Simulation studies on through-wall radar imaging

Tham, Wilson Wei Hsen

2006

Tham, W. W. H. (2006). Simulation studies on through-wall radar imaging. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/46943>

<https://doi.org/10.32657/10356/46943>

Nanyang Technological University

Downloaded on 26 Apr 2023 23:31:07 SGT

~~9435684~~

9438540

Simulation Studies on Through-Wall

Radar Imaging

Tham Wei Hsen, Wilson

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University

in fulfillment of the requirement for the degree of

Master of Engineering

2006

TK
5103.4
T366
2006

Acknowledgements

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. Anthony T. S. Ho. If not for his guidance, encouragement and support, I would not have achieved what I have today. His weekly update meetings have kept me on my toes throughout the course and have ensured that I made steady progress in my work. Working with him has been a great pleasure of mine.

Next, I would like to thank Prof. Low Kay Soon. During our weekly meetings, he always gave suggestions on how to improve on the accuracy of the results. His questions have always served to highlight shortcomings in my work as well as aspects which I have overlooked.

Special thanks to the laboratory executives, Mr. Mui, Suat Geok and Sandy, of Information Systems Research Lab (ISRL) for providing technical and administrative support.

Last, but most definitely not least, I have my parents to thank for all that I have. Their continuous support and encouragement throughout my course have been invaluable. Without them, I cannot imagine where I would be now. They really are the best !

Contents

Contents

ACKNOWLEDGEMENTS I

CONTENTS.....II

LIST OF FIGURES V

LIST OF TABLES X

ABSTRACT XI

CHAPTER 1 INTRODUCTION..... 1

1.1. OBJECTIVES 2

1.2. MAJOR CONTRIBUTIONS OF THIS THESIS..... 3

1.3. OVERVIEW OF THESIS 4

CHAPTER 2 RELATED RESEARCH 5

CHAPTER 3 FINITE DIFFERENCE TIME DOMAIN (FDTD) METHOD 7

3.1. INTRODUCTION 7

3.2. MAXWELL’S EQUATIONS 7

3.3. MAXWELL’S EQUATIONS IN 2 DIMENSIONS..... 8

3.4. FINITE DIFFERENCE FORMULATION 9

3.5. YEE CELL..... 11

3.6. GRID SIZE AND STABILITY CRITERION 12

3.7. BERENGER’S PERFECTLY MATCHED LAYER (PML) 12

3.7.1. Introduction..... 12

3.7.2. Definition of the PML Medium 13

3.8. SUMMARY 16

CHAPTER 4 FDTD SIMULATION USING MATLAB 17

4.1. INTRODUCTION 17

4.2. SETUP PARAMETERS 17

4.3. UWB INPUT PULSE..... 18

4.4. SIMULATION RUN 19

4.5. RECEIVED SIGNAL..... 20

Contents

4.6.	SUMMARY	22
CHAPTER 5	IMAGE RECONSTRUCTION	23
5.1.	INTRODUCTION	23
5.2.	NMO CORRECTION	25
5.3.	VELOCITY SEARCH	26
5.4.	DIX FORMULA.....	28
5.5.	STACKING	30
5.6.	PULSE DETECTION	30
5.6.1.	<i>Cross-correlation.....</i>	<i>30</i>
5.6.2.	<i>Hilbert Transform.....</i>	<i>32</i>
5.6.3.	<i>Result of Pulse Detection.....</i>	<i>33</i>
5.7.	SUMMARY	33
CHAPTER 6	RESULTS FOR IMAGE RECONSTRUCTION.....	35
6.1.	INTRODUCTION	35
6.2.	APPLICATION OF NMO CORRECTION	35
6.2.1.	<i>First Layer of Wall.....</i>	<i>36</i>
6.2.2.	<i>Second Layer of Wall.....</i>	<i>36</i>
6.2.3.	<i>First Layer of Object.....</i>	<i>37</i>
6.3.	APPLICATION OF STACKING	38
6.4.	EFFECT OF NOISE ON STACKING	41
6.5.	APPLICATION OF NOISE FILTER ON STACKING	43
6.6.	IMAGE RECONSTRUCTION ON OBJECTS OF DIFFERENT SHAPES.....	47
6.6.1.	<i>Cylindrical Object.....</i>	<i>47</i>
6.6.2.	<i>Two Partially Overlapped Flat Objects.....</i>	<i>51</i>
6.7.	SUMMARY	55
CHAPTER 7	SIGNAL CLASSIFICATION AND IDENTIFICATION.....	56
7.1.	INTRODUCTION	56

