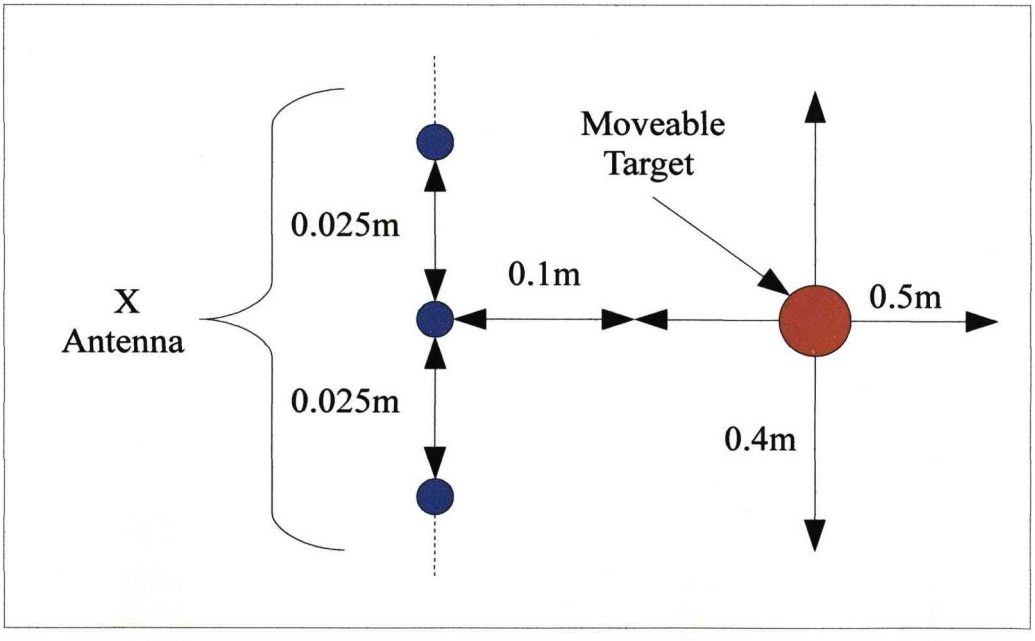

Figure 4.4 - Simulation Set-up

The simulation is set-up according to Figure 4.4 with the target moved over a range of values and the number of antenna elements also varied. From the range of target positions three will be looked at, (-0.2, 0.1), (0, 0.3) and (0.1, 0.3). The corresponding angles (identical on transmitting and receiving sides) are -1.107 rad, 0 rad and 0.322

rad. The frequency remained constant at 6GHz with half wavelength antenna element spacing in air. The H-matrix was generated using Eq. (4.3) transmitting and receiving at the same array with the targets reflection coefficient equal to 1. Result images have been altered so the transmission angle matches the received angle for a given target location (the output has the transmission angle matched to the negative received angle so the alteration is a reversal in polarity of one axis).

4.3.2 3-Element Array

The 3-element array has a limited resolution of only 3 transmit and 3 receive angles for 9 possibilities total. For transmitting and receiving from the same location with a single target only the 3 diagonal values of equal transmit/receive angles are relevant.

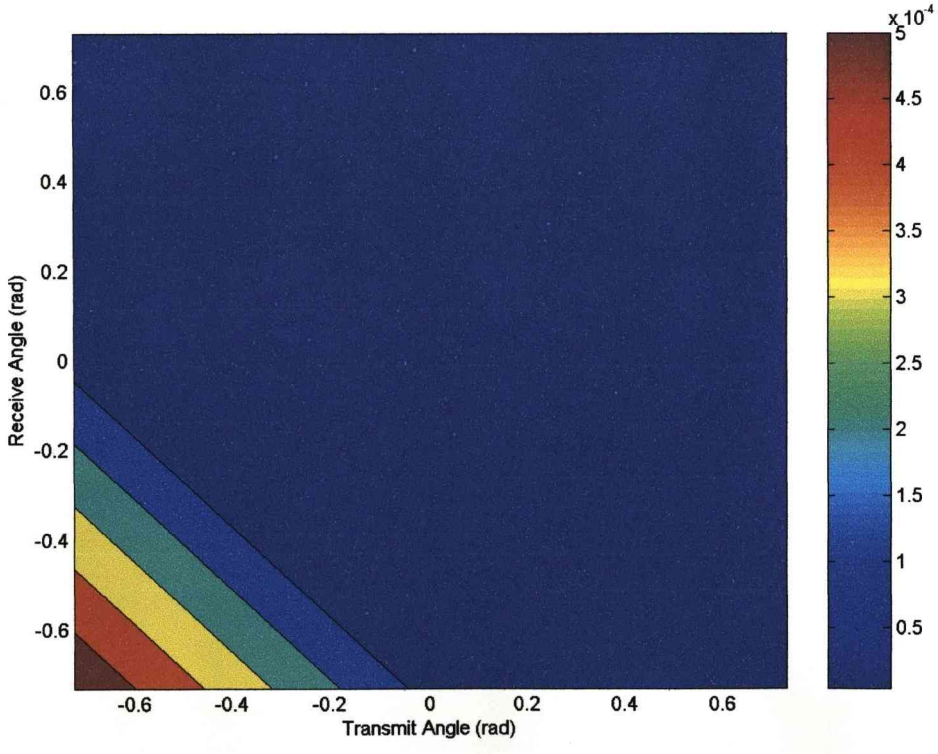

Figure 4.5 - 3-Element, (-0.2, 0.1) target

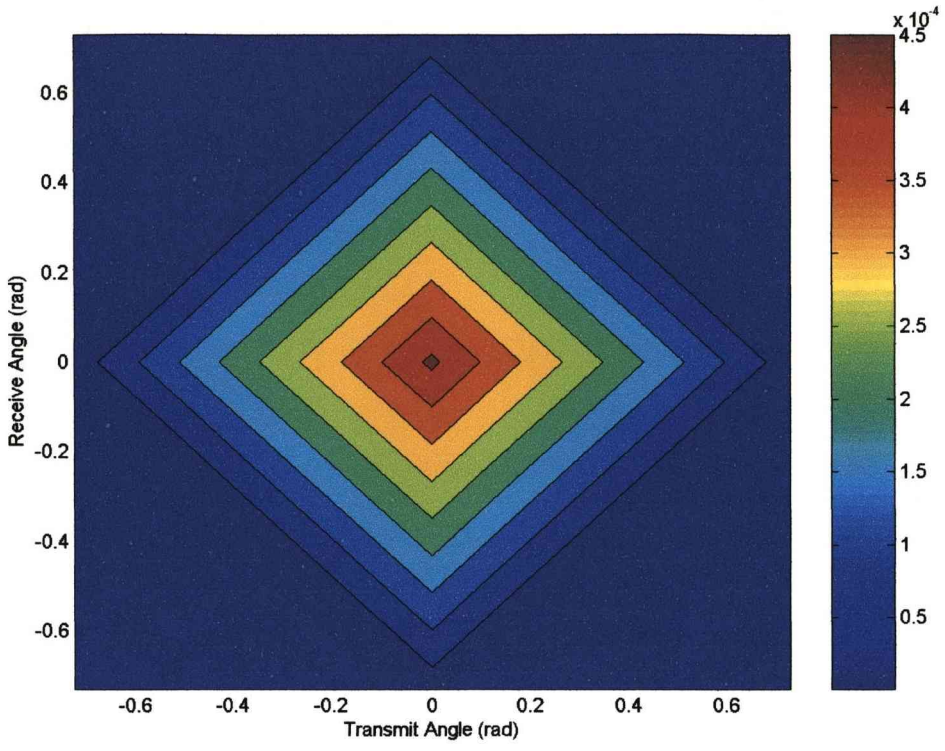

Figure 4.6 - 3-Element, (0, 0.3) target

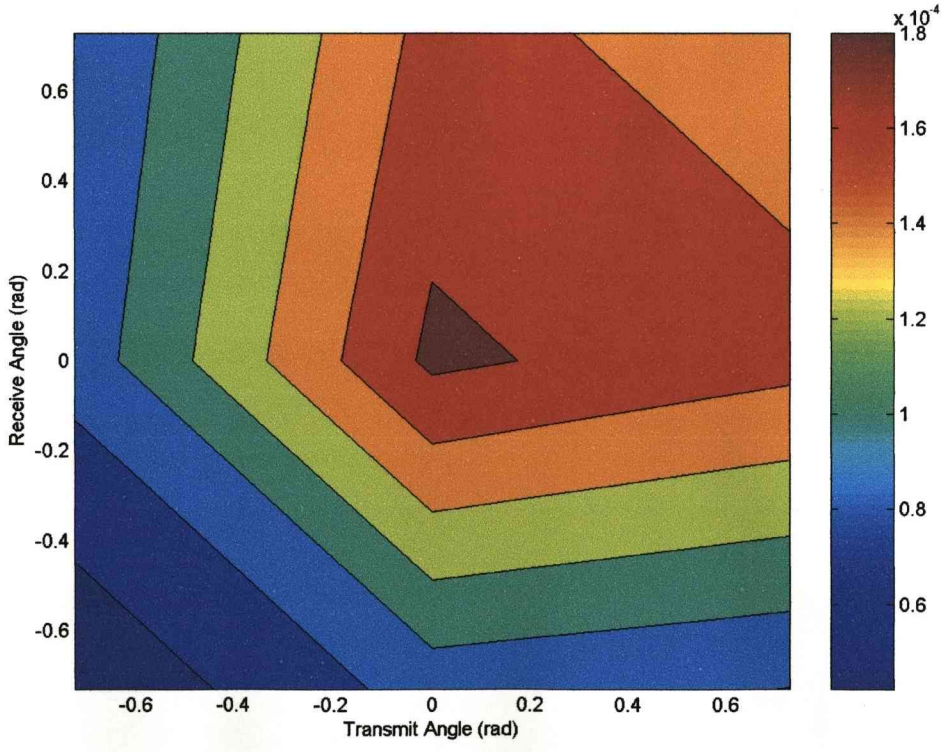

Figure 4.7 - 3-Element, (0.1, 0.3) target

Figure 4.5 shows the target is in one of the outer two angles (in this case negative side) and Figure 4.6 where the target is directly ahead of the array making it symmetrical. Figure 4.7 is a bit more interesting showing when the target is approaching the boundary between two angles giving a more spread out result. Overall the target is shown in the correct angular combination but the resolution is poor.

4.3.3 7-Element Array

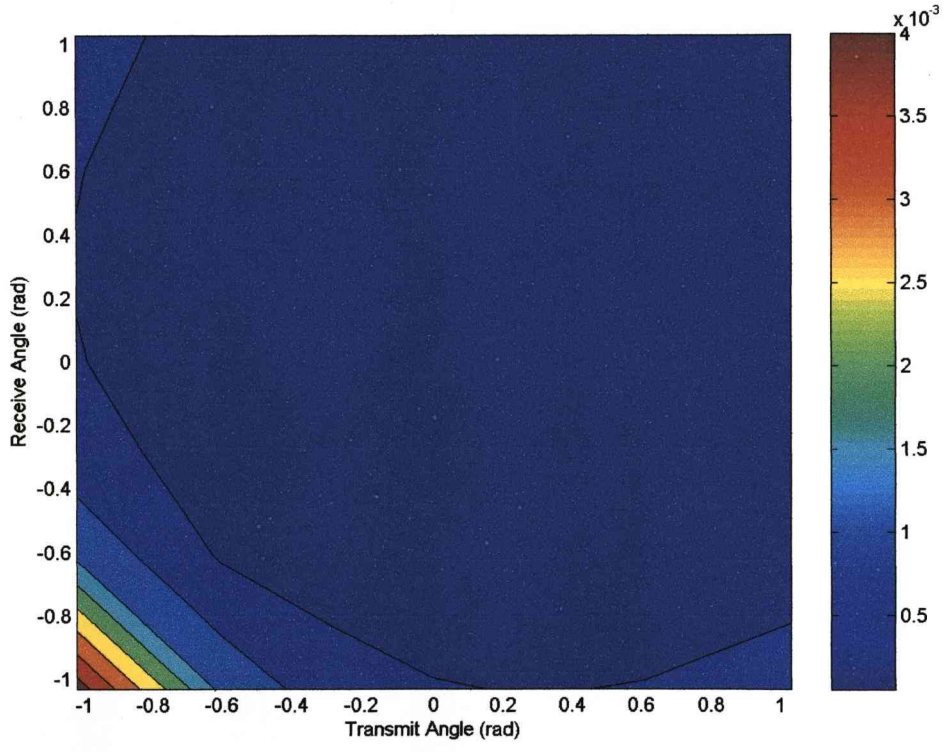

Figure 4.8 - 7-Element, (-0.2, 0.1) target

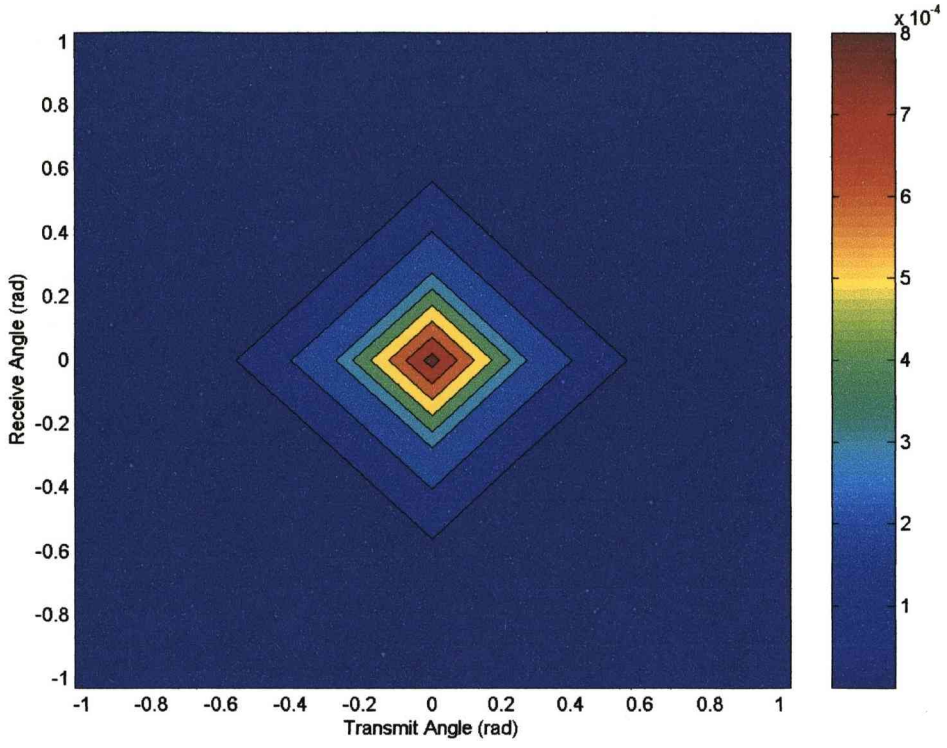

Figure 4.9 - 7-Element, (0, 0.3) target

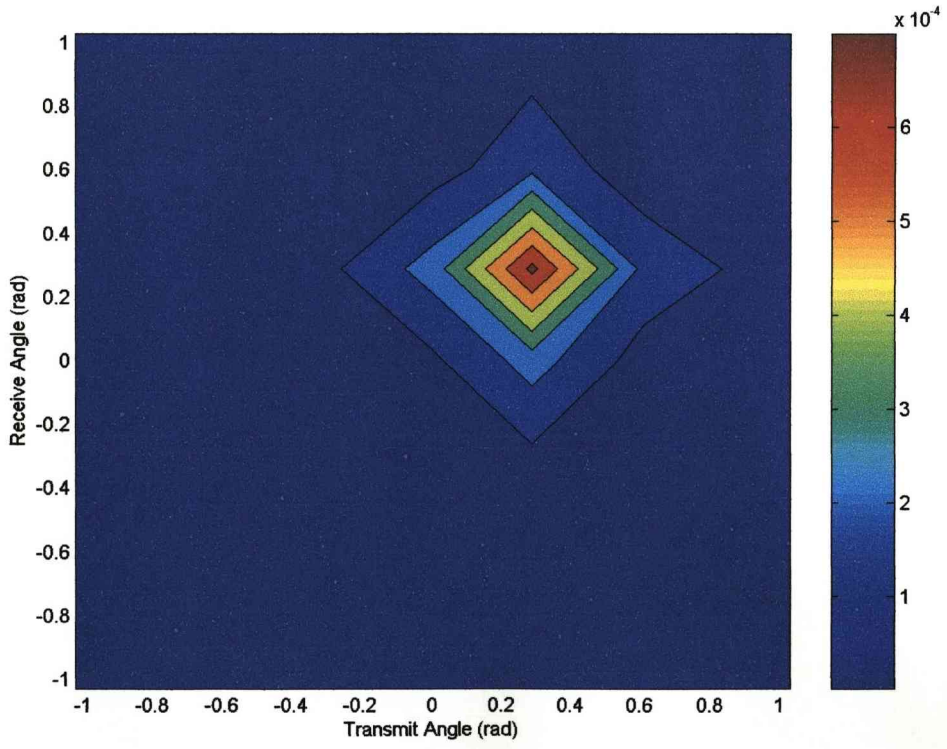

Figure 4.10 - Element, (0.1, 0.3) target

Again in Figure 4.8 the target is in the one of the outer angles. Looking at Figure 4.9 and Figure 4.10 shows the smaller target that comes with better resolution (7-element array gives 7 potential angles).

4.3.4 13-Element Array

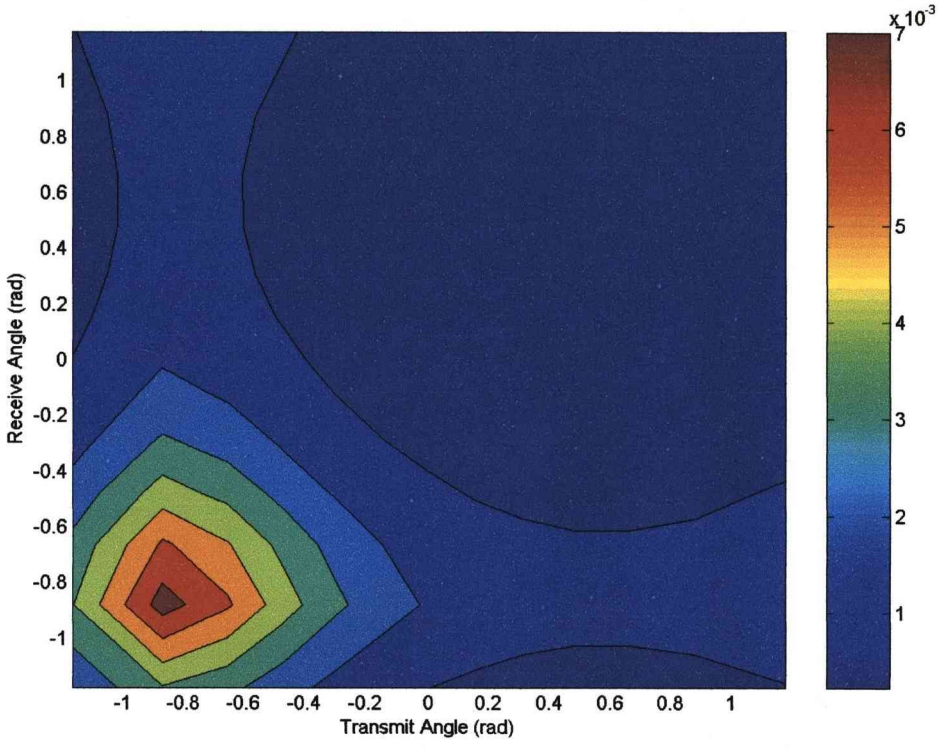

Figure 4.11 - 13-Element Array – (-0.2, 0.1) target

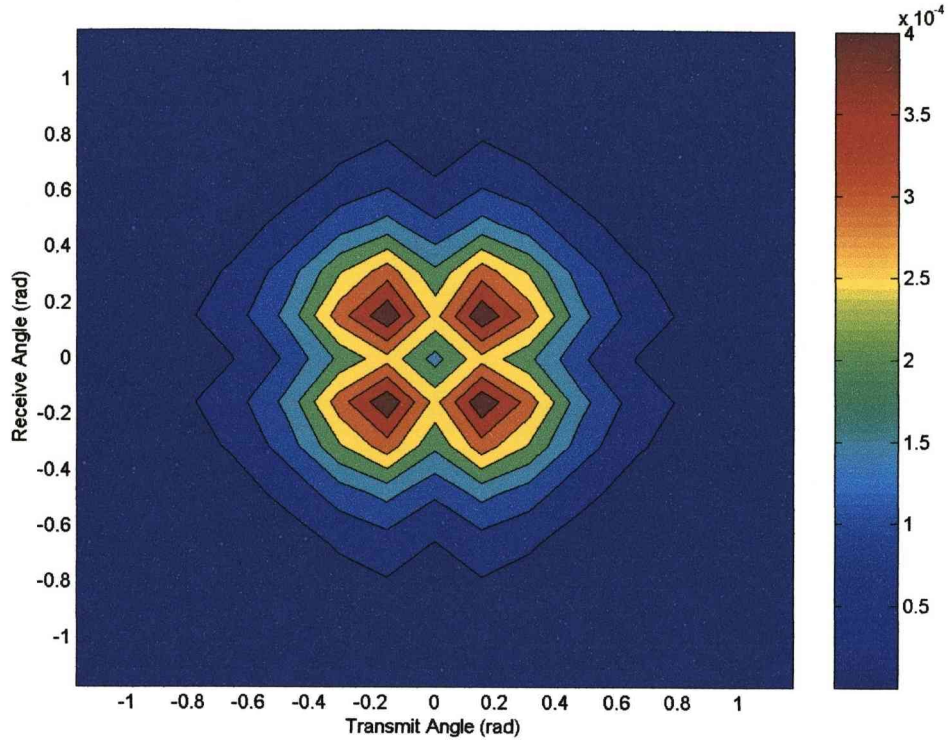

Figure 4.12 - 13-Element Array – (0, 0.3) target

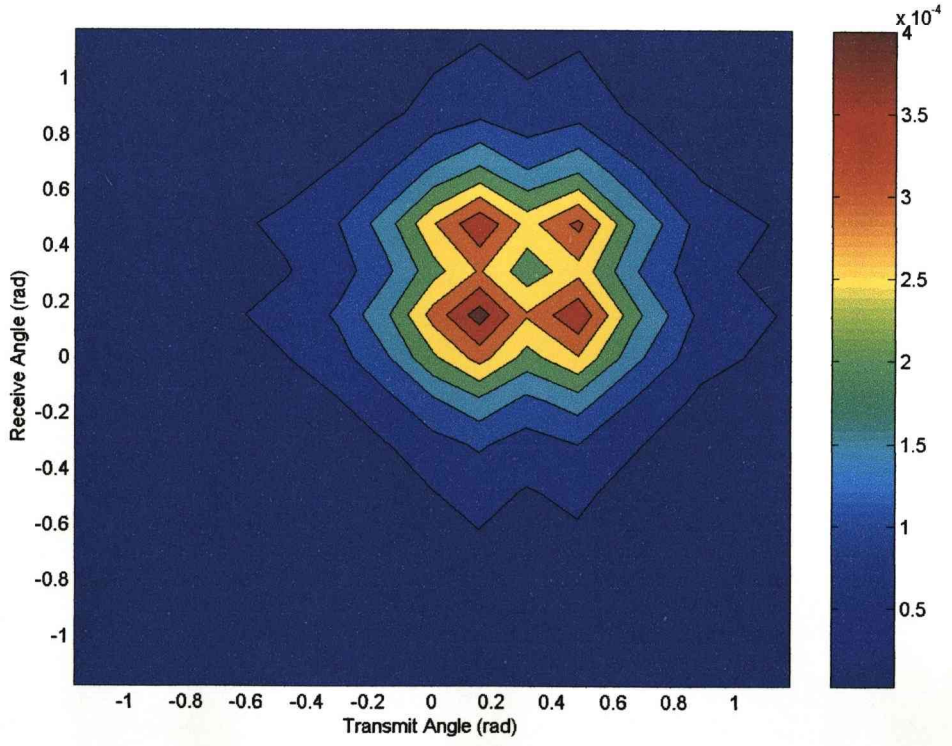

Figure 4.13 - 13-Element Array – (0.1, 0.3) target

With more antenna elements Figure 4.11 shows that the target is now more visible, it could potentially be due to the increased resolution and is no longer in one of the outer angles but its angle is wrong so it fits in with the next two images. Figure 4.12 and Figure 4.13 show the target breaking up over multiple adjacent angles. This is because the target is too close to the antenna array, there is an effective limit to how close the target can be and still be detected as a single detection. Moving the target further away in Figure 4.14 (angle 0 rad) and Figure 4.15 (angle 0.38 rad) shows that the single detection reappears at an appropriate distance.

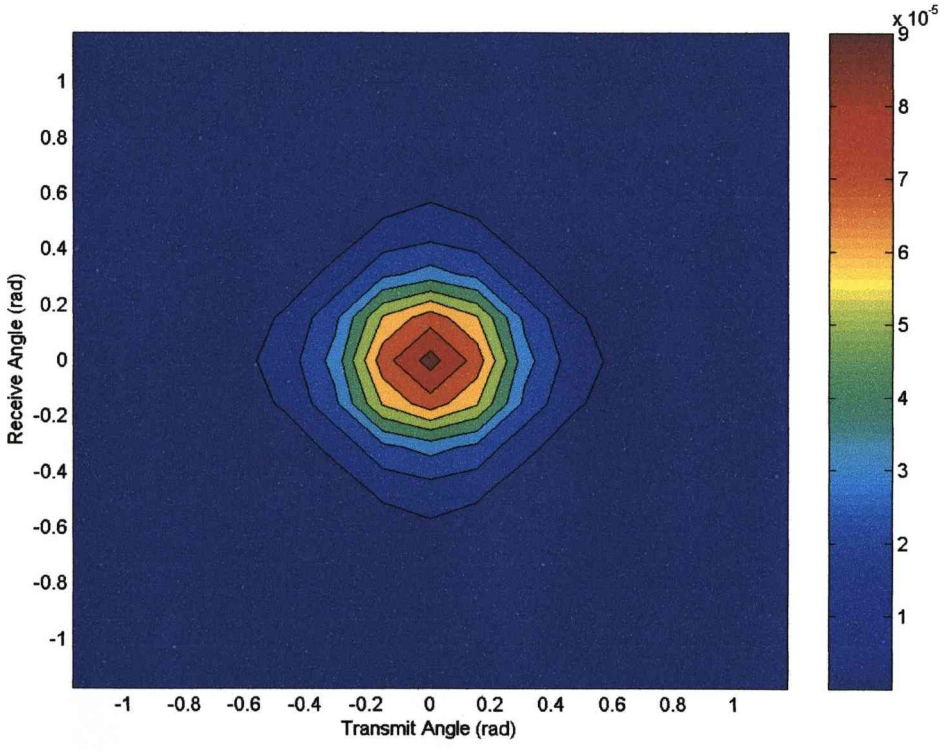

Figure 4.14 - 13-Element Array – (0, 0.5) target

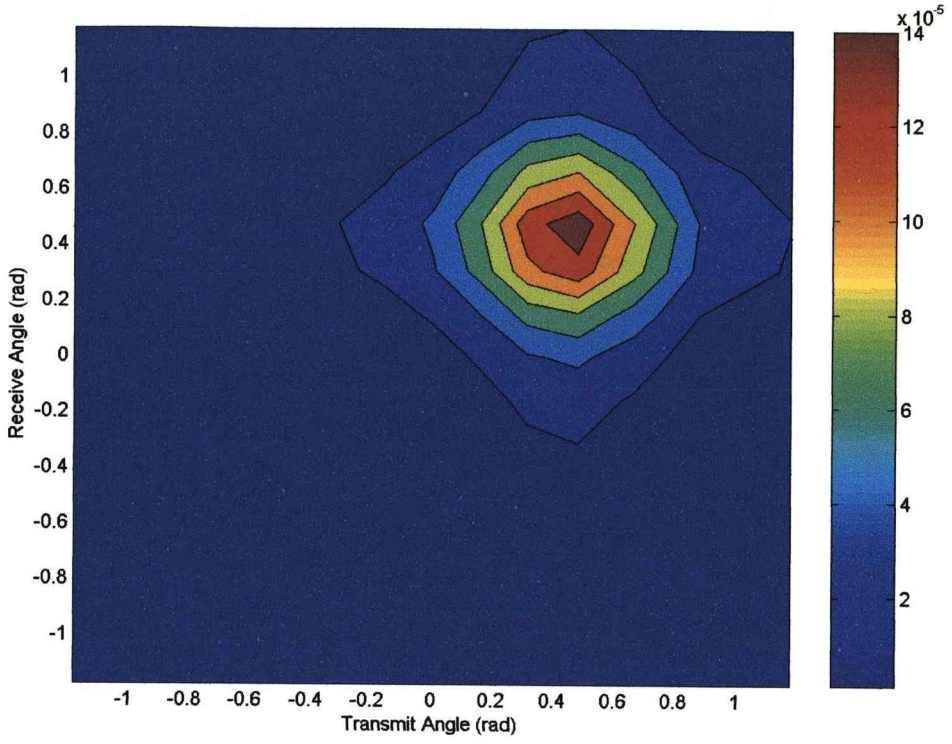

Figure 4.15 - 13-Element Array – (0.2, 0.5) target

4.3.5 Pre-Sum Phase Plots

A 9-element array was chosen with a target contained in the angle (0.46, 0.46) rad. A plot of the H-matrix in a magnitude and angle form is given in Figure 4.16.

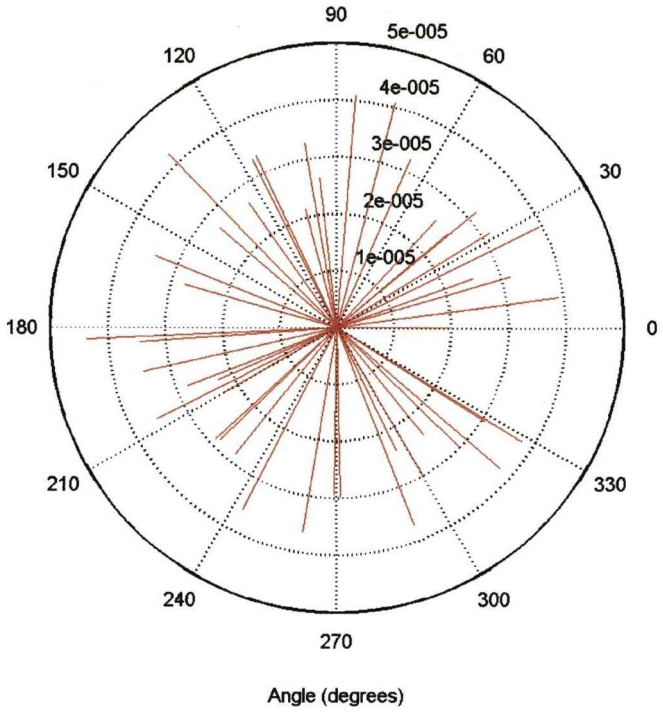

Figure 4.16 - H-Matrix – Magnitude and Phase

The processed matrix for the transmitted and received angles matching that of the targets direction before the complex sum is given in Figure 4.17. It shows that the phases of all the elements converge towards a common phase. Experimental results don't show perfect convergence but it is enough so that the complex sum gives a maximum. When there isn't a match the elements spread out over the whole range leading to partial cancelling and a much smaller complex sum as shown in Figure 4.18.

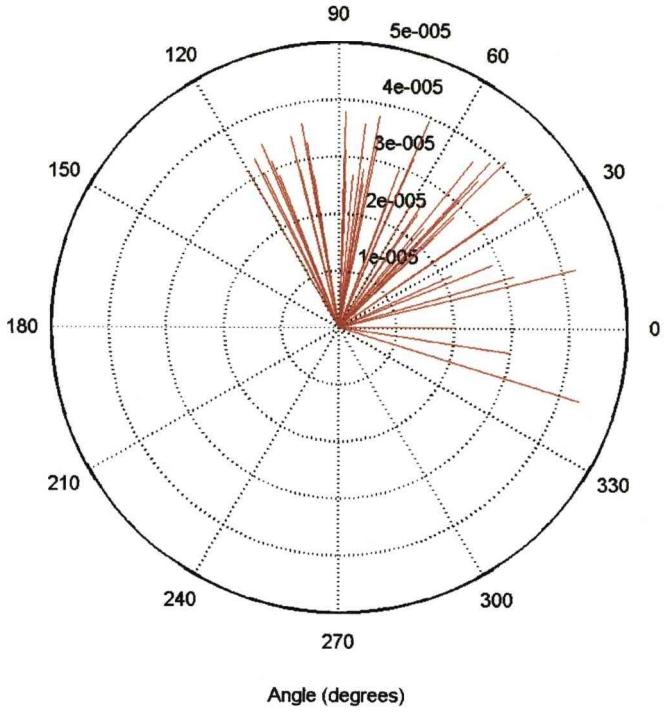

Figure 4.17 - Correct angle choice – Processed Magnitude and Phase

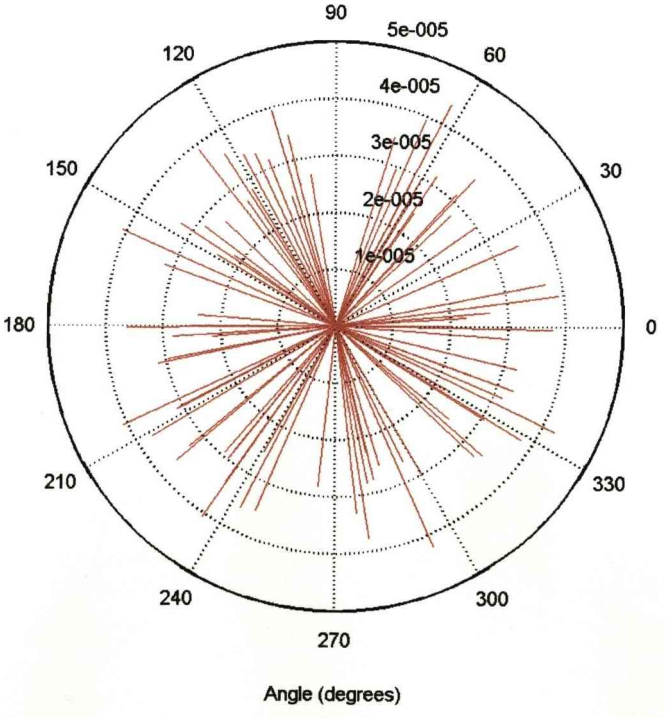

Figure 4.18 - Incorrect angle choice – Processed Magnitude and Phase

4.3.6 Range Limit

As shown there is a minimum range limit at which a single target is resolved as a single target. This limit is important to recognise and can be experimentally acquired as it shows variability with array size. Using the same simulated set up a target is moved from close by the antenna array away along the zero angle transmit/received line. The range at which the adjacent angles show half strength compared to the main detection is recorded and the whole process is repeated for antenna array sizes from 3 to 21.

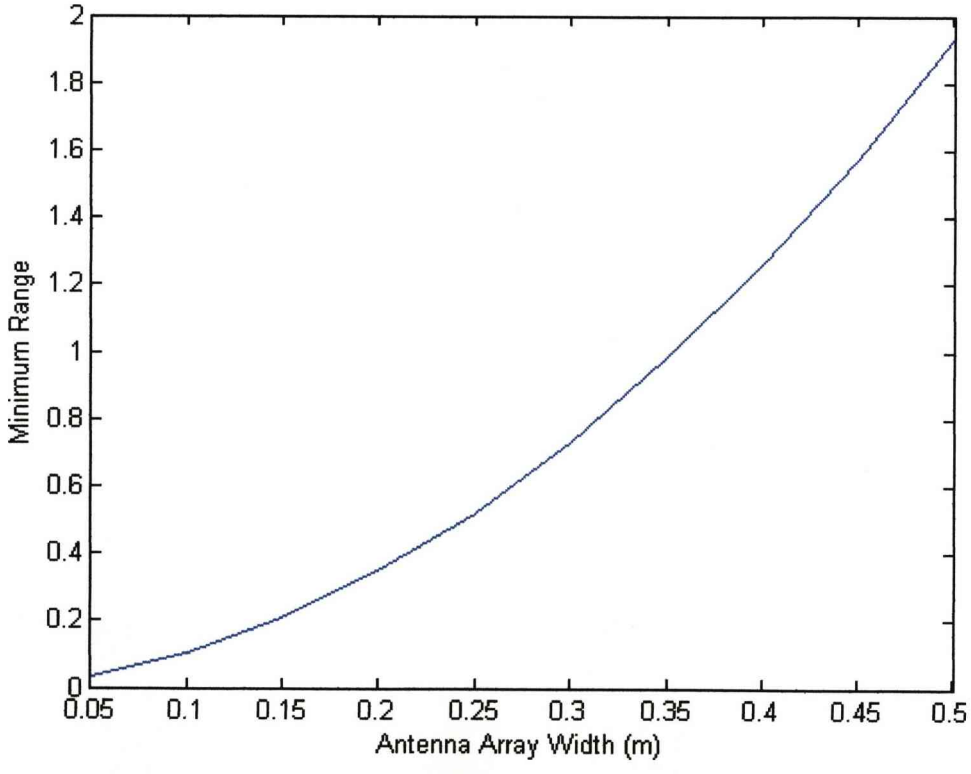

Figure 4.19 - Minimum range vs. antenna array width

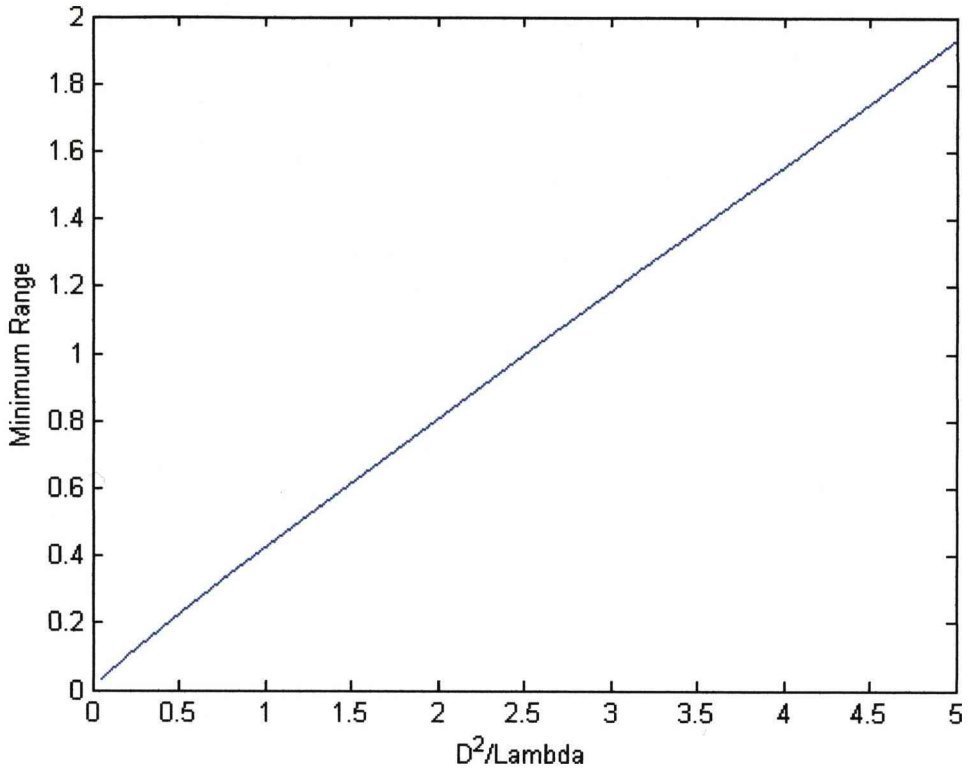

Figure 4.20 - Minimum range vs. D^2/λ

The minimum range shows an upward curve with array width in Figure 4.19. The second plot, Figure 4.20 shows that the minimum range is approximately proportional to D^2/λ. This relationship is usually used to describe the far-field to near-field relation for antenna arrays. It is also something of a problem because better resolution requires a larger number of antenna elements and hence a larger array leading to a potential problem with minimum range.

4.4 Modification

The transform alters the phase of the received signals to match the transmitting or receiving directions in order to maximise the magnitude of the sum for a given output element (that couples a particular transmitting angle with a receiving angle). Alteration of the transformation arrays Eq. (4.11) to create a version that incorporates

a distance measure should be possible. There are two potential benefits, the first being particular distance values producing stronger coupling values hence giving a depth perception. Secondly it would allow larger arrays to focus closer reducing the far field limit to that of the antenna elements instead of the total array.