Contents	
7.2.	PRONY'S METHOD 56
7.3.	SINGULAR VALUE DECOMPOSITION (SVD) 58
7.4.	FISHER LINEAR DISCRIMINANT (FLD)..... 59
7.5.	SUMMARY 60
CHAPTER 8 RESULTS FOR SIGNAL CLASSIFICATION AND IDENTIFICATION 61	
8.1.	INTRODUCTION 61
8.2.	NUMBER OF POLES AND RESIDUES USED TO ESTIMATE A SIGNAL 61
8.3.	EFFECT OF NOISE ON RESIDUE AMPLITUDE 65
8.4.	EFFECT OF NOISE ON RECONSTRUCTED SIGNAL 66
8.5.	IMPROVEMENT USING SVD..... 69
8.6.	DOMINANT POLES..... 73
8.6.1.	<i>Residue vs. Frequency</i> 73
8.6.2.	<i>Attenuation vs. Frequency</i> 74
8.7.	DOMINANT POLES OF OBJECTS OF DIFFERENT MATERIAL 76
8.8.	SUMMARY..... 80
CHAPTER 9 CONCLUSION AND FUTURE DIRECTION 81	
9.1.	FUTURE WORK..... 82
AUTHOR'S PUBLICATIONS 84	
REFERENCES..... 85	
APPENDIX A: DERIVATION OF PRONY'S EQUATIONSA-1	
APPENDIX B: MATLAB CODE FOR FDTD SIMULATIONB-1	

List of Figures

List of Figures

FIGURE 1-1 OVERALL FLOW PROCESS.....	3
FIGURE 3-1 POSITION OF THE VARIOUS ELECTRIC AND MAGNETIC FIELD COMPONENTS WITHIN A SINGLE YEE CELL.....	11
FIGURE 3-2 SEQUENCE SHOWING THE RADIATED WAVE REFLECTING OFF THE BOUNDARY OF THE SIMULATION DOMAIN.	12
FIGURE 3-3 DIAGRAM OF PML MEDIUM SURROUNDING THE SIMULATION DOMAIN AND THE DEFINITIONS OF THE TERMS USED IN THE UPDATE EQUATIONS.	15
FIGURE 3-4 SEQUENCE SHOWING THE RADIATED WAVE BEING ABSORBED BY BERENGER'S PERFECTLY MATCHED LAYER AT THE BOUNDARY OF THE SIMULATION DOMAIN.	15
FIGURE 4-1 LAYOUT OF THE SIMULATION DOMAIN	17
FIGURE 4-2 TRANSMITTED UWB PULSE AND ITS FREQUENCY SPECTRUM.	18
FIGURE 4-3 SIMULATION OF ELECTROMAGNETIC FIELD PASSING THROUGH A WALL AND INTERACTING WITH AN OBJECT	20
FIGURE 4-4 RECEIVED SIGNAL FROM A SINGLE RECEIVER AS WELL AS THE TIME-GATED PORTION WHICH IS THE RESULT OF REFLECTING OFF THE OBJECT	21
FIGURE 4-5 IMAGE FORMED FROM SIGNALS OF MANY RECEIVERS.....	22
FIGURE 5-1 SIGNAL ACQUISITION PROCESS USING A SINGLE TRANSMITTER AND RECEIVER PAIR	23
FIGURE 5-2 SINGLE TRANSMITTER AND ARRAY OF RECEIVERS USED IN RADAR IMAGING SYSTEM	24
FIGURE 5-3 TIME-DELAY DIFFERENCE DUE TO VARYING DISTANCE OF DIFFERENT RECEIVERS FROM THE TRANSMITTER.....	24
FIGURE 5-4 DEFINITION OF TERMS USED IN NMO EQUATION FOR A MULTI-LAYERED MEDIUM	26