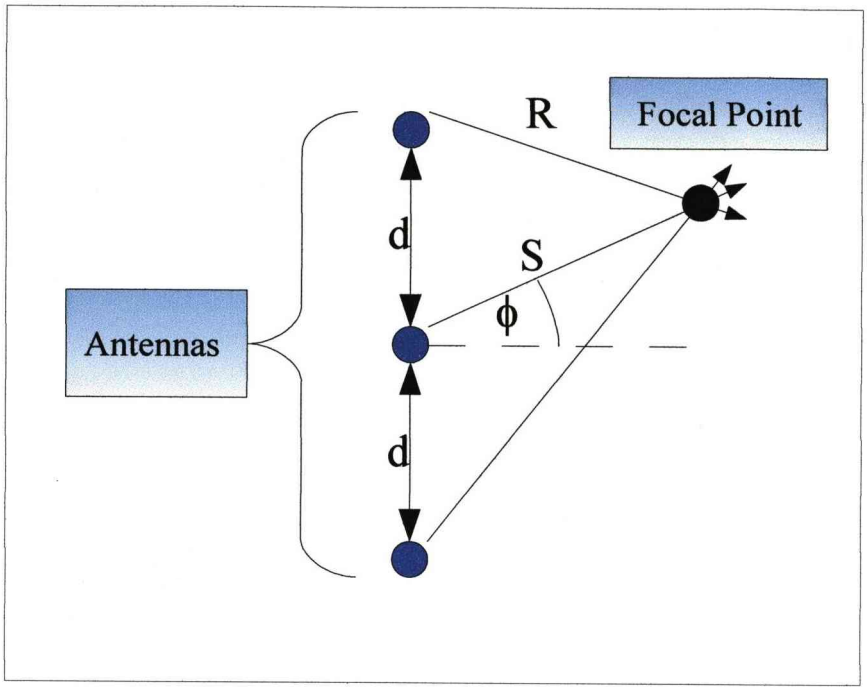

Figure 4.21 - Antenna Array

Using the same antenna array as in Figure 4.21, instead of having all the antenna element beams parallel they are moved inward to a focus point. The focus point is a distance S away from the centre element giving the distance for the adjacent element of R. The key measure that determines the phase alterations of the transform is the difference between the path lengths of S and R for a given angle ϕ. R is calculated using the cosine rule in terms of d, S and f in Eq. (4.34) (R_u is for antenna elements above the centre element and R_l for below) with the path length phase difference, θ, then given in Eq. (4.35)

$$\begin{aligned} R_u &= \sqrt{d^2 + S^2 - 2\,dS\sin(\varphi)} \\ R_l &= \sqrt{d^2 + S^2 + 2\,dS\sin(\varphi)} \end{aligned} \tag{4.34}$$

$$\theta = \frac{S-R}{\lambda}$$
$$\theta = \frac{S-\sqrt{d^2+S^2-2dS\sin(\varphi)}}{\lambda} \tag{4.35}$$

Then substituting the new θ value into an array steering/response vector and altering it for antenna positioning based on variable D gives Eq. (4.36) where D is given in Eq. (4.37)

$$\boldsymbol{\alpha}(\varphi) = \frac{1}{\sqrt{T}}\begin{bmatrix} 1 \\ e^{-j2\pi\left(\frac{S-\sqrt{D_1^2+S^2-2DS\sin(\varphi)}}{\lambda}\right)} \\ \vdots \\ e^{-j2\pi\left(\frac{S-\sqrt{D_T^2+S^2-2DS\sin(\varphi)}}{\lambda}\right)} \end{bmatrix} \tag{4.36}$$

$$D_n = nd \text{ for } -\frac{(T-1)}{2} \le n \le \frac{(T-1)}{2} \tag{4.37}$$

Note that Eq. (4.36) also has a change of variable from θ to ϕ when compared to Eq. (4.6). The correspondence between θ and ϕ is no longer as in Eq. (4.7) so the array steering and response vectors that make up the transform matrices have to use ϕ as the basis. The first version uses the same θ range substituted in for ϕ. For this version the final form is given in Eq. (4.38-40) with the reference antenna element set to antenna 1 although in practice the centre antenna element ($(T-1)/2$) is usually preferable for this method.

$$H_{v_{(x,y)}} = \frac{1}{\sqrt{Q}\sqrt{T}}\sum_{q=1}^{Q}\sum_{t=1}^{T}\alpha_{(t,q)}e^{-j2\pi\beta_{(t,q)}} \tag{4.38}$$

$$\alpha_{(t,q)}=\left[\begin{array}{l}\dfrac{1}{\left(2S_t-\sqrt{\left(d(t-1)\right)^2+S_t^2+2S_t\lambda(t-1)\left(\dfrac{f-\dfrac{T+1}{2}}{T}\right)}\right)^2}\\ \times\dfrac{\Gamma\lambda^2}{(4\pi)^3}\\ \times\dfrac{1}{\left(2S_q-\sqrt{\left(d(q-1)\right)^2+S_q^2+2S_q\lambda(q-1)\left(\dfrac{g-\dfrac{Q+1}{2}}{Q}\right)}\right)^2}\end{array}\right] \tag{4.39}$$

$$\beta_{(t,q)}=\left(\frac{2S_t+2S_q}{\lambda}\right)$$

$$-\frac{1}{\lambda}\left(\begin{array}{l}\sqrt{\left(d(t-1)\right)^2+S_t^2+2S_t\lambda(t-1)\left(\dfrac{f-\dfrac{T+1}{2}}{T}\right)}\\ +\sqrt{\left(d(q-1)\right)^2+S_q^2+2S_q\lambda(q-1)\left(\dfrac{g-\dfrac{Q+1}{2}}{Q}\right)}\end{array}\right) \tag{4.40}$$

$$+\frac{1}{\lambda}\left(\begin{array}{l}\sqrt{\left(d(t-1)\right)^2+K_t^2+2K_t\lambda(t-1)\left(\dfrac{x-\dfrac{T+1}{2}}{T}\right)}\\ +\sqrt{\left(d(q-1)\right)^2+K_q^2+2K_q\lambda(q-1)\left(\dfrac{y-\dfrac{Q+1}{2}}{Q}\right)}\end{array}\right)$$

The actual distance of the target from the reference antenna is given by S (subscript t for transmitter and q for receiver). The corresponding term in the modification part is denoted by variable K. Note that the maximum now occurs when x equals f, y equals

g, K_t equals S_t and K_q equals S_q. The assumption that the target is far enough away to simplify the magnitude term though no longer applies so there is no simple solution.

4.4.1 Results

4.4.1.1 Single Target

Using the same data as that for the previous results section based around the set-up in Figure 4.4 the new modified focussed method is used. There is an additional parameter that needs to be specified, that of the focussing distance (K). There would normally be independence between K for the transmitting and receiving antenna arrays but for a single combined transmit receive array (mono-static) used here a single value can be used. Note that the results for a large value of K approach that of the parallel method. The value of K chosen for long focussing is K=100m, in most cases the images for this value of K are approximately equal to those produced using the parallel version. For short focussing a value of K=0.25m is used putting the focus distance in the middle of the range of target locations.

4.4.1.2 3-Element Array

The three element array like the parallel version has a limited resolution. With only a limited number of elements there is no visible improvement between the long focussing (K=100m) and short focussing (K=0.25m). Comparisons between this and the parallel version show slight variation for the (-0.2, 0.1) target shown in Figures 4.22-23 but no visible difference for the other two target positions shown in Figures 4.24-27.

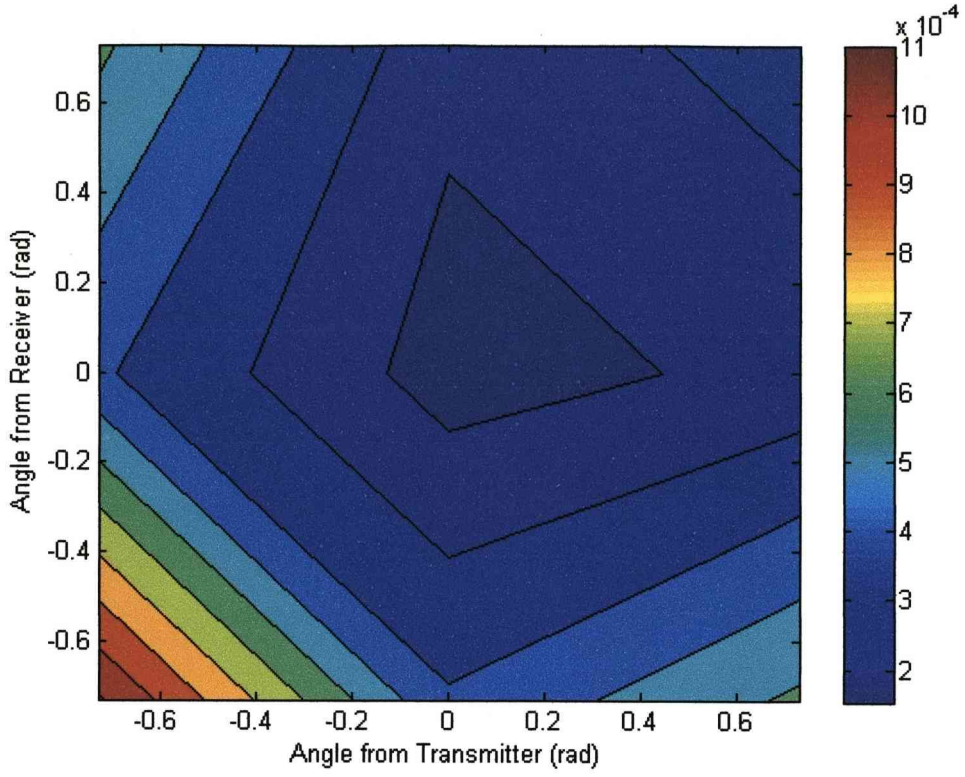

Figure 4.22 - 3-Element array – (-0.2, 0.1) target, K=100m

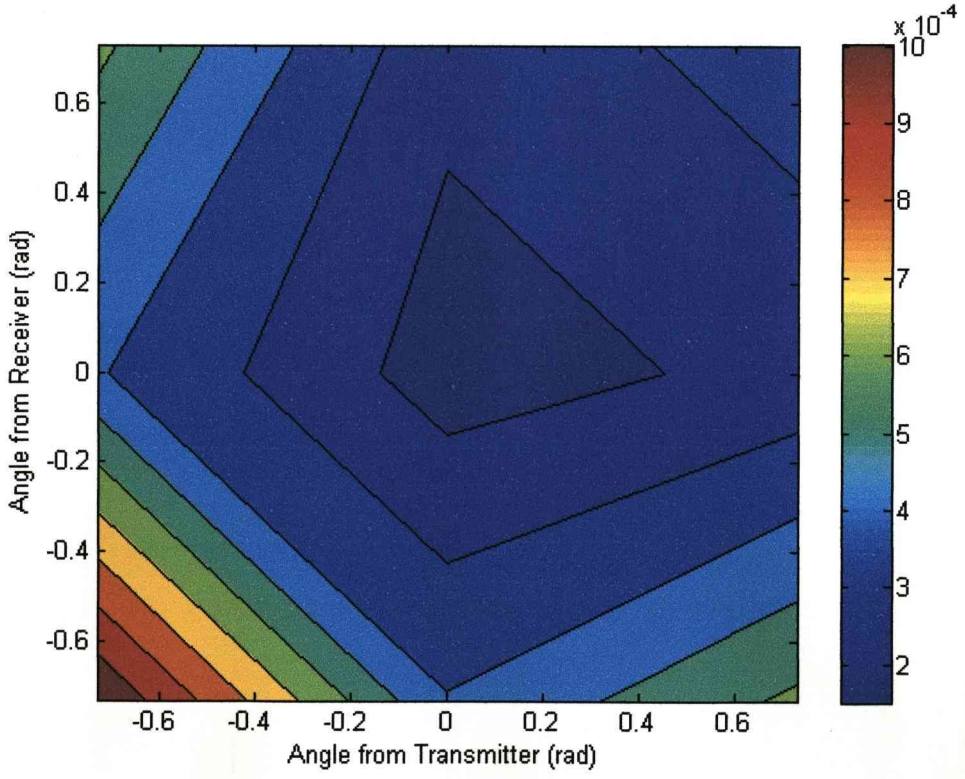

Figure 4.23 - 3-Element array – (-0.2, 0.1) target, K=0.25m

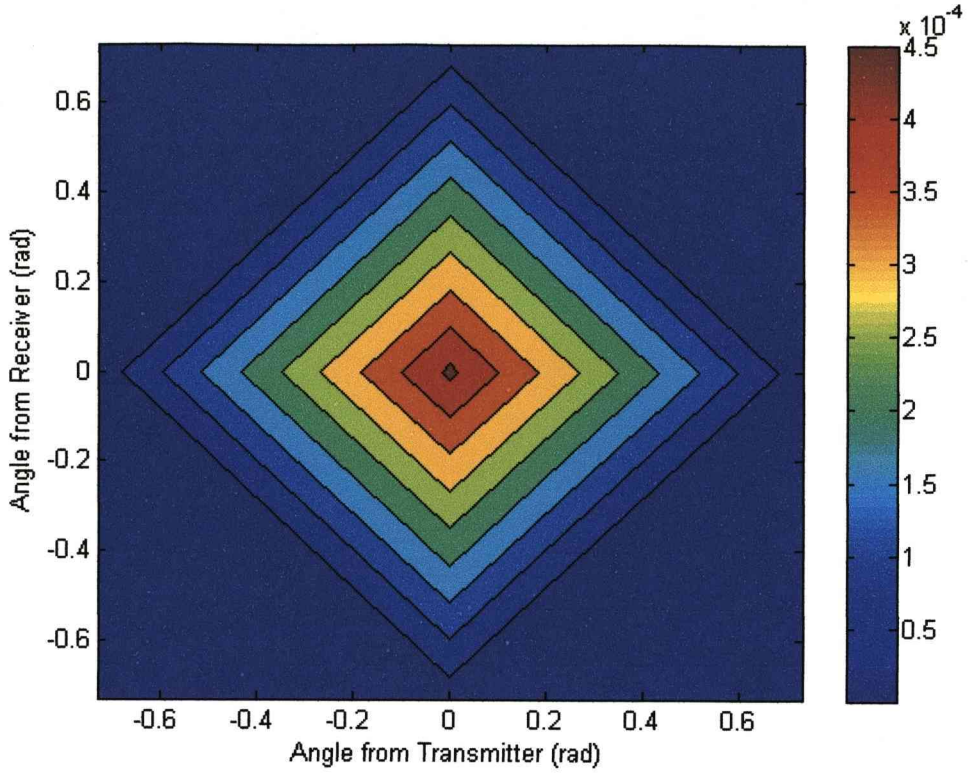

Figure 4.24 - 3-Element array – (0, 0.3) target, K=100m

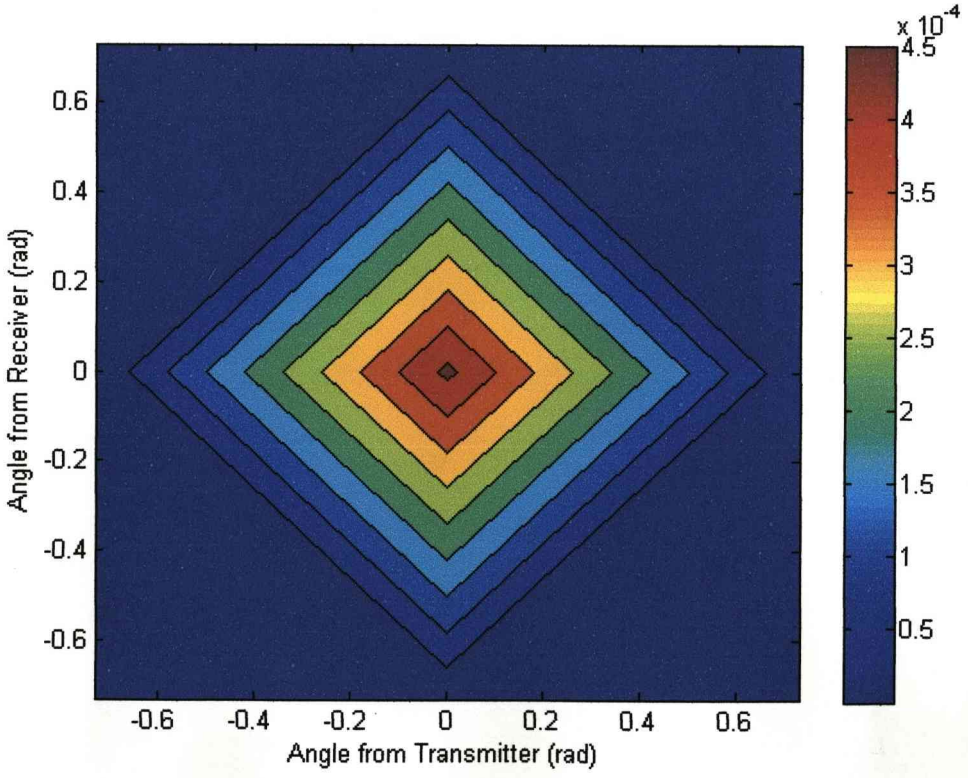

Figure 4.25 - 3-Element array – (0, 0.3) target, K=0.25m

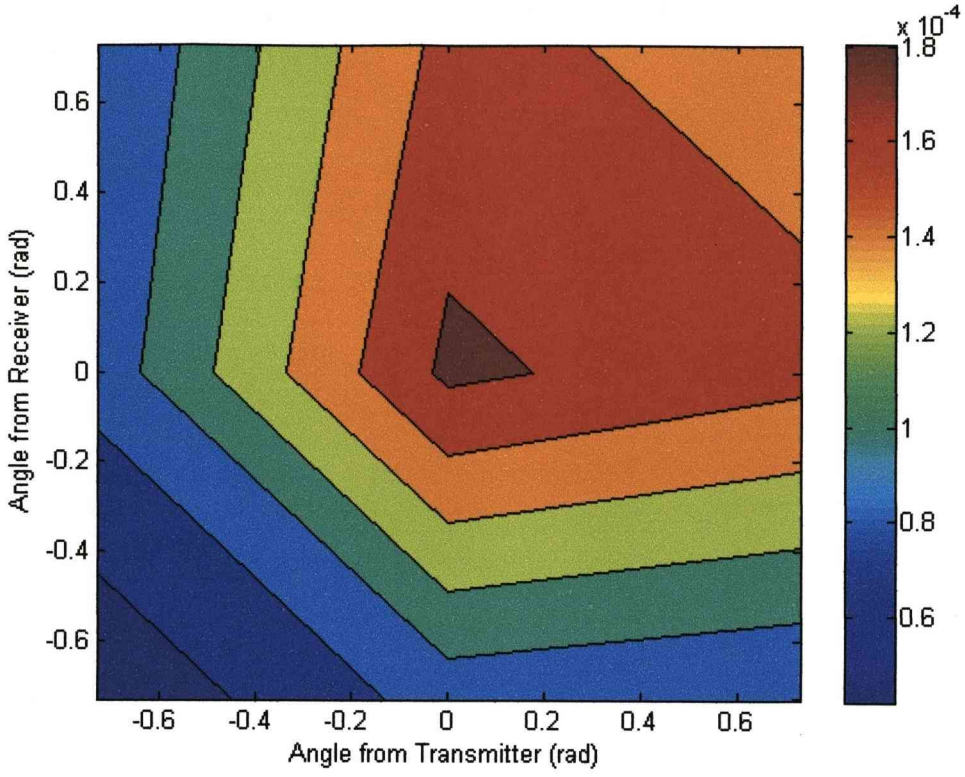

Figure 4.26 - 3-Element array – (0.1, 0.3) target, K=100m

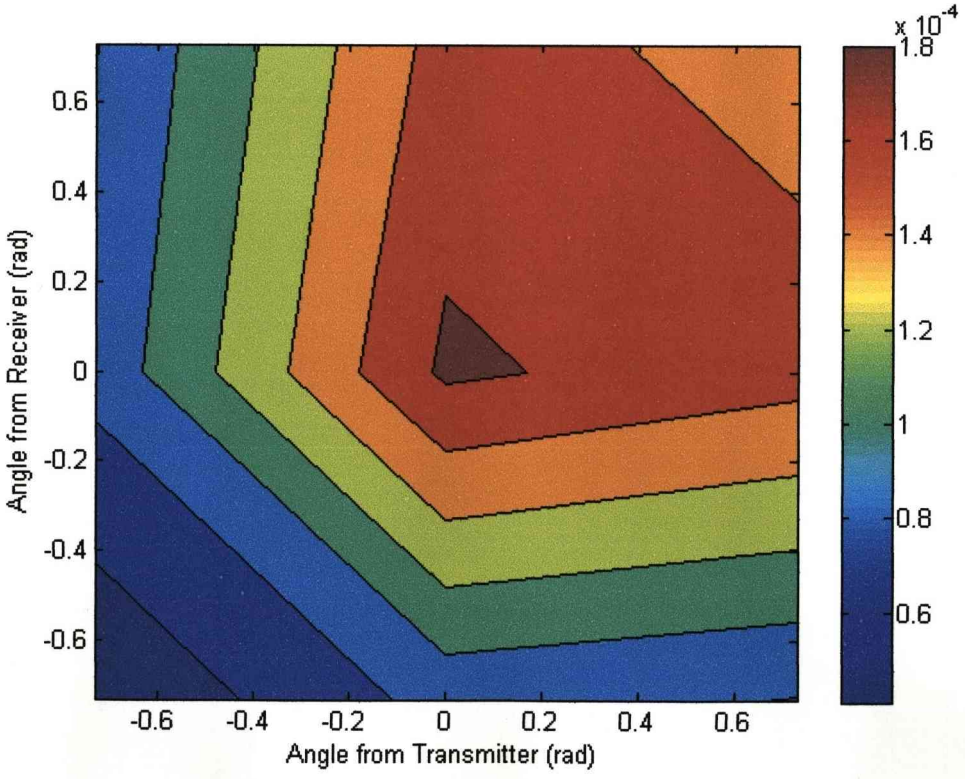

Figure 4.27 - 3-Element array – (0.1, 0.3) target, K=0.25m

4.4.1.3 7-Element Array

This array size starts showing some improvements over the parallel version. The first target at (-0.2, 0.1) shows a slight shrinkage of the target between the long (Figure 4.28) and short (Figure 4.29) focussing but it is relatively minor. The second target at (0, 0.3) has a fairly large reduction in target size between the two focussing values (Figures 4.30-31) which also occurs for the last target at (0.1, 0.3) in Figures 4.32-33.

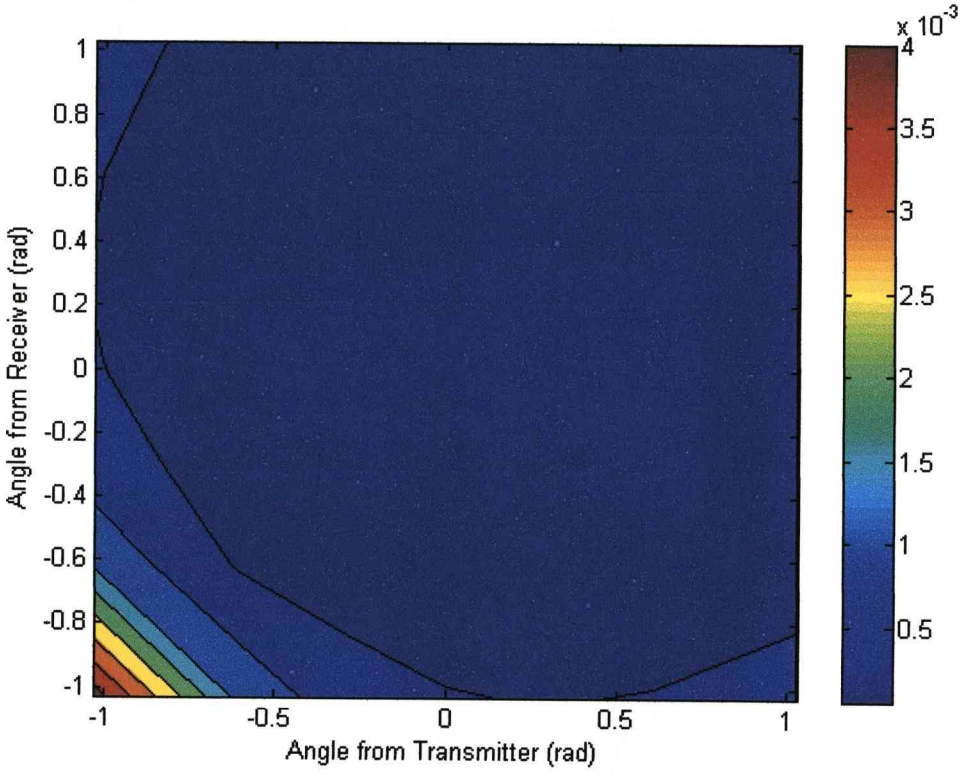

Figure 4.28 - 7-Element array – (-0.2, 0.1) target, K=100m

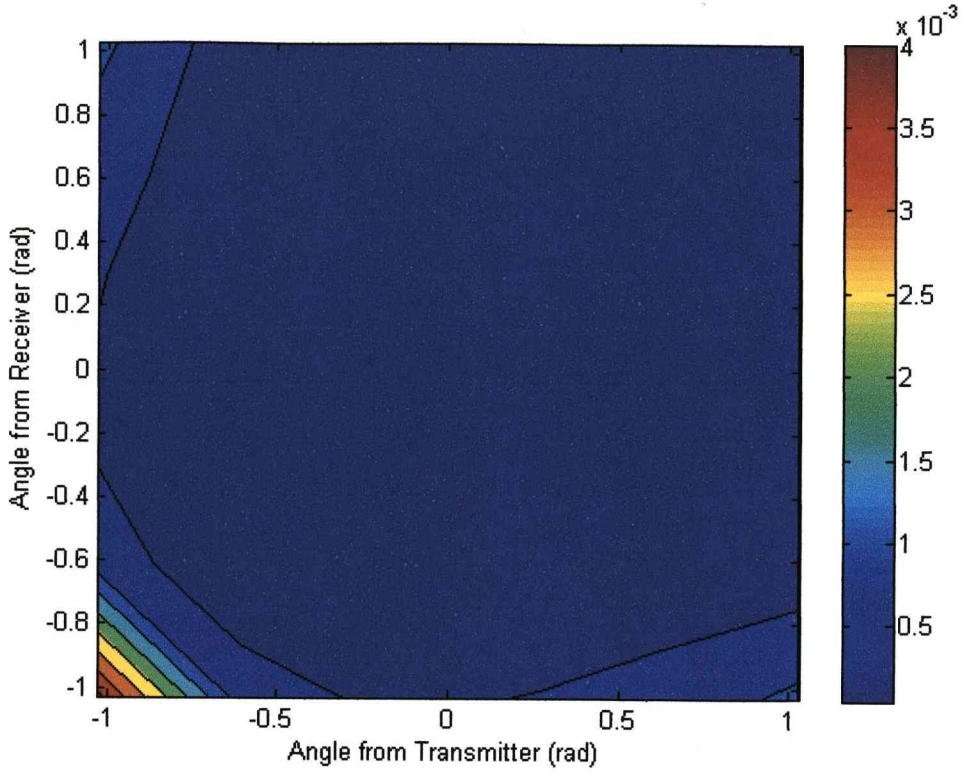

Figure 4.29 - 7-Element array – (-0.2, 0.1) target, K=0.25m

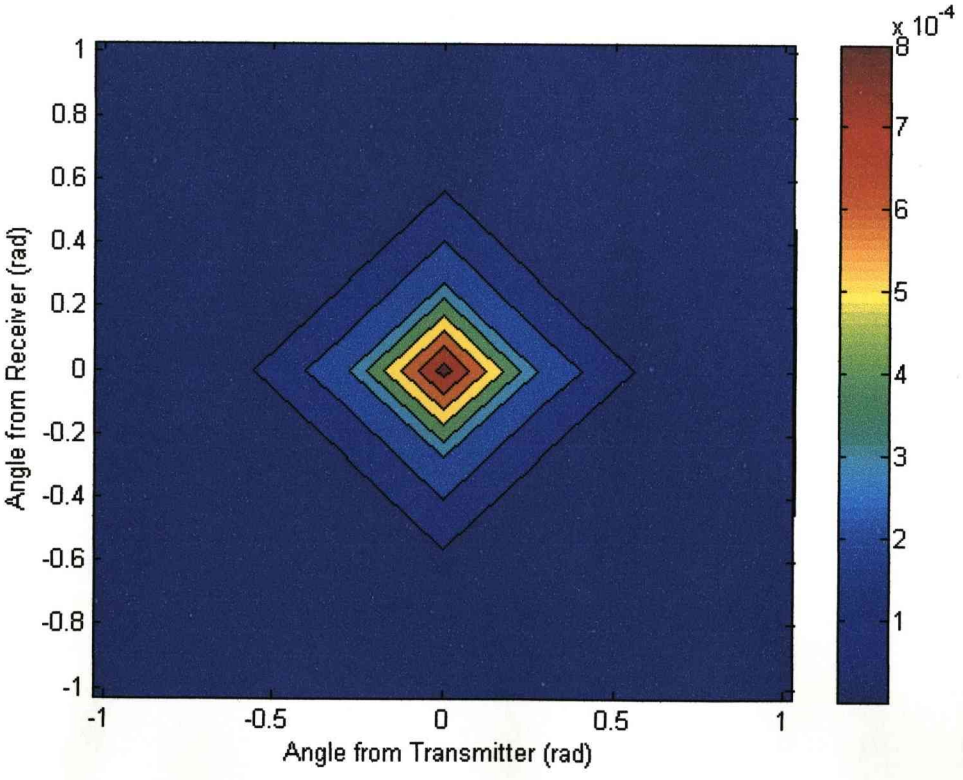

Figure 4.30 - 7-Element array – (0, 0.3) target, K=100m

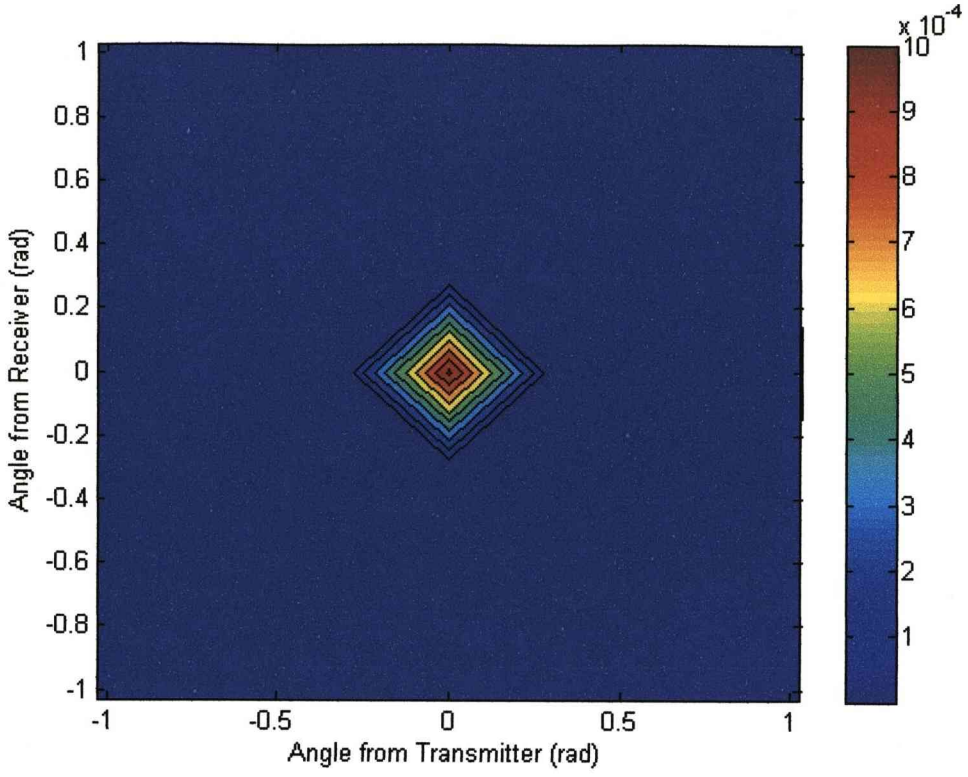

Figure 4.31 - 7-Element array – (0, 0.3) target, K=0.25m

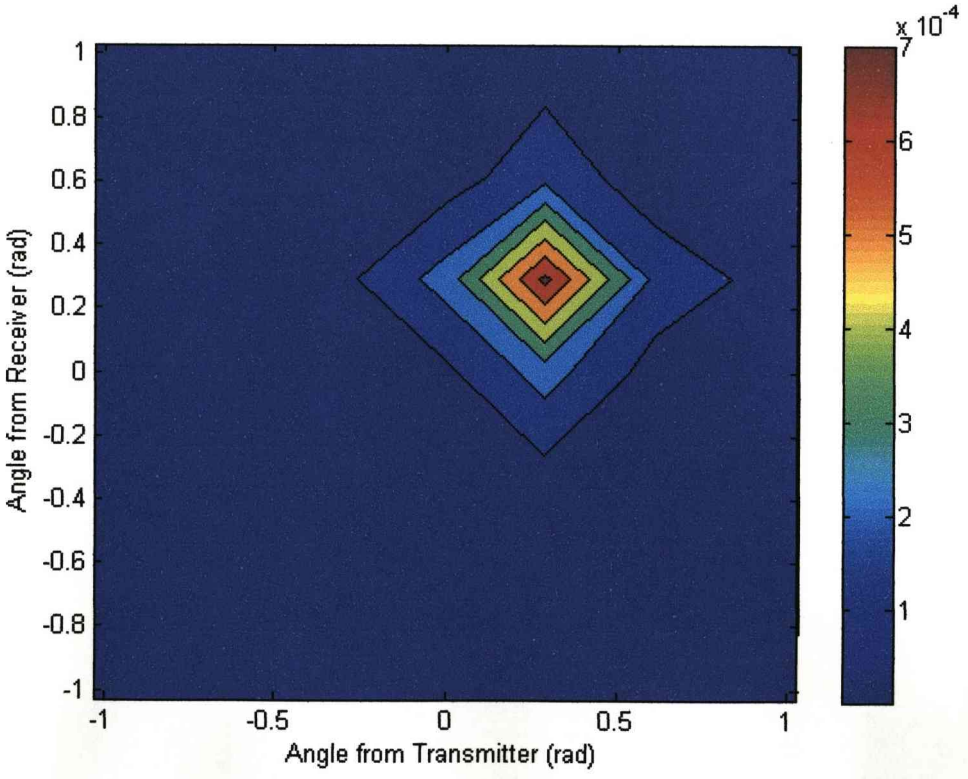

Figure 4.32 - 7-Element array – (0.1, 0.3) target, K=100m

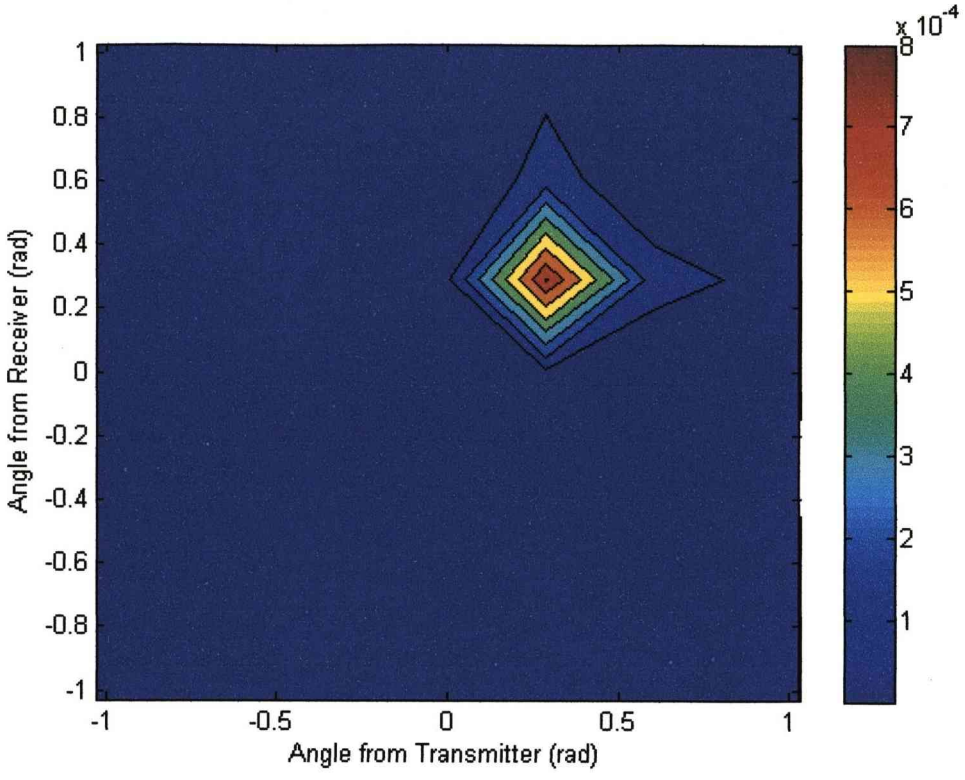

Figure 4.33 - 7-Element array – (0.1, 0.3) target, K=0.25m

4.4.1.4 13-Element Array

With the parallel version this array showed problems with targets too close to the array breaking up with some positioning distortion. The modified focussed version should fix these problems. The break up of the target is clearly visible with the long focus in Figure 4.36 and Figure 4.38 while the target in Figure 4.33 is offset from its true angle. The corresponding short focus images have considerably smaller targets (Figure 4.37 and Figure 4.39) and correct angular positions (Figure 4.35).

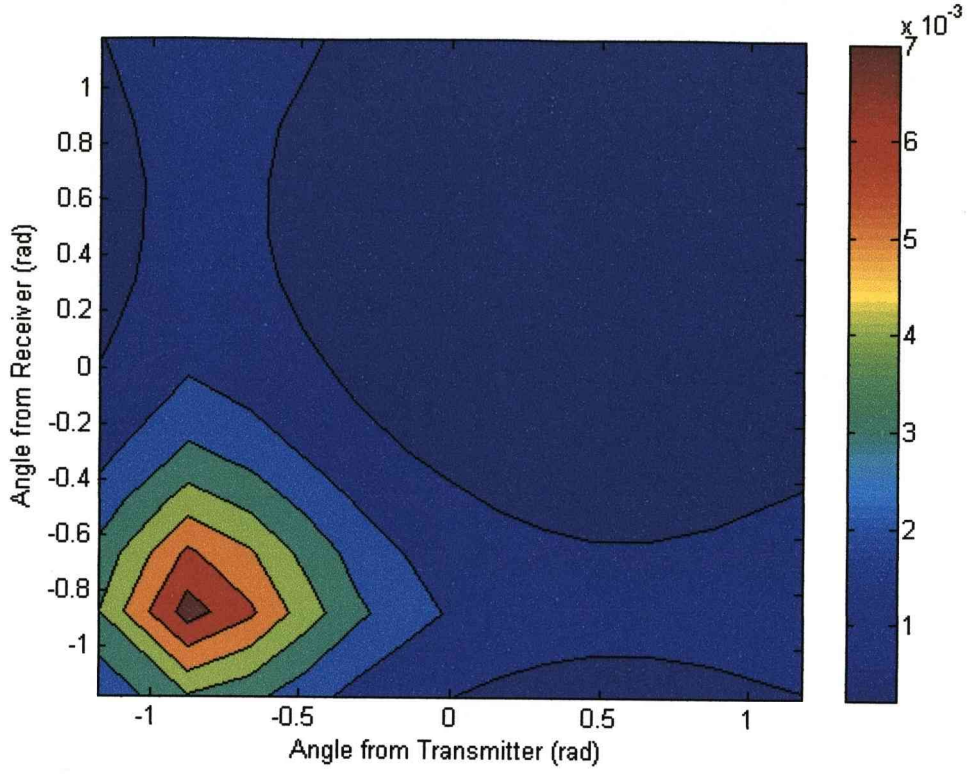

Figure 4.34 - 13-Element array – (-0.2, 0.1) target, K=100m

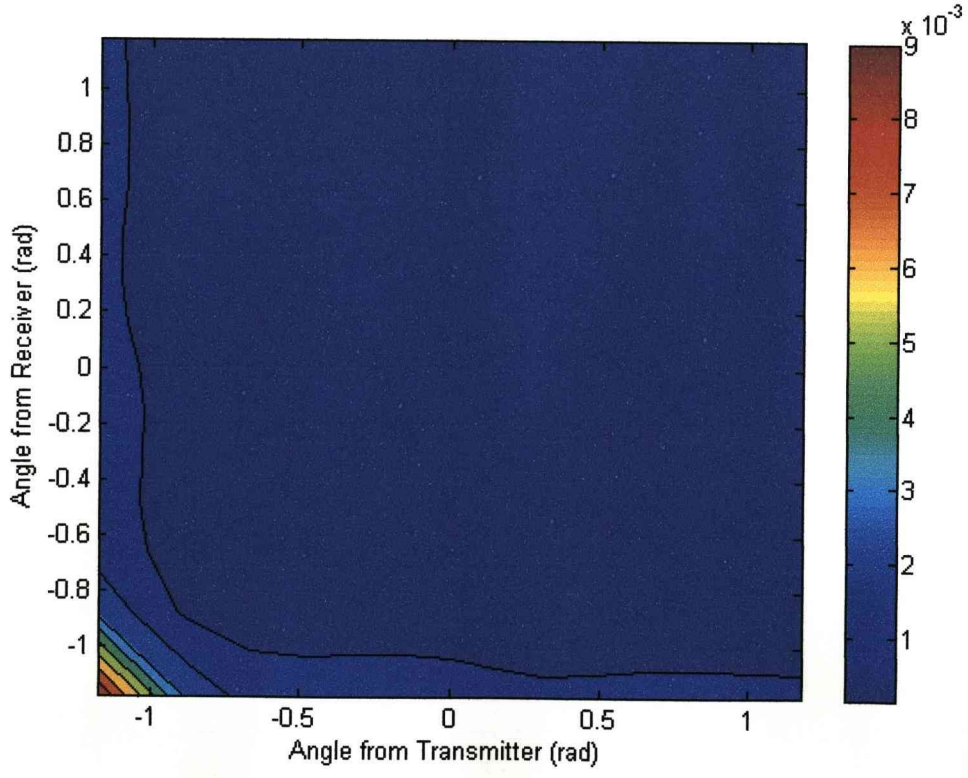

Figure 4.35 - 13-Element array – (-0.2, 0.1) target, K=0.25m

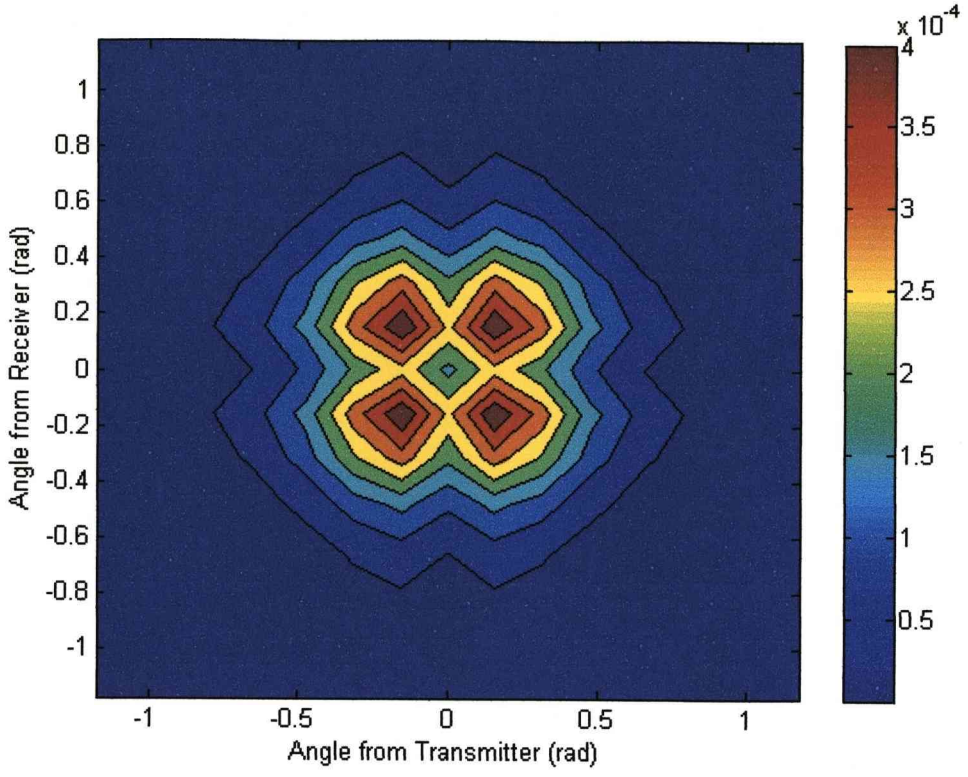

Figure 4.36 - 13-Element array – (0, 0.3) target, K=100m

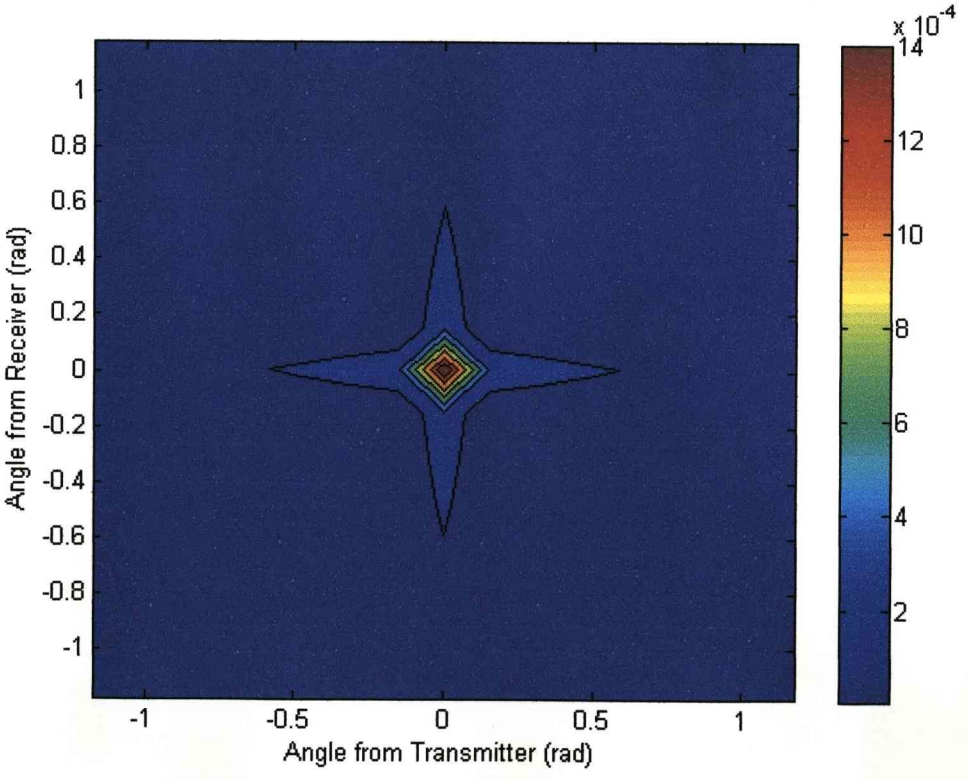

Figure 4.37 - 13-Element array – (0, 0.3) target, K=0.25m

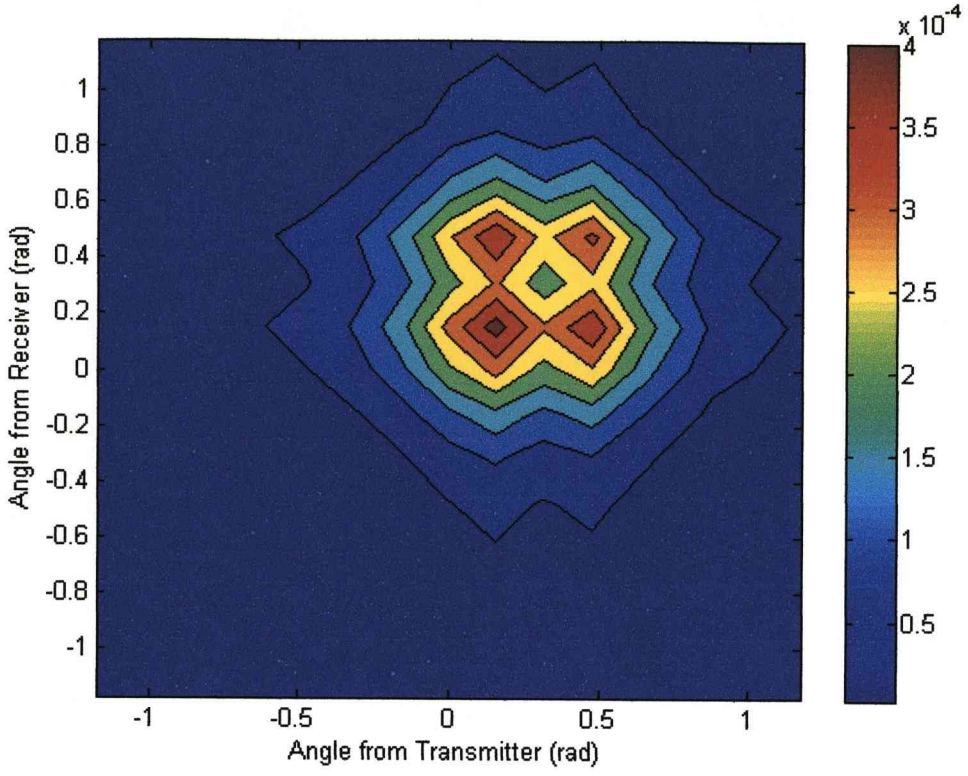

Figure 4.38 - 13-Element array – (0.1, 0.3) target, K=100m

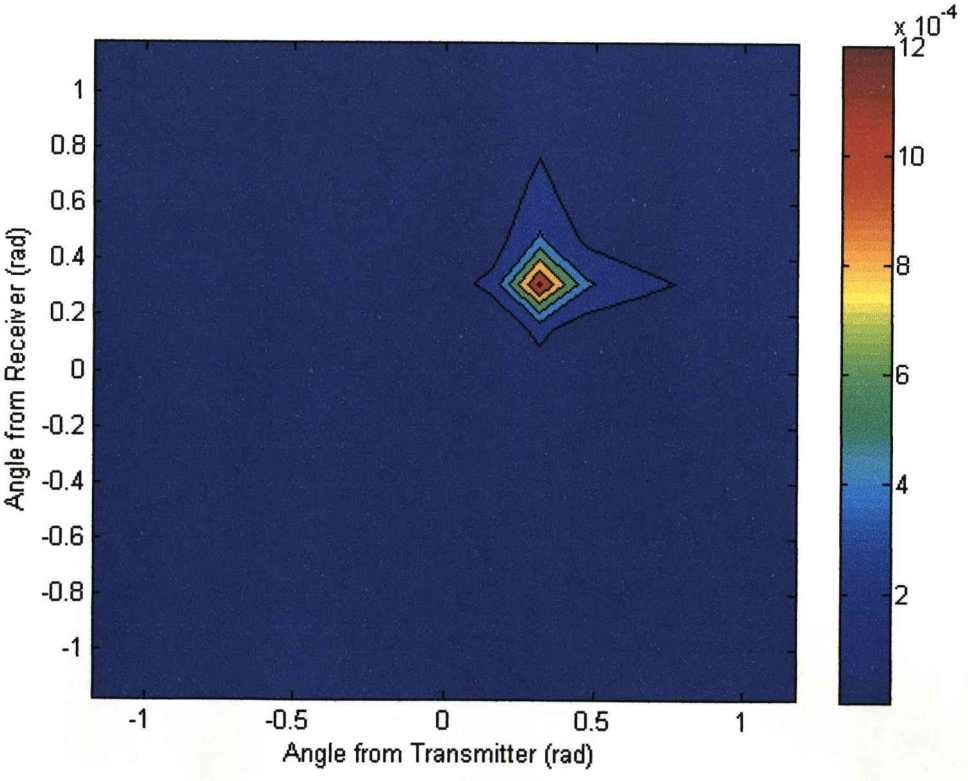

Figure 4.39 - 13-Element array – (0.1, 0.3) target, K=0.25m

4.4.1.5 Range Limit and Focussing Distance

The range limit to achieve good target detection for the parallel version and will also apply to the longer focussed method is approximately proportional to D^2/λ. This will clearly not apply to the short focussed version as that works with close targets when a short focussing distance is used. The same method as before is used, with a number of different array sizes calculated and the minimum distance used is when the values for angles adjacent to the target angle are less than half of the target angles value. In addition the focussing distance is assumed to match that of the target distances (and should give best case scenario results).

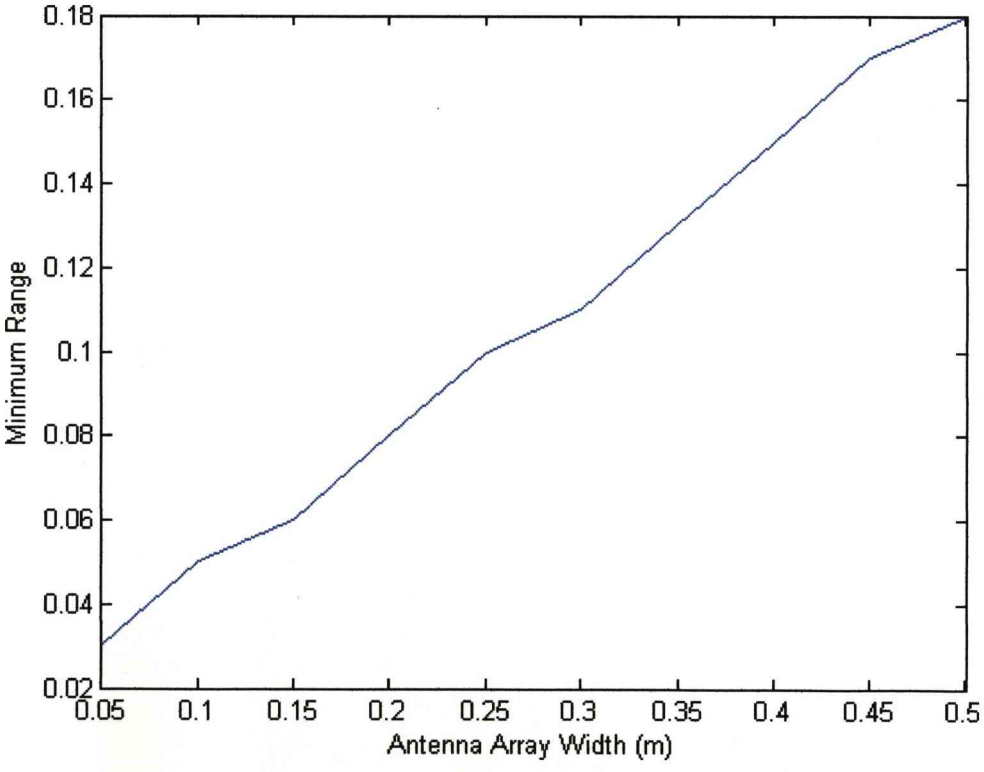

Figure 4.40 - Minimum range vs. antenna array width using matched focussing

The minimum distance is greatly reduced in Figure 4.40 to an approximately straight line (Integer number of array elements at half wavelength spacing causes graph stepping) as opposed to the upward curve in Figure 4.19. This new minimum gives an

approximate proportionality to D instead of D^2 making an overall improvement at correct detection of close targets.

4.4.1.5.1 Influence of *K*

The selection of K has an impact on the image produced. Large values tend towards the parallel method with its limitations and there is potential for problems with very small values relative to the targets actual distance. Two versions are looked at, both using a single target at 0.5m distance, 13 antenna elements used to transmit and receive and all at half wavelength spacing at 6GHz. One target is perpendicular to the array centre (0 rad) and the other at an offset angle ($\pi/4$ rad). The images are calculated using a range of values of K from 0.025m to 4m. Several 2D and 1D cuts through the arrays are presented to show the variation with K.

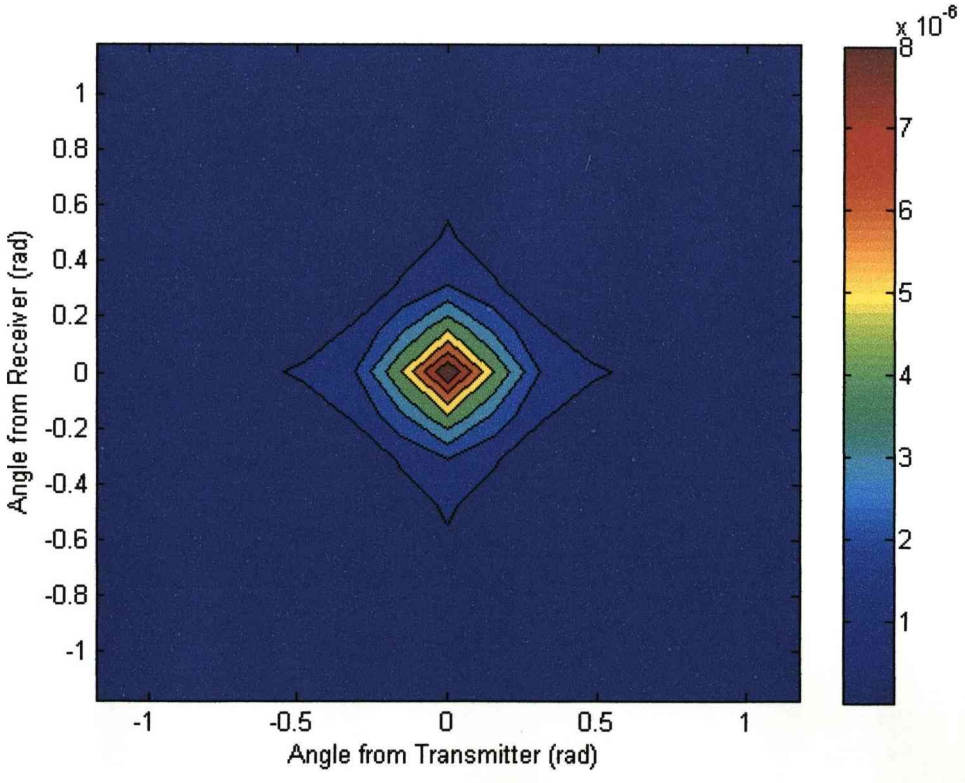

Figure 4.41 - 0 rad target, K=0.5

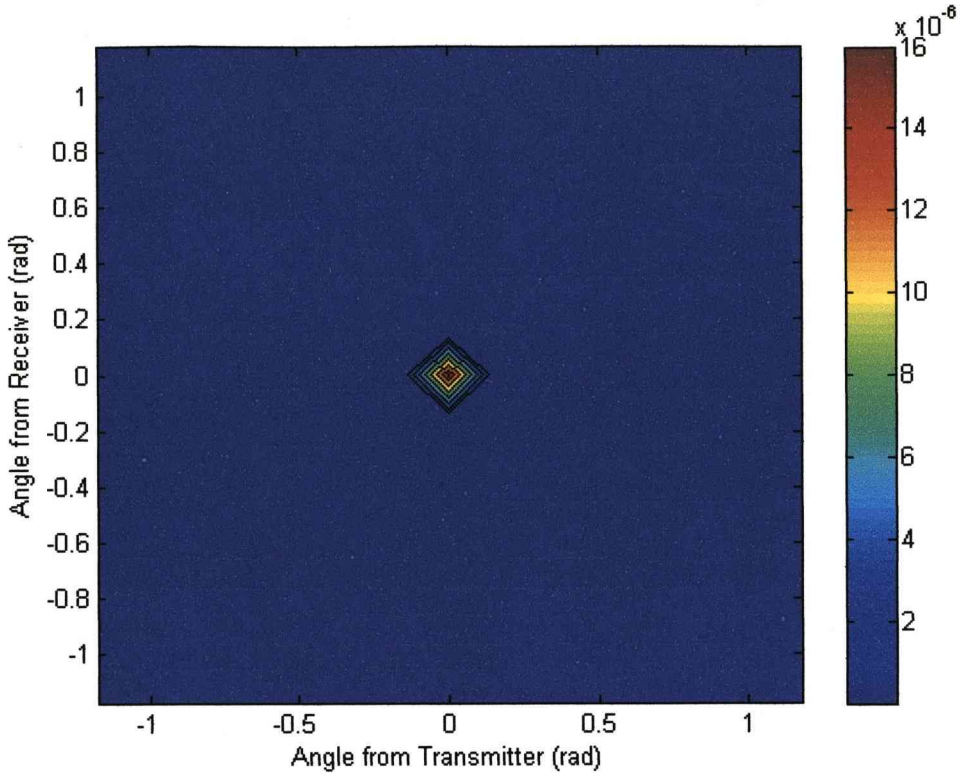

Figure 4.42 - 0 rad target, K=1.275

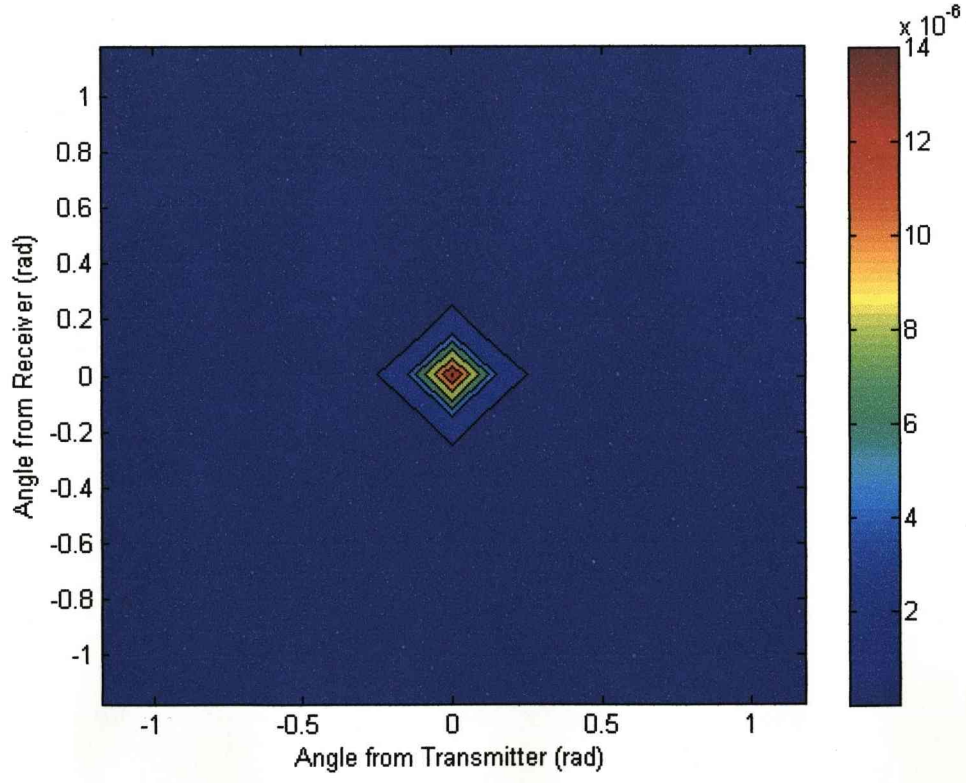

Figure 4.43 - 0 rad target, K=5.025

Figures 4.41-43 show the differences associated with K in regard to the target bleeding through to adjacent angles. Note that the optimum choice of K like in Figure 4.41 is not the value that corresponds to the targets distance exactly as shown by improved results for a larger K value as in Figure 4.42. Figure 4.43 moves closer to the parallel version and shows similar results to that.

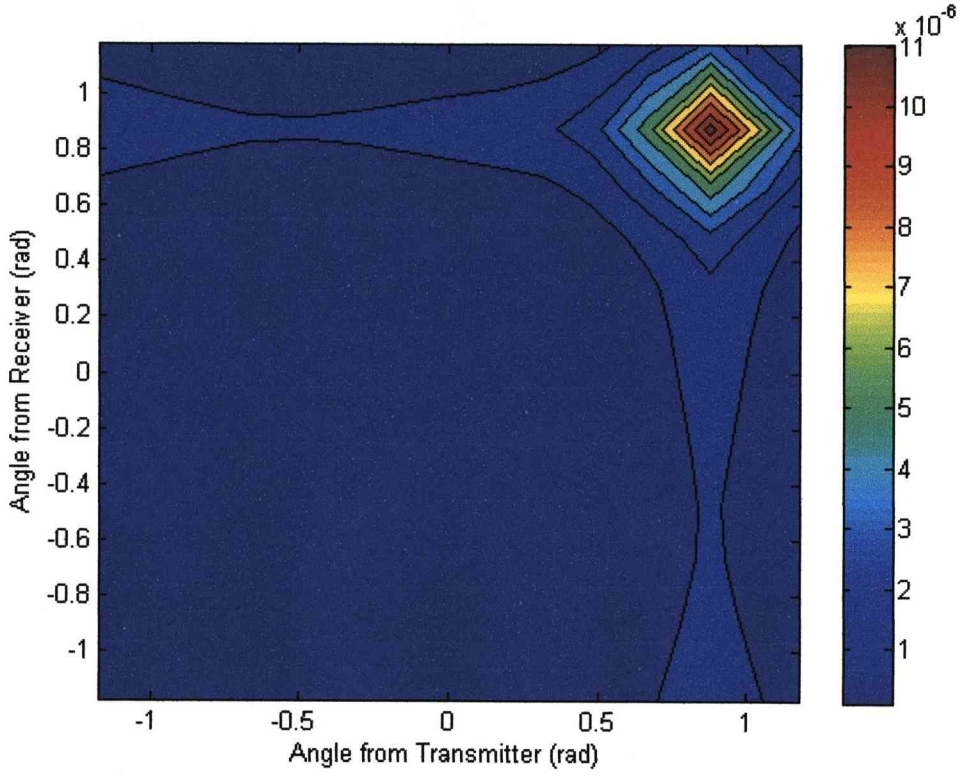

Figure 4.44 - $\pi/4$ rad target, K=0.5

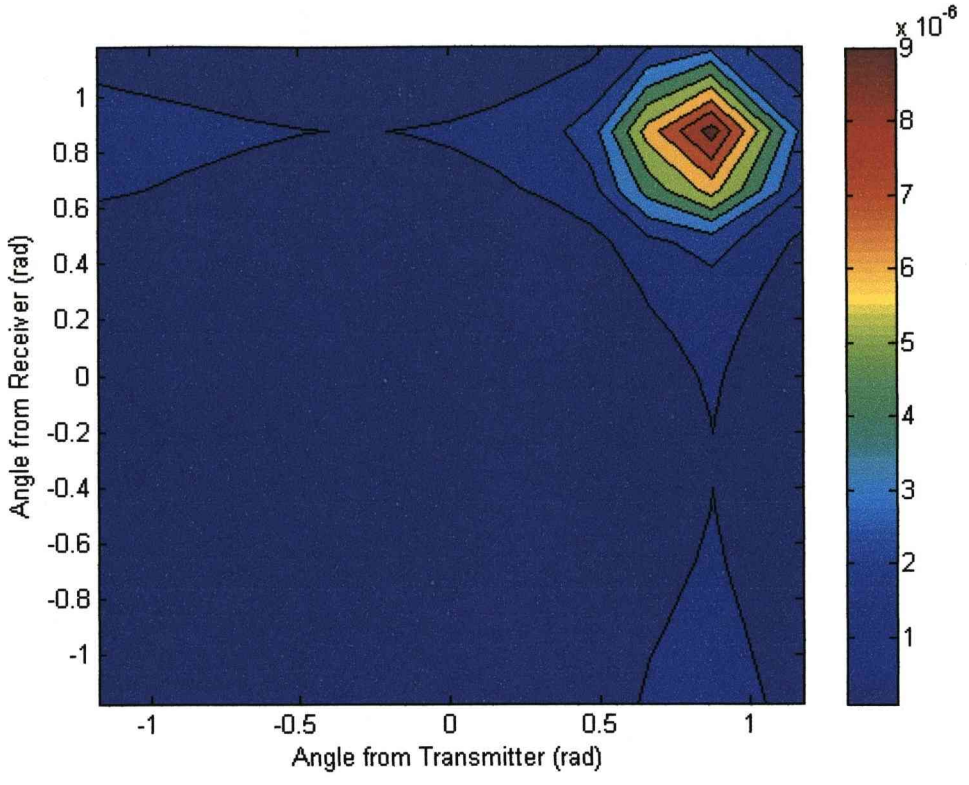

Figure 4.45 - $\pi/4$ rad target, K=1.275

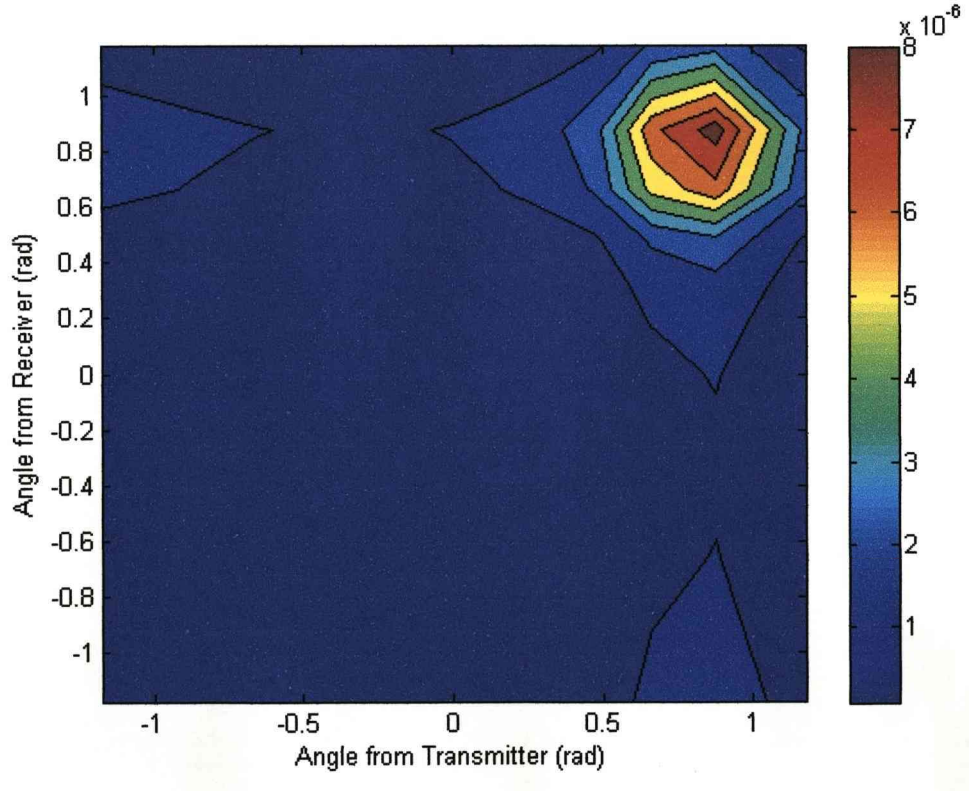

Figure 4.46 - $\pi/4$ rad target, K=5.025

Figures 4.44-46 show the effects of K for an off angle target that doesn't lie in the centre of one of the angles used. In this case the best results are when the target distance and K match like in Figure 4.44 with slightly degrading results for larger value of K.

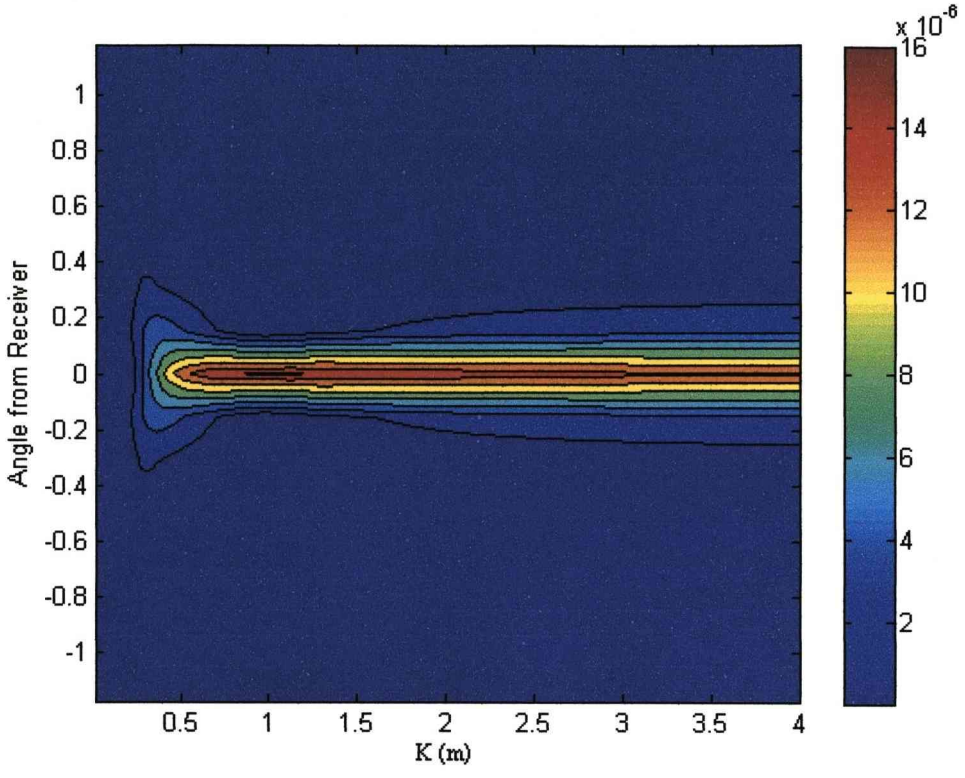

Figure 4.47 - 2D vertical cut through the 0 rad target centre

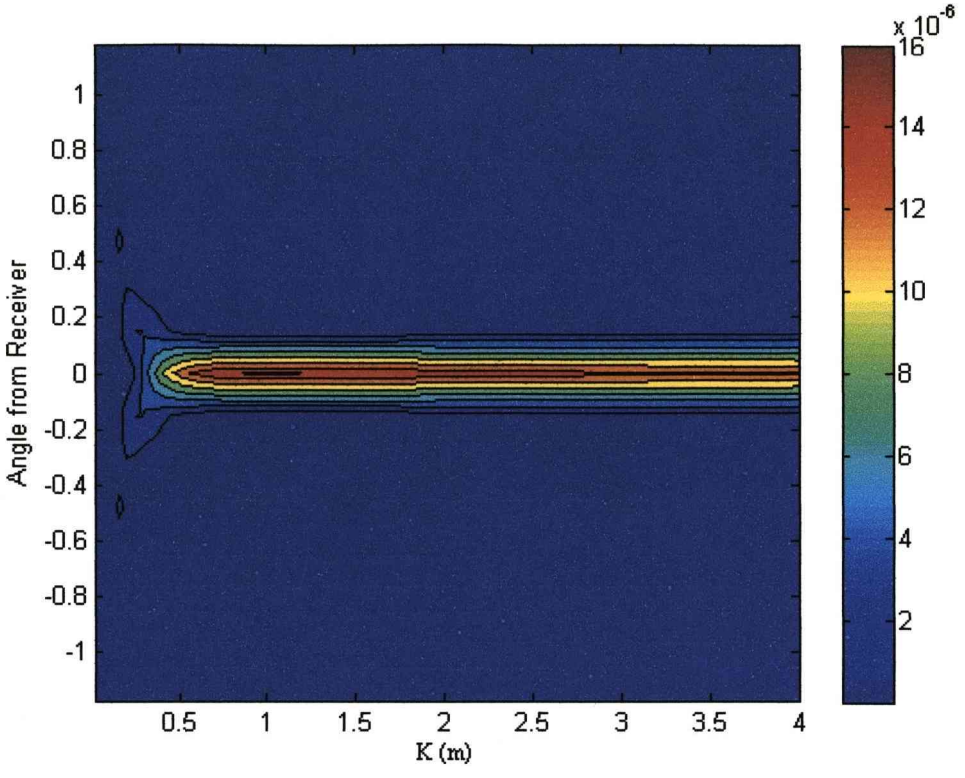

Figure 4.48 - 2D diagonal cut through the 0 rad target centre

The vertical cut in Figure 4.47 and diagonal cut in Figure 4.48 show several key points. The first point is that very small values of K that cause focussing a significant distance in front of the target do not work at all. The vertical cut in Figure 4.47 shows a stronger bleeding to adjacent angles than the diagonal cut in Figure 4.48, particularly for larger K values. The optimum value appears in the K=0.7 to K=1.2 range which is slightly larger than the targets distance of 0.5m.

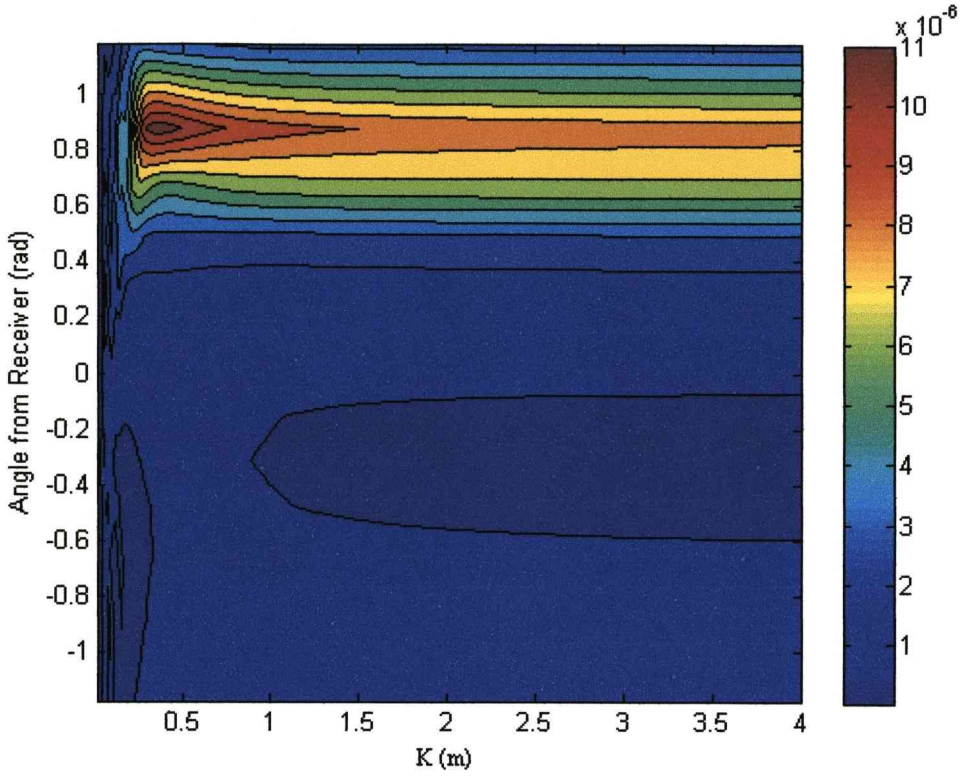

Figure 4.49 - 2D vertical cut through the π/4 rad target centre

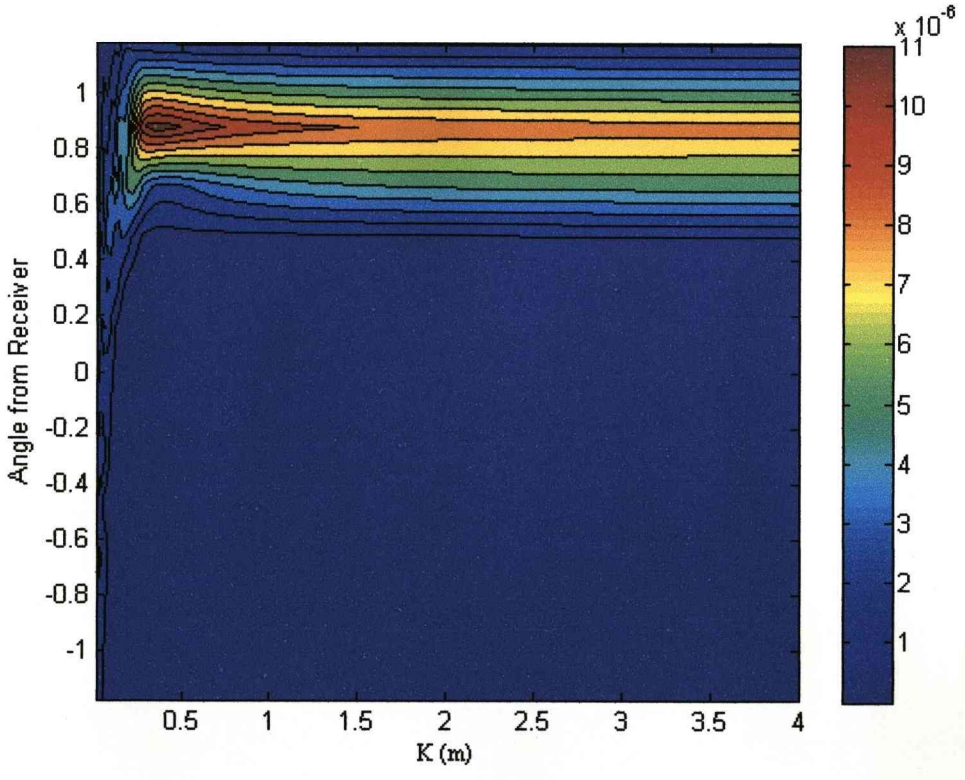

Figure 4.50 - 2D diagonal cut through the π/4 rad target centre

For the offset target the optimum value of K differs to the range K=0.4 to K=0.5. Much smaller values than these show considerable clutter in Figures 4.49-50 and larger values show a steeper decrease from the peak than for the 0 rad target. Again the vertical cut shows slightly more adjacent angle bleeding in Figure 4.49 than the diagonal cut in Figure 4.50.

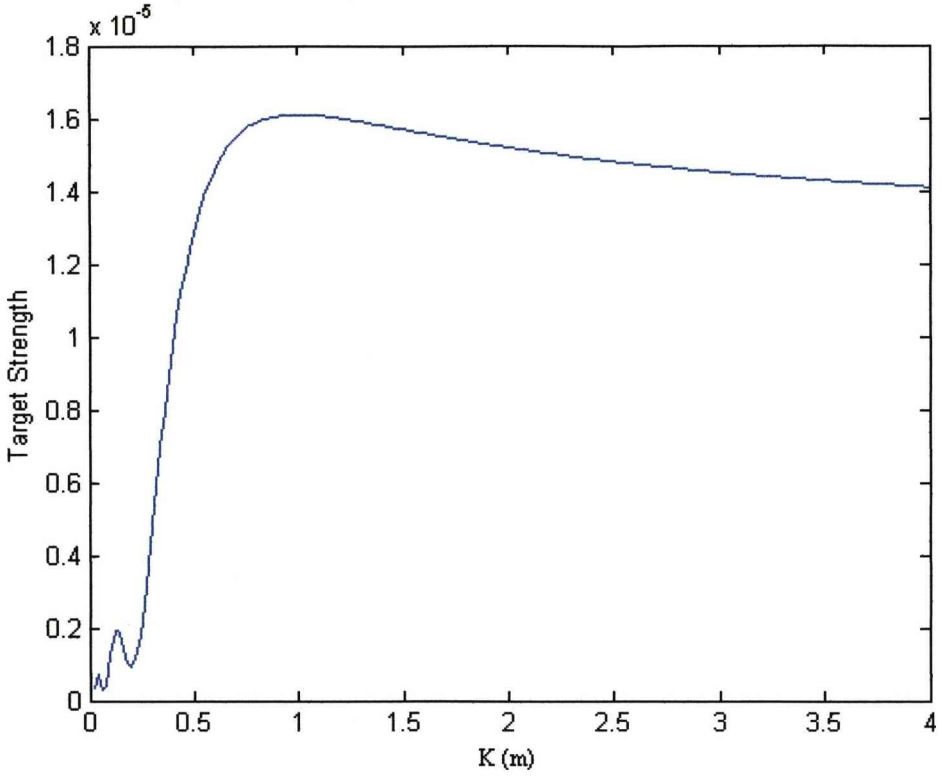

Figure 4.51 - 1D cut through the 0 rad target centre

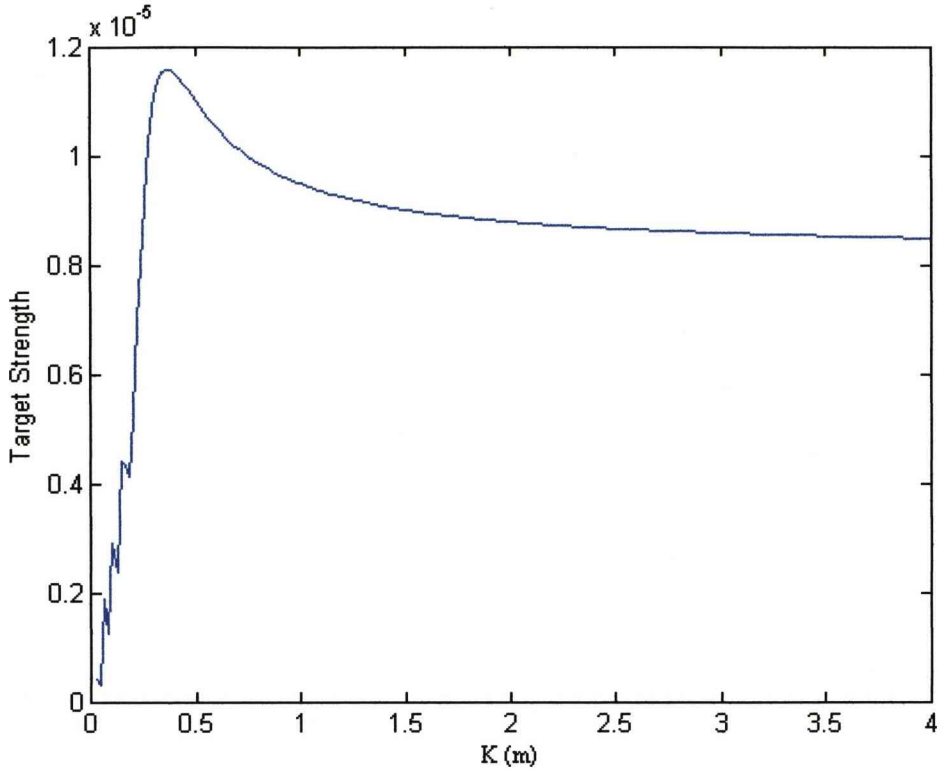

Figure 4.52 - 1D cut through the π/4 rad target centre

Taking a 1D cut through targets centre shows a strong initial upward section with a few clutter elements but then goes to a peak and a slow decrease to a steady state value, that being the parallel versions solution. The 0 rad target in Figure 4.51 has a slower ascent with peak significantly further away than the actual targets distance. For the π/4 rad target in Figure 4.52 the peak is much closer to the actual targets distance but does precede it and the peak is also proportionally larger than the eventual steady state value compared to the 0 rad target.

4.5 Comparison and Analysis

Ignoring the individual processes and values used to calculate $\boldsymbol{H}_v$ as well as the power related complex conjugate the fundamental version can be given as Eq. (4.63) where a, b, c and d are array indices.

$$H_{v_{(c,d)}} = \sum_{a=1}^{T} \sum_{b=1}^{Q} H_{(a,b)} A_{(c,a)} A_{(d,b)} \tag{4.41}$$

A is given by Eq. (4.64) and $\theta_{(x,y)}$ given by Eq. (4.65) for the parallel implementation and Eq. (4.66) for the focussed implementation.

$$A_{(x,y)} = \frac{1}{\sqrt{X}} e^{-j2\pi\theta_{(x,y)}} \tag{4.42}$$

$$\theta_{(x,y)} = (x-1)\frac{\left(y - \frac{X-1}{2} - 1\right)}{X} \tag{4.43}$$

$$\theta_{(x,y)} = P - \sqrt{\left(d\left(x - \frac{X-1}{2} - 1\right)\right)^2 + P^2 - \frac{2P\lambda}{X}\left(x - \frac{X-1}{2} - 1\right)\left(y - \frac{X-1}{2} - 1\right)} \tag{4.44}$$

$\theta_{(x,y)}$ of either version is merely the path length difference between the current element (x) and a reference element for a given value of y (in this case element 1, the lowest element for Eq. (4.65) and the centre element for Eq. (4.66)). y is used for a linear angular spread in Eq. (4.65) and as a component part in Eq. (4.66).

Assuming a 5×5 array gives the following values for $\boldsymbol{H}_v$ in Eq. (4.67), $A_{(x,y)}$ in Eq. (4.68), $\theta_{(x,y)}$ parallel in Eq. (4.69) and $\theta_{(x,y)}$ focussed in Eq. (4.70).

$$\begin{aligned}
H_{v_{(1,1)}} = {} & H_{(1,1)} A_{(1,1)} A_{(1,1)} + H_{(1,2)} A_{(1,1)} A_{(1,2)} + H_{(1,3)} A_{(1,1)} A_{(1,3)} \\
& + H_{(1,4)} A_{(1,1)} A_{(1,4)} + H_{(1,5)} A_{(1,1)} A_{(1,5)} + H_{(2,1)} A_{(1,2)} A_{(1,1)} \\
& + H_{(2,2)} A_{(1,2)} A_{(1,2)} + H_{(2,3)} A_{(1,2)} A_{(1,3)} + H_{(2,4)} A_{(1,2)} A_{(1,4)} \\
& + H_{(2,5)} A_{(1,2)} A_{(1,5)} + H_{(3,1)} A_{(1,3)} A_{(1,1)} + H_{(3,2)} A_{(1,3)} A_{(1,2)} \\
& + H_{(3,3)} A_{(1,3)} A_{(1,3)} + H_{(3,4)} A_{(1,3)} A_{(1,4)} + H_{(3,5)} A_{(1,3)} A_{(1,5)} \\
& + H_{(4,1)} A_{(1,4)} A_{(1,1)} + H_{(4,2)} A_{(1,4)} A_{(1,2)} + H_{(4,3)} A_{(1,4)} A_{(1,3)} \\
& + H_{(4,4)} A_{(1,4)} A_{(1,4)} + H_{(4,5)} A_{(1,4)} A_{(1,5)} + H_{(5,1)} A_{(1,5)} A_{(1,1)} \\
& + H_{(5,2)} A_{(1,5)} A_{(1,2)} + H_{(5,3)} A_{(1,5)} A_{(1,3)} + H_{(5,4)} A_{(1,5)} A_{(1,4)} \\
& + H_{(5,5)} A_{(1,5)} A_{(1,5)}
\end{aligned} \tag{4.45}$$

$$A_{(x,y)} = \frac{1}{\sqrt{5}} e^{-j2\pi\theta_{(x,y)}} \tag{4.46}$$

$$\theta_{(x,y)} = (x-1)\frac{(y-3)}{5} \tag{4.47}$$

$$\theta_{(x,y)} = P - \sqrt{(d(x-3))^2 + P^2 - \frac{2P\lambda}{5}(x-3)(y-3)} \tag{4.48}$$

Looking specifically at each element a pattern exists. The ninth element of Eq. (4.67) has the following procedure

- Take the value of $H(2, 4)$
- $H(2, 4)$ is the signal from transmitter 2 to the target and back to receiver 4.
- Perform a multiplication by $A(1, 2)$
- $A(1, 2)$ is the phase difference between a transmission from transmitter 2 and a transmission from the reference transmitter (T_x 3)
- Perform a multiplication by $A(1, 4)$
- $A(1, 4)$ is the phase difference between reception from receiver 4 and reception from the reference receiver (R_x 3)

Basically the value of $\boldsymbol{H_{(x,y)}}$ is phase altered by the expected difference between the element transmitter and receiver and the reference transmitter and receiver before being complex summed. As the target can appear anywhere within angle to angle range the system very rarely produces an exact phase match over all received signals after processing.