List of Figures

FIGURE 5-5 NMO CORRECTION APPLIED ON RECEIVER SIGNAL	26
FIGURE 5-6 VELOCITIES OF THE DIFFERENT MEDIUMS	27
FIGURE 5-7 VELOCITY SEARCH.....	28
FIGURE 5-8 ILLUSTRATION OF TERMS USED IN DIX’S FORMULA.....	29
FIGURE 5-9 (A) COMMON MID-POINTS FOR 3 DIFFERENT TRANSMITTER POSITIONS (B) COMMON MID-POINT GATHER.....	29
FIGURE 5-10 (A) SIGNAL FROM RECEIVER (B) TRANSMITTED PULSE AND (C) CROSS- CORRELATION RESULT.....	31
FIGURE 5-11 ENVELOP DETECTION USING HILBERT TRANSFORM.....	32
FIGURE 5-12 TIME-STEP IN WHICH TRANSMITTED PULSE WAS DETECTED IN THE RECEIVED SIGNAL.	33
FIGURE 6-1 RECEIVED SIGNALS FOR A SINGLE TRANSMITTER POSITION	35
FIGURE 6-2 NMO CORRECTION APPLIED TO FIRST LAYER OF WALL.....	36
FIGURE 6-3 NMO CORRECTION APPLIED TO SECOND LAYER OF WALL.....	37
FIGURE 6-4 NMO CORRECTION APPLIED TO FIRST LAYER OF OBJECT	37
FIGURE 6-5 (A) IMAGE FROM A SINGLE TRANSMITTER AND RECEIVER PAIR (B) IMPROVED IMAGE USING AN ARRAY OF RECEIVERS AND APPLYING STACKING. NOTE: AMPLITUDE HAS BEEN NORMALIZED TO 1.....	38
FIGURE 6-6 SINGLE RECEIVER READING FROM (A) SINGLE TRANSMITTER AND RECEIVER PAIR AND (B) ARRAY OF RECEIVERS WITH STACKING APPLIED. NOTE: AMPLITUDE HAS BEEN NORMALIZED TO 1.....	39
FIGURE 6-7 STACKED SIGNALS (A) WITHOUT NOISE AND (B) NOISE WITH SNR 20 dB.....	41
FIGURE 6-8 SINGLE STACKED SIGNAL (A) WITHOUT NOISE AND (B) NOISE WITH SNR 20 dB	42
FIGURE 6-9 EFFECT OF FILTER SIZE ON AMPLITUDE GAIN.....	44

List of Figures

FIGURE 6-10 STACKED SIGNALS (20 dB NOISE ADDED) (A) WITHOUT FILTERING AND (B) WITH FILTERING APPLIED.	45
FIGURE 6-11 SINGLE STACKED SIGNAL (20 dB NOISE ADDED) (A) WITHOUT FILTERING AND (B) WITH FILTERING APPLIED.....	46
FIGURE 6-12 IMPROVEMENT IN THE RECEIVED SIGNAL AMPLITUDE WITH THE APPLICATION OF FILTER FOR VARIOUS NOISE LEVELS.....	47
FIGURE 6-13 LAYOUT OF SIMULATION DOMAIN WITH CYLINDRICAL OBJECT.	48
FIGURE 6-14 RECEIVED SIGNAL FROM A SINGLE TRANSMITTER POSITION.....	48
FIGURE 6-15 NMO CORRECTION APPLIED TO FIRST LAYER OF CYLINDRICAL OBJECT.	49
FIGURE 6-16 (A) IMAGE FROM A SINGLE TRANSMITTER AND RECEIVER PAIR (B) IMPROVED IMAGE USING AN ARRAY OF RECEIVERS AND APPLYING STACKING.....	50
FIGURE 6-17 SINGLE RECEIVER READING FROM (A) SINGLE TRANSMITTER AND RECEIVER PAIR AND (B) ARRAY OF RECEIVERS WITH STACKING APPLIED. NOTE: AMPLITUDE HAS BEEN NORMALIZED TO 1.....	51
FIGURE 6-18 LAYOUT OF SIMULATION DOMAIN WITH TWO PARTIALLY OVERLAPPED FLAT OBJECTS.	52
FIGURE 6-19 RECEIVED SIGNAL FROM A SINGLE TRANSMITTER POSITION.....	52
FIGURE 6-20 NMO CORRECTION APPLIED TO FIRST LAYER OF THE TWO PARTIALLY OVERLAPPED FLAT OBJECTS.	53
FIGURE 6-21 (A) IMAGE FROM A SINGLE TRANSMITTER AND RECEIVER PAIR (B) IMPROVED IMAGE USING AN ARRAY OF RECEIVERS AND APPLYING STACKING.....	53
FIGURE 6-22 SINGLE RECEIVER READING FROM (A) SINGLE TRANSMITTER AND RECEIVER PAIR AND (B) ARRAY OF RECEIVERS WITH STACKING APPLIED. NOTE: AMPLITUDE HAS BEEN NORMALIZED TO 1.....	54
FIGURE 8-1 (A) MEAN SQUARED ERROR OF RECONSTRUCTED SIGNAL USING DIFFERENT INITIAL NUMBER OF POLE AND RESIDUE ESTIMATES (B) TIME-GATED PORTION OF THE	