4.6 Angular to Cartesian

One of the possible problems is that sometimes a Cartesian based reference is easier to work with than the angular version the method produces. A straight conversion will

lead to a simple triangular idea of where the target is getting larger as distance increases like in Figure 4.53. A more useful idea is to segment the H-matrix and generate many angular images with slightly different origin points before combining them all to narrow down the angular spreading after the conversion. For this a 9-element array with half wavelength spacing at 6GHz is used. A perfectly reflecting target is positioned 0.4m away from the centre of the array and offset by 0.2m (0.4m, 0.2m on the graphs). The H-matrix is generated using the radar equation and is then split into small subsets of data, with each subset having the transform applied to it (parallel version). The subset is a group of adjacent elements and the transmitting and receiving subsets don't need to be the same (but for simplicity have the same number of elements) This leads to 49 groups of 3, 36 groups of 4, 25 groups of 5, 16 groups of 6, 9 groups of 7, 4 groups of 8 and 1 group of 9. These groups have different angular spreads and different origin points.

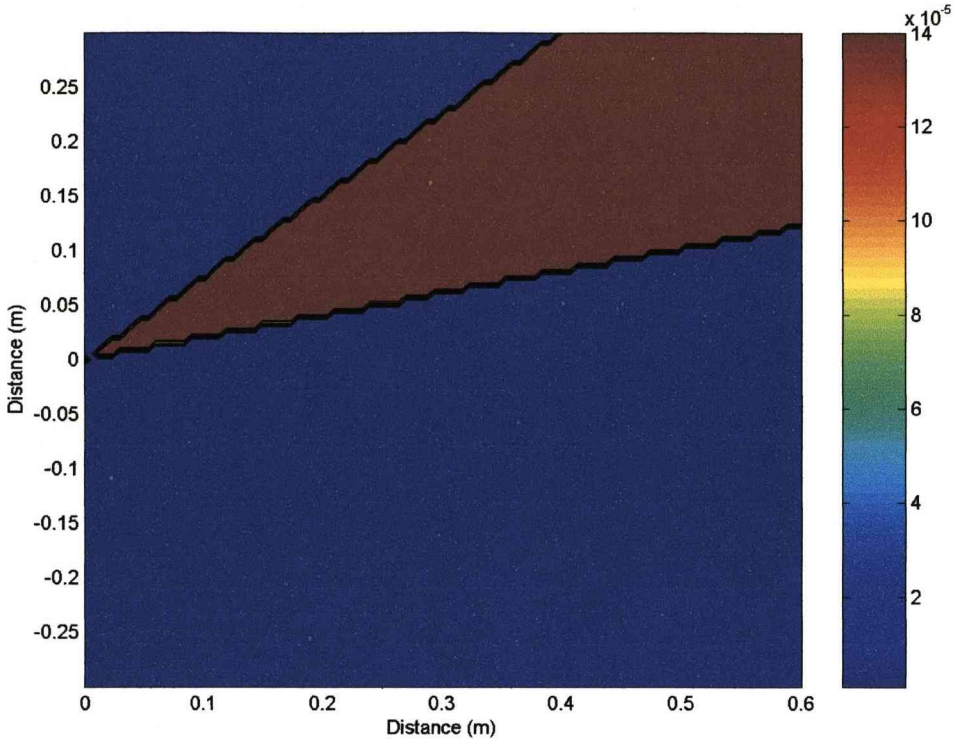

Figure 4.53 - 5-Element array in Cartesian form

The 5-element array is a good indicator of the sort of thing seen, with a single sharp triangle of potential locations stretching away from the array. When you take the sum of a 5-element array from the top edge of a 9-element array and a 5-element array from the bottom edge the potential location of any target is reduced as in figure 4.54. Similarly the sum of all 5-element subsets of a 9-element array has a comparable result with a smoother transition.

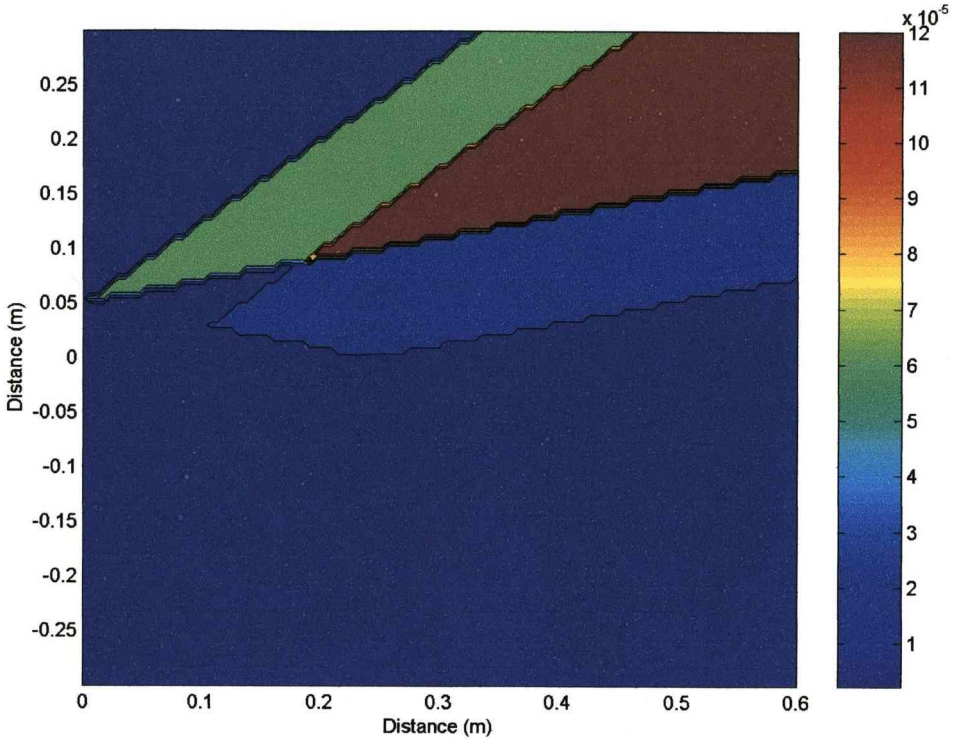

Figure 4.54 - Sum of edge 5-Element arrays

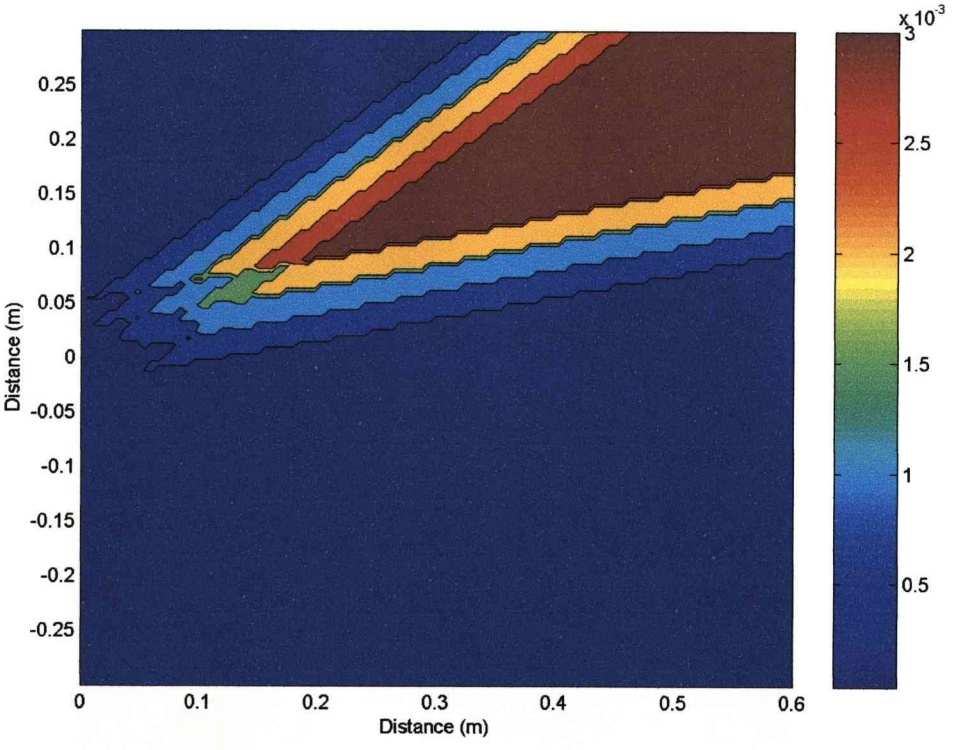

Figure 4.55 - Sum of all 5-Element arrays

There are then various combinations to work through with the same method, the goal being to separate the origin points of the centre of the subsets used to gain an idea of distance by using a single array instead of two separate arrays. The sum of all 3 to 9-element subset groupings gives the result in Figure 4.56.

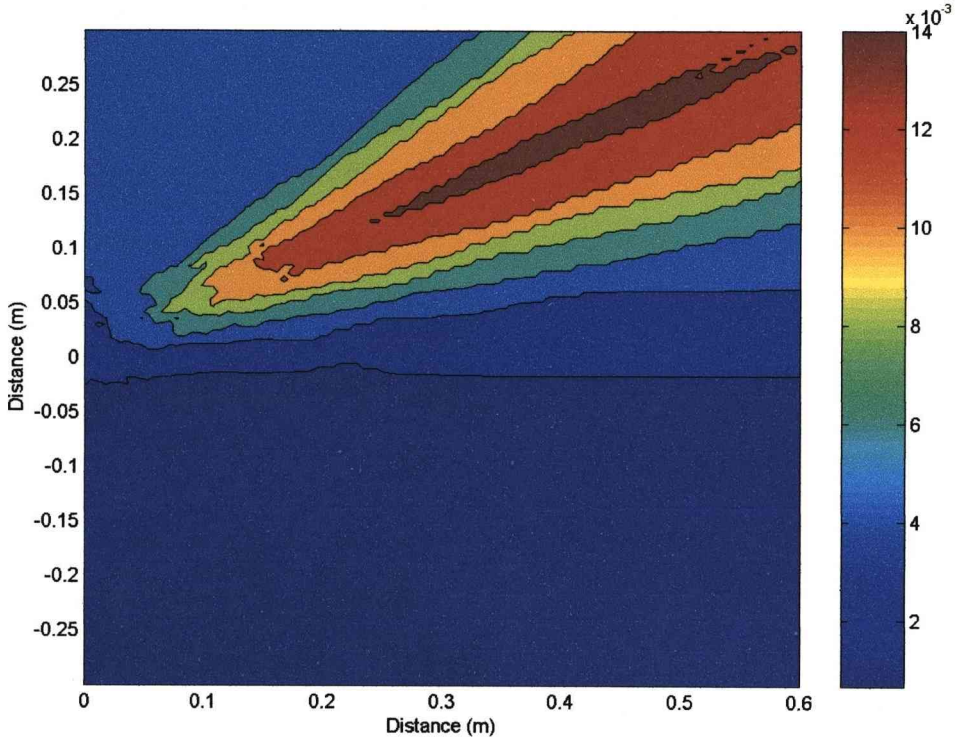

Figure 4.56 - Sum of all 3-9-Element subset groupings

The most likely target location is narrowed to a thin sliver centred over the actual target location showing that a limited distance measure is possible. It is also possible to pick out some different frequencies from a signal, using them with appropriate antenna element spacing (like every second element for half frequency) and combining it later with subsets from other frequencies. The major disadvantage though to this method is the high speed for the transform itself is lost in the conversion process making it no faster than a target localising method in the Cartesian domain.

4.7 Chapter Conclusions

The virtual H-Matrix as a tool for imaging has been largely unexplored. This chapter has looked at the overall performance for imaging and introduced a variation that improves close range focussing. The main benefit to a direct transform of the H-matrix is the speed at which it operates but it comes with significant drawbacks. This method offers a fast way to generate imaging results that pinpoint the direction of the target in terms of angles from the transmitter and receiver array. The results show clear targets and a focussing method allows larger arrays to be used without closer targets breaking up.

A change from angular to Cartesian form shows that some distance measure is also possible (while still only using a single frequency) at the expense of the quick processing speed of the transform and further narrowing down the angular location of the target. The main problem is the low resolution related to the number of antennas in the array and hence the array size required for higher resolutions. Looking at the influence of the focal distance value (k) shows that the matching of focal distance to target distance isn't perfect and that a distance measure is not easily obtainable via observation of k. Low resolution is really the main weakness leading to a suggestion that it would be best employed to give angular information that is then combined with another method for distance and a more finely tuned direction. One more important aspect to consider is the potential benefits for pattern recognition as the angular image produced looks completely different to Cartesian images. Setting a conventional system up like in the following chapter so that it is compatible with this method could lead to hybrid methods or additional characterisation information for pattern recognition systems.

Chapter 5

Conventional Imaging Methods

5.1 Calibration

This calibration method is designed to not require an initial reference set of data obtained by performing a target scan with no target present which means it can be used where this initial scan would be problematic. It is similar to a range of skin subtraction algorithms covered in [9]. As such the results show more clutter than would be present with a no target scan subtraction calibration method. This calibration uses an average subtraction to remove unwanted direct path signals between transmitter and receiver as shown in Figure 5.1. The receiver or receivers are placed equidistant from the transmitter and moved as a group to all scanning locations. The average of all received signals is then subtracted from each individual signal. Every signal has a common component from the direct signal that should be identical because the relative location difference between transmitter and receiver stays the same. So the common direct signal component is removed leaving only the external wanted signal component to be processed. Any other signal components that remain constant over all the scanning locations such as a uniform linear surface when scanning locations run parallel to it will also be removed.

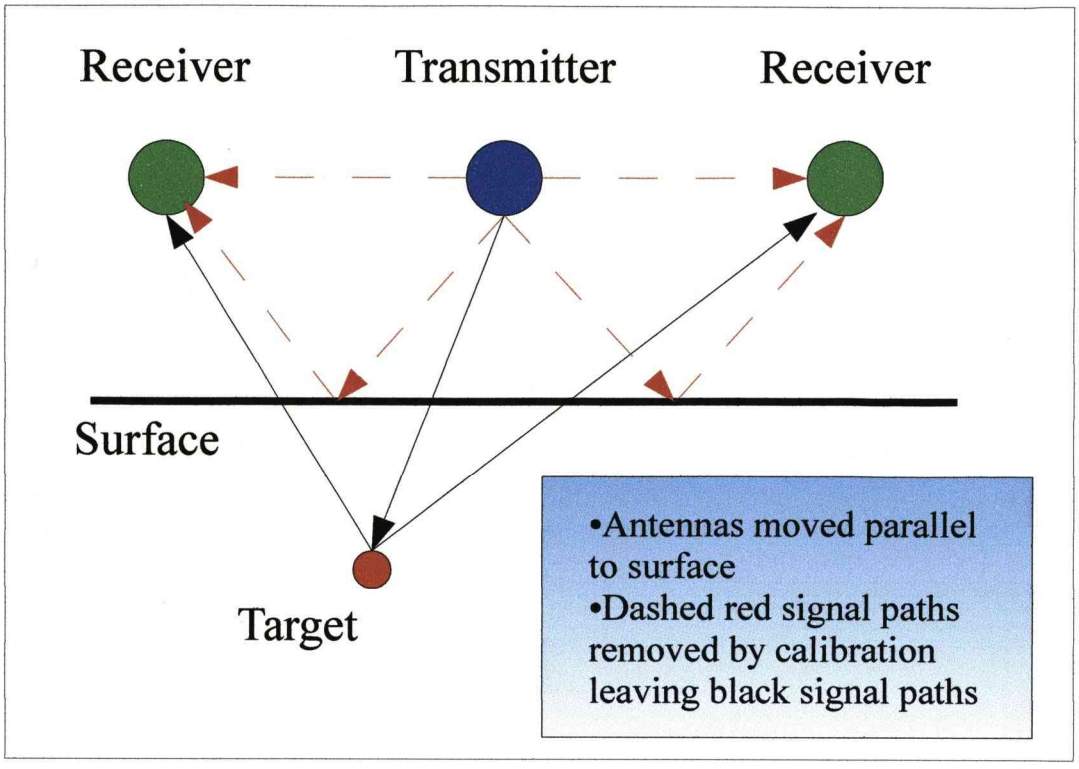

Figure 5.1 - Calibration Set-up

5.2 System Set-up

The data is generated using an FDTD algorithm. Compared to a standard wideband pulse the encoded PN sequence requires a much longer transmission for effectiveness and hence a longer simulation. Coupled with the large simulation area and multiple simulations for each transmitter location, a 2D FDTD simulation is the optimum solution to data generation. Comparisons between conventional wideband pulsed excitation and an encoded PN sequence require a simulation for each excitation as well. Multiple set-ups are also used, each with a common set of transmitter locations but with the addition of flat/curved skin surfaces, round/square targets and variable tissue properties.

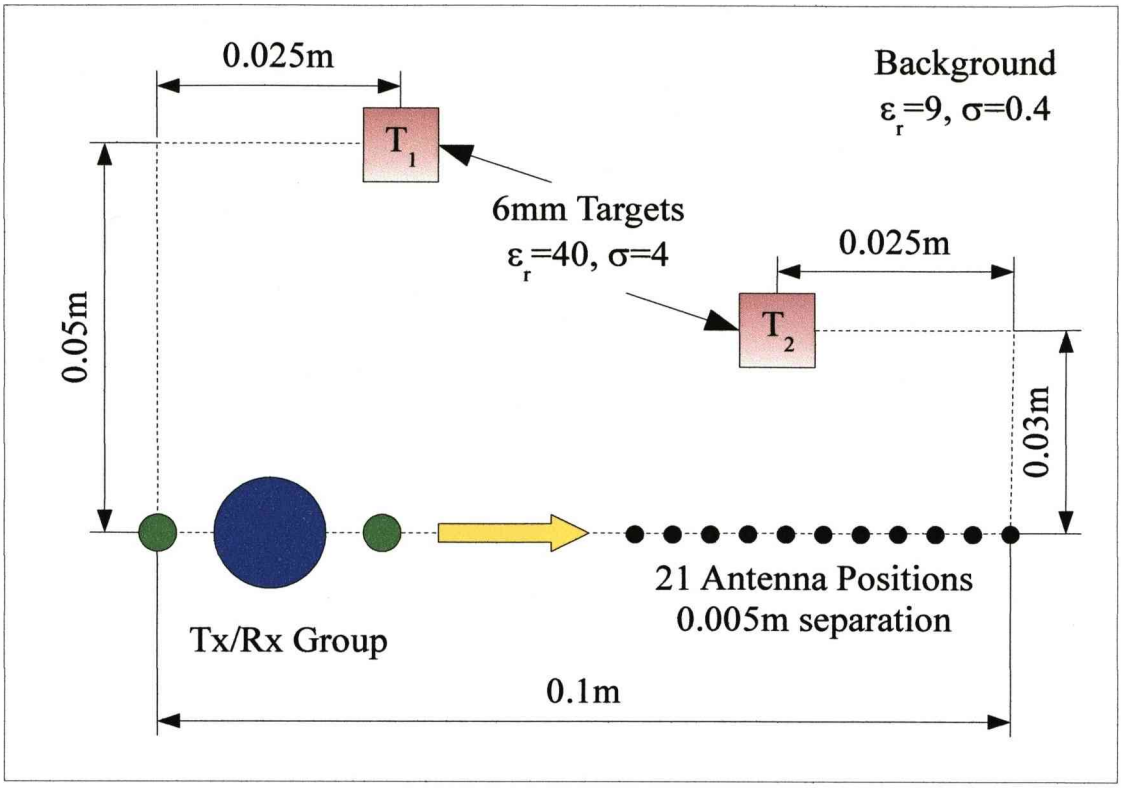

Figure 5.2 - Basic Set-up

In addition to the basic set-up given in Figure 5.2 there is the addition of a skin surface either flat and parallel to the transmitter locations (at 0.01m distance) or curved (of radius 0.05m, 0.06m distance from the centre of the transmitter locations). The background material can also be altered by applying a variation in relative permittivity and conductivity to each cell that makes up the tissue (inside the skin surface).

5.3 Excitation

$$y=\left(t-300\times10^{-12}\right)e^{\frac{\left(t-300\times10^{-12}\right)^2}{\left(75\times10^{-12}\right)}} \tag{5.1}$$

A differentiated Gaussian pulse shown in Figure 5.3 and generated using Eq. 5.1 was used as the conventional wideband pulse excitation and a BPSK encoded PN sequence shown in Figure 5.4 as the new excitation. As the PN sequence needs to be

as long as possible for best performance the signal used was twice as long as the differentiated Gaussian pulse with the differentiated Gaussian pulse effectively zero over this extra length.

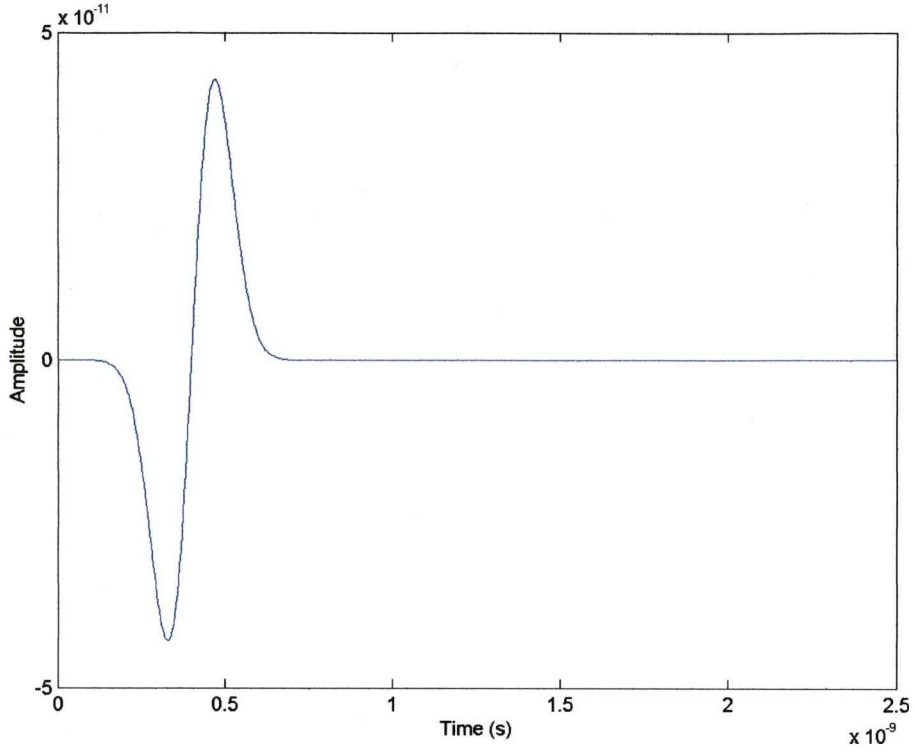

Figure 5.3 - Differentiated Gaussian Pulse Excitation

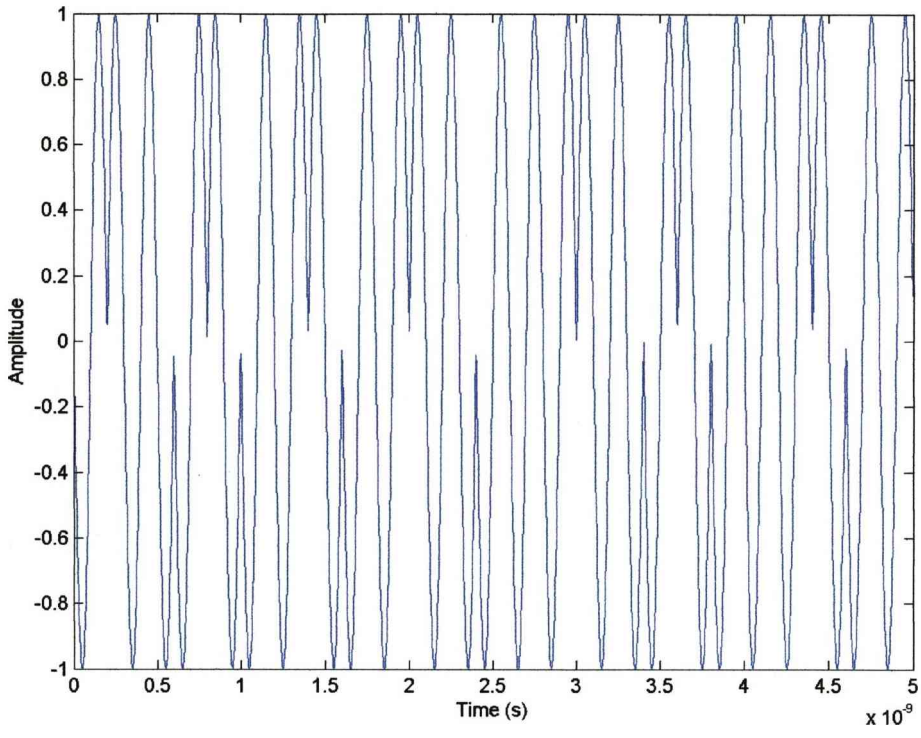

Figure 5.4 - BPSK encoded PN Sequence Excitation

5.4 Localisation

The localisation technique is a standard shift and sum algorithm [25].

$$p(x,y,z)=\sum_{k=1}^{N} c_k s_k(T_k) \tag{5.2}$$

So for a point p the value is given by the sum of values from each signal (s_k) at the corresponding delay time (T_k). This delay time is calculated using an estimate of the propagation velocity and point p's position, taking into account travel time from the transmitter to point p and back to the receiver. There is also a compensation factor (c_k) that is used to provide compensation for radial spreading loss

$$\begin{aligned} c_k=&((Tx_k(x)-p(x))^2+(Tx_k(y)-p(y))^2+(Tx_k(z)-p(z))^2) \\ &\times((p(x)-Rx_k(x))^2+(p(y)-Rx_k(y))^2+(p(z)-Rx_k(z))^2) \end{aligned} \tag{5.3}$$

A modification is made to (5.2) for the wideband pulse as excitation (5.4). As distortion of the pulse occurs when travelling through a medium the integral along the pulse length starting at the delay time is used. A maximum occurs when the calculated delay time matches up exactly with the reflected signal. In addition an offset is introduced to the delay time to ensure the centre of any reflected signal is detected.

$$p(x, y, z) = \int_0^{\tau} \left| c_k s_x(t + T_k) \right| \tag{5.4}$$

5.4.1 Results and Discussions

These results are the signals that will be used for localisation. The wideband pulse signal is used directly while the encoded PN sequence uses filter output. The filter output is designed to offer a direct way to pick up the surface of the target while the wideband pulse and integration is to give the strongest detection. All have been calibrated prior to processing as appropriate. The set-up involves two 6mm square targets in locations already given above in Figure 5.2 with no skin present or tissue variation. Whenever a reflecting target is present a characteristic hyperbola shape is observed when the signals are presented directly side by side as in Figure 5.5 and 5.6, the strength of which shows the strength of detection. The integrated pulse shows a good detection of one of the targets but the second further target is mostly obscured and the overall detection is quite wide.

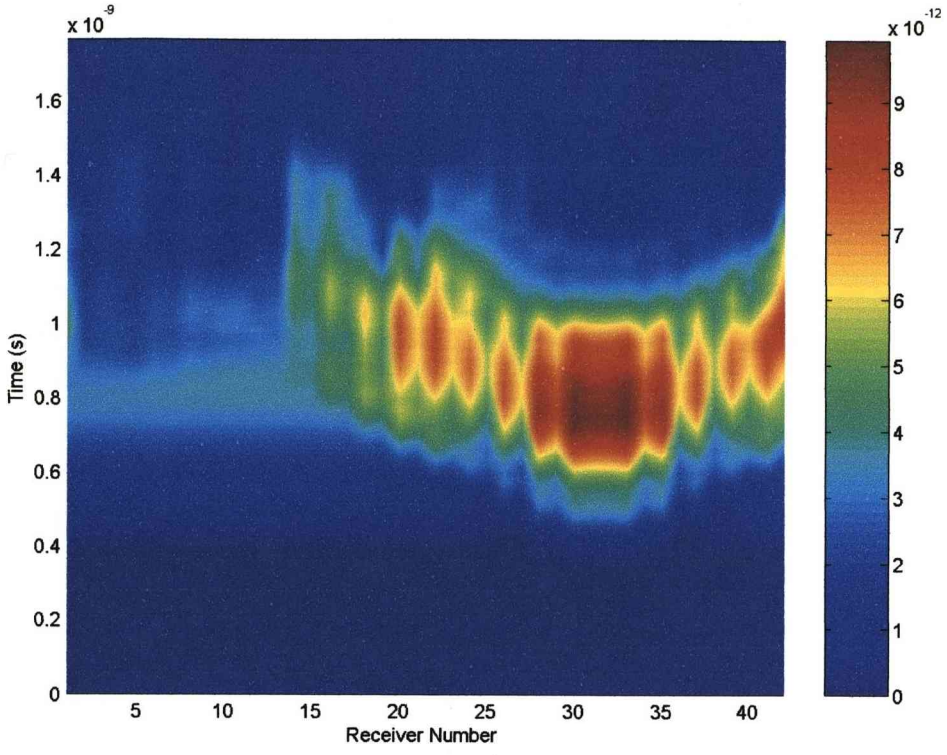

Figure 5.5 - Side by side plot of calibrated signals for integrated pulse

Using a PN sequence and filter leads to a much narrower but weaker detection with more noticeable calibration artefacts. The second target though is visible and recognisable as a target.

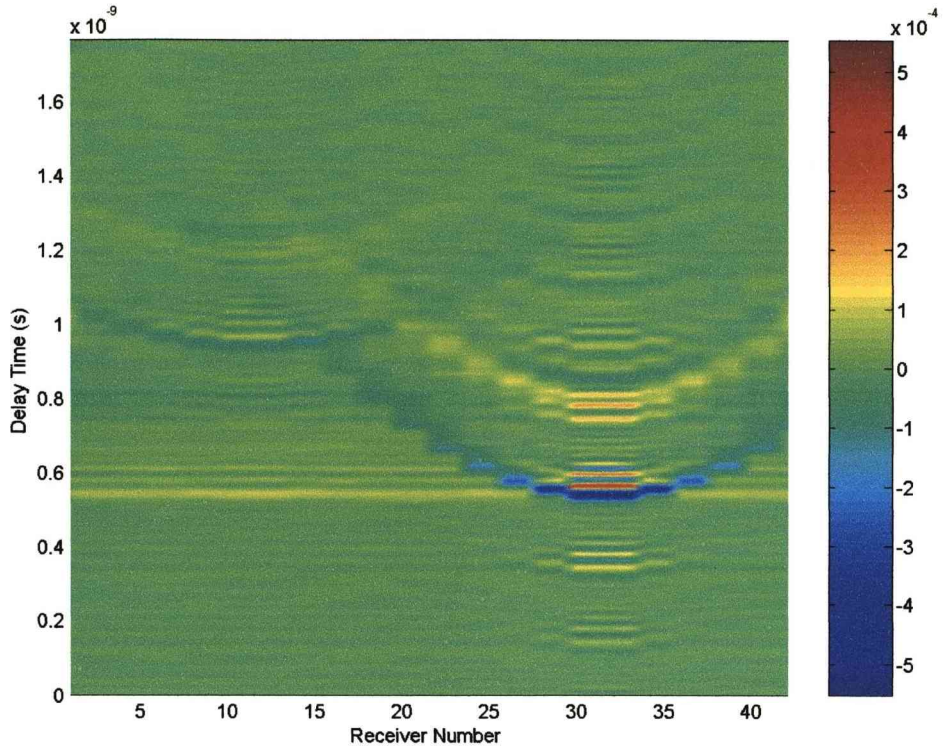

Figure 5.6 - Side by side plot of calibrated signals for PN sequence and filter

Targets further away are naturally reduced due to spreading and attenuation losses. Spreading loss can be accounted for in the localisation algorithm. Also affecting detection is the strength of reflection due to dielectric permittivity interface variation and target size. The filter output gives a direct value for the delay while the wideband pulse has an offset due to the pulse centre not being at time zero. The first test uses square targets with the two different excitation signals and processing methods. For each image the transmitter locations are between 0.05m and 0.15m in the x-axis and at 0.02m in the y-axis. The targets are located at (0.075,0.07) m and (0.125,0.05) m respectively.

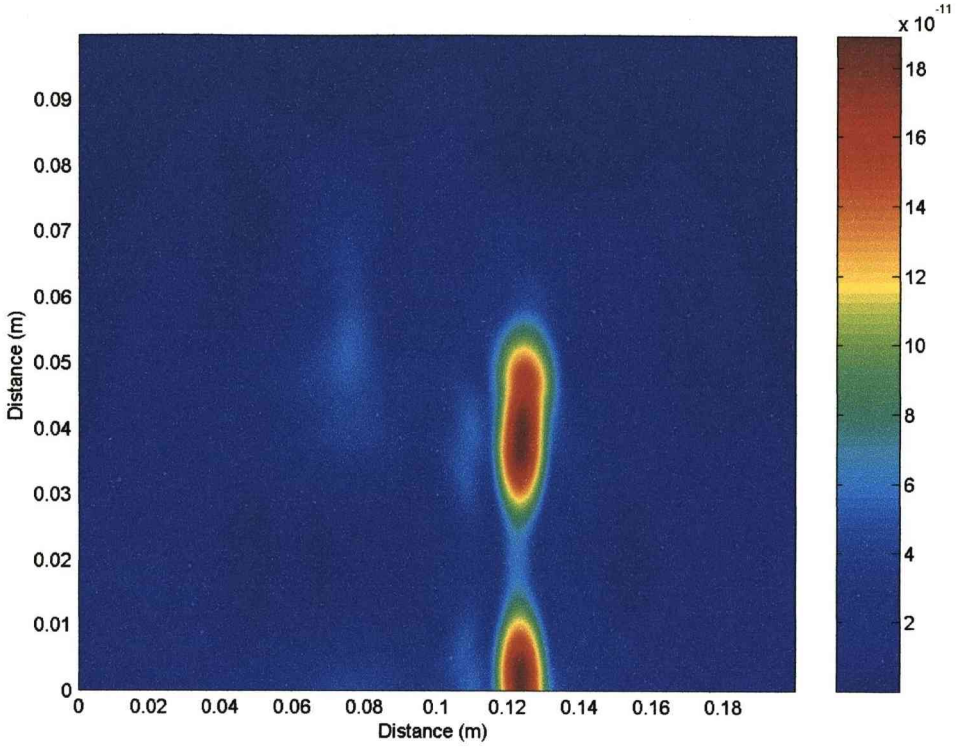

Figure 5.7 - Pulse, 6mm square targets without spreading loss compensation

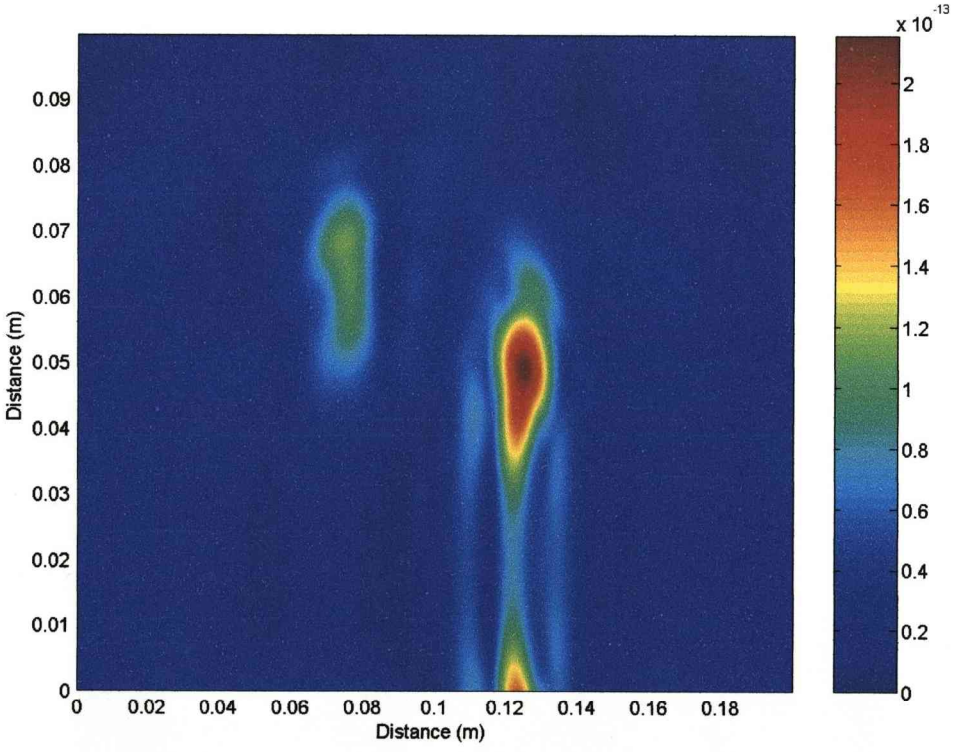

Figure 5.8 - Pulse, 6mm square targets with spreading loss compensation

The main target in Figure 5.7 shows as quite a large anomaly with some smearing along the y-direction and a general distortion of the target location. The further target is also very weak and hard to locate. Figure 5.8 when spreading loss is compensated for brings the first target back to its correct location and reduces smearing. The second target also shows up more but it is still slightly weaker and smeared.

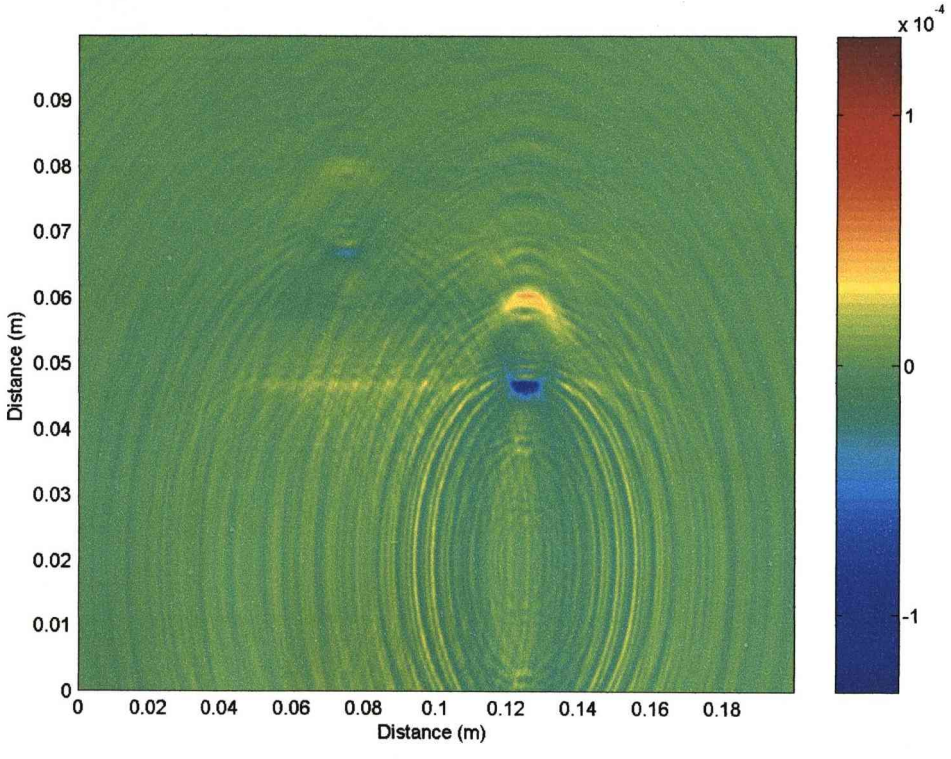

Figure 5.9 - PN sequence, 6mm square targets without spreading loss compensation

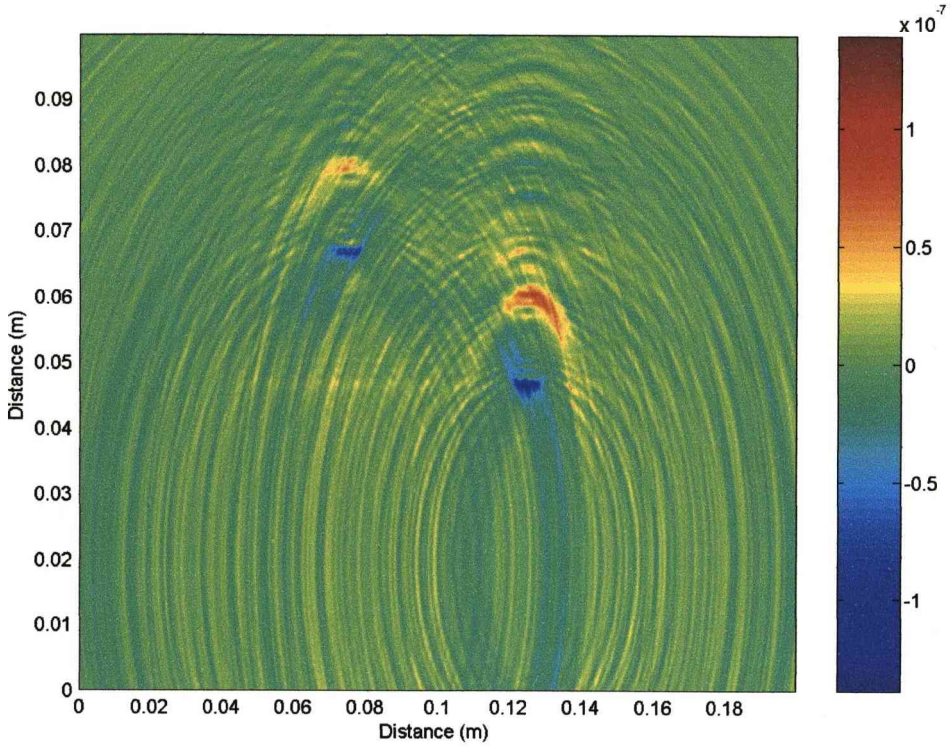

Figure 5.10 - PN sequence, 6mm square targets with spreading loss compensation

The filter using a PN code signal gives very sharp results in Figure 5.9. The front and rear reflections are detected and are complete with the correct reflection coefficient sign. The front reflection is exactly where it is supposed to be with the rear reflection displaced backwards by a small amount. The second target is also weak like with the pulse but when using spreading compensation in Figure 5.10 this is much improved with both targets approaching the same level of detection strength. There is though a small amount of additional clutter. As the target looks like it has angular features, shape identification may be possible so 3mm radius circular targets were used in another simulation. This time spreading loss compensation is applied to all the images shown as it gives the best overall image and in addition a correlation image from the PN sequence is included to provide the filter method with a same signal comparison.