List of Figures

REFLECTED PULSE FROM THE OBJECT. (C) RECONSTRUCTED SIGNAL BASED ON 170 POLES AND RESIDUES. (D) RECONSTRUCTED SIGNAL BASED ON 130 POLES AND RESIDUES	63
FIGURE 8-2 (A) MEAN SQUARED ERROR (B) MEAN ABSOLUTE ERROR AND (C) MEAN ABSOLUTE PERCENTAGE ERROR, VERSUS DIFFERENT INITIAL POLE AND RESIDUE ESTIMATES FOR DIFFERENT NOISE LEVELS.....	64
FIGURE 8-3 RESIDUE VALUES OF 80 POLES FROM A SIGNAL (A) WITHOUT NOISE (B) WITH SNR OF 20 DB	66
FIGURE 8-4 (A) SIGNAL WITH NO NOISE (B) SIGNAL WITH SNR 20 DB (C) RECONSTRUCTED SIGNAL BASED ON 300 POLES AND RESIDUES.	67
FIGURE 8-5 (A) MEAN-SQUARED ERROR (B) MEAN ABSOLUTE ERROR AND (C) MEAN ABSOLUTE PERCENTAGE ERROR BETWEEN RECONSTRUCTED SIGNAL AND ORIGINAL SIGNAL (WITH NO NOISE) WITH VARYING AMOUNTS OF NOISE ADDED.....	68
FIGURE 8-6 SINGULAR VALUES OF SIGNAL WITH NO NOISE.	70
FIGURE 8-7 SINGULAR VALUES OF SIGNAL WITH SNR OF 20 DB.	70
FIGURE 8-8 RECONSTRUCTED SIGNAL (A) WITHOUT AND (B) WITH THE APPLICATION OF SVD.....	71
FIGURE 8-9 IMPROVEMENT IN THE (A) MEAN-SQUARED ERROR (B) MEAN ABSOLUTE ERROR AND (C) MEAN ABSOLUTE PERCENTAGE ERROR, WITH THE USE OF SVD FOR VARYING AMOUNTS OF NOISE	72
FIGURE 8-10 (A) POSITION OF POLES WITH HIGH RESIDUES AND (B) DOMINANT POLES.....	74
FIGURE 8-11 (A) POLE POSITIONS OF SIGNALS WITH VARYING AMOUNTS OF NOISE (B) MAGNIFIED VIEW.....	75
FIGURE 8-12 (A) DOMINANT POLES (B) MAGNIFIED VIEW.	76
FIGURE 8-13 RECEIVED SIGNAL FOR DIFFERENT OBJECT MATERIALS	77

List of Figures

FIGURE 8-14 DOMINANT POLES OF RECEIVED SIGNAL FROM OBJECTS OF DIFFERENT MATERIAL. 78

FIGURE 8-15 DOMINANT POLES OF RECEIVED SIGNAL FROM OBJECTS OF DIFFERENT MATERIAL. 79

FIGURE 8-16 USE OF FISHER LINEAR DISCRIMINANT TO CLASSIFY THE DIFFERENT MATERIALS. 79

List of Tables

List of Tables

TABLE 8-1 MATERIAL PROPERTIES OF THE DIFFERENT OBJECTS USED77

Abstract

Abstract

Through-wall radar imaging has become a keen research topic due to the emphasis on homeland security in recent years. Applications for this area include domestic counter-terrorism, urban warfare, hostage rescue as well as search and rescue operations.

An electromagnetic wave simulation program, using the finite difference time domain (FDTD) method was developed to model ultra-wideband (UWB) transient signals passing through walls and reflecting off objects of different materials. The reflected signals were then recorded by multiple receivers located at various positions on the same side of the wall as the transmitter.

The difference in positions between the multiple receivers and the single transmitter results in varying time-delays in the received signals. Normal moveout (NMO) correction was applied to compensate for the time-delays. Multiple readings were taken by re-positioning the transmitter and multiple receiver array. All the received signals were then combined to form an image which represents the reconstruction of the position and shape of the objects behind the wall.

Classification and identification of objects (of different material) behind the wall was also carried out. The Prony's method was used to decompose the reflected signal from the object into its constituent residues and poles. The pole positions and residue amplitudes were then used to classify the objects behind the wall based on their material medium. The effect of noise on the pole positions was also investigated and singular value decomposition (SVD) was used to enhance (mean squared error was reduced by 37% for a signal with signal-to-noise ratio of 30 dB) the pole stability in the presence of noise.

Chapter 1

Introduction

There are many occasions when it is desirable to inspect the interior of a room or building before physically entering it. This is especially so when entering the room might pose a serious danger to the person involved. These situations include civil engineering, search and rescue operations and law enforcement scenarios [1]. An example would be law enforcement officers faced with a hostage situation in a room or building. In such a scenario, there is a critical need to know the precise location of the hostages as well as the hostage-takers in order for the police officers to attain better situational awareness before making a well-coordinated rescue attempt.

Through-wall radar imaging is the technology which addresses these problems and has become a keen research topic in recent years. Its importance is due to the increased emphasis on homeland security, which involves many key areas such as intelligence gathering, concealed weapons detection [2] and domestic counter-terrorism. Other applications for through-wall imaging include urban warfare, imaging of objects that are located on the other side of a wall and detection of objects contained within a wall.

The study and use of ultra-wideband (UWB) technology has also gained momentum in the last few years due to the approval of its use commercially by the Federal Communications Commission (FCC) in 2002. The large available bandwidth for UWB and its performance in dispersive environments make it ideal for indoor use in high-clutter environments. Coupled with the fact that UWB provides high spatial resolution, the application of UWB in through-wall radar imaging can be seen as an obvious choice.

Chapter 1 Introduction

Currently, a few through-wall imaging system prototypes have already been developed [3-6] and numerous experimental and numerical simulation studies [7-16] have been researched upon. In numerical simulation studies, various methods [7-12] were used to model the transmission of electromagnetic signals through a wall and other object materials. In the current research work, the finite difference time domain (FDTD) [17, 18] method was used as the simulation model of choice due to its ease of implementation and its ability to model complex object geometries.

Through-wall image reconstruction was carried out by several researchers [11,12,14,15]. In this work, the received signals were first time-aligned using normal-moveout (NMO) correction [17-19], a term borrowed from the field of seismology. The time-aligned signals were then combined to form a reconstructed image of the objects behind the wall.

Analysis and classification of through-wall radar signals also poses many difficulties due to the presence of the wall as well as multiple reflections due to many objects within the room. It is thus desirable to extract certain dominant characteristics from the received signals which correspond to the different types of object materials within the room.

The Prony's method [23-25] was used to decompose the received reflected signals into their characteristic poles and residues. These pole positions and residue amplitudes were then analyzed to determine the "dominant" poles which correspond to certain material properties. Singular value decomposition (SVD) [26] was also used to improve on the pole stability in the presence of noise

1.1. Objectives

Through-wall imaging is still in its infancy and presents itself as an important direction in radar microwave imaging. Unique requirements of through wall imaging are high cross-

Chapter 1 Introduction

range resolution, low power, small size, low cost and high-speed evaluation. To achieve such requirements, research and development efforts in hardware as well as in analysis and software are still needed.

This work has three goals. The first is to design and implement a computer simulation program to model the transmission of an UWB pulse through a wall and to record the reflected signal using an array of receivers. Second, a post-processing technique will have to be applied in order to time-align the recorded signals so that they can be combined to form an image of the position and shape of the objects behind the wall. Lastly, classification and identification of the objects behind the wall will be performed based on the object’s material composition. Figure 1-1 illustrates the flow process in accomplishing the three objectives as stated above.

Figure 1-1 Overall flow process.

1.2. Major Contributions of this Thesis

This thesis presents the development of a 2D simulation program to model the transmission of an UWB pulse through walls.

Image reconstruction was performed on the simulation data and it was shown that the image of partially overlapping flat objects could be reconstructed successfully. To overcome the effects of the presence of the wall, a velocity search technique was

Chapter 1 Introduction

introduced. To the best of the author's knowledge, no previous research has been done on using a velocity search technique to correct for the time-delay in the received signals in the presence of an obstacle such as the wall. Effects of noise on the signal reconstruction was also studied in detail.

The reflected signals of objects of various materials behind the wall was decomposed using Prony's method. The dominant poles produced using Prony's method was then studied in detail to classify them according to their material properties. It was shown that objects made of wood, glass and metal could be distinctly classified based on their dominant poles.

1.3. Overview of Thesis

In Chapter 2, related research works will be presented. The finite difference time domain (FDTD) method and its implementation using Matlab will be introduced in Chapters 3 and 4.

The use of normal moveout (NMO) correction in image reconstruction as well as the results obtained are shown in Chapters 5 and 6 respectively. The use of Prony's method in signal classification and identification, along with the results obtained will be discussed in Chapters 7 and 8, respectively. Finally, the conclusion and future directions will be presented in Chapter 9.