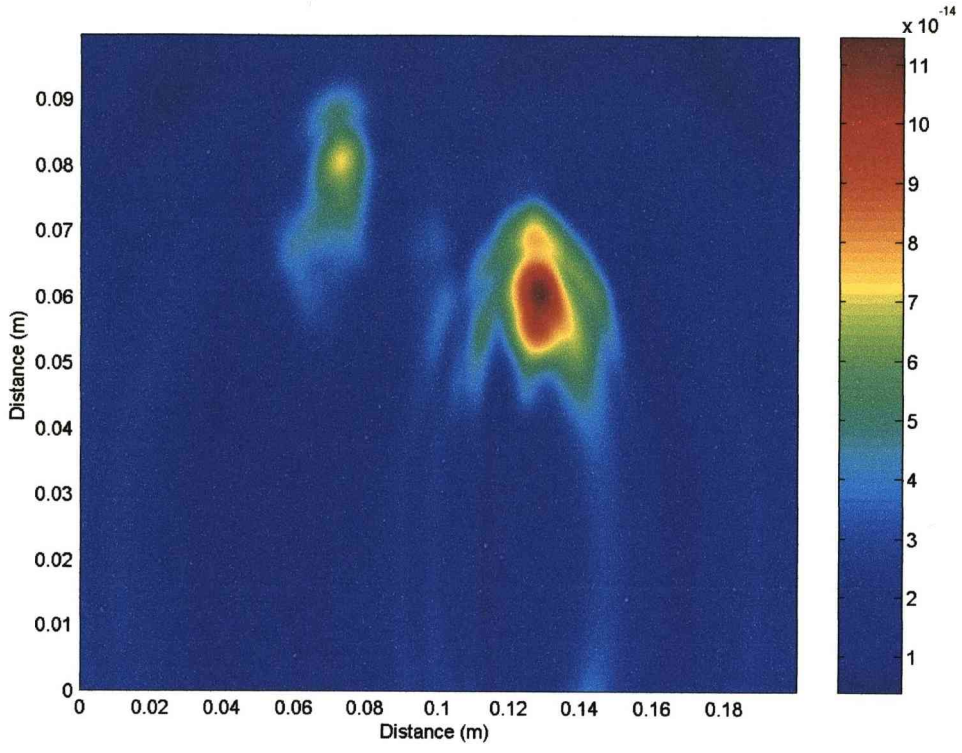

Figure 5.11 - Pulse, 3mm radius circular targets

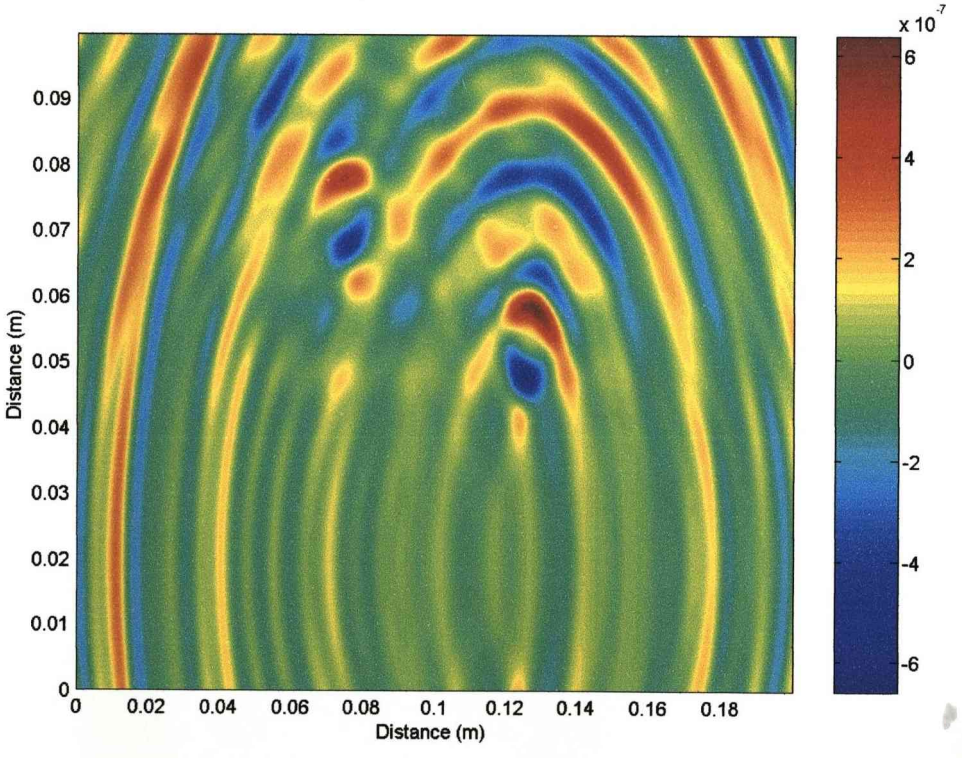

Figure 5.12 - PN sequence, 3mm radius circular targets, correlation

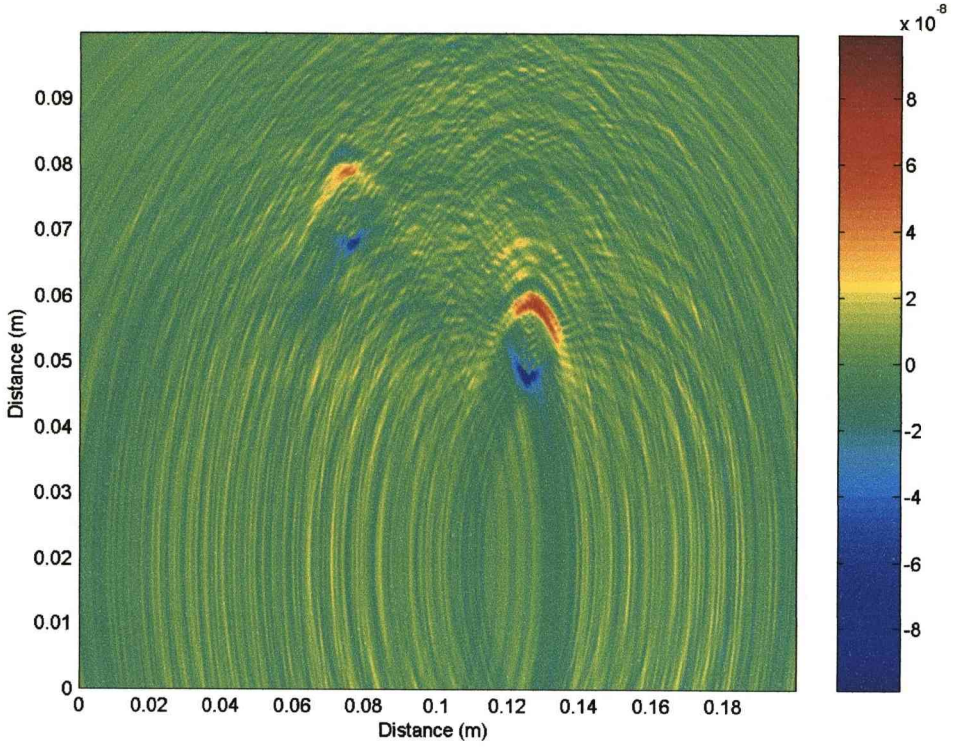

Figure 5.13 - PN sequence, 3mm radius circular targets, filter

Figure 5.11 shows the circular targets with less smearing than the square targets but overall is similar in relative strength of detection between the closer and further targets. Correlation in Figure 5.12 shows the targets but has a significant amount of trailing clutter and is probably the worst of the three images. The targets in Figure 5.13 show up as more rounded than for the 6mm square targets. Image strength of the round targets is about 70% of the square targets. Up to now skin has been omitted but it is an important factor. A flat skin surface is introduced to assess how well the detection works after penetrating the skin and also the usefulness of the calibration procedure.

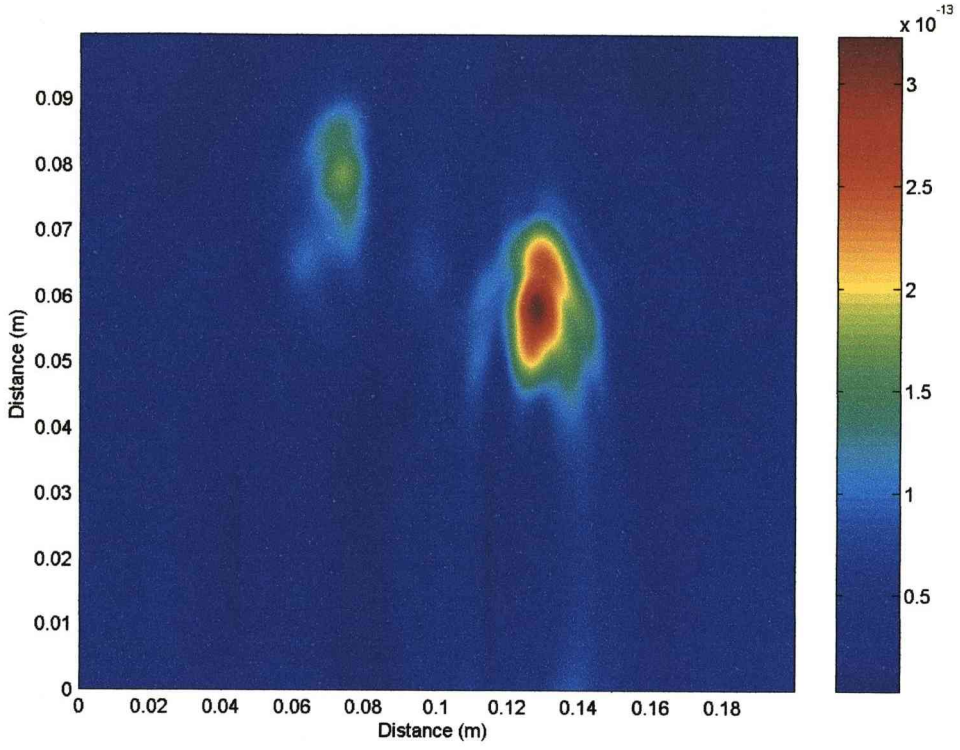

Figure 5.14 - Pulse, 3mm radius circular targets with skin

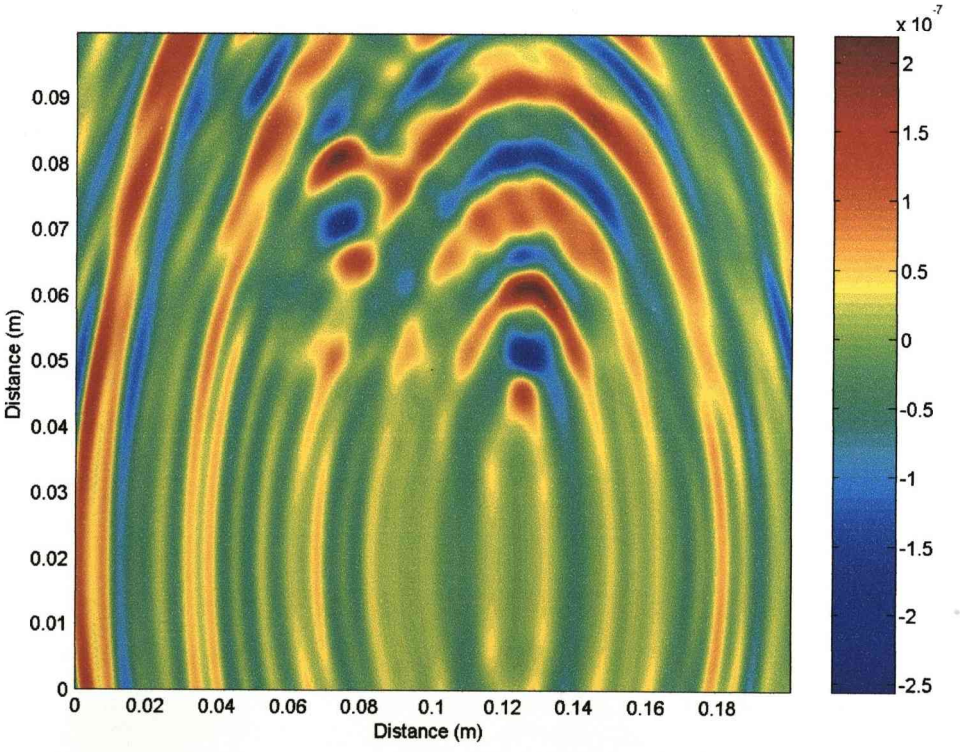

Figure 5.15 - PN sequence, 3mm radius circular targets with skin, correlation

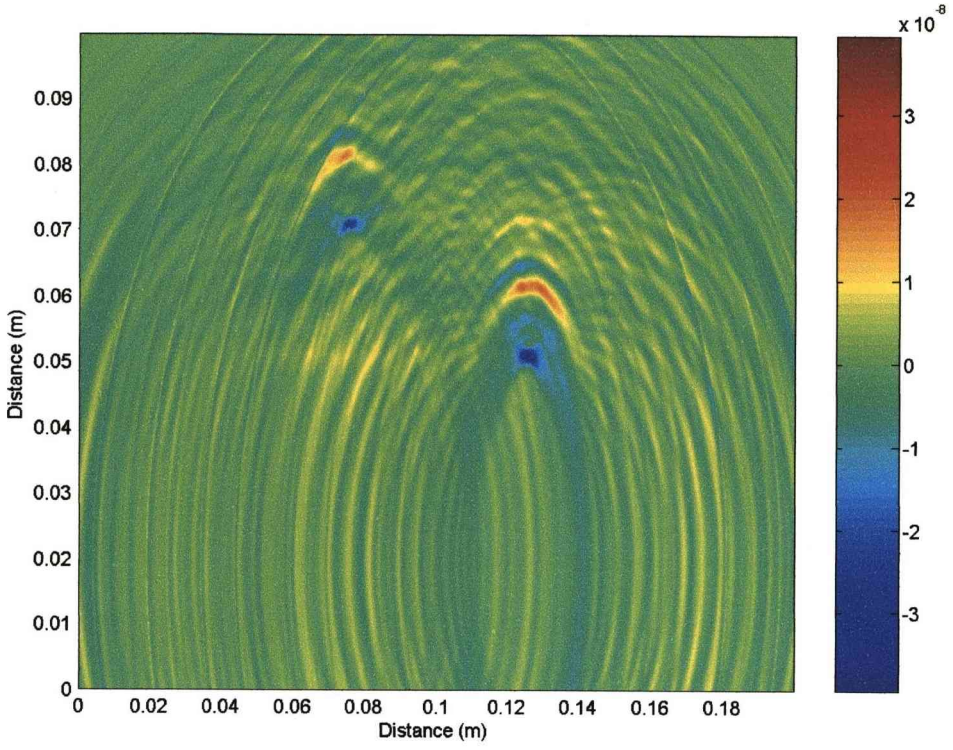

Figure 5.16 - PN sequence, 3mm radius circular targets with skin, filter

The skin causes some problems including an increase in clutter for the PN sequence methods in Figure 5.15 and 5.16 along with a general decrease in target image strength and a small offset. The image strength is reduced compared to the corresponding without skin image. This fits in with an expected reduction due to reflection from the skin travelling to and from the target. There is also an offset due to a slowdown caused by travelling through the skin itself. The offset though is slight at less than 3mm. By comparison the integrated pulse method doesn't show a great deal of difference to the without skin version as shown by Figure 5.14. Due to the nature of the calibration method a flat array does not work with a curved skin surface as indicated by Figure 5.17 where a fragmented image of the skin is visible (including mirror image at the bottom) but the targets are very faint and it's very cluttered.

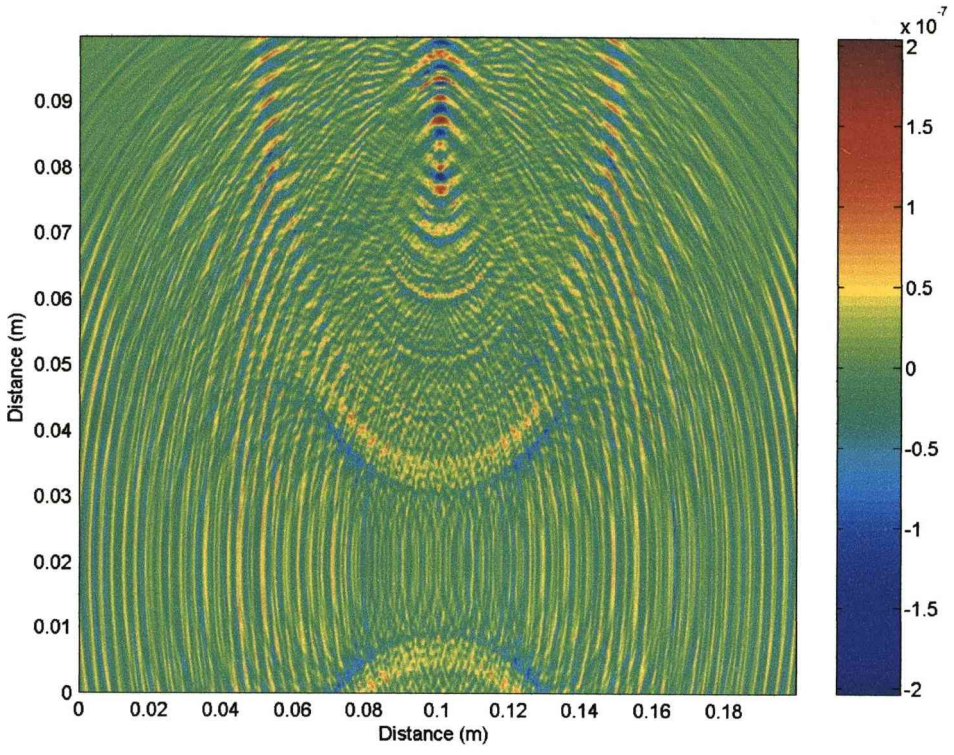

Figure 5.17 - PN sequence, 3mm radius circular targets with curved skin, filter

Due to the presence of the rear reflection of the target visible in the PN sequence based methods it is interesting to look at a high contrast example where the targets permittivity is increased to an arbitrarily large value.

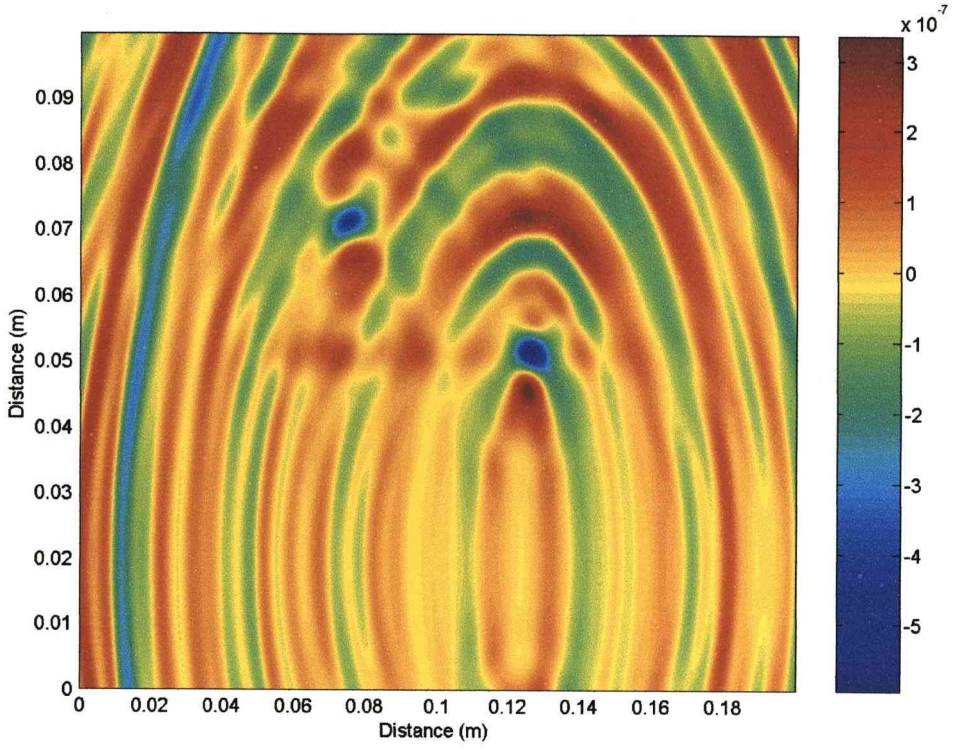

Figure 5.18 - PN sequence, 3mm radius high contrast circular targets with skin, correlation

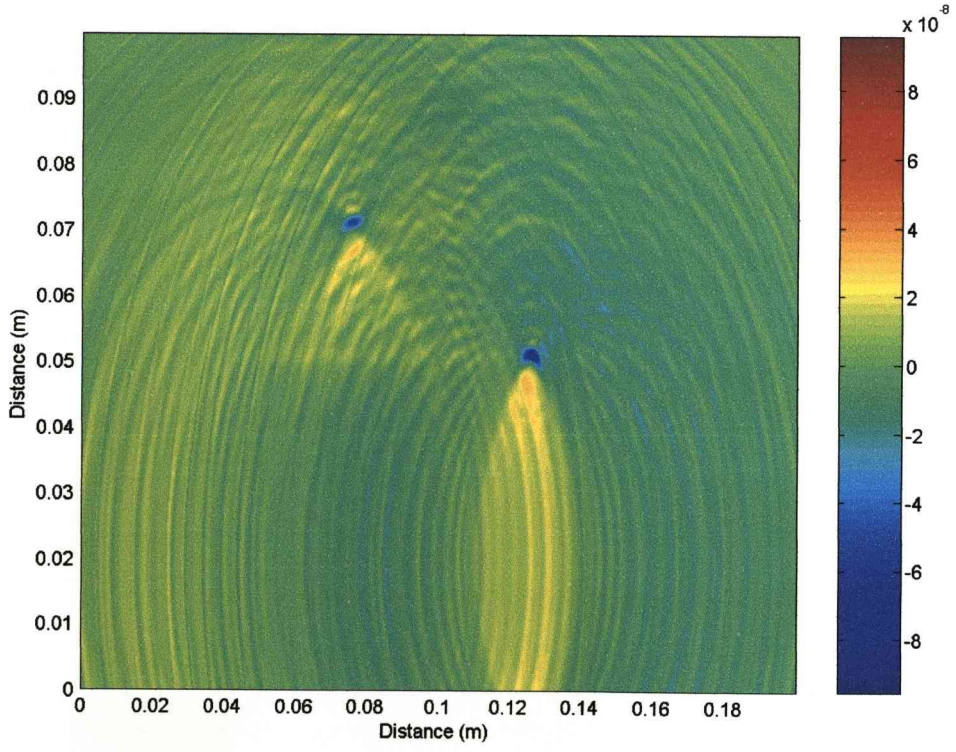

Figure 5.19 - PN sequence, 3mm radius high contrast circular targets with skin, filter

When compared to earlier images there appears to be a front positive shadow particularly the correlation in Figure 5.18. The rear reflection disappears completely with the likely explanation that the speed through the target is now so slow that the signal doesn't return to a receiver in time via the back reflection (the actual rear reflection would then be off image). The target detection strength on both is also significantly increased due to the better target reflection.

5.5 Identification of Target Shape

With a sharp edge detection an array of transmitters encompassing a target over a full (or nearly full) 360 degrees should allow an accurate representation of the shape of the target to be discerned. With only one sided scanning so far it is possible to tell a square target from a round one (comparing Figures 5.10 to 5.13) due to the square targets flat rather than rounded inside edge. Extrapolated to all around a target the same features should be visible allowing for shape identification. In total 5 shapes will be tested given in Figure 5.20, ranging from the three simple geometric shapes (Circle, square and triangle) to the more complex circle cluster and enclosed square.

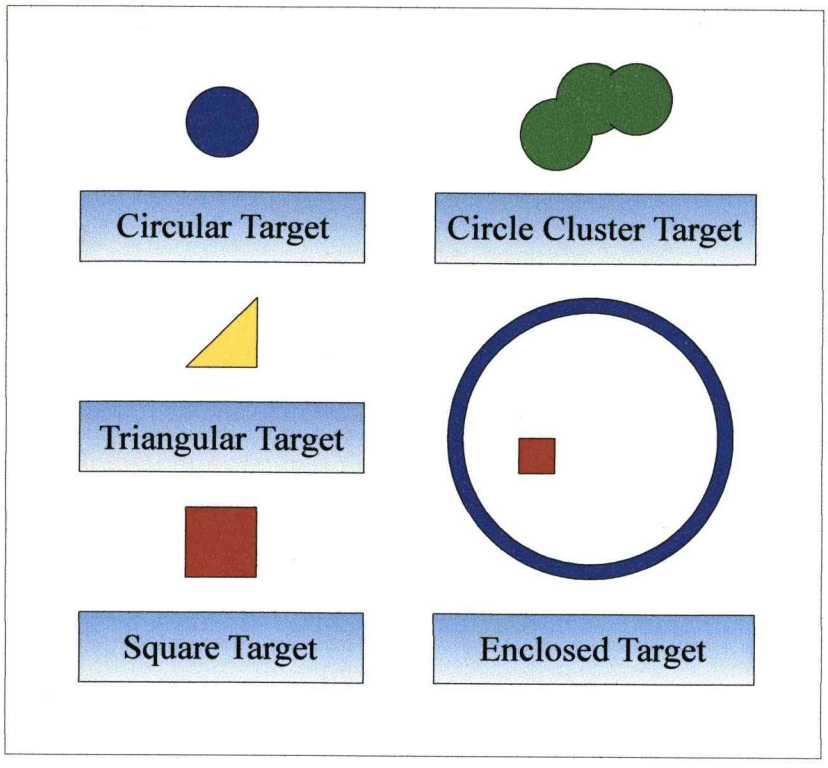

Figure 5.20 - Target shapes tested

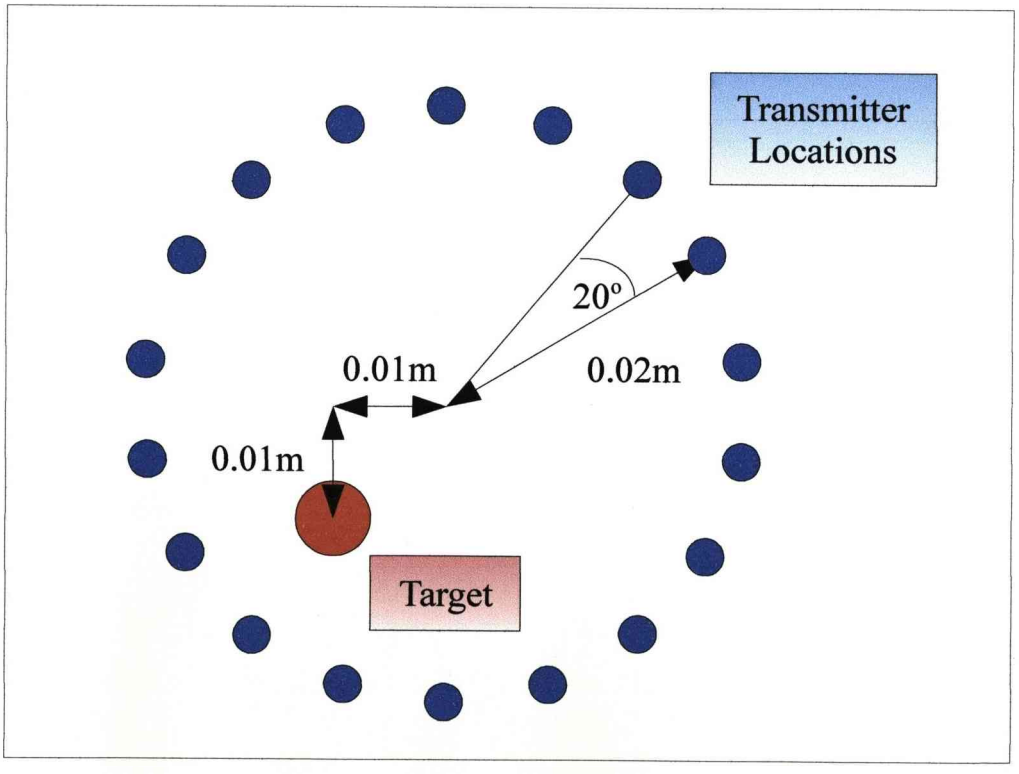

Figure 5.21 - Simulation Set-up

The simulation set-up given in Figure 5.21 consists of a ring of transmitter locations with a radius of 0.02m and 20° increments. At each location a transmission is made with the adjacent transmitter locations acting as ideal receivers. The calibration is performed as normal and processed to give the localization graph before the areas outside of the transmitter location ring are removed. This leaves a graph containing only the area within the transmitter ring. The background material is set to ε_r=9, σ=0 with the targets as ε_r=40, σ=4. With a complete ring around the target and over a short distance spreading loss compensation is not needed.

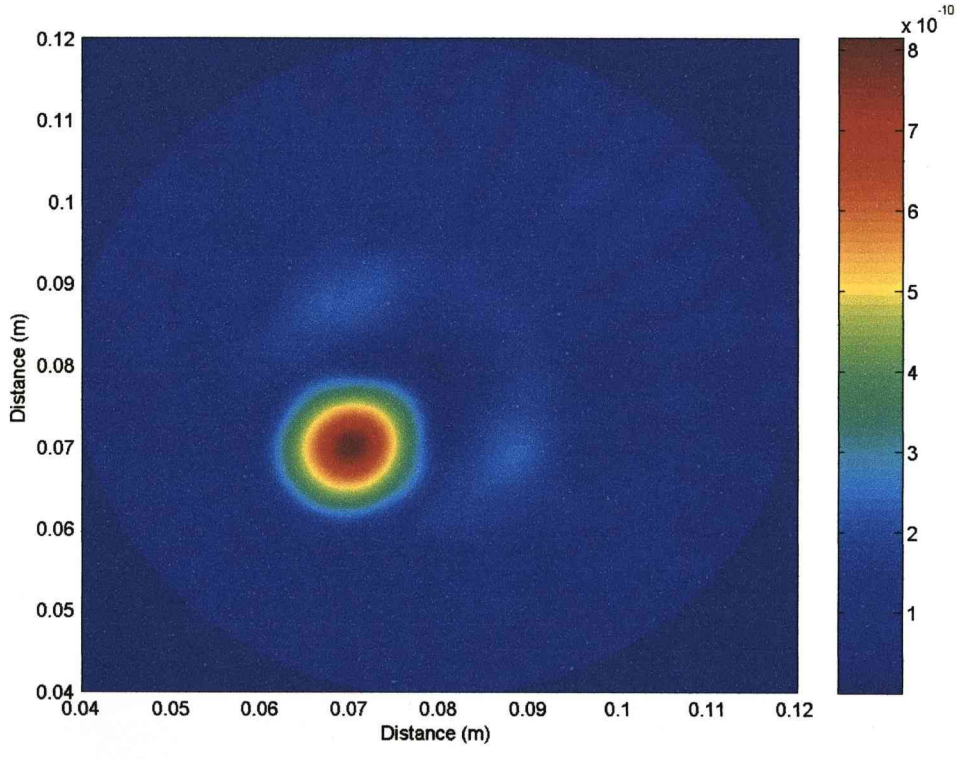

Figure 5.22 - Pulse, circle target

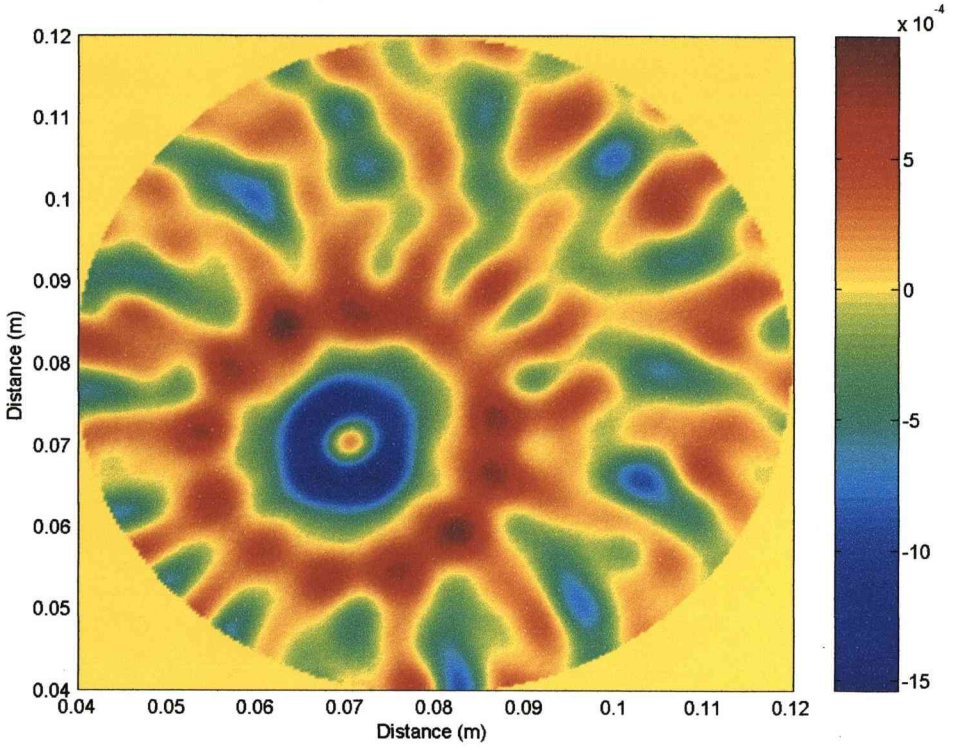

Figure 5.23 - PN sequence, circle target, correlation

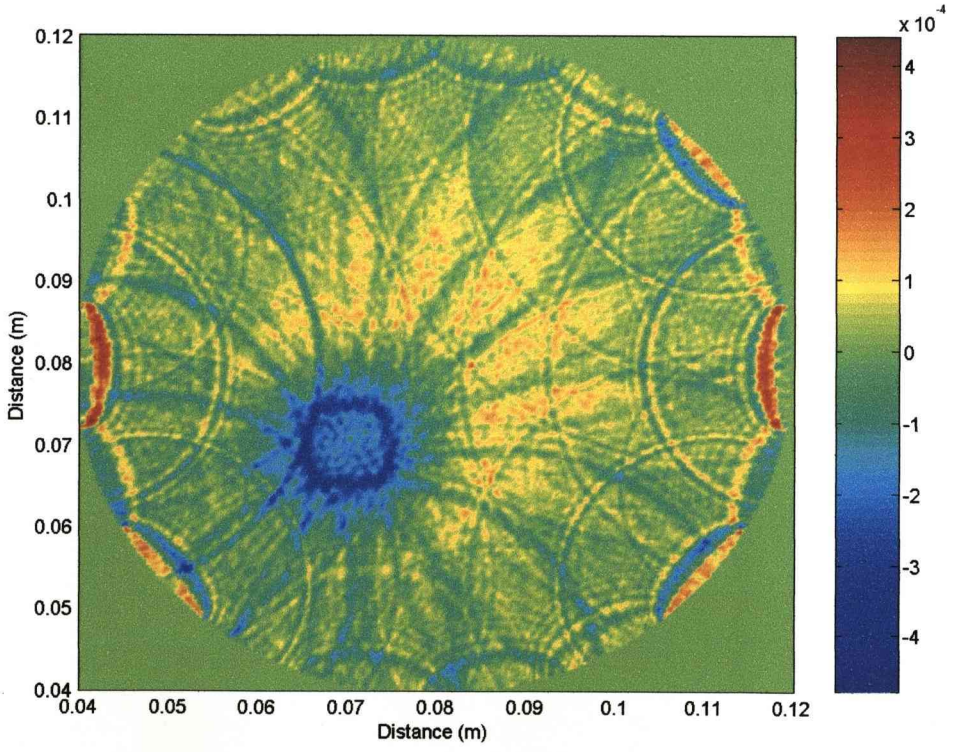

Figure 5.24 - PN sequence, circle target, filter

Unsurprisingly the Gaussian pulse does give a rounded target detection with an approximately accurate size in Figure 5.22. The filter though clearly shows a round edge consistent with a circle target in Figure 5.24 although some calibration artefacts can be seen around the edges. The correlation in Figure 5.23 looks similar to the filter but with a thicker stronger detection. Also note that the correlation and filter show a positive outer ring around the negative target detection representing the reflection from inside the target travelling out (analogous to the rear reflection from the previous section)

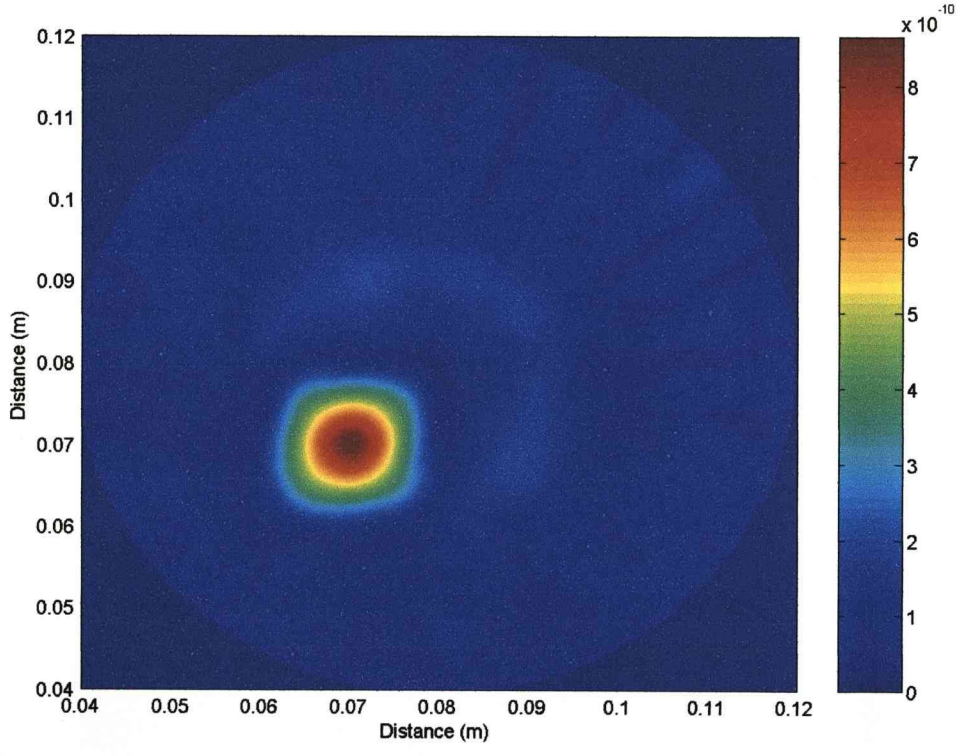

Figure 5.25 - Pulse, square target

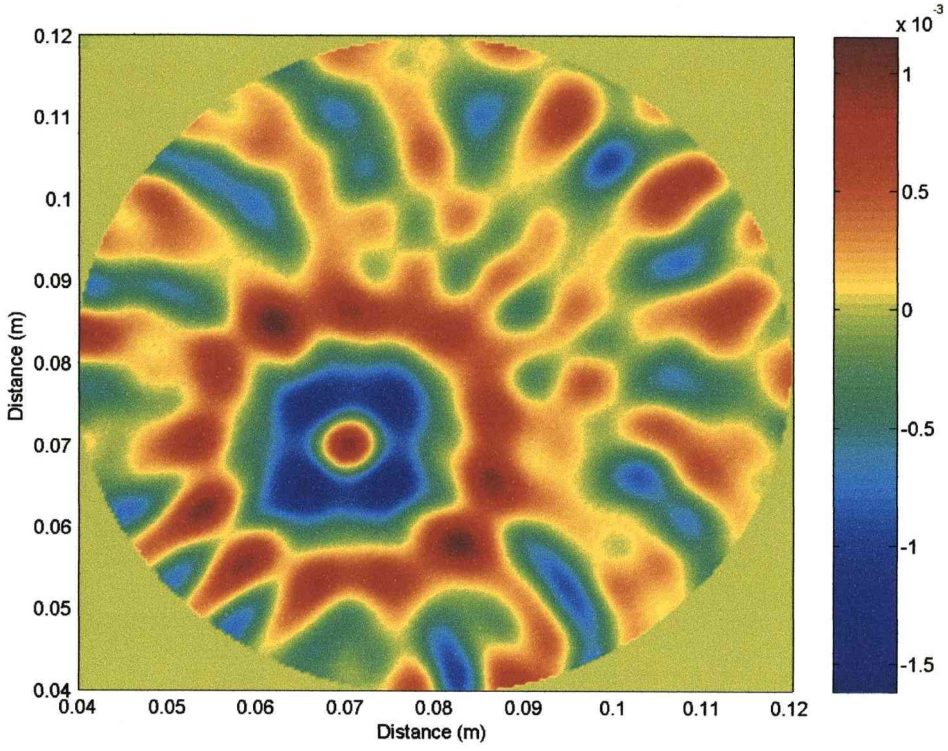

Figure 5.26 - PN sequence, square target, correlation

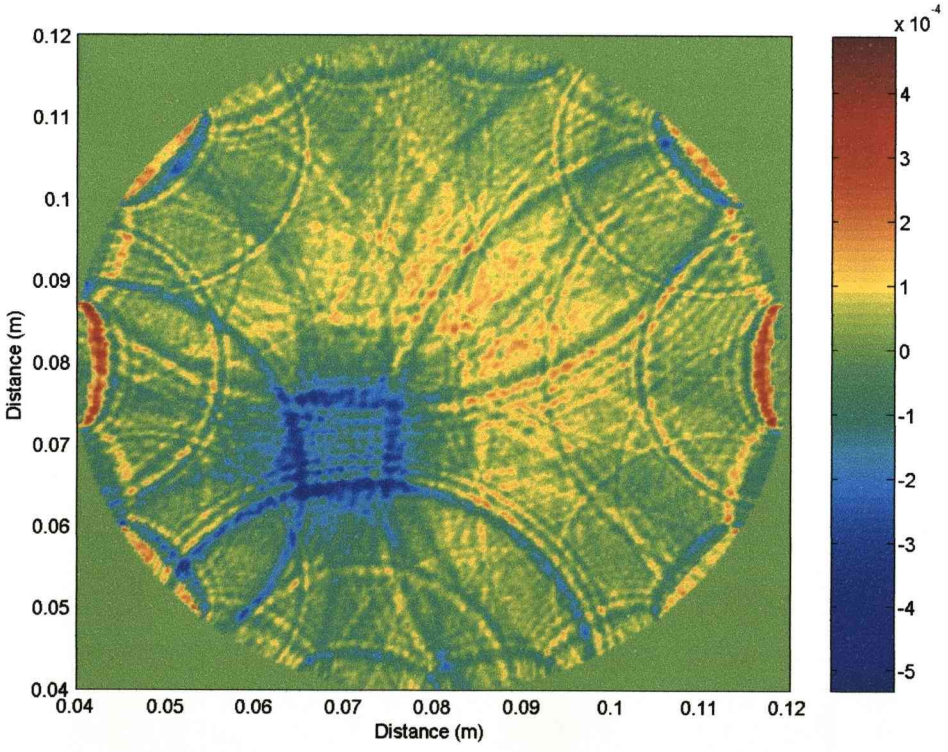

Figure 5.27 - PN sequence, square target, filter

The pulse in Figure 5.25 shows a strong detection but is still mostly round with only slightly more angular edges compared to the round target in Figure 5.22 so it's not possible to determine the shape. Correlation in Figure 5.26 shows something of an angular square shape but has protruding corners. The filter again gives a clearly square target image in Figure 5.27 but continues with more clutter present.