Chapter 2

Related Research

Several through-wall imaging prototypes have already been developed commercially in the past few years. Time Domain Corporation is one such company which has developed a time-modulated ultra-wide band (TM-UWB) wall penetrating impulse radar [1,2]. Their TM-UWB radar is able to generate two-dimensional interferometric images representing the location of moving people behind non-metallic walls. They make use of a moving target detection (MTD) filter [2] to detect targets moving at different velocities. However, their system is meant for detection of moving targets and does not allow for imaging of stationary objects behind the wall.

AKELA [3] developed a high range resolution radar operating between 450 MHz and 2GHz to produce images and detect motion behind walls. Their radar consists of a single transmitter and a synthetic array of multiple receivers. A backprojection algorithm was used to reconstruct the image of the object behind the wall. A limitation of their technique is the presence of ghost images which is the result of the geometry of their antenna array.

A robot-mounted motion-detection radar developed by SRI International [4] employs pulse-Doppler techniques to locate moving targets behind walls. It acts as a real-time surveillance system, scanning and detecting the presence of human activity. Its classification and identification of target motion is based on the analysis of the energy in the Doppler waveform and the kurtosis of the time-domain waveform.

Stratis and Demetriou [7] conducted a numerical study using the FDTD method. However, their investigation was mainly on the effects of wall thickness and material on the reflection and transmission coefficients of the wall. Other methods of simulation

Chapter 2 Related Research

included frustrum ray tracing technique (FRTT) [8], modal transmission-line (MTL) method [10] and the microwave tomography technique [11].

In the area of through-wall imaging, Benjamin et. al. [14] made use of the real-aperture synthetically organised radar (RASOR) to transmit and record the received transient signals. The recorded received signals were then time-aligned before they were combined to form the reconstructed image. However, the time-alignment algorithm needed prior knowledge of the estimated path delay in the signal. It was stated that this method would fail if the material were highly inhomogeneous. On the other hand, the velocity search method (section 5.3), as applied in this thesis, enables the received signals to be time-aligned even with the presence of inhomogeneous materials.

Barrie [15] studied the use of an ultra-wideband (UWB) synthetic aperture radar (SAR) to implement a through-wall imaging system. The backprojection algorithm was used to reconstruct the image. However, the algorithm is computationally expensive. For example, given there are T transmitter positions, R receivers and N discrete sampling points, the the backprojection algorithm will require $T \times R^2 \times N$ computations to form the final reconstructed image. The NMO method, as applied in this thesis, will only require $T \times R \times N$ computations. Also, throughout the study, only a single object was used and no wall was included. As such, it is unclear how the presence of a wall will affect the imaging process using the backprojection algorithm.

In the area of signal classification and identification, Hoffman [32] investigated the detection of metal and plastic objects underground using signals from a ground penetrating radar. He used the Prony's method to extract model parameters from the signals. Feature vectors were then formed from the model parameters and classification of the object materials was based on the difference in distance between the feature vectors. It was shown that the feature vectors of a metal and plastic object were clearly distinguishable.

Chapter 3

Finite Difference Time Domain (FDTD) Method

3.1. *Introduction*

The finite difference time domain (FDTD) method was first introduced by Yee [17] and has since been very widely used in the simulation of electromagnetic waves interacting with material media. A great amount of work has been done subsequently by Taflov et. al. [18,27] and Luebbers et. al. [28] to improve and apply the technique.

Using the FDTD method, the diffraction of waves from a source is modeled by repeatedly solving the finite difference equivalent of Maxwell's equations. Subsequently, both electric and magnetic fields are solved in time and space simultaneously.

FDTD provides several advantages over other electromagnetic field modeling methods. One of which is its ease of implementation. With FDTD, there is no need to solve simultaneous equations explicitly or to perform any large matrix inversions. Another advantage is that there is no need to specifically define the boundary conditions of the problem. Instead, only the material permittivity, permeability and conductivity need to be specified at each point in space. This enables the simulation of complicated materials with varying material properties to be modeled with ease.

3.2. *Maxwell's Equations*

Maxwell's equations for linear, isotropic, non-dispersive materials are:

$$\frac{\partial B}{\partial t} = -\nabla \times E \quad (1)$$

$$\frac{\partial D}{\partial t} = \nabla \times H - \sigma E \quad (2)$$

Chapter 3 Finite Difference Time Domain (FDTD) Method

$$B = \mu H \quad (3)$$

$$D = \epsilon E \quad (4)$$

where E is the electric field, D is the electric flux density, H is the magnetic field, B is the magnetic flux density and σ is the electric conductivity.