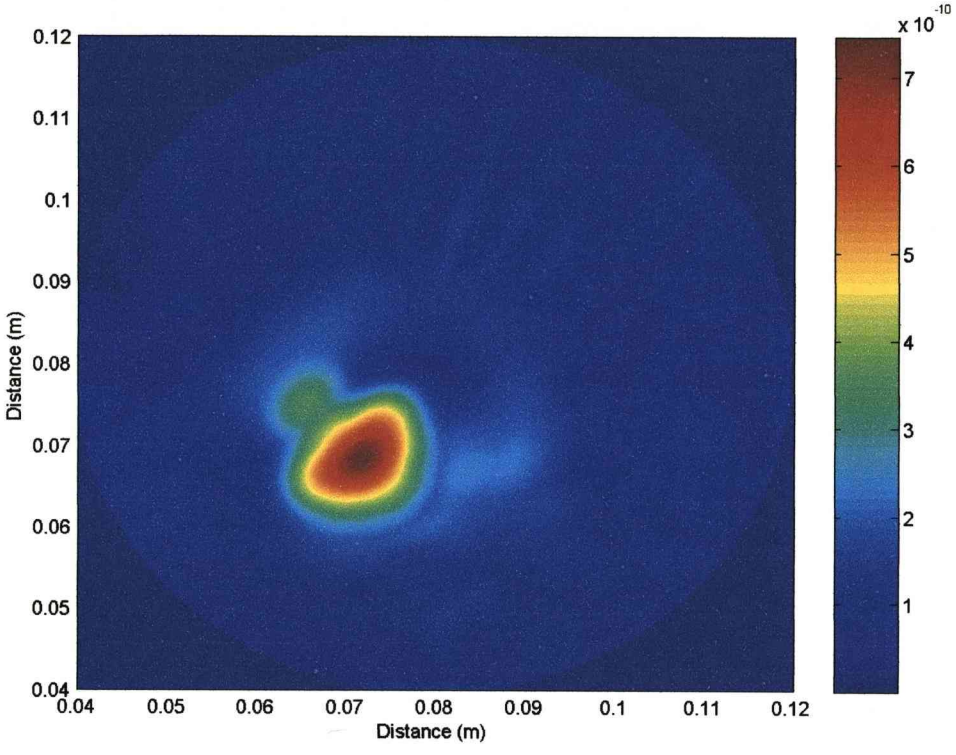

Figure 5.28 - Pulse, triangle target

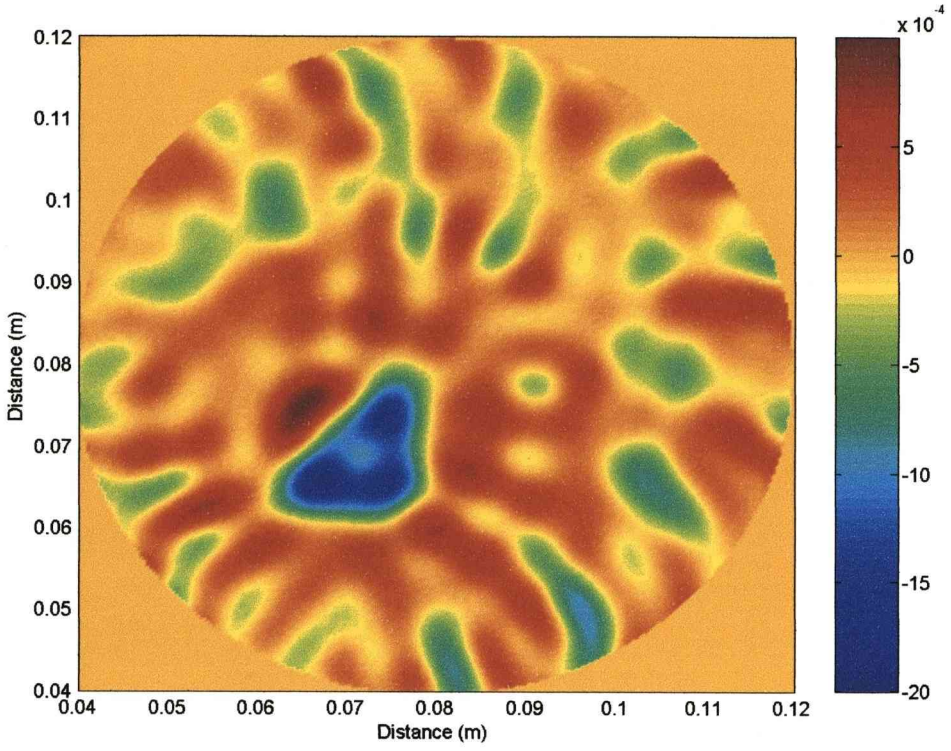

Figure 5.29 - PN sequence, triangle target, correlation

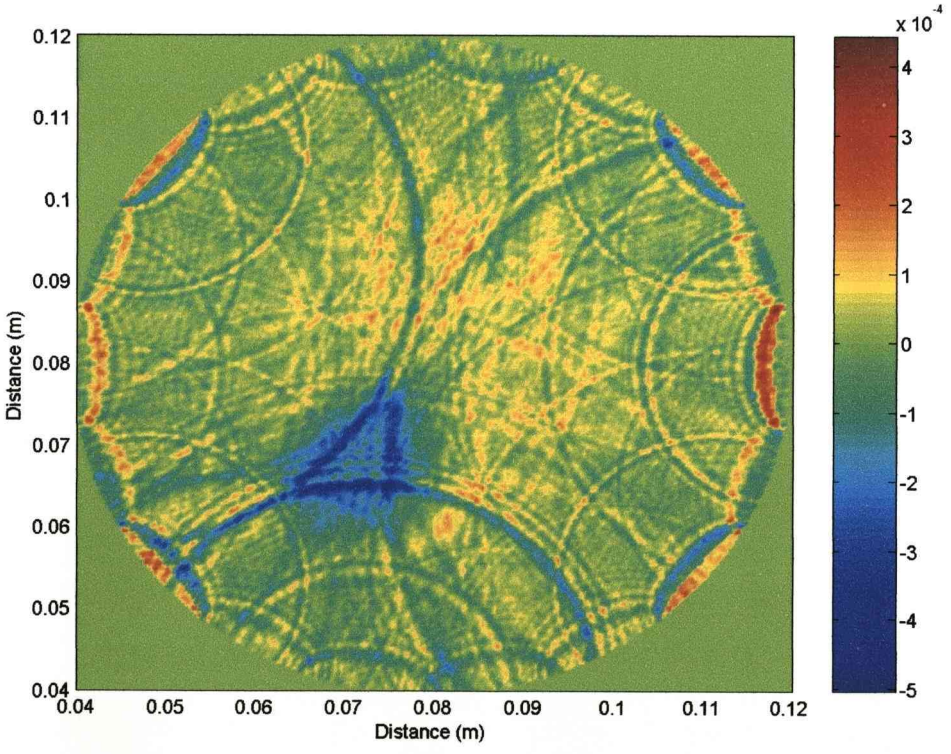

Figure 5.30 - PN sequence, triangle target, filter

The pulse method shows a target that is recognisably triangular with somewhat rounded edges in Figure 5.28. Correlation in Figure 5.29 matches more closely to the pulse method than filter with a nearly solid triangular detection and additional component just off the long edge (outward reflection). The filter results though show the edges of the triangle target clearly in Figure 5.30 but with clear contributory arcs showing as clutter.

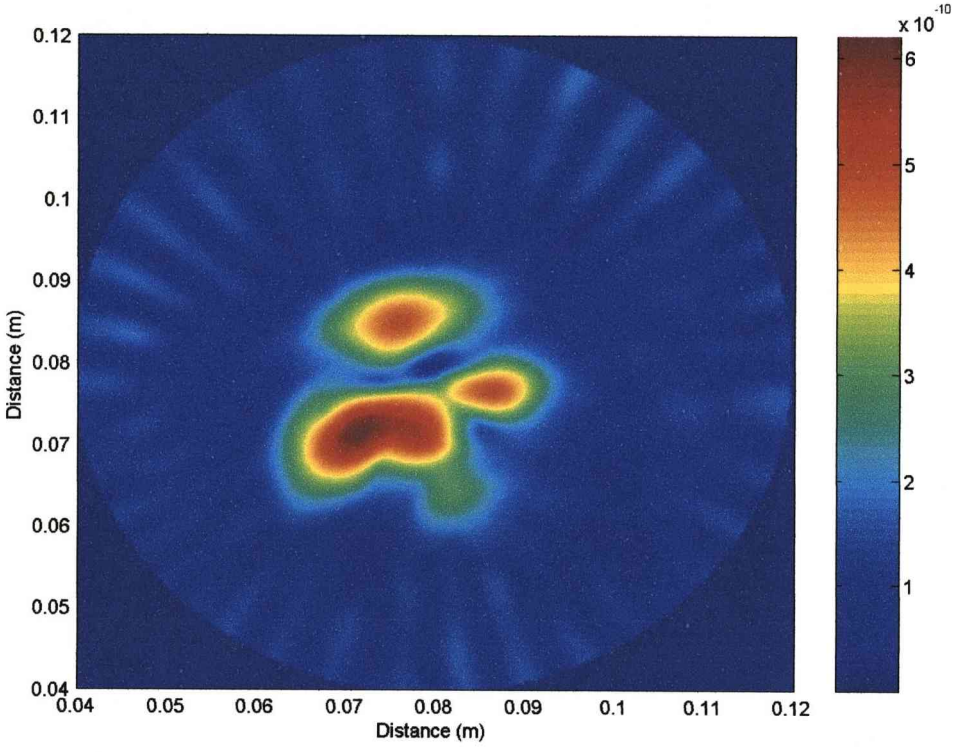

Figure 5.31 - Pulse, cluster target

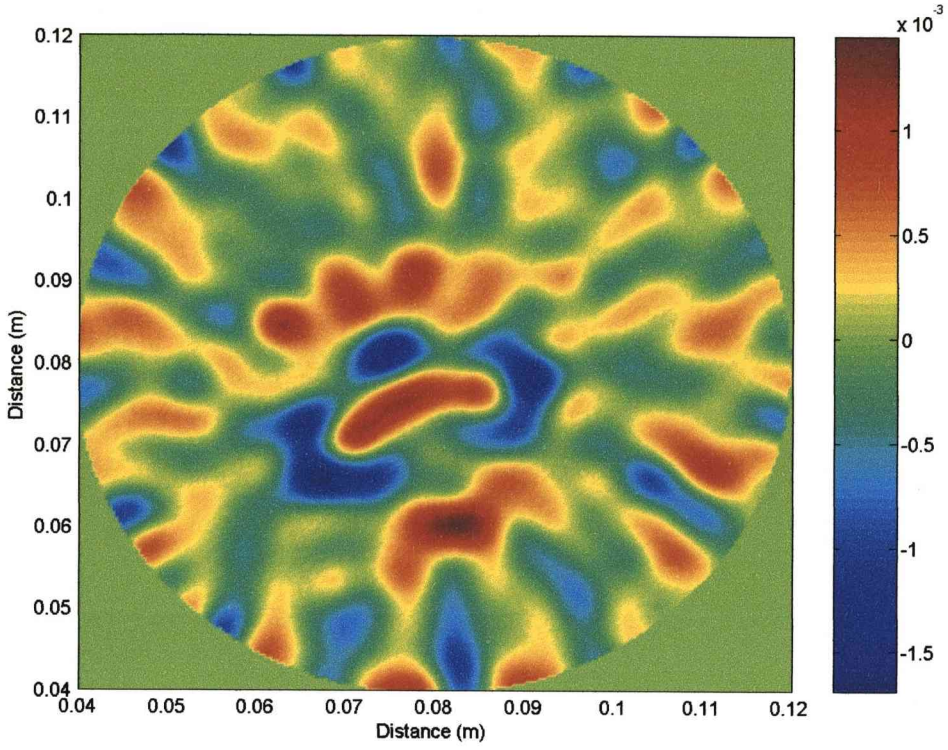

Figure 5.32 - PN sequence, cluster target, correlation

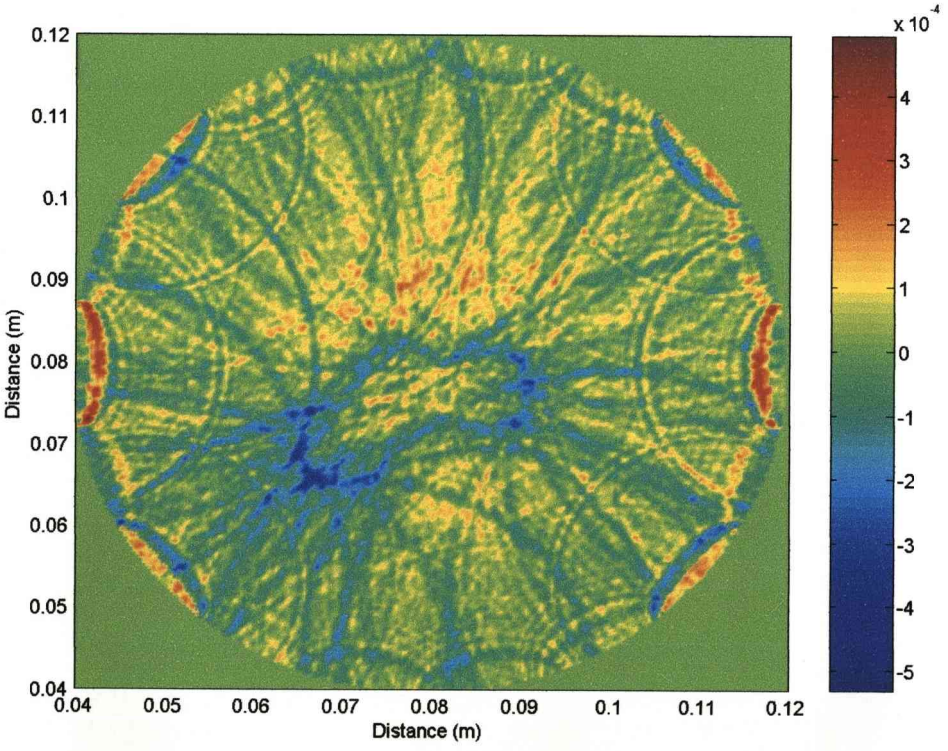

Figure 5.33 - PN sequence, cluster target, filter

This is the first of the harder shapes to detect. The pulse result in Figure 5.31 shows a target but shape is poor indicating a much larger and fragmented target. Correlation gives a good detection picking out some of the edges in Figure 5.32 but the filter result in Figure 5.33 shows quite visible ends of the shape with a weaker but still visible outline of the target shape connecting them. Clutter is still fairly high though.

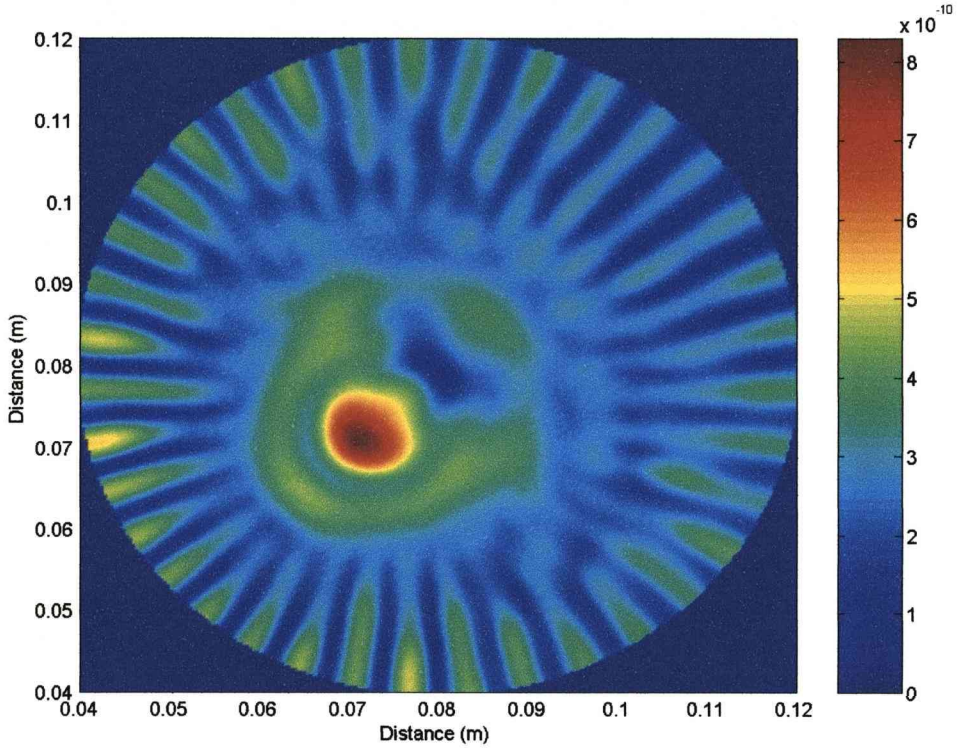

Figure 5.34 - Pulse, enclosed target

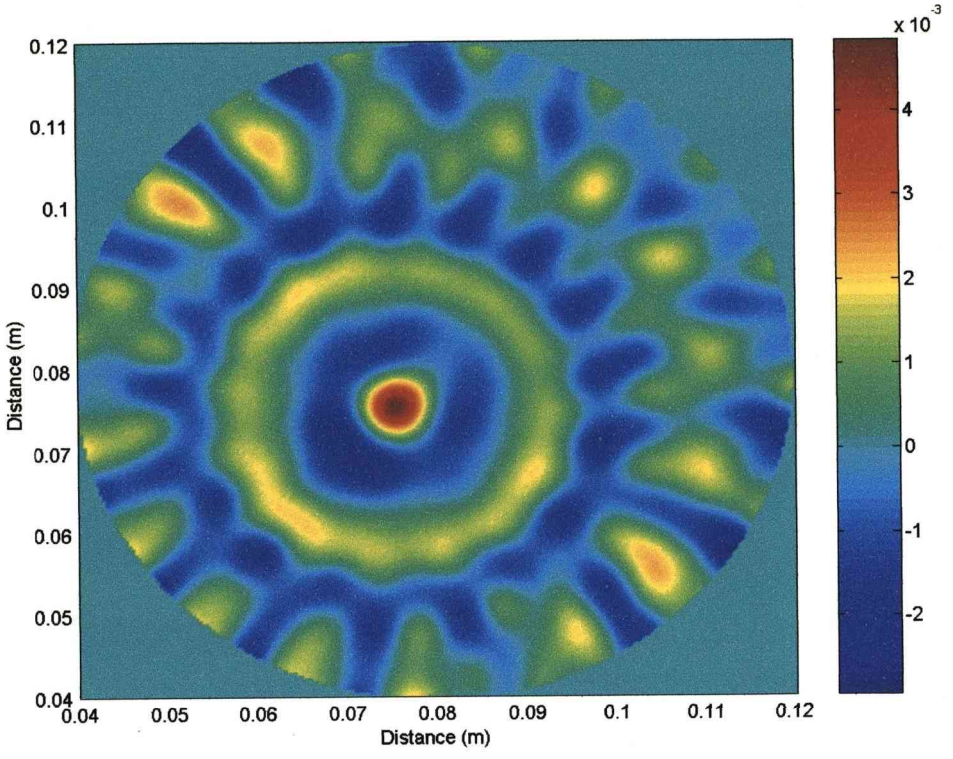

Figure 5.35 - PN sequence, enclosed target, correlation

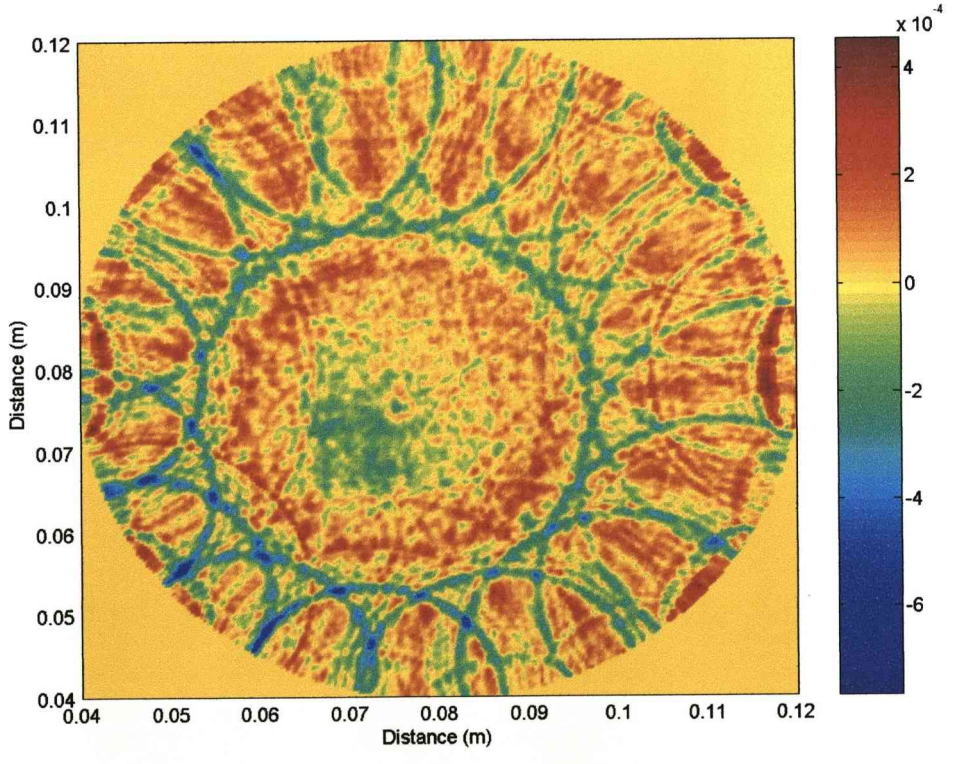

Figure 5.36 - PN sequence, enclosed target, filter

An enclosed target appears to be the trickiest target to image. In essence the outer shell is imaged first and then the internal target is imaged through the shell. The filter gives mixed results in Figure 5.36 with the outer circle picked out well with the inner square target roughly square shaped but fairly weak detection wise. Correlation in Figure 5.35 has a strong positive anomaly that doesn't quite pick out the actual target which should be negative and offset down and to the left (some of it is partially visible but not recognisably the target). The pulse results in Figure 5.34 show the square target strongly but no shape information and the outer circle is lost completely.

5.5.1 Section Conclusions

It is clear that the reasons for using a pulse method are for the excellent detection of a target. Although there are sometimes some anomalies around the target location (such as the cluster target) the target is clear in every image. Use of a PN sequence with correlation or filter processing produces a weakened target detection but the shape of the target is an important differentiating factor from the pulse method. Shape could be very useful in many imaging applications as it gives an indication of what the target could be as well as its location. With more identifiable information available further processing involving shape such as pattern recognition can be used. One of the noticeable benefits is the detection of both the outer circular shape and the internal square shape with the enclosed target test. The pulse method missed the outer target but it was picked up by the other methods and layered detections like these better aid identification. A further aid to identification is the polarity of the detections as that can give indications of the differences between the two media and so provides additional information that can help to identify what the target is composed of.

5.5.2 Trading Fineness for Strength of Detection

The fineness of detection means that sometimes the detection itself is weak. An object's shape is of no use if the object cannot be detected. Use of averaging or smoothing filters should spread the detection out over a longer period maximising the overlap and hence detection strength of the localisation algorithm. Correlation would give a similar result but it is sometimes too harsh. It is already used for the Gaussian method so some tests on the filter output should indicate if this could improve performance.

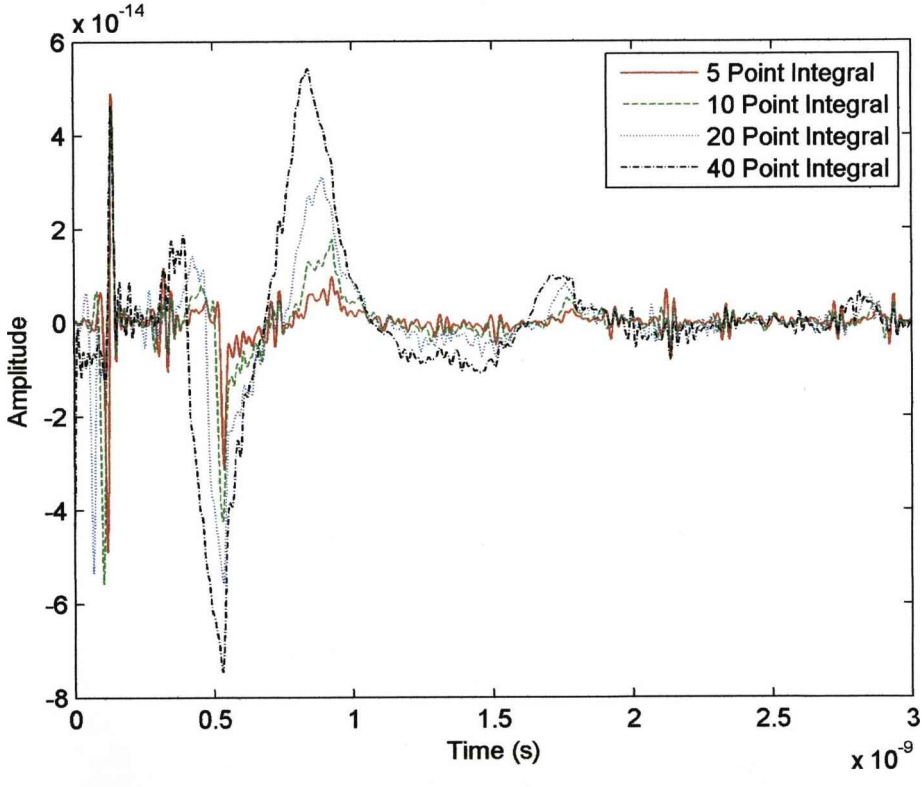

Figure 5.37 - Integral length comparison for a circle target

A single received signal for the circle target was selected, calibrated and ran through the filter. The output was then subjected to integration over various window lengths shown in Figure 5.37. The target detection is easily identified as the region between 0.5 and 1ns (front side reflection negative, rear side positive). Increasing the integral

length leads to a relative increase in detection strength over surrounding clutter and calibration artefacts. It also widens the potential location of the target's edges. 5, 10, 20 and 40 points where then used on the whole data set with the localisation performed after to give a new set of images. These images show some improvements and some degradation over the initial results depending on integral length. The best of these results are shown with an indication of the integral length used, again without spreading loss compensation applied.

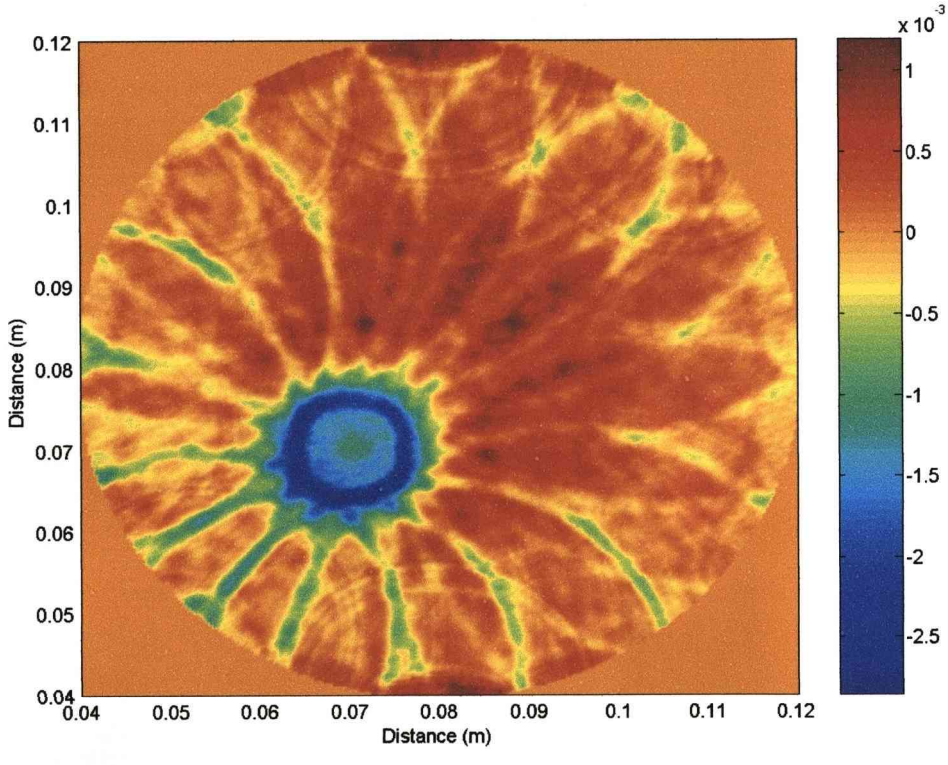

Figure 5.38 - Circle target – Integral length 10

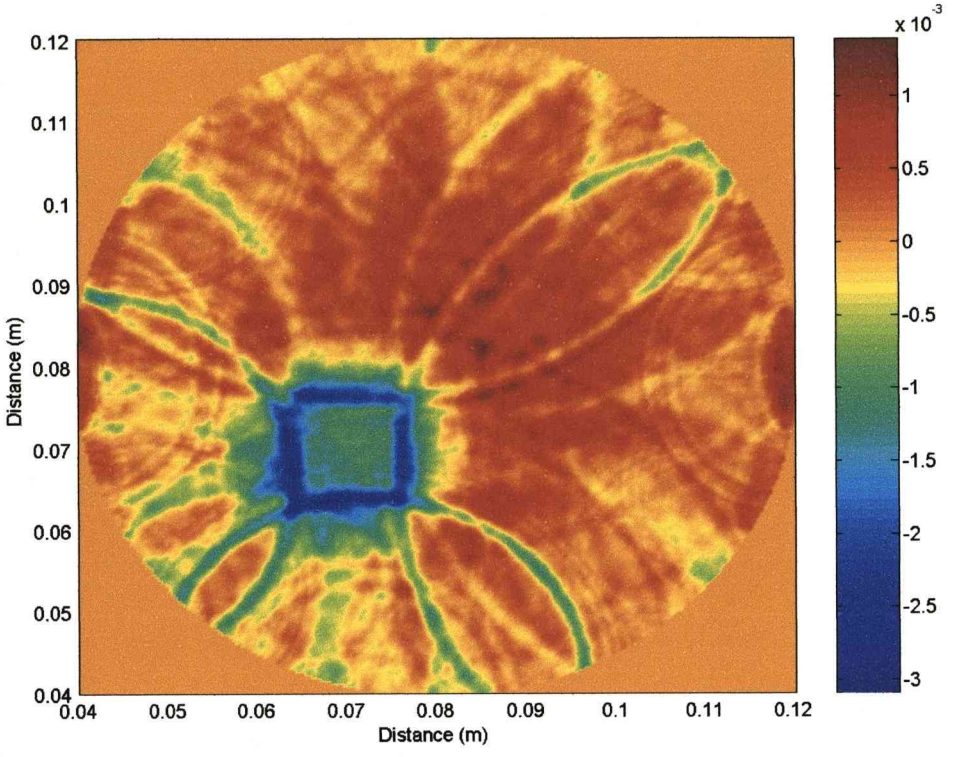

Figure 5.39 - Square target – Integral length 10

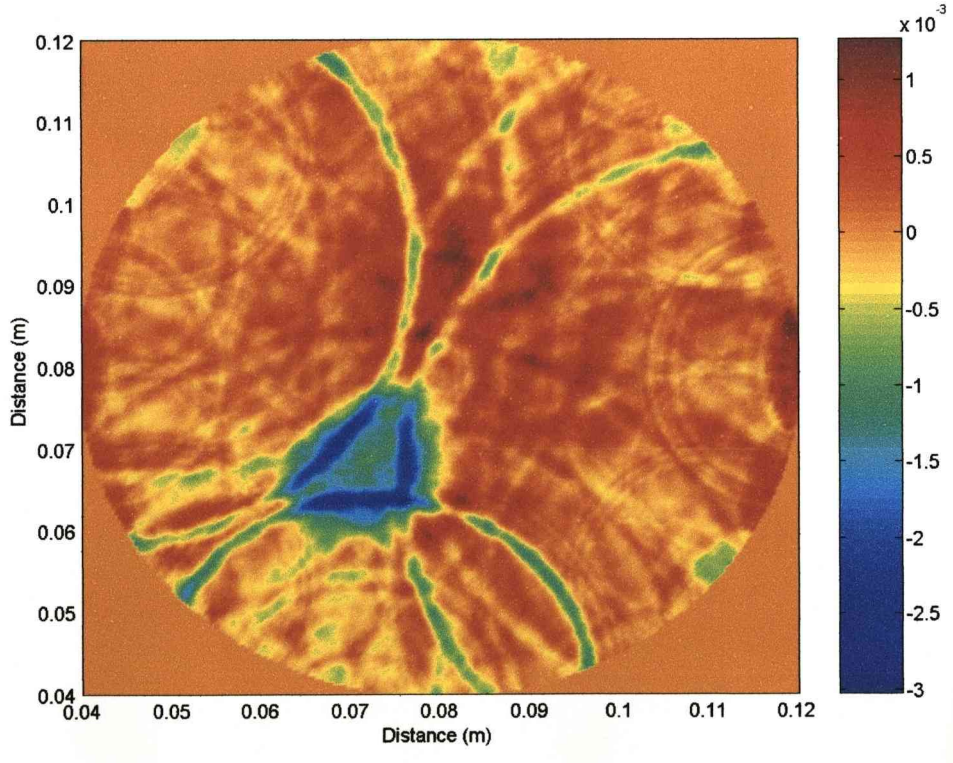

Figure 5.40 - Triangle target – Integral length 10

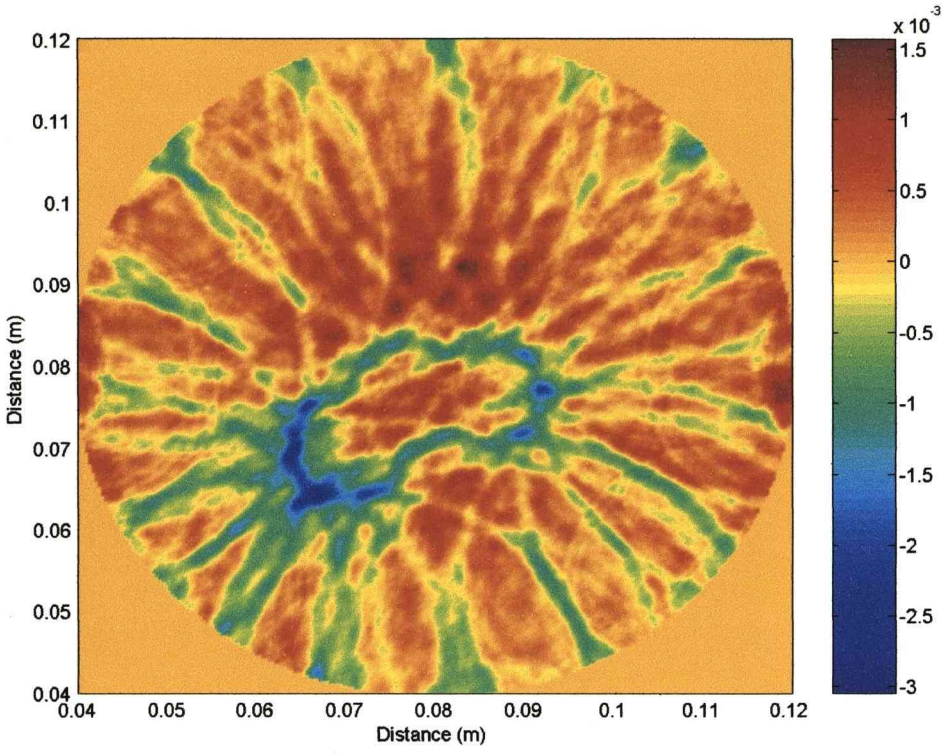

Figure 5.41 - Cluster target – Integral length 10

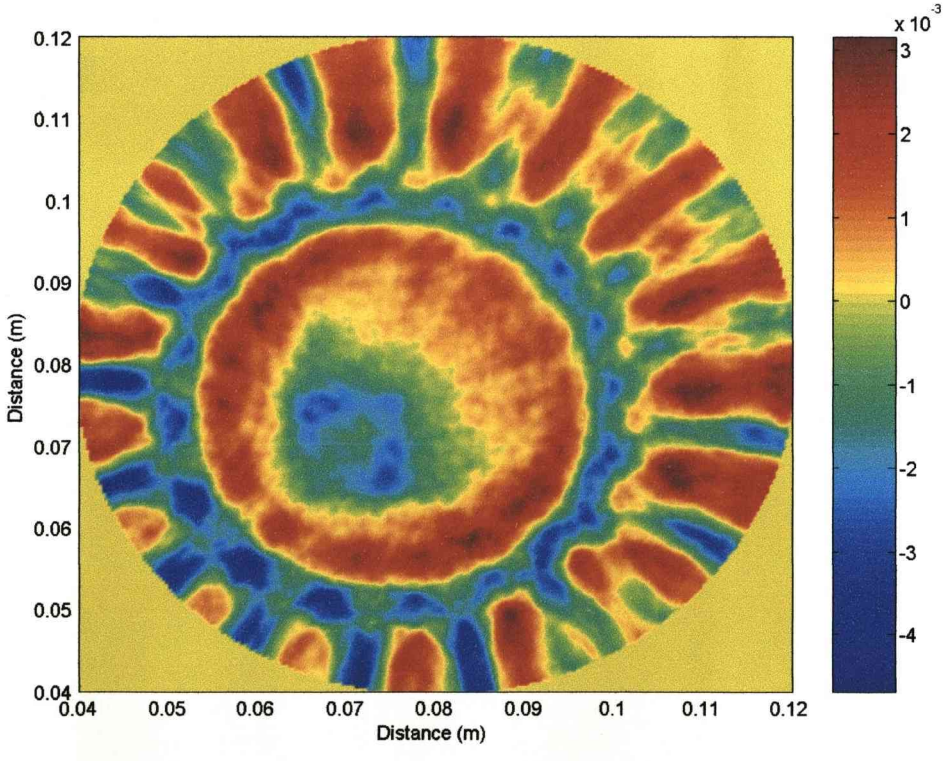

Figure 5.42 - Enclosed target – Integral length 20

Use of short integral window lengths doesn't lead to much improvement. At an integral length of 10 a significant improvement can be made in detection strength with minimal loss of fineness as shown in Figures 5.38 to 5.41. For the most difficult to image target where the target is enclosed, a longer integral length of 20 gives better results showing the outer shell and a clearly square inner target in Figure 5.42. Use of the filter method to discern target shape works well. The simpler shapes give best results but through trading-off some of the fineness for a stronger detection the more complex shapes also image well. Use of a differentiated Gaussian pulse with a GPR method leads to generally stronger detection but shape information is mostly lost. The trade off is user preference and in most cases a shorter integral will give best results but for imaging through another target a slightly longer integral brings out the weaker inner target better.

5.6 Identification of Relative Permittivity

As the filter method can detect the front and back reflections from a target object it should be possible to image an unknown object to identify its relative permittivity value and maybe also its conductivity. A target object of unknown relative permittivity of a selected shape is suspended in a known medium. It is scanned as for detection purposes but the rear reflection will appear displaced from its actual location because the target object will slow or speed up the transmitted wave speed.

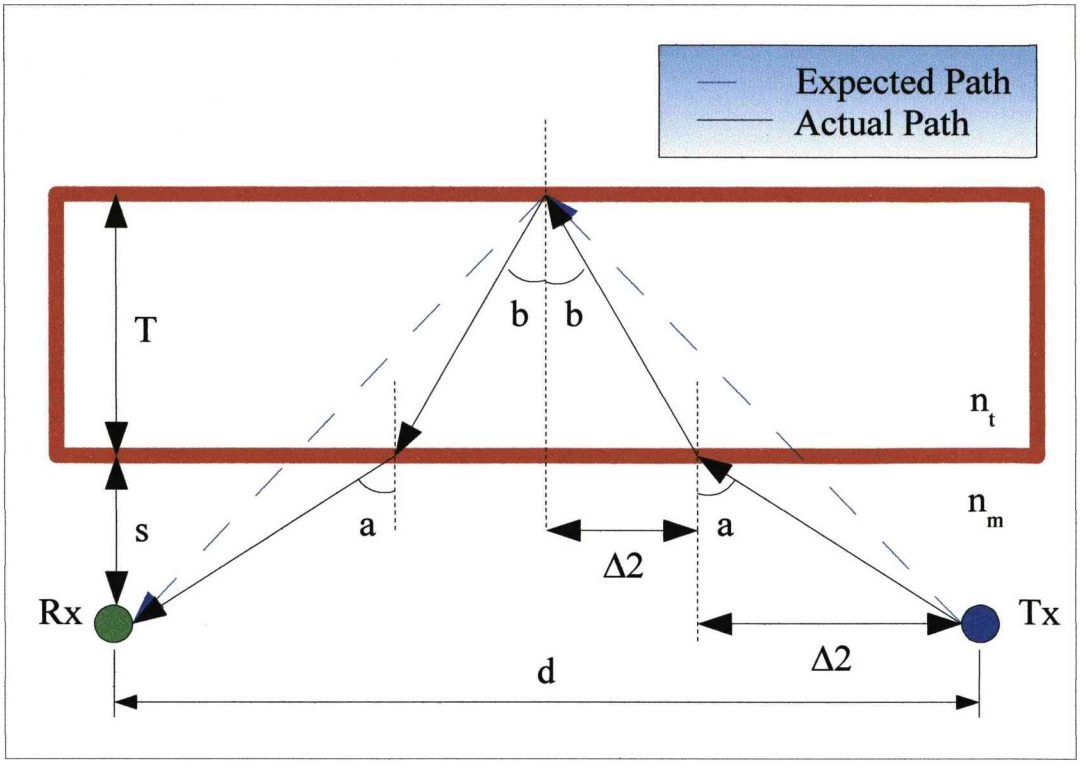

Figure 5.43 - Relative Permittivity Identification

The shift and sum processing method uses an estimate of the wave propagation velocity to take the time signal and convert it into distance. The wave, on hitting the target material will both refract and be altered in speed. Hence any use of an estimate of wave propagation velocity would lead to an inaccurate position of the rear reflection from the target. However the target material has a known thickness identified as T in Figure 5.43 so the difference between the actual and predicted rear reflection gives an idea about the relative permittivity of the target object.

$$\begin{aligned} n &= \sqrt{\epsilon_r \mu_r} \\ b &= \frac{n_t}{n_m} \sin(a) \\ \Delta 1 &= s \tan(a) \\ \Delta 2 &= T \tan(b) \end{aligned} \tag{5.5}$$

The distance between transmitter (Tx) and receiver (Rx), the value d, is fixed. Target to Tx/Rx separation (s) is also fixed. Angles a and b are related and depend only on

the indices of refraction (n_t for the target and n_m for the medium). The medium is pre-selected so the refractive index is known leaving the only unknown as the value of the refractive index for the target n_t. This is directly related to the dielectric permittivity of the medium. Combining some of the equations leads to a relationship given in Eq. (5.6) that can be solved iteratively for any given value of target ε_r to obtain a value for angle a, which leads to b.

$$d = 2(s\tan(a) + T\tan(b)) \tag{5.6}$$

The total travelled distance by the signal can then be worked out

$$\text{Distance}(D) = 2\sqrt{(s\tan(a))^2 + s^2} + 2\sqrt{\left(T\tan\left(\frac{\sqrt{\epsilon_t}}{\sqrt{\epsilon_m}}\sin(a)\right)\right)^2 + T^2} \tag{5.7}$$

This distance value could then be used for loss calculations or predictions. The distance in the known medium is expressed as the first part as in Eq. (5.8) and the section in the target as the second part as in Eq. (5.9).

$$\text{Distance in Known Medium } (D_m) = 2\sqrt{(s\tan(a))^2 + s^2} \tag{5.8}$$

$$\text{Distance in Unknown Medium } (D_t) = 2\sqrt{\left(T\tan\left(\frac{\sqrt{\varepsilon_t}}{\sqrt{\varepsilon_m}}\sin(a)\right)\right)^2 + T^2} \tag{5.9}$$

The front distance is through the known medium so wave speed is reduced by a known amount while the wave speed for the second section through the unknown medium will be a variable. Converting distance to time based on the estimated wave speeds yields an equation for the total time travelled.

$$\text{Time Travelled} = \frac{D_m}{V_m} + \frac{D_t}{V_t}$$
$$\text{where } V_m = \frac{1}{\sqrt{\varepsilon_0 \varepsilon_m \mu_0 \mu_m}} \text{ and } V_t = \frac{1}{\sqrt{\varepsilon_0 \varepsilon_t \mu_0 \mu_t}} \tag{5.10}$$

The system by which the target is localised to a point or points in space does not know nor take into account the unknown targets properties. It merely converts back to distance using the estimated wave speed, which is that through the known medium. The total perceived distance travelled then is given by

$$\begin{aligned}\text{Perceived Distance Travelled } (P) &= V_m\left(\frac{D_m}{V_m}+\frac{D_t}{V_t}\right)\\ &= D_m+\frac{V_m D_t}{V_t}=D_m+\frac{D_t\sqrt{\varepsilon_t}}{\sqrt{\varepsilon_m}}\end{aligned} \tag{5.11}$$

The position of the perceived rear reflection from the unknown target will be given by

$$\text{Perceived Rear Reflection } =\sqrt{\left(\frac{P}{2}\right)^2-\left(\frac{d}{2}\right)^2} \tag{5.12}$$

The actual rear reflection should be at $T+s$. The difference between the actual rear reflection and the perceived rear reflection would enable the deduction of the relative permittivity (ε_r) of an unknown material through the use of a graph of ε_r vs. reflection difference. A series of simulations were carried out with a slightly different set-up given in Figure 5.44. The dielectric permittivity (ε_r) was altered allowing a look at the effects this causes and to compare the rear reflection location as predicted by theory to the simulated results.

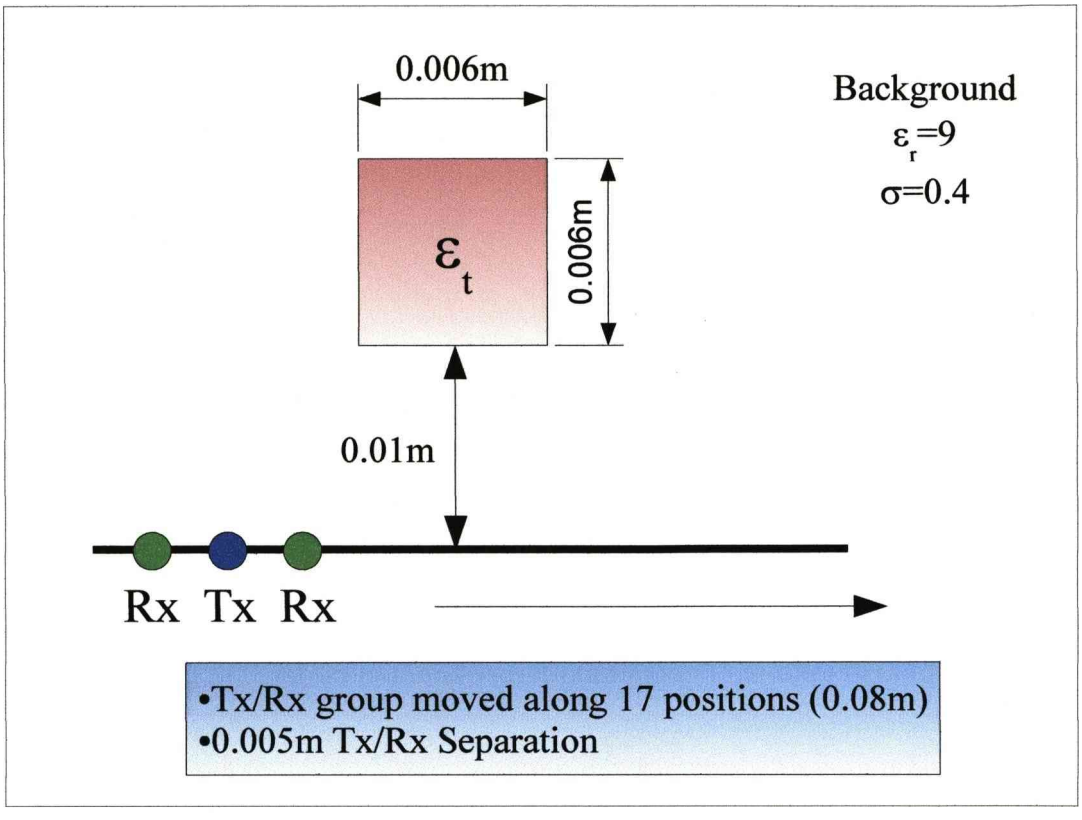

Figure 5.44 - Simulation Set-up

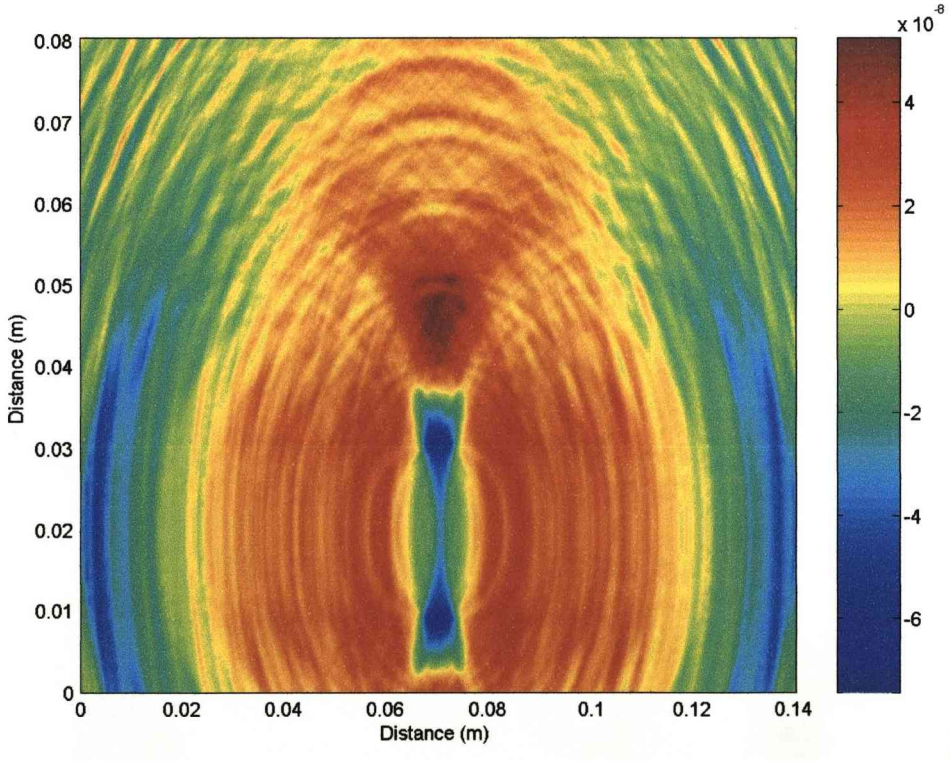

Figure 5.45 - Image of target with ε_r=10

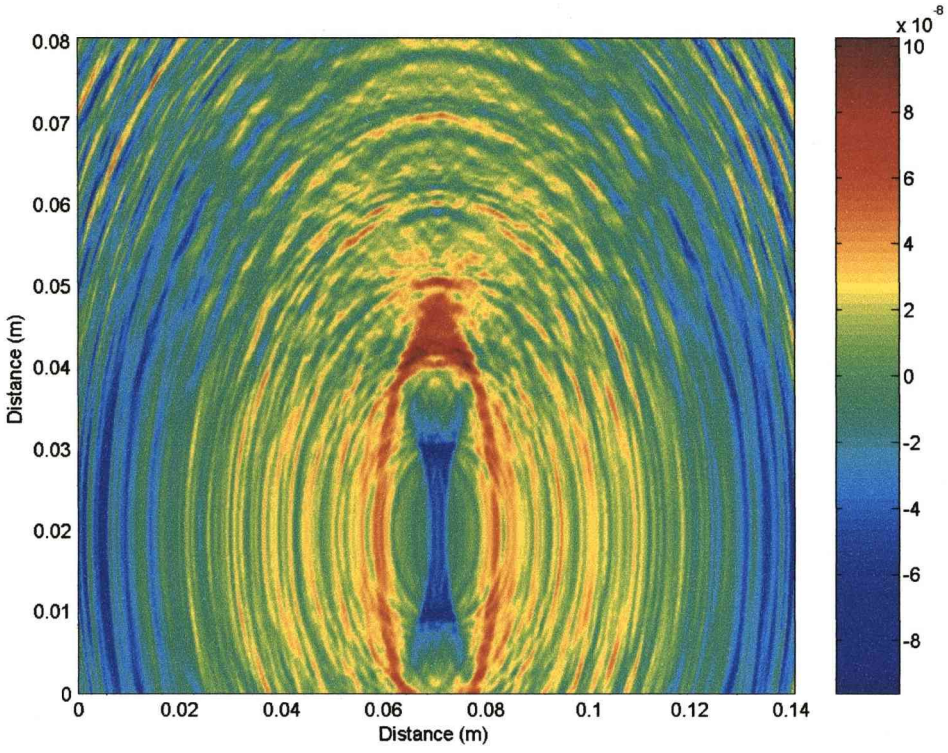

Figure 5.46 - Image of target with ε_r=30

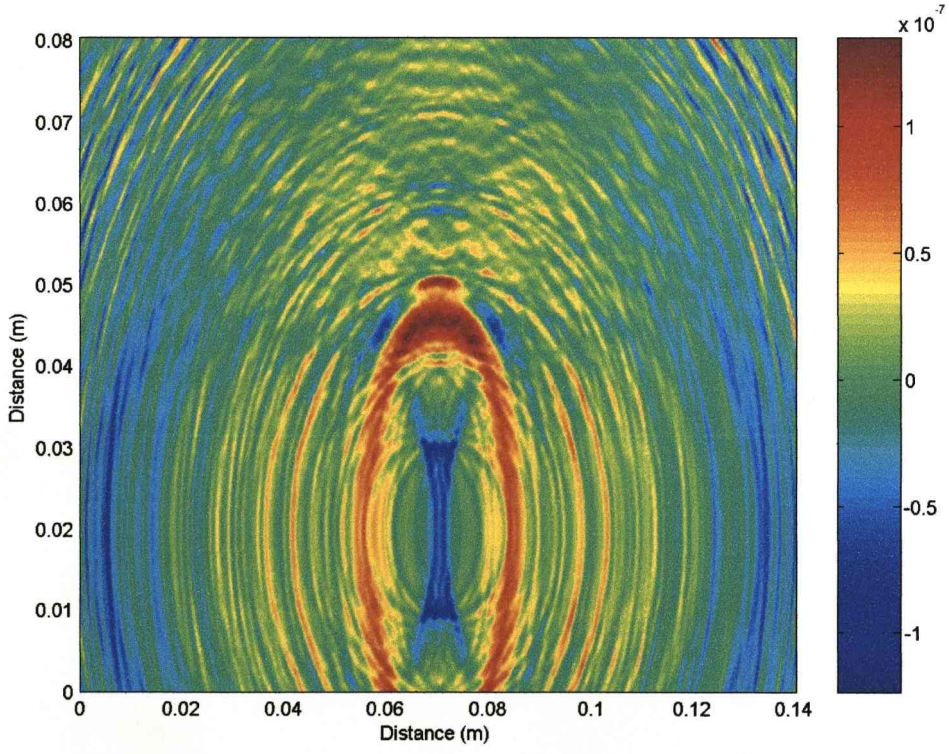

Figure 5.47 - Image of target with ε_r=50

The minute difference between a background dielectric value of 9 and a target value of 10 still enables detection as shown in Figure 5.45. The front reflection is strong (at 0.03m) if a little fuzzy but the rear reflection is indistinct. Remarkably the detection strength for the target has increased by a relatively small amount when compared to a large change in the targets dielectric permittivity in Figure 5.46. It does show an increase though of approximately $2x10^{-8}$. The front of the target is now clearly visible as a flat edge and the rear reflection has taken shape. Another small increase in detection by $2x10^{-8}$ and an even clearer rear reflection shown in Figure 5.47 are present.

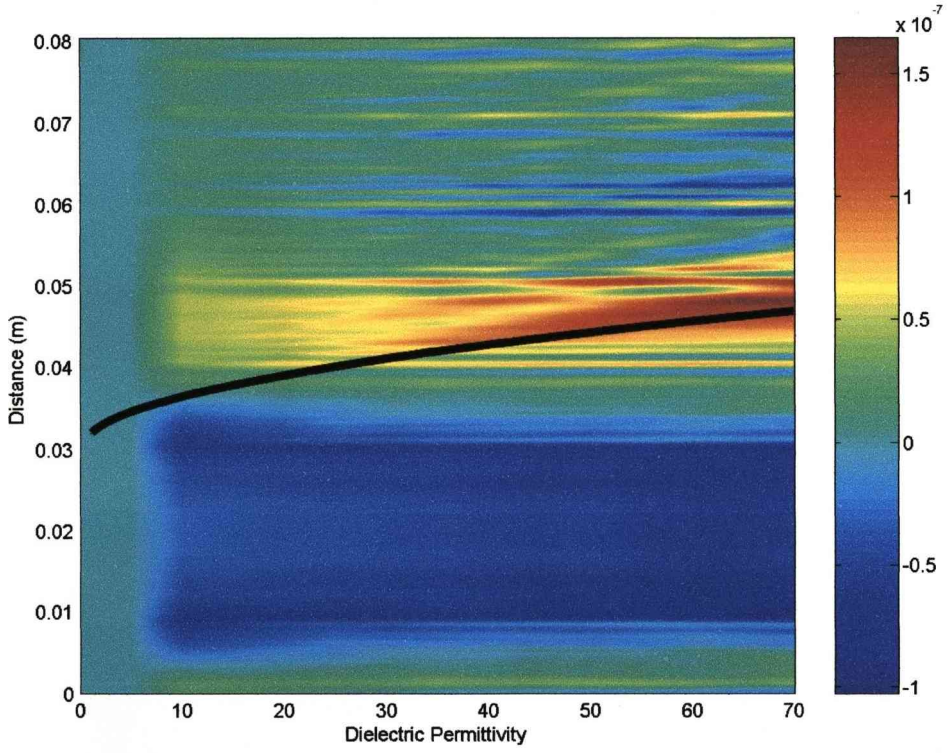

Figure 5.48 - Rear Reflection Comparison Cut showing predicted location of rear reflection

Multiple simulations were performed and then a cut was taken through the data to observe the change in rear reflection location with dielectric permittivity shown in Figure 5.48. A predicted location is also traced based on the theory above. The front

reflection remains at a constant position across the dielectric range observed although it does increase in strength. The rear reflection shows a strong correlation with the theory at the more clearly defined larger permittivity values. Although complex, various permutations lead to a generalised equation that would give an approximate estimate of the other parameters through known parameters and an experimental result (in this case the known dielectric permittivity or target thickness would lead to the other if the other parameters are known also). It appears possible then to image an unknown material under controlled conditions to identify its dielectric permittivity, at least for homogeneous materials.

5.7 Conclusions

A limited calibration method that generates its own reference signal for calibration subtraction has been shown to offer satisfactory results. Comparisons between integration of a differentiated Gaussian pulse to correlation or filter processing of a PN sequence has shown that the filter offers a better detection of a target's edges than either of the other methods. Further investigation has shown that as well as a detection of a target, the targets shape can also be discerned. Flat arrays only on one side of the target can show that the target is flat or curved while 360 degree arrays pick up the full shape of the target. One of the faults of the filter method is that of weaker detections, a potential solution to this is using a smoothing operation conducted on the data before localisation. The results show that this brings out a stronger detection but doesn't greatly impact the shape, leaving a good level of both detection and shape information when compared to using the pre-smoothed data with the filter or correlation methods. The PN sequence with filter method offers an alternative to

using a wideband pulse and integration that specifically aims to show a targets shape. This makes it useful where shape information is important or as a complimentary method that can be used after a target has been detected to identify a targets shape. The PN sequence and filter method was also the only one to image both the outer shell and inner target while retaining the shape of both.

Chapter 6

Conclusions and Further Work

6.1 Conclusions

In summary a set of techniques and methods that show benefit to imaging in general with particular reference to breast cancer detection has been presented. The majority of the work is on improving conventional methods but a new alternative angular method using a H-matrix transform has also been looked at. A summary and key conclusions for each section are reproduced below.

Comparisons between the commercial package CST Microwave Studio and the FDTD code has shown that the FDTD gives comparable results usable for simulation purposes to generate data although there are still some limitations on domain size and frequency with a limited boundary condition. The M-sequence has been introduced and explained with some of the potential benefits shown as well as its generation. Methods based on its use including correlation and filtering have been presented along with idealised performance in a range of situations including missing symbols, noise and filter parameter effects. Taking the FDTD generated data and using M-sequences for a range of tests led to use of a satisfactory calibration method that generates its own reference signal for calibration subtraction. Comparisons between pulse integration, correlation and filtering has shown that the correlation and filtering methods using M-sequence data lead to a detection of a target's edges at the cost of detection strength. Pulse integration still has the best detection strength but differentiating targets based on shape is difficult. Comparing the two methods using

M-sequence data has shown that the filter performs better overall by limiting the size of the sidelobes associated with correlation.

Theoretical work has been carried out on a H-matrix transform that bridges the gap between the abstract H-matrix and the environment its generated from. The original work [6] has been expanded and extended for imaging purposes. This transform gives an angular representation of the H-matrix that shows coupling of angles instead of antennas and a full investigation into its use as an imaging tool has been carried out. A modified version has been presented that allows focussing down to antenna limits instead of array limits. The transform is very fast but low image resolution limits its use.

6.2 Further Work

Data generation is a key part of this project and there are certainly improvements that could be made to the process by which this data was generated. The simulation itself could be much improved by a number of methods. Looking at the large array of available commercial packages for electromagnetic simulations is something that should be done, the pace of progress in these software packages is rapid so there is a good chance of suitable packages being available. Improvements to the FDTD code used is an area of improvement. Usability is poor so some improvements could be made there like building a user interface and an upgrade to 3D FDTD instead of just 2D. The boundary condition is something that needs to be updated to a better performing version (like the Perfect Boundary Condition) and incorporation of a method to build antennas into the simulation instead of using single cell sources would be another good inclusion.

The use of M-sequences needs more investigation with reference to correlation and the matching filter. A look at how the methods perform under various bandwidth limited scenarios would be useful to determine which method is better and under what circumstances. A look at which symbols to use and their effects on the correlation or filter output should be looked at as the standard Gaussian type pulses don't appear to work very well. Limitations on the LMS method for the filter is an obvious area for improvement to increase robustness, particularly with reference to noisy scenarios. One of the advantages of M-sequences is that multiple can be transmitted and received simultaneously, transmitted signals are long so a speed up can be produced. Finding the right balance between M-sequence length and number of simultaneous transmissions possible is an area to look at and is well suited to experimental testing. Also how much of an improvement if any does an M-sequence offer over an averaging of the same symbol transmitted multiple times sequentially.

Looking at the calibration method and curved arrays there exists a possibility to use more than just the adjacent antennas to receive by averaging all equidistant receiving antenna. This would offer improvements by using many more signals but how does only part of a curved array effect it compared to a full 360° array and will some cross target signals skew the calibration leading to performance degradation. Finally the H-matrix transform has several options open to look into. The first is looking at differently shaped antenna arrays that may offer better performance. Trying to combine it with other methods such as those conventional methods covered is another option. Probably the easiest action for further study is testing it in a simulation such as using FDTD. This would also provide a good test for multiple simultaneous M-sequences as they are ideal for generating H-matrices. Then a proper

comparison to conventional methods could be made using the same data set.

Appendix

A.1 Mathematical Work

A.1.1 Pulse Correlation

This is the workings for the sections for the pulse correlation problem in Chapter 3. Eq. (A.1) splits the problem down into the required sections and lists the integrals to solve. Sections 1, 2, 6 and 7 are always zero leaving 3 sections to integrate (with two parts for section 4). The solutions to those sections are given in Eqs.(A.2-5)

$$
\begin{aligned}
&(1) \quad R(\Delta) = \int_{-\infty}^{d-\frac{a}{b}} (0)(0)\,dx \\
&(2) \quad R(\Delta) = \int_{d-\frac{a}{b}}^{d-\frac{a}{b}+\Delta} (a+b(x-d))(0)\,dx \\
&(3) \quad R(\Delta) = \int_{d-\frac{a}{b}+\Delta}^{d} (a+b(x-d))(a+b(x-d-\Delta))\,dx \\
&(4a) \quad R(\Delta) = \int_{d}^{d+\Delta} (a-b(x-d))(a+b(x-d-\Delta))\,dx \\
&(4b) \quad R(\Delta) = \int_{d-\frac{a}{b}+\Delta}^{d+\frac{a}{b}} (a-b(x-d))(a+b(x-d-\Delta))\,dx \\
&(5) \quad R(\Delta) = \int_{d+\Delta}^{d+\frac{a}{b}} (a-b(x-d))(a-b(x-d-\Delta))\,dx \\
&(6) \quad R(\Delta) = \int_{d+\frac{a}{b}}^{d+\frac{a}{b}+\Delta} (0)(a-b(x-d-\Delta))\,dx \\
&(7) \quad R(\Delta) = \int_{d+\frac{a}{b}+\Delta}^{\infty} (0)(0)\,dx
\end{aligned}
\tag{A.1}
$$

A.1.1.1 Section 3

$$R(\Delta)=\int_{d-\frac{a}{b}+\Delta}^{d}(a+b(x-d))(a+b(x-d-\Delta))dx$$

$$=\int_{d-\frac{a}{b}+\Delta}^{d}(b^2x^2+x(2ab-b^2\Delta-2b^2d)+(a^2-ab\Delta-2abd+b^2d\Delta+b^2d^2))dx$$

$$=\left[\frac{b^2x^3}{3}+\frac{x^2}{2}(2ab-b^2\Delta-2b^2d)+x(a^2-ab\Delta-2abd+b^2d\Delta+b^2d^2)\right]_{d-\frac{a}{b}+\Delta}^{d}$$

$$=\frac{b^2d^3}{3}+\frac{d^2}{2}(2ab-b^2\Delta-2b^2d)+d(a^2-ab\Delta-2abd+b^2d\Delta+b^2d^2) \qquad \text{(A.2)}$$

$$-\left(\frac{b^2\left(d-\frac{a}{b}+\Delta\right)^3}{3}+\frac{\left(d-\frac{a}{b}+\Delta\right)^2}{2}(2ab-b^2\Delta-2b^2d)\right)$$

$$-\left(d-\frac{a}{b}+\Delta\right)(a^2-ab\Delta-2abd+b^2d\Delta+b^2d^2)$$

$$=\frac{b^2\Delta^3}{6}-\frac{a^2\Delta}{2}+\frac{a^3}{3b}$$

A.1.1.2 Section 4a

$$R(\Delta)=\int_{d}^{d+\Delta}(a-b(x-d))(a+b(x-d-\Delta))dx$$

$$=\int_{d}^{d+\Delta}(-b^2x^2+x(b^2\Delta+2b^2d)+(a^2-ab\Delta-b^2d\Delta-b^2d^2))dx$$

$$=\left[-\frac{b^2x^3}{3}+\frac{x^2}{2}(b^2\Delta+2b^2d)+x(a^2-ab\Delta-b^2d\Delta-b^2d^2)\right]_{d}^{d+\Delta} \qquad \text{(A.3)}$$

$$=-\frac{b^2(d+\Delta)^3}{3}+\frac{(d+\Delta)^2}{2}(b^2\Delta+2b^2d)+(d+\Delta)(a^2-ab\Delta-b^2d\Delta-b^2d^2)$$

$$-\left(\frac{b^2d^3}{3}+\frac{d^2}{2}(b^2\Delta+2b^2d)+d(a^2-ab\Delta-b^2d\Delta-b^2d^2)\right)$$

$$=\frac{b^2\Delta^3}{6}-ab\Delta^2+a^2\Delta$$

A.1.1.3 Section 4b

$$
\begin{aligned}
R(\Delta) &= \int_{d-\frac{a}{b}+\Delta}^{d+\frac{a}{b}} (a-b(x-d))(a+b(x-d-\Delta))dx \\
&= \int_{d-\frac{a}{b}+\Delta}^{d+\frac{a}{b}} (-b^2x^2+x(b^2\Delta+2b^2d)+(a^2-ab\Delta-b^2d\Delta-b^2d^2))dx \\
&= \left[-\frac{b^2x^3}{3}+\frac{x^2}{2}(b^2\Delta+2b^2d)+x(a^2-ab\Delta-b^2d\Delta-b^2d^2)\right]_{d-\frac{a}{b}+\Delta}^{d+\frac{a}{b}} \\
&= -\frac{b^2\left(d+\frac{a}{b}\right)^3}{3}+\frac{\left(d+\frac{a}{b}\right)^2}{2}(b^2\Delta+2b^2d)+\left(d+\frac{a}{b}\right)(a^2-ab\Delta-b^2d\Delta-b^2d^2) \\
&\quad -\left(\frac{b^2\left(d-\frac{a}{b}+\Delta\right)^3}{3}+\frac{\left(d-\frac{a}{b}+\Delta\right)^2}{2}(b^2\Delta+2b^2d)\right) \\
&\quad -\left(d-\frac{a}{b}+\Delta\right)(a^2-ab\Delta-b^2d\Delta-b^2d^2) \\
&= -\frac{b^2\Delta^3}{6}+ab\Delta^2-2a^2\Delta+\frac{4a^3}{3b}
\end{aligned}
\tag{A.4}
$$

A.1.1.4 Section 5

$$
\begin{aligned}
R(\Delta) &= \int_{d+\Delta}^{d+\frac{a}{b}} (a-b(x-d))(a-b(x-d-\Delta))dx \\
&= \int_{d+\Delta}^{d+\frac{a}{b}} (b^2x^2+x(-2ab-b^2\Delta-2b^2d)+(a^2+ab\Delta+2abd+b^2d\Delta+b^2d^2))dx \\
&= \left[\frac{b^2x^3}{3}+\frac{x^2}{2}(-2ab-b^2\Delta-2b^2d)+x(a^2+ab\Delta+2abd+b^2d\Delta+b^2d^2)\right]_{d+\Delta}^{d+\frac{a}{b}} \\
&= \frac{b^2\left(d+\frac{a}{b}\right)^3}{3}+\frac{\left(d+\frac{a}{b}\right)^2}{2}(-2ab-b^2\Delta-2b^2d) \\
&\quad +\left(d+\frac{a}{b}\right)(a^2+ab\Delta+2abd+b^2d\Delta+b^2d^2) \\
&\quad -\left(\frac{b^2(d+\Delta)^3}{3}+\frac{(d+\Delta)^2}{2}(-2ab-b^2\Delta-2b^2d\right) \\
&\quad -(d+\Delta)(a^2+ab\Delta+2abd+b^2d\Delta+b^2d^2) \\
&= \frac{b^2\Delta^3}{6}-\frac{a^2\Delta}{2}+\frac{a^3}{3b}
\end{aligned}
\tag{A.5}
$$

A.2 Matlab Code

A.2.1 2D FDTD

This FDTD code was used to generate most of the simulated data with minor modifications to give alternate transmitter or receiver locations and target locations or values. It is based on the 3D FDTD code given later.

```
% It outputs 2 received signals, the time axis and the transmitted
signal. It's
% inputs are the transmitter and receiver coordinates, the target
type flag,
% signal flag and a variation flag
[output1,output2,time_axis,signal_z]=FDTD_2D_flat_c(tx_x,rx1_x,rx2_x,
      tx_y,rx1_y,rx2_y,t_type,sig_val,variation)
% Constants, Epsilon and Sigma both vary within the grid
% Mu is constant

eps_0=8.8542e-12; mu_0=1.2566e-6;
tissue_eps=9; tissue_mu=1; tissue_sigma=0;
tumour_eps=40; tumour_mu=1; tumour_sigma=4;
wave_speed=1/sqrt(eps_0*tissue_eps*mu_0*tissue_mu);

% Cell size ideally at least 1/20 of a wavelength
cell_size=0.0005;

% Courant stability condition calculation for the time stepping
time_step = 1.0/(wave_speed*sqrt(1/(cell_size^2)+1/(cell_size^2)+1));

% Set this for the length of time to simulate
simulation_time=5e-9;

% This next section generates the PN sequence, 31 bits.  It uses a
shift
% register with taps XOR'd together to give the next input.

PN_bandwidth=5e9;
seq_length=(2^5)-1;
counter=1;
seed=[0 1 1 0 1];
for q=1:seq_length
      for p=1:5
            if p==1
                  temp(p)=xor(seed(5),seed(3));
                  if temp(p)==1
                        PN_sequence(counter)=1;
                  elseif temp(p)==0
                        PN_sequence(counter)=-1;
                  end
                  counter=counter+1;
            else
                  temp(p)=seed(p-1);
            end
      end
```

```
      seed=temp;
end

%Specifying the size of the working domain
Nx=round(0.16/cell_size);
Ny=round(0.16/cell_size);

% Cell locations for the transmitter and 2 receivers
tx_cell=round(tx_x/cell_size);
ty_cell=round(tx_y/cell_size);
rx_cell_1=round(rx1_x/cell_size);
rx_cell_2=round(rx2_x/cell_size);
ry_cell_1=round(rx1_y/cell_size);
ry_cell_2=round(rx2_y/cell_size);

% Basic Tissue parameters
eps_z=(ones(Nx-1,Ny-1)*tissue_eps*eps_0);
sigma_z=(ones(Nx-1,Ny-1)*tissue_sigma);

% Setting up the targets.  It generates a range of targets based on
% input parameters.  It also allows randomised tissue parameters
% within a circular
% area and generates a circular skin surface.  The targets are
% 1 - Circle
% 2 - Square
% 3 - Triangle
% 4 - 3 linked filled circles
% 5 - A square with enclosing circle

for p=1:Nx-1
    for q=1:Ny-1
        circle_val=sqrt((p-(0.08/cell_size))^2
                        +(q-(0.08/cell_size))^2);
        target_val_1=sqrt((p-(0.07/cell_size))^2
                        +(q-(0.07/cell_size))^2);
        tv1=sqrt((p-(0.07/cell_size))^2+(q-(0.07/cell_size))^2);
        tv2=sqrt((p-(0.0765/cell_size))^2+(q-(0.0765/cell_size))^2);
        tv3=sqrt((p-(0.0857/cell_size))^2+(q-(0.0765/cell_size))^2);
        encloser=sqrt((p-(0.075/cell_size))^2
                        +(q-(0.075/cell_size))^2);
        if circle_val>=(0.04/cell_size)
                        & circle_val<=(0.042/cell_size)
            eps_z(p,q)=tumour_eps*eps_0;
            sigma_z(p,q)=tumour_sigma;
        end
        if circle_val<(0.04/cell_size) & variation==1
            eps_z(p,q)=((((rand-0.5)/5)*tissue_eps)+tissue_eps)
                                                    *eps_0;
            sigma_z(p,q)=((((rand-0.5)/5)*tissue_sigma)
                                                +tissue_sigma);
        end
        if t_type==1 & target_val_1<=(0.005/cell_size)
            eps_z(p,q)=tumour_eps*eps_0;
            sigma_z(p,q)=tumour_sigma;
        elseif t_type==2 & p>=(0.065/cell_size)
                & p<=(0.075/cell_size) & q>=(0.065/cell_size)
                                    & q<=(0.075/cell_size)
            eps_z(p,q)=tumour_eps*eps_0;
            sigma_z(p,q)=tumour_sigma;
```

```
        elseif t_type==3 & p>=(0.065/cell_size)
                  & p<=(0.075/cell_size) & q>=(0.065/cell_size)
                                    & q<=(0.075/cell_size) & p>=q
            eps_z(p,q)=tumour_eps*eps_0;
            sigma_z(p,q)=tumour_sigma;
        elseif t_type==4 & (tv1<=(0.005/cell_size) |
                  tv2<=(0.005/cell_size) | tv3<=(0.005/cell_size))
            eps_z(p,q)=tumour_eps*eps_0;
            sigma_z(p,q)=tumour_sigma;
        elseif t_type==5 & ((p>=(0.065/cell_size)
                  & p<=(0.075/cell_size) & q>=(0.065/cell_size)
                  & q<=(0.075/cell_size))|(encloser>=(0.02/cell_size)
                                    & encloser<=(0.022/cell_size)))
            eps_z(p,q)=tumour_eps*eps_0;
            sigma_z(p,q)=tumour_sigma;
        end

    end
end


% Setting up the variables required
counter=1;

Ez_grid=zeros(Nx+1,Ny+1);
Ez_grid_core=zeros(Nx-1,Ny-1);
Hx_grid=zeros(Nx+1,Ny);
Hx_grid_core=zeros(Nx-1,Ny);
Hy_grid=zeros(Nx,Ny+1);
Hy_grid_core=zeros(Nx,Ny-1);

old_adj_Ez_left=zeros(Ny+1);
old_Ez_left=zeros(Ny+1);
old_adj_Ez_right=zeros(Ny+1);
old_Ez_right=zeros(Ny+1);

old_adj_Ez_back=zeros(Nx+1);
old_Ez_back=zeros(Nx+1);
old_adj_Ez_forward=zeros(Nx+1);
old_Ez_forward=zeros(Nx+1);

% Grid Mapping
% H Grids start at n=1 corresponding to (n-0.5)*time_step
% E Grids start at n=1 corresponding to (n-1)*time_step

% Signal generation, either a Differentiated Gaussian Pulse
% (sig_val==1) or PN  % sequence (sig_val==2)
pulse_unit=round(0.5e-9/(6*time_step));
for n=1:(round((simulation_time)/(time_step)))
    time_axis(n)=(n-1)*time_step;
    if sig_val==1
        signal_z(n)=(((n-1)*time_step)-(4*75e-12))
            *exp(-((((n-1)*time_step)-(4*75e-12))^2)/(75e-12^2));
    elseif sig_val==2
        signal_z(n)=sin(2*pi*5e9.*time_axis(n))
            *PN_sequence(floor(time_axis(n)/(1/PN_bandwidth))+1);
    end
end
```

```
% The simulation itself.  It uses a matrix multiplication method to
% speed up the % Matlab process over loops, although it does require
% additional variables.
for n=1:(round((simulation_time)/(time_step)))

    Ez_grid(tx_cell,ty_cell)=signal_z(n);

    mu_x=tissue_mu*mu_0;
    alpha_x=time_step/(cell_size*mu_x);
    old_Hx = Hx_grid;
    Hx_grid=old_Hx+alpha_x*(-diff(Ez_grid,1,2));

    mu_y=tissue_mu*mu_0;
    alpha_y=time_step/(cell_size*mu_y);
    old_Hy = Hy_grid;
    Hy_grid=old_Hy+alpha_y*(diff(Ez_grid,1,1));

    % 1st Order Mur boundary Conditions for H
    % Due to the nature of the grid there is never any need to
    % calculate the boundary conditions for the H field because the E
    % field vectors that are required for correct calculation at the
    % edges of the H grid are present.

    for p=2:Nx
        Hx_grid_core(p-1,:)=Hx_grid(p,:);
    end
    for q=2:Ny
        Hy_grid_core(:,q-1)=Hy_grid(:,q);
    end

    beta_z=(1-((sigma_z*time_step)./(2*eps_z)))
                        ./(1+((sigma_z*time_step)./(2*eps_z)));
    delta_z=((time_step./eps_z))./(1+((sigma_z*time_step)
                        ./(2*eps_z)))./cell_size;

    old_Ez_core = Ez_grid_core;
    Ez_grid_core=(beta_z.*old_Ez_core)
      +(delta_z.*(diff(Hy_grid_core,1,1)- diff(Hx_grid_core,1,2)));

    for l=2:Nx
        for m=2:Ny
            Ez_grid(l,m,:)=Ez_grid_core(l-1,m-1,:);
        end
    end

    mur_coeff=((wave_speed*time_step)-cell_size)
                        /((wave_speed*time_step)+cell_size);

    % Grid Limit Terminolgy for Mur ABC
    %
    %            /\                 _______
    %           /  \              /      /|
    %       Top  ||              /_____/  |          bottom  ||
    %                            |      | /                \  /
    %                            |_____|/                  \/
    %
    %                                                    z |
    %                                                      |
    %                            _______                   |_____ x
    %                           /      /|
```

```
%       Left  <==           /_____/ |   Right ==>         /
%                           |     | /                 y /
%                           |_____|/
%
%
%                            _____
%                           /     /|                  /\
%       Foward //           /_____/ |           back //
%             //            |     | /               //
%            \/             |_____|/
%

% 1st order Mur boundary conditions for E

for j=1:Ny+1
    Ez_grid(1,j)=old_adj_Ez_left(j)
              +mur_coeff*(Ez_grid(2,j)-old_Ez_left(j));
    Ez_grid(Nx+1,j)=old_adj_Ez_right(j)
              +mur_coeff*(Ez_grid(Nx,j)-old_Ez_right(j));
    old_adj_Ez_left(j)=Ez_grid(2,j);
    old_Ez_left(j)=Ez_grid(1,j);
    old_adj_Ez_right(j)=Ez_grid(Nx,j);
    old_Ez_right(j)=Ez_grid(Nx+1,j);
end

for i=1:Nx+1
    Ez_grid(i,1)=old_adj_Ez_back(i)
              +mur_coeff*(Ez_grid(i,2)-old_Ez_back(i));
    Ez_grid(i,Ny+1)=old_adj_Ez_forward(i)
              +mur_coeff*(Ez_grid(i,Ny)-old_Ez_forward(i));
    old_adj_Ez_back(i)=Ez_grid(i,2);
    old_Ez_back(i)=Ez_grid(i,1);
    old_adj_Ez_forward(i)=Ez_grid(i,Ny);
    old_Ez_forward(i)=Ez_grid(i,Ny+1);
end

% The following comments are removed sections that allow extraction
% of each time step frame and graphing in order to construct a video
% file showing the signal propagation

%    k=5*counter; % Number (in this case 5) denotes which frames to
%    plot and add to animation

%    if n==k

        fprintf('Time step is %1.0d of
                        %1.0d\n',n,round(simulation_time/time_step));
        %Graph Plotter for Animation
%        for i=1:Nx+1
%            for j=1:Ny+1

%                Ez_plot(i,j)=Ez_grid(i,j);

%            end
%        end
%        surf(Ez_plot)
        %caxis([-1 1])
        %caxis([-0.1 0.1])
%        caxis([-1 1])
%        view([0 90])
```

```
%         anim(counter) = getframe;
%         counter=counter+1;
%     end

    output1(n,1)=Ez_grid(rx_cell_1,ry_cell_1);
    output2(n,1)=Ez_grid(rx_cell_2,ry_cell_2);
    % for f=1:abs((tx_cell-rx_cell)/2)-5;
    % output2(n,f)=Ez_grid(tx_cell+f,ty_cell+f,tz_cell);
end
% distance_axis=[1:abs(tx_cell-rx_cell)]*cell_size;
% movie2avi(anim,'c:\barry\FDTD.avi','compression','none','FPS',10);
```