The above curl operators can be expanded into their three dimensional vector components in Cartesian coordinates as follows:

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu} \left[\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \right] \quad (5)$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu} \left[\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \right] \quad (6)$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu} \left[\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right] \quad (7)$$

$$\frac{\partial E_x}{\partial t} = \frac{1}{\epsilon} \left[\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - \sigma E_x \right] \quad (8)$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\epsilon} \left[\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - \sigma E_y \right] \quad (9)$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\epsilon} \left[\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z \right] \quad (10)$$

3.3. Maxwell's Equations in 2 dimensions

To simplify the simulation, it is assumed that the electric and magnetic fields do not vary in the z -direction (i.e. the partial derivatives with respect to z are zero). Then the set of Maxwell's equations can be reduced to two dimensions and subsequently simplified and grouped according to the transverse-magnetic mode (TM _{z})

Chapter 3 Finite Difference Time Domain (FDTD) Method

$$\frac{\partial H_x}{\partial t} = -\frac{1}{\mu} \left[\frac{\partial E_z}{\partial y} \right] \quad (11)$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu} \left[\frac{\partial E_z}{\partial x} \right] \quad (12)$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon} \left[\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z \right] \quad (13)$$

as well as the transverse-electric mode (TE_z)

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu} \left[\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right] \quad (14)$$

$$\frac{\partial E_x}{\partial t} = \frac{1}{\varepsilon} \left[\frac{\partial H_z}{\partial y} - \sigma E_x \right] \quad (15)$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\varepsilon} \left[-\frac{\partial H_z}{\partial x} - \sigma E_y \right] \quad (16)$$

The transverse-magnetic mode (TM_z) was used in the simulation.

3.4. Finite Difference Formulation

Yee [17] then replaced Maxwell's equations with a set of finite difference equations.

Based on Yee's notation, a grid point in space is defined as

$$(i, j, k) = (i\Delta x, j\Delta y, k\Delta z) \quad (17)$$

and a function of space and time is defined as

$$F|_{i,j,k}^n = F(i\Delta x, j\Delta y, k\Delta z, n\Delta t) \quad (18)$$

Based on the above notation, the finite difference expressions for the time and space derivatives are respectively

Chapter 3 Finite Difference Time Domain (FDTD) Method

$$\frac{\partial F|_{i,j,k}^n}{\partial x} = \frac{F|_{i+1/2,j,k}^n - F|_{i-1/2,j,k}^n}{\Delta x} \quad (19)$$

$$\frac{\partial F|_{i,j,k}^n}{\partial t} = \frac{F|_{i,j,k}^{n+1/2} - F|_{i,j,k}^{n-1/2}}{\Delta t} \quad (20)$$

Next, consider equation (8) which can be rewritten in terms of finite differences as

$$\frac{E_x|_{i,j+1/2,k+1/2}^{n+1/2} - E_x|_{i,j+1/2,k+1/2}^{n-1/2}}{\Delta t} = \frac{1}{\epsilon_{i,j+1/2,k+1/2}} \left[\frac{H_z|_{i,j+1,k+1/2}^n - H_z|_{i,j,k+1/2}^n}{\Delta y} - \frac{H_y|_{i,j+1/2,k+1}^n - H_y|_{i,j+1/2,k}^n}{\Delta z} - \sigma_{i,j+1/2,k+1/2} E_x|_{i,j+1/2,k+1/2}^n \right] \quad (21)$$

where the E_x field associated with the material conductivity can be approximated (based on [18]) as follows

$$E_x|_{i,j+1/2,k+1/2}^n = \frac{E_x|_{i,j+1/2,k+1/2}^{n+1/2} + E_x|_{i,j+1/2,k+1/2}^{n-1/2}}{2} \quad (22)$$

Substituting equation (22) into equation (21), rearranging and gathering terms together, the following is obtained

$$E_x|_{i,j+1/2,k+1/2}^{n+1/2} = \left(\frac{1 - \frac{\sigma_{i,j+1/2,k+1/2} \Delta t}{2\epsilon_{i,j+1/2,k+1/2}}}{1 + \frac{\sigma_{i,j+1/2,k+1/2} \Delta t}{2\epsilon_{i,j+1/2,k+1/2}}} \right) E_x|_{i,j+1/2,k+1/2}^{n-1/2} + \left(\frac{\Delta t}{1 + \frac{\sigma_{i,j+1/2,k+1/2} \Delta t}{2\epsilon_{i,j+1/2,k+1/2}}} \right) \left(\frac{H_z|_{i,j+1,k+1/2}^n - H_z|_{i,j,k+1/2}^n}{\Delta y} - \frac{H_y|_{i,j+1/2,k+1}^n - H_y|_{i,j+1/2,k}^n}{\Delta z} \right) \quad (23)$$