A.2.2 3D FDTD

The 3D FDTD code is the basis for the previous 2D code but there were some perceived problems with the output results as explained in Chapter 2 but it is possible that it is normal behaviour. The speed of simulation is much slower and memory requirements are much higher than for 2D.

```
% It outputs 2 received signals, the time axis and the transmitted
% signal. It's inputs are the transmitter and receiver coordinates
% and signal flag
Function [output1,output2,time_axis,signal_z] =
FDTD_3D_c(tx_x,rx1_x,rx2_x,tx_y,rx1_y,rx2_y,tx_z,rx1_z,rx2_z,sig_val)

% Constants, Epsilon and Sigma both vary within the grid
% Mu is constant

eps_0=8.8542e-12; mu_0=1.2566e-6;
tissue_eps=9; tissue_mu=1; tissue_sigma=0;%0.4;
tumour_eps=40; tumour_mu=1; tumour_sigma=4;
wave_speed=1/sqrt(eps_0*tissue_eps*mu_0*tissue_mu);

% Cell size ideally at least 1/20 of a wavelength but difficult due
% to 3D domain size and memory limits
cell_size=0.001;

% Courant stability condition calculation for the time stepping
time_step = 1.0/(wave_speed*sqrt(1/(cell_size^2)+1/(cell_size^2)
                 +1/(cell_size^2)))

% Set this for the length of time to simulate
simulation_time=5e-9;

% This next section generates the PN sequence, 31 bits.  It uses a
% shift register with taps XOR'd together to give the next input.

PN_bandwidth=5e9;
seq_length=(2^5)-1;
counter=1;
seed=[0 1 1 0 1];
for q=1:seq_length
      for p=1:5
```

```
            if p==1
                  temp(p)=xor(seed(5),seed(3));
                  if temp(p)==1
                        PN_sequence(counter)=1;
                  elseif temp(p)==0
                        PN_sequence(counter)=-1;
                  end
                  counter=counter+1;
            else
                  temp(p)=seed(p-1);
            end
      end
      seed=temp;
end

%Specifying the size of the working domain
Nx=round(0.16/cell_size);
Ny=round(0.16/cell_size);
Nz=round(0.16/cell_size);

% Cell locations for the transmitter and 2 receivers
tx_cell=round(tx_x/cell_size);
ty_cell=round(tx_y/cell_size);
tz_cell=round(tx_z/cell_size);
rx_cell_1=round(rx1_x/cell_size);
ry_cell_1=round(rx1_y/cell_size);
rz_cell_1=round(rx1_z/cell_size);
rx_cell_2=round(rx2_x/cell_size);
ry_cell_2=round(rx2_y/cell_size);
rz_cell_2=round(rx2_z/cell_size);

% Basic Tissue parameters
sigma=(ones(Nx,Ny,Nz))*tissue_sigma;
eps=(ones(Nx,Ny,Nz))*tissue_eps*eps_0;
eps_x=(ones(Nx,Ny-1,Nz-1)*tissue_eps*eps_0);
sigma_x=(ones(Nx,Ny-1,Nz-1)*tissue_sigma);
eps_y=(ones(Nx-1,Ny,Nz-1)*tissue_eps*eps_0);
sigma_y=(ones(Nx-1,Ny,Nz-1)*tissue_sigma);
eps_z=(ones(Nx-1,Ny-1,Nz)*tissue_eps*eps_0);
sigma_z=(ones(Nx-1,Ny-1,Nz)*tissue_sigma);

% Setting up the targets.  Much simpler than the 2D FDTD targets that
% were added later to that code only
target_x=round(0.07/cell_size);
target_y=round(0.07/cell_size);
target_z=round(0.07/cell_size);
target_width=round(0.005/cell_size);

for r=-target_width:target_width
    for s=-target_width:target_width
        for t=-target_width:target_width
            eps_x(target_x+r,target_y-1+s,target_z-1+t)
                                        =tumour_eps*eps_0;
            sigma_x(target_x+r,target_y-1+s,target_z-1+t)
                                        =tumour_sigma;
            eps_y(target_x-1+r,target_y+s,target_z-1+t)
                                        =tumour_eps*eps_0;
            sigma_y(target_x-1+r,target_y+s,target_z-1+t)
```

```
                                                =tumour_sigma;
            eps_z(target_x-1+r,target_y-1+s,target_z+t)
                                                =tumour_eps*eps_0;
            sigma_z(target_x-1+r,target_y-1+s,target_z+t)
                                                =tumour_sigma;
        end
    end
end

% Setting up the grid variables
counter=1;

Ex_grid=zeros(Nx,Ny+1,Nz+1);
Ex_grid_core=zeros(Nx,Ny-1,Nz-1);
Ey_grid=zeros(Nx+1,Ny,Nz+1);
Ey_grid_core=zeros(Nx-1,Ny,Nz-1);
Ez_grid=zeros(Nx+1,Ny+1,Nz);
Ez_grid_core=zeros(Nx-1,Ny-1,Nz);
Hx_grid=zeros(Nx+1,Ny,Nz);
Hx_grid_core=zeros(Nx-1,Ny,Nz);
Hy_grid=zeros(Nx,Ny+1,Nz);
Hy_grid_core=zeros(Nx,Ny-1,Nz);
Hz_grid=zeros(Nx,Ny,Nz+1);
Hz_grid_core=zeros(Nx,Ny,Nz-1);

old_adj_Ex_bottom=zeros(Nx,Ny+1,1);
old_adj_Ey_bottom=zeros(Nx+1,Ny,1);
old_adj_Ez_bottom=zeros(Nx+1,Ny+1,1);
old_Ex_bottom=zeros(Nx,Ny+1,1);
old_Ey_bottom=zeros(Nx+1,Ny,1);
old_Ez_bottom=zeros(Nx+1,Ny+1,1);
old_adj_Ex_top=zeros(Nx,Ny+1,1);
old_adj_Ey_top=zeros(Nx+1,Ny,1);
old_adj_Ez_top=zeros(Nx+1,Ny+1,1);
old_Ex_top=zeros(Nx,Ny+1,1);
old_Ey_top=zeros(Nx+1,Ny,1);
old_Ez_top=zeros(Nx+1,Ny+1,1);

old_adj_Ex_left=zeros(1,Ny+1,Nz+1);
old_adj_Ey_left=zeros(1,Ny,Nz+1);
old_adj_Ez_left=zeros(1,Ny+1,Nz);
old_Ex_left=zeros(1,Ny+1,Nz+1);
old_Ey_left=zeros(1,Ny,Nz+1);
old_Ez_left=zeros(1,Ny+1,Nz);
old_adj_Ex_right=zeros(1,Ny+1,Nz+1);
old_adj_Ey_right=zeros(1,Ny,Nz+1);
old_adj_Ez_right=zeros(1,Ny+1,Nz);
old_Ex_right=zeros(1,Ny+1,Nz+1);
old_Ey_right=zeros(1,Ny,Nz+1);
old_Ez_right=zeros(1,Ny+1,Nz);

old_adj_Ex_back=zeros(Nx,1,Nz+1);
old_adj_Ey_back=zeros(Nx+1,1,Nz+1);
old_adj_Ez_back=zeros(Nx+1,1,Nz);
old_Ex_back=zeros(Nx,1,Nz+1);
old_Ey_back=zeros(Nx+1,1,Nz+1);
old_Ez_back=zeros(Nx+1,1,Nz);
old_adj_Ex_forward=zeros(Nx,1,Nz+1);
```

```
old_adj_Ey_forward=zeros(Nx+1,1,Nz+1);
old_adj_Ez_forward=zeros(Nx+1,1,Nz);
old_Ex_forward=zeros(Nx,1,Nz+1);
old_Ey_forward=zeros(Nx+1,1,Nz+1);
old_Ez_forward=zeros(Nx+1,1,Nz);

% Grid Mapping
% H Grids start at n=1 corresponding to (n-0.5)*time_step
% E Grids start at n=1 corresponding to (n-1)*time_step

% Signal generation, either a Differentiated Gaussian Pulse
% (sig_val==1) or PN  % sequence (sig_val==2)
pulse_unit=round(0.5e-9/(6*time_step));
for n=1:(round((simulation_time)/(time_step)))
    time_axis(n)=(n-1)*time_step;
    if sig_val==1
        signal_z(n)=(((n-1)*time_step)-(4*75e-12))
            *exp(-((((n-1)*time_step)-(4*75e-12))^2)/(75e-12^2));
    elseif sig_val==2
        signal_z(n)=sin(2*pi*5e9.*time_axis(n))
            *PN_sequence(floor(time_axis(n)/(1/PN_bandwidth))+1);
    end
end

% The simulation itself.  It uses a matrix multiplication method to
% speed up the Matlab process over loops, although it does require
% additional variables which in the 3D case represents significant
% additional memory requirements
for n=1:(round((simulation_time)/(time_step)))
    time_axis(n)=(n-1)*time_step;
    Ez_grid(tx_cell,ty_cell,tz_cell)=signal_z(n);

    mu_x=tissue_mu*mu_0;
    alpha_x=time_step/(cell_size*mu_x);
    old_Hx = Hx_grid;
    Hx_grid=old_Hx+alpha_x*(diff(Ey_grid,1,3)-diff(Ez_grid,1,2));

    mu_y=tissue_mu*mu_0;
    alpha_y=time_step/(cell_size*mu_y);
    old_Hy = Hy_grid;
    Hy_grid=old_Hy+alpha_y*(diff(Ez_grid,1,1)-diff(Ex_grid,1,3));

    mu_z=tissue_mu*mu_0;
    alpha_z=time_step/(cell_size*mu_z);
    old_Hz = Hz_grid;
    Hz_grid=old_Hz+alpha_z*(diff(Ex_grid,1,2)-diff(Ey_grid,1,1));

    % 1st Order Mur boundary Conditions for H
    % Due to the nature of the grid there is never any need to
    % calculate the boundary conditions for the H field because the E
    % field vectors that are required for correct calculation at the
    % edges of the H grid are present.

    Hx_grid_core(1:Nx-1,:,:)=Hx_grid(2:Nx,:,:);
    Hy_grid_core(:,1:Ny-1,:)=Hy_grid(:,2:Ny,:);
    Hz_grid_core(:,:,1:Nz-1)=Hz_grid(:,:,2:Nz);

    beta_x=(1-((sigma_x*time_step)./(2*eps_x)))
```

```
        ./(1+((sigma_x*time_step)./(2*eps_x)));
delta_x=((time_step./eps_x))
        ./(1+((sigma_x*time_step)./(2*eps_x)))/cell_size;

Ex_grid_core=(beta_x.*Ex_grid_core)
  +(delta_x.*(diff(Hz_grid_core,1,2)-diff(Hy_grid_core,1,3)));

Ex_grid(:,2:Ny,2:Nz)=Ex_grid_core(:,1:Ny-1,1:Nz-1);

beta_y=(1-((sigma_y*time_step)./(2*eps_y)))
        ./(1+((sigma_y*time_step)./(2*eps_y)));
delta_y=((time_step/eps_y))
        ./(1+((sigma_y*time_step)./(2*eps_y))./cell_size;

Ey_grid_core=(beta_y.*Ey_grid_core)
  +(delta_y.*(diff(Hx_grid_core,1,3)-diff(Hz_grid_core,1,1)));

Ey_grid(2:Nx,:,2:Nz)=Ey_grid_core(1:Nx-1,:,1:Nz-1);

beta_z=(1-((sigma_z*time_step)./(2*eps_z)))
        ./(1+((sigma_z*time_step)./(2*eps_z)));
delta_z=((time_step/eps_z))
        ./(1+((sigma_z*time_step)./(2*eps_z))./cell_size;

Ez_grid_core=(beta_z.*Ez_grid_core)
  +(delta_z.*(diff(Hy_grid_core,1,1)-diff(Hx_grid_core,1,2)));

Ez_grid(2:Nx,2:Ny,:)=Ez_grid_core(1:Nx-1,1:Ny-1,:);

mur_coeff=((wave_speed*time_step)-cell_size)
             /((wave_speed*time_step)+cell_size);


% Grid Limit Terminolgy for Mur ABC
%
%          /\                  _______
%         /  \               /      /|
%     Top  ||               /_____/ |          bottom  ||
%                           |     | /                 \  /
%                           |_____|/                   \/
%
%                                                     z |
%                            _______                    |
%                           /      /|                   |_______ x
%     Left  <==            /_____/ |   Right ==>            /
%                          |     | /                   y /
%                          |_____|/
%
%                            _______
%                           /      /|                  /\
%     Foward //            /_____/ |           back //
%          //              |     | /                 //
%         \/               |_____|/
%

% 1st order Mur boundary conditions for E

Ex_grid(1:Nx,1:Ny+1,1)=old_adj_Ex_bottom(1:Nx,1:Ny+1,1)
        +mur_coeff*(Ex_grid(1:Nx,1:Ny+1,2)
```

```
        -old_Ex_bottom(1:Nx,1:Ny+1,1));
Ex_grid(1:Nx,1:Ny+1,Nz+1)=old_adj_Ex_top(1:Nx,1:Ny+1,1)
        +mur_coeff*(Ex_grid(1:Nx,1:Ny+1,Nz)
        -old_Ex_top(1:Nx,1:Ny+1,1));
old_adj_Ex_bottom(1:Nx,1:Ny+1,1)=Ex_grid(1:Nx,1:Ny+1,2);
old_Ex_bottom(1:Nx,1:Ny+1,1)=Ex_grid(1:Nx,1:Ny+1,1);
old_adj_Ex_top(1:Nx,1:Ny+1,1)=Ex_grid(1:Nx,1:Ny+1,Nz);
old_Ex_top(1:Nx,1:Ny+1,1)=Ex_grid(1:Nx,1:Ny+1,Nz+1);

Ey_grid(1:Nx+1,1:Ny,1)=old_adj_Ey_bottom(1:Nx+1,1:Ny,1)
        +mur_coeff*(Ey_grid(1:Nx+1,1:Ny,2)
        -old_Ey_bottom(1:Nx+1,1:Ny,1));
Ey_grid(1:Nx+1,1:Ny,Nz+1)=old_adj_Ey_top(1:Nx+1,1:Ny,1)
        +mur_coeff*(Ey_grid(1:Nx+1,1:Ny,Nz)
        -old_Ey_top(1:Nx+1,1:Ny,1));
old_adj_Ey_bottom(1:Nx+1,1:Ny,1)=Ey_grid(1:Nx+1,1:Ny,2);
old_Ey_bottom(1:Nx+1,1:Ny,1)=Ey_grid(1:Nx+1,1:Ny,1);
old_adj_Ey_top(1:Nx+1,1:Ny,1)=Ey_grid(1:Nx+1,1:Ny,Nz);
old_Ey_top(1:Nx+1,1:Ny,1)=Ey_grid(1:Nx+1,1:Ny,Nz+1);

Ez_grid(1:Nx+1,1:Ny+1,1)=old_adj_Ez_bottom(1:Nx+1,1:Ny+1,1)
        +mur_coeff*(Ez_grid(1:Nx+1,1:Ny+1,2)
        -old_Ez_bottom(1:Nx+1,1:Ny+1,1));
Ez_grid(1:Nx+1,1:Ny+1,Nz)=old_adj_Ez_top(1:Nx+1,1:Ny+1,1)
        +mur_coeff*(Ez_grid(1:Nx+1,1:Ny+1,Nz-1)
        -old_Ez_top(1:Nx+1,1:Ny+1,1));
old_adj_Ez_bottom(1:Nx+1,1:Ny+1,1)=Ez_grid(1:Nx+1,1:Ny+1,2);
old_Ez_bottom(1:Nx+1,1:Ny+1,1)=Ez_grid(1:Nx+1,1:Ny+1,1);
old_adj_Ez_top(1:Nx+1,1:Ny+1,1)=Ez_grid(1:Nx+1,1:Ny+1,Nz-1);
old_Ez_top(1:Nx+1,1:Ny+1,1)=Ez_grid(1:Nx+1,1:Ny+1,Nz);

Ex_grid(1,1:Ny+1,1:Nz+1)=old_adj_Ex_left(1,1:Ny+1,1:Nz+1)
        +mur_coeff*(Ex_grid(2,1:Ny+1,1:Nz+1)
        -old_Ex_left(1,1:Ny+1,1:Nz+1));
Ex_grid(Nx,1:Ny+1,1:Nz+1)=old_adj_Ex_right(1,1:Ny+1,1:Nz+1)
        +mur_coeff*(Ex_grid(Nx-1,1:Ny+1,1:Nz+1)
        -old_Ex_right(1,1:Ny+1,1:Nz+1));
old_adj_Ex_left(1,1:Ny+1,1:Nz+1)=Ex_grid(2,1:Ny+1,1:Nz+1);
old_Ex_left(1,1:Ny+1,1:Nz+1)=Ex_grid(1,1:Ny+1,1:Nz+1);
old_adj_Ex_right(1,1:Ny+1,1:Nz+1)=Ex_grid(Nx-1,1:Ny+1,1:Nz+1);
old_Ex_right(1,1:Ny+1,1:Nz+1)=Ex_grid(Nx,1:Ny+1,1:Nz+1);

Ey_grid(1,1:Ny,1:Nz+1)=old_adj_Ey_left(1,1:Ny,1:Nz+1)
        +mur_coeff*(Ey_grid(2,1:Ny,1:Nz+1)
        -old_Ey_left(1,1:Ny,1:Nz+1));
Ey_grid(Nx+1,1:Ny,1:Nz+1)=old_adj_Ey_right(1,1:Ny,1:Nz+1)
        +mur_coeff*(Ey_grid(Nx,1:Ny,1:Nz+1)
        -old_Ey_right(1,1:Ny,1:Nz+1));
old_adj_Ey_left(1,1:Ny,1:Nz+1)=Ey_grid(2,1:Ny,1:Nz+1);
old_Ey_left(1,1:Ny,1:Nz+1)=Ey_grid(1,1:Ny,1:Nz+1);
old_adj_Ey_right(1,1:Ny,1:Nz+1)=Ey_grid(Nx,1:Ny,1:Nz+1);
old_Ey_right(1,1:Ny,1:Nz+1)=Ey_grid(Nx+1,1:Ny,1:Nz+1);

Ez_grid(1,1:Ny+1,1:Nz)=old_adj_Ez_left(1,1:Ny+1,1:Nz)
        +mur_coeff*(Ez_grid(2,1:Ny+1,1:Nz)
        -old_Ez_left(1,1:Ny+1,1:Nz));
Ez_grid(Nx+1,1:Ny+1,1:Nz)=old_adj_Ez_right(1,1:Ny+1,1:Nz)
        +mur_coeff*(Ez_grid(Nx,1:Ny+1,1:Nz)
        -old_Ez_right(1,1:Ny+1,1:Nz));
```

```
    old_adj_Ez_left(1,1:Ny+1,1:Nz)=Ez_grid(2,1:Ny+1,1:Nz);
    old_Ez_left(1,1:Ny+1,1:Nz)=Ez_grid(1,1:Ny+1,1:Nz);
    old_adj_Ez_right(1,1:Ny+1,1:Nz)=Ez_grid(Nx,1:Ny+1,1:Nz);
    old_Ez_right(1,1:Ny+1,1:Nz)=Ez_grid(Nx+1,1:Ny+1,1:Nz);

    Ex_grid(1:Nx,1,1:Nz+1)=old_adj_Ex_back(1:Nx,1,1:Nz+1)
            +mur_coeff*(Ex_grid(1:Nx,2,1:Nz+1)
            -old_Ex_back(1:Nx,1,1:Nz+1));
    Ex_grid(1:Nx,Ny+1,1:Nz+1)=old_adj_Ex_forward(1:Nx,1,1:Nz+1)
            +mur_coeff*(Ex_grid(1:Nx,Ny,1:Nz+1)
            -old_Ex_forward(1:Nx,1,1:Nz+1));
    old_adj_Ex_back(1:Nx,1,1:Nz+1)=Ex_grid(1:Nx,2,1:Nz+1);
    old_Ex_back(1:Nx,1,1:Nz+1)=Ex_grid(1:Nx,1,1:Nz+1);
    old_adj_Ex_forward(1:Nx,1,1:Nz+1)=Ex_grid(1:Nx,Ny,1:Nz+1);
    old_Ex_forward(1:Nx,1,1:Nz+1)=Ex_grid(1:Nx,Ny+1,1:Nz+1);

    Ey_grid(1:Nx+1,1,1:Nz+1)=old_adj_Ey_back(1:Nx+1,1,1:Nz+1)
            +mur_coeff*(Ey_grid(1:Nx+1,2,1:Nz+1)
            -old_Ey_back(1:Nx+1,1,1:Nz+1));
    Ey_grid(1:Nx+1,Ny,1:Nz+1)=old_adj_Ey_forward(1:Nx+1,1,1:Nz+1)
            +mur_coeff* (Ey_grid(1:Nx+1,Ny-1,1:Nz+1)
            -old_Ey_forward(1:Nx+1,1,1:Nz+1));
    old_adj_Ey_back(1:Nx+1,1,1:Nz+1)=Ey_grid(1:Nx+1,2,1:Nz+1);
    old_Ey_back(1:Nx+1,1,1:Nz+1)=Ey_grid(1:Nx+1,1,1:Nz+1);
    old_adj_Ey_forward(1:Nx+1,1,1:Nz+1)=Ey_grid(1:Nx+1,Ny-1,1:Nz+1);
    old_Ey_forward(1:Nx+1,1,1:Nz+1)=Ey_grid(1:Nx+1,Ny,1:Nz+1);

    Ez_grid(1:Nx+1,1,1:Nz)=old_adj_Ez_back(1:Nx+1,1,1:Nz)
            +mur_coeff*(Ez_grid(1:Nx+1,2,1:Nz)
            -old_Ez_back(1:Nx+1,1,1:Nz));
    Ez_grid(1:Nx+1,Ny+1,1:Nz)=old_adj_Ez_forward(1:Nx+1,1,1:Nz)
            +mur_coeff*(Ez_grid(1:Nx+1,Ny,1:Nz)
            -old_Ez_forward(1:Nx+1,1,1:Nz));
    old_adj_Ez_back(1:Nx+1,1,1:Nz)=Ez_grid(1:Nx+1,2,1:Nz);
    old_Ez_back(1:Nx+1,1,1:Nz)=Ez_grid(1:Nx+1,1,1:Nz);
    old_adj_Ez_forward(1:Nx+1,1,1:Nz)=Ez_grid(1:Nx+1,Ny,1:Nz);
    old_Ez_forward(1:Nx+1,1,1:Nz)=Ez_grid(1:Nx+1,Ny+1,1:Nz);

% The following comments are removed sections that allow extraction
% of each time step frame and graphing in order to construct a video
% file showing the signal propagation

    % k=5*counter; % Number (in this case 5) denotes which frames to
    % plot and add to animation

    % if n==k

        fprintf('Time step is %1.0d of
                 %1.0d\n',n,round(simulation_time/time_step));
    %    Graph Plotter for Animation
    %    for i=1:Nx+1
    %        for j=1:Ny+1

    %            Ez_plot(i,j)=Ez_grid(i,j,tz_cell);

    %        end
    %    end
    %    surf(Ez_plot)
    %    caxis([-500e-1 500e-1])
```

```
    %    caxis([-0.1 0.1])
    %    caxis([-0.5e-3 0.5e-3])
    %    view([0 90])
    %    anim(counter) = getframe;
    %    counter=counter+1;
    %end

    output1(n,1)=Ez_grid(rx_cell_1,ry_cell_1,rz_cell_1);
    output2(n,1)=Ez_grid(rx_cell_2,ry_cell_2,rz_cell_2);
    %for f=1:abs((tx_cell-rx_cell)/2)-5;
    %    output2(n,f)=Ez_grid(tx_cell+f,ty_cell+f,tz_cell);
    %end
end

%distance_axis=[1:abs(tx_cell-rx_cell)]*cell_size;
%movie2avi(anim,'c:\barry\FDTD.avi','compression','none','FPS',10);
```

A.2.3 PN Generator

The PN generator code creates a maximal length PN sequence for a given bit length (4-10 and 16) with an error return message for out of bounds values for the bit length chosen. It uses a shift register and XOR'd taps to generate the next input leading to the sequences generation.

```
function [PN_sequence]=PN_generator(L)

seq_length=(2^L)-1;

switch L
    case 4
        counter=1;
        seed=[0 1 1 0];
        for q=1:seq_length
            for p=1:L
                if p==1
                    temp(p)=xor(seed(4),seed(3));
                    if temp(p)==1
                        PN_sequence(counter)=1;
                    elseif temp(p)==0
                        PN_sequence(counter)=-1;
                    end
                    counter=counter+1;
                else
                    temp(p)=seed(p-1);
                end
            end
            seed=temp;
        end

    case 5
```

```
    counter=1;
    seed=[0 1 1 0 1];
    for q=1:seq_length
        for p=1:L
            if p==1
                temp(p)=xor(seed(5),seed(3));
                if temp(p)==1
                    PN_sequence(counter)=1;
                elseif temp(p)==0
                    PN_sequence(counter)=-1;
                end
                counter=counter+1;
            else
                temp(p)=seed(p-1);
            end
        end
        seed=temp;
    end

case 6
    counter=1;
    seed=[0 1 1 1 1 0];
    for q=1:seq_length
        for p=1:L
            if p==1
                temp(p)=xor(seed(6),seed(5));
                if temp(p)==1
                    PN_sequence(counter)=1;
                elseif temp(p)==0
                    PN_sequence(counter)=-1;
                end
                counter=counter+1;
            else
                temp(p)=seed(p-1);
            end
        end
        seed=temp;
    end

case 7
    counter=1;
    seed=[0 1 1 1 1 0 1];
    for q=1:seq_length
        for p=1:L
            if p==1
                temp(p)=xor(seed(7),seed(6));
                if temp(p)==1
                    PN_sequence(counter)=1;
                elseif temp(p)==0
                    PN_sequence(counter)=-1;
                end
                counter=counter+1;
            else
                temp(p)=seed(p-1);
            end
        end
        seed=temp;
    end
```

```
case 8
    counter=1;
    seed=[0 1 1 1 0 0 1 0];
    for q=1:seq_length
        for p=1:L
            if p==1
                temp(p)=xor(xor(xor(seed(8),seed(7)),
                                                seed(6)),seed(1));
                if temp(p)==1
                    PN_sequence(counter)=1;
                elseif temp(p)==0
                    PN_sequence(counter)=-1;
                end
                counter=counter+1;
            else
                temp(p)=seed(p-1);
            end
        end
        seed=temp;
    end

case 9
    counter=1;
    seed=[0 1 1 1 1 0 0 1 1];
    for q=1:seq_length
        for p=1:L
            if p==1
                temp(p)=xor(seed(9),seed(5));
                if temp(p)==1
                    PN_sequence(counter)=1;
                elseif temp(p)==0
                    PN_sequence(counter)=-1;
                end
                counter=counter+1;
            else
                temp(p)=seed(p-1);
            end
        end
        seed=temp;
    end

case 10
    counter=1;
    seed=[0 1 1 1 1 0 0 1 1 0];
    for q=1:seq_length
        for p=1:L
            if p==1
                temp(p)=xor(seed(10),seed(7));
                if temp(p)==1
                    PN_sequence(counter)=1;
                elseif temp(p)==0
                    PN_sequence(counter)=-1;
                end
                counter=counter+1;
            else
                temp(p)=seed(p-1);
            end
        end
        seed=temp;
```

```
        end

    case 16
        counter=1;
        seed=[0 1 1 0 1 0 0 0 1 1 1 1 0 0 1 1];
        for q=1:seq_length
            for p=1:L
                if p==1
                    temp(p)=xor(xor(xor(seed(16),seed(14)),
                                                    seed(13)),seed(11));
                    if temp(p)==1
                        PN_sequence(counter)=1;
                    elseif temp(p)==0
                        PN_sequence(counter)=-1;
                    end
                    counter=counter+1;
                else
                    temp(p)=seed(p-1);
                end
            end
            seed=temp;
        end
    otherwise
        fprintf('The shift register length entered to calculate the
            PN sequence is not recognised (Choices:4-10,16)');
        return
end
```

A.2.4 Filter Generation

Filter generation uses the LMS algorithm to match an input signal (variable in) to a desired signal (variable desired). It outputs the filter coefficients as well as the resultant filter output for the given input and the error signal (difference between filter output and desired output).

```
function [coeff,y_out,error]=filter_gen_2b
            (desired,in,spacing,num_coeffs,offset,num_iterations)

L_in=length(in);
coeff=zeros(num_coeffs,1);
new_coeff=zeros(num_coeffs,1);

% Step size is an important parameter that has an impact on stability
% and final result
step_size=0.001;

% Variable required for alternate absolute sum error tracking by
% iteration abs_error_sum=1000000;

for iterations=1:num_iterations
```

```
    for k=1:L_in
        for p=1:num_coeffs

% temp is a holder to allow for the correct location of the filter
% coefficients based on the coefficient spacing and an offset (both
% input variables)
            temp=spacing*(p-1)+offset;
            if k>temp & k-temp<=L_in
                x_val(p,1)=in(k-temp);
            else
                x_val(p,1)=0;
            end
        end

% The LMS algorithm
        y_out(k,1)=coeff'*x_val;
        error(k,1)=desired(k,1)-y_out(k,1);
        new_coeff=coeff+(step_size*error(k,1)'*x_val);
        coeff=new_coeff;

    end

% Method that allows reduction in the step size for a finer result
% when the absolute error sum stabilizes.  Number of iterations
% required is usually very large before it activates and is slow
    % new_abs_error_sum=sum(abs(error));
    % if new_abs_error_sum<abs_error_sum
    %    abs_error_sum=new_abs_error_sum;
    % elseif new_abs_error_sum>abs_error_sum
    %    step_size=step_size/2;
    % end

% These are for recording the absolute error sum and coefficients
% over a number of iterations, they need to be added to the output
% variables if needed
    % abs_error_sum=sum(abs(error));
    % recorder(iterations)=abs_error_sum;
    % coeff_it(:,iterations)=coeff;
end
```

A.2.5 Localisation

The localisation code has had many iterations, this is the last and was used for the shape detection results with 2D FDTD simulation results. It covers both filter (and correlation) along with integration over the pulse length.

```
% time is a vector of time values for the data values.  The values
% variable contains the signal data and contains any number of
% signals [n signals, signal length].  The antenna variable contains
% the antenna locations for each of the signals [n signals, 6]  in
% metres and 3D.
function [full_array]=location_3D_shape(time,values,antenna)

c=3e8;
eps_0=8.8542e-12;
```

```
mu_0=1.2566e-6;
eps_r=9;
mu_r=1;
f=6e9;
w=2*pi*f;
sigma=0.4;
lambda=c/f;

y=((mu_0*mu_r*eps_0*eps_r)/2);
z=(sigma/(w*eps_0*eps_r));

alpha=w*sqrt(y*(sqrt(1+z^2)-1));
beta=w*sqrt(y*(sqrt(1+z^2)+1));

gamma=alpha+(j*beta);

time_step=time(2)-time(1);
distance_step=1e8*time_step;

% offset is a variable to aid alignment of signals, filter uses 0
offset=round(0.25e-9/time_step);%0.55
%offset=0;%round(0.2788e-9/time_step); %for migrated

box_x=round(0.15/distance_step);
box_x_axis=[1:box_x]*distance_step;
box_y=round(0.15/distance_step);
box_y_axis=[1:box_y]*distance_step;
box_z=round(0.1/distance_step);
box_z_axis=[1:box_z]*distance_step;

full_array=zeros(box_x,box_y,1);
[dmax,null]=size(antenna);
max_transit=length(values(1,:));
k_length=round(0.5e-9/time_step);

% Uncomment for 3D generation
% for a=1:box_z

a=round(0.0575/distance_step);
    for b=1:box_y
        for c=1:box_x

% Used for filter
            %temp_holder=0;

% Used for integration over a pulse
            temp_holder=zeros(1,k_length+1);

            for d=1:dmax
                tx=round(antenna(d,1)/distance_step);
                ty=round(antenna(d,2)/distance_step);
                tz=round(antenna(d,3)/distance_step);
                rx=round(antenna(d,4)/distance_step);
                ry=round(antenna(d,5)/distance_step);
                rz=round(antenna(d,6)/distance_step);

                transit_1=sqrt(((tz-a)^2)+((ty-b)^2)+((tx-c)^2));
                transit_2=sqrt(((rz-a)^2)+((ry-b)^2)+((rx-c)^2));
                total_transit=round(transit_1+transit_2+offset);
```

```
                transit_tx_real=round(transit_1)*distance_step;
                transit_rx_real=round(transit_2)*distance_step;

% First block is for integration over a pulse
% This is an alternate spreading loss compensation factor, in
% temp_holder the divide by sp_loss_factor can be replaced by divide
% by E
%               E=abs((exp(-gamma*transit_rx_real))/transit_rx_real)
                  +abs((exp(-gamma*transit_tx_real))
                        /transit_tx_real);
                sp_loss_factor=(1/transit_tx_real)
                                        *(1/transit_rx_real);
                for k=1:k_length+1
                    if total_transit+k<=max_transit
                        temp_holder(1,k)=temp_holder(1,k)
                              +((1/sp_loss_factor)
                              *values(d,total_transit+k));
                    end
                end

% Second block is for filter
%               E=abs((exp(-gamma*transit_rx_real))/transit_rx_real)
                  +abs((exp(-gamma*transit_tx_real))
                        /transit_tx_real);
%               sp_loss_factor=1;%(1/transit_tx_real)
                                        *(1/transit_rx_real);
%               if total_transit<=max_transit
%                     temp_holder=temp_holder+((1/sp_loss_factor)
                        *values(d,total_transit)/dmax);
%               end
            end
% For Filter
%           full_array(c,b,1)=temp_holder;
% For pulse integration
            full_array(c,b,1)=trapz(abs(temp_holder));
        end
    end
% end to match the for a=... statement above for 3D generation
  %end
```

A.2.6 H-Matrix Transform

The H-Matrix transforms are both given, selection is by selecting which alphas to use.

The original is marked (1), the modified with (2).

```
function [Hv,angle_r_phi,angle_t_phi,alpha_r,alpha_t] =
                        H_matrix_gen_D(c,f,Tx,Rx,H_matrix,dist)

lambda=c/f;
alpha_Tx=(Tx(1,2)-Tx(1,1))/lambda;
alpha_Rx=(Rx(1,2)-Rx(1,1))/lambda;

Q=length(Rx);
P=length(Tx);
```

```
P_bar=(P-1)/2;
Q_bar=(Q-1)/2;

angle_t=[-P_bar:P_bar]/P;
angle_r=[-Q_bar:Q_bar]/Q;
angle_r_phi=asin(angle_r/alpha_Rx);
angle_t_phi=asin(angle_t/alpha_Tx);

alpha_t=zeros(P,P);
alpha_r=zeros(Q,Q);

for p=1:P
    for q=1:P
% Two variants, the original (1) and the modified (2).  Uncomment as
% required
% (1)   alpha_t(q,p)=(1/sqrt(P))*exp(-j*2*pi*(q-1)*((p-1-P_bar)/P));
% (2) alpha_t(p,q)=(1/sqrt(P))*exp(-j*2*pi*(dist-sqrt(((Tx(1,p)).^2)
%   +((dist)^2)-2.*(Tx(1,p)).*(dist).*sin(angle_t_phi(q))))/lambda);

    end
end
for p=1:Q
    for q=1:Q
% Two variants, the original (1) and the modified (2).  Uncomment
% matching
% (1)  alpha_r(q,p)=(1/sqrt(Q))*exp(-j*2*pi*(q-1)*((p-1-Q_bar)/Q));
% (2) alpha_r(p,q)=(1/sqrt(Q))*exp(-j*2*pi*(dist-sqrt(((Rx(1,p)).^2)
%   +((dist)^2)-2.*(Rx(1,p)).*(dist).*sin(angle_r_phi(q))))/lambda);
    end
end

% Virtual H-matrix transform
Hv=alpha_r.'*H_matrix*alpha_t;
```

References

[1] Compact Oxford English Dictionary, http://www.askoxford.com

[2] H. Hashemi, "Impulse response modeling of indoor radio propagation channels", *IEEE Journal on Selected Areas in Communications*, **Vol. 11**, No. 7, pp.967-978, September 1993

[3] H. Hashemi, "The indoor radio propagation channel", *Proceedings of the IEEE*, **Vol. 81, No.7**, pp.943-968, July 1993

[4] P. S. H. Leather and D. Parsons, "Equalization for antenna-pattern measurements: Established Technique – New Application", *IEEE Antennas and Propagation Magazine*, **Vol. 45, No.2**, pp.154-161, April 2003

[5] P. S. H. Leather, J.D. Parsons, J. Romeu, S. Blanch and A. Aguasca, "Correlation techniques applied to antenna pattern measurement", *Electronics letters*, **Vol. 40, No. 10**, pp.154-161, 13th May 2004

[6] A. M. Sayeed, "Deconstructing multiantenna fading channels", *IEEE Trans. Signal Processing*, **Vol. 50, No. 10**, pp.2563-2579, Oct 2002

[7] E. C. Fear, X. Li, S. C. Hagness and M. A. Stuchly, "Confocal microwave imaging for breast cancer detection: Localization of tumours in three dimensions", *IEEE Transactions on Biomedical Engineering*, **Vol. 49, No. 8**, pp.812-822, August 2002

[8] R. Nilavalan, A. Gbedemah, I. J. Craddock, X. Li, S. C. Hagness, "Numerical investigation of breast tumour detection using multi-static radar", *Electronics Letters*, **Vol**. 39, No. 25, December 2003

[9] E. C. Fear and M. A. Stuchly, "Microwave detection of breast tumors: Comparison of skin subtraction algorithms," *Proc. SPIE*, **vol. 4129**, pp. 207–217, 2000.

[10] Cancer Research UK, http://www.cancerresearchuk.org

[11] M. Brown, F. Houn, E. Sickles, and L. Kessler, "Screening mammography in community practice," *Amer. J. Roentgen.*, **vol. 165**, pp.1373–1377, Dec. 1995.

[12] V. P. Jackson, R. E. Hendrick, S. A. Feig, and D. B. Kopans, "Imaging of the radiographically dense breast," *Radiology*, vol. 188, pp. 297–301, Aug. 1993.

[13] S. Kashikar-Zuck, F. J. Keefe, P. Kornguth, P. Beaupre, A. Holzberg, D. Delong, "Pain coping and the pain experience during mammography: A preliminary study" Pain 73, pp. 165–172, 1997

[14] M.E. Keemers-Gels, R.P.R. Groenendijk, J.H.M. van den Heuvel, C. Boetes, P.G.M. Peer and Th. Wobbes, "Pain experienced by women attending breast cancer screening", *Breast Cancer Research and Treatment* 60: pp. 235–240, 2000

[15] E.R. Adair, R.C. Petersen, "Biological effects of radiofrequency/microwave radiation", *IEEE Trans. Microwave Theory and Techniques*, **Vol. 50** Issue:3, pp. 953-962, Mar 2002

[16] U. Fischer, K. P. Hermann, F. Baum, "Digital mammography: current state and future aspects", *Eur. Radiol. 16*, pp. 38–44, 2006

[17] J. C. Birnholz, "An approach to specific tumor diagnosis by ultrasound imaging", 1972 Ultrasonics Symposium, pp. 31 - 32

[18] L Esserman, D Wolverton and N Hylton, "Magnetic resonance imaging for primary breast cancer management: current role and new applications", *Endocrine-Related Cancer 9*, pp. 141–153, 2002

[19] D. Saslow, C. Boetes, W. Burke, et al. "American Cancer Society guidelines for breast screening with MRI as an adjunct to mammography". *CA Cancer J Clin*, 57(2), pp. 75-89, Mar-Apr 2007

[20] L. E. Larsen and J. H. Jacobi, Eds., "Medical Applications of Microwave Imaging", *Piscataway, NJ*: IEEE Press, 1986.

[21] M. Lazebnik, D. Popovic, L. McCartney, C. B. Watkins, M. J. Lindstrom, J. Harter, S. Sewall, T. Ogilvie, A. Magliocco, T. M. Breslin, W. Temple, D. Mew, J. H. Booske, M. Okoniewski, S. C. Hagness, "A large-scale study of the ultrawideband microwave dielectric properties of normal, benign and malignant breast tissues obtained from cancer surgeries", Physics in Medicine and Biology, Vol. 52, pp. 6093-6115, 2007

[22] A. M. Cambell and D. V. Land, "Dielectric properties of female human breast tissue measured in vitro at 3.2 GHz", *Physics in Medicine and Biology*, **Vol. 37. No. 1**, pp. 193-210, January 1992

[23] M. Lazebnik, L. McCartney, D. Popovic, C. B. Watkins, M. J. Lindstrom, J. Harter, S. Sewall, T. Ogilvie, A. Magliocco, J. H. Booske, M. Okoniewski, S. C. Hagness, "A large-scale study of the ultrawideband microwave dielectric properties of normal breast tissue obtained from reduction surgeries", *Physics in Medicine and Biology*, **Vol. 52**, pp. 2637-2636, 2007

[24] M. Klemm, I.J. Craddock, J. Leendertz, A.W. Preece and R. Benjamin, "Breast cancer detection using symmetrical antenna array", *Proceedings of The European Conference on Antennas and Propagation: EuCAP 2007,* Edinburgh, UK, November 2007

[25] D. J. Daniels, "Ground Penetrating Radar" 2nd Edition, *IEE Radar, Sonar and Navigation Series 15*, pp.247-352

[26] X. Li and S. C. Hagness, "A confocal microwave imaging algorithm for breast cancer detection", *IEEE Microwave and Wireless Components Letters*, **Vol. 11,**

No. 3, pp.130-132, March 2001

[27] E. C. Fear, S. C. Hagness, P.M. Meaney, M. Okoniewski and M. A. Stuchly, "Enhancing breast tumour detection with near field imaging", *IEEE Microwave Potentials*, pp.48-56, March 2002

[28] S. K. Davis, E. J. Bond, X. Li, S. C. Hagness and B. D. Van Veen, "Microwave imaging via space-time beamforming for early detection of breast cancer", *IEEE Transactions on Antennas and Propagation*, **Vol. 51, No. 8**, pp.1690-1705, August 2003

[29] E. C. Fear, P.M. Meaney and M. A. Stuchly, "Microwaves for breast cancer detection", *IEEE Potentials*, pp.12-18, February/March 2003

[30] D. Li, P. M. Meaney and K. D. Paulsen, "Conformal microwave imaging for breast cancer detection", *IEEE Transactions on Microwave Theory and Techniques*, **Vol. 51, No. 4,** pp. 1179-1186, April 2003

[31] I. J. Craddock, R. Nilavalan, J. Leenderz, A. Preece and R. Benjamin, "Experimental investigation of real aperture synthetically organised radar for breast cancer detection", *IEEE AP-S International Symposium and USNC/URSI National Radio Science Meeting Proceedings*, 2005

[32] I. J. Craddock, R. Nilavalan, J. Leenderz, A. Preece and R. Benjamin, "Breast cancer tumour detection using microwave radar techniques", *URSI EMTS 2004*

[33] W. Shao, B, Zhou, G, Wang, "Early breast tumour imaging via UWB microwave method: Study on multi-target detection", *IEEE AP-S International Symposium and USNC/URSI National Radio Science Meeting Proceedings*, 2005

[34] W. Shao, B, Zhou, G, Wang, "On the resolution of UWB microwave imaging of tumours in random breast tissue", *IEEE AP-S International Symposium and*

USNC/URSI National Radio Science Meeting Proceedings, 2005

[35] P. M. van den Berg and R. E. Kleinman, "A contrast source inversion method", *Inverse Problems 13*, pp. 1607-1620, 1997

[36] A. Abubakar and P. M. van den Berg, "The contrast source inversion method for location and shape reconstructions", *Inverse Problems 18*, pp. 495-510, 2002

[37] L. Song, C. Yu and Q. H. Liu, "Image reconstruction from measured scattered data", *IEEE AP-S International Symposium and USNC/URSI National Radio Science Meeting Proceedings*, 2005

[38] L. Song, C. Yu and Q. H. Liu, "2-D nonlinear image reconstruction for objects buried in layered media", *IEEE AP-S International Symposium and USNC/URSI National Radio Science Meeting Proceedings*, 2005

[39] Q. Zhu and L. M. Collins, "Application of feature extraction methods for landmine detection using the Wichmann/Niitek ground-penetrating radar", *IEE Transactions on Geoscience and Remote Sensing*, **Vol. 43, No. 1**, pp. 81-85, January 2005

[40] S. K. Davis, E. J. Bond, X. Li, S. C. Hagness and B. D. Van Veen, "Microwave imaging via space-time beamforming for early detection of breast cancer: Beamformer design in the frequency domain", *Journal of Electromagnetics, Waves and Applications*, **Vol. 17, No. 2**, pp.357-381 , 2003

[41] X. Li, S. K. Davis, S. C. Hagness, D. W. van der Weide and B. D. Van Veen, "Microwave imaging via space-time beamforming: Experimental investigation of tumour detection in multilayer breast phantoms", *IEEE Transactions on Microwave Theory and Techniques*, **Vol. 52, No. 8**, pp.1856-1865, August 2004

[42] A. Taflove, "Computational Electrodynamics: The Finite-Difference Time-

Domain Method", *Artech House Publishers*, pp. 59-66, 1995

[43] CST Microwave Studio V5, Computer Simulation Technology, http://www.cst.com

[44] A. Taflove, "Computational Electrodynamics: The Finite-Difference Time-Domain Method", *Artech House Publishers*, pp. 158-160, 1995

[45] Neal Zierler, "Linear recurring sequence", *Journal of the Society for Industrial and Applied Mathematics*, **Vol. 7, No. 1**. (Mar., 1959), pp.31-48.

[46] Solomon W. Golomb, "Shift-register sequences and spread-spectrum communications", Keynote Address, *IEEE Third International Symposium on Spread Spectrum Techniques & Applications*, Oulu, Finland, July 4-6, 1994

[47] Prof. A.K. Nandi, ELEC430, Adaptive Signal Processing, Lecture Notes, University of Liverpool, 2004

[48] Prof. Q. H. Wu, ELEC304, Optimisation, Lecture Notes, University of Liverpool, 2004

[49] A. S. Mudukutore, V. Chandrasekar and R. J. Keeler, "Pulse compression for weather radars", *IEEE trans. on Geoscience and Remote Sensing*, **Vol. 36, No. 1**, Jan 1998

Bibliography and Software

1. D. J. Daniels, **"**Ground Penetrating Radar 2nd Edition**"**, *IEE Radar, Sonar and Navigation Series 15*, pp.247-352
2. A. Taflove, "Computational Electrodynamics: The Finite-Difference Time-Domain Method", *Artech House Publishers*, 1995
3. J. L. Martins, "Approximate analytical formulas for Kirchhoff migration operator", *Applied Mathematics and Computation 163*, pp.1181-1195, 2005

4. P. Kosmas, C. M. Rappaport and E. Bishop, "Modeling With the FDTD Method for Microwave Breast Cancer Detection", *IEEE Transactions on Microwave Theory and Techniques*, **Vol. 52, No. 8** pp. 1890-1897, August 2004
5. H. Sun and G. T. Schuster, "3D wavepath migration", *Geophysical Prospecting*, **Vol 51**, pp. 421-430, 2003
6. I. J. Craddock, R. Nilavalan, J. Leenderz, A. Preece and R. Benjamin, "A Wideband Planar Antenna for In-body Imaging", *IEEE AP-S International Symposium and USNC/URSI National Radio Science Meeting Proceedings*, 2005
7. T. Huang and A. S. Mohan, "Application of Particle Swarm Optimisation for Microwave Imaging of Lossy Dielectric Objects", *IEEE AP-S International Symposium and USNC/URSI National Radio Science Meeting Proceedings*, 2005
8. http://www.cancerhelp.org.uk
9. http://www.wikipedia.org
10. Matlab V6.5-7.4, The Mathworks Inc.
11. CST Microwave Studio V5, Computer Simulation Technology, http://www.cst.com
12. http://imagine.gsfc.nasa.gov/docs/science/know_l1/emspectrum.html