The above simply states the formulation of how to update the E_x field component of a particular cell, based on the previous E_x value as well as the surrounding H_z and H_y field

Chapter 3 Finite Difference Time Domain (FDTD) Method

components calculated previously. Similar updating equations for E_y , E_z , H_x , H_y and H_z can be formulated based on the principle described above.

3.5. Yee Cell

The notation in the previous section can be illustrated in figure 3-1. This is commonly known as the Yee cell [17]. By visualizing many of these Yee cells forming a three dimensional space, it can be observed that every E-field component is surrounded by four circulating H-field components and similarly, every H-field component is surrounded by four circulating E-field components.

Figure 3-1 Position of the various electric and magnetic field components within a single Yee cell.

Chapter 3 Finite Difference Time Domain (FDTD) Method

3.6. Grid Size and Stability Criterion

For stability and accuracy, the grid size, Δx , must be a small fraction of the minimum expected wavelength of the propagating wave during the simulation. This will ensure that the electromagnetic field will not change significantly over a unit cell within a unit time-step. Once the grid size has been determined, the time-step must then satisfy the following criteria [27] to maintain computational stability

$$c\Delta t \leq \frac{1}{\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}} \quad (24)$$

3.7. Berenger's Perfectly Matched Layer (PML)

3.7.1. Introduction

A problem with using the FDTD method is that as the scattered electromagnetic fields reach the boundary of the simulation domain, they are reflected back (analogous to reflecting off a metal wall). This is shown in figure 3-2. As a result, after a certain time, the reflected waves will begin to interfere with the relevant calculated data within the simulation domain, thus corrupting the simulated data.

Figure 3-2 Sequence showing the radiated wave reflecting off the boundary of the simulation domain.

A possible solution would be to enlarge the simulation domain such that the outgoing electromagnetic waves will take a very long period of time to reflect off the

Chapter 3 Finite Difference Time Domain (FDTD) Method

simulation boundary and thus will not interfere with the relevant simulation data. However, this solution will not be feasible if a steady-state solution is required. Also, the computation time for a larger simulation domain increases dramatically. Therefore, there is a need to “absorb” the electromagnetic waves impinging at the boundary. This is similar to the padding on the walls of an anechoic chamber to absorb the sound waves.

Absorbing boundary conditions (ABCs) developed by Berenger [29,30] and Mur [31] have been widely used in many FDTD models to solve for the boundary problem. In the current simulation model, Berenger’s perfectly matched layer (PML) was chosen due to its more effective absorption of the electromagnetic waves at any frequency and at any incidence angle.

3.7.2. Definition of the PML Medium

Berenger’s [29,30] PML medium will be considered in two dimensions in the transverse magnetic (TM_z) mode. For the PML medium, the electric component E_z will be split into two components, E_{zx} and E_{zy}. Equations (15)-(17) will then be represented as

$$\mu_0 \frac{\partial H_x}{\partial t} + \sigma_y^* H_x = -\frac{\partial(E_{zx} + E_{zy})}{\partial y} \quad (25)$$

$$\mu_0 \frac{\partial H_y}{\partial t} + \sigma_x^* H_y = \frac{\partial(E_{zx} + E_{zy})}{\partial x} \quad (26)$$

$$\epsilon_0 \frac{\partial E_{zx}}{\partial t} + \sigma_x E_{zx} = -\frac{\partial H_y}{\partial x} \quad (27)$$

$$\epsilon_0 \frac{\partial E_{zy}}{\partial t} + \sigma_y E_{zy} = \frac{\partial H_x}{\partial y} \quad (28)$$

where σ_x , σ_x^* , σ_y , σ_y^* are analogous to the electric and magnetic conductivities.

The updating equations for the PML medium can then be formulated as

Chapter 3 Finite Difference Time Domain (FDTD) Method

$$H_x^{n+1}(i+1/2, j) = e^{-\sigma_y^*(i)\Delta t / \mu_0} H_x^n(i+1/2, j) - \frac{(1 - e^{-\sigma_y^*(i)\Delta t / \mu_0})}{\sigma_y^*(i)\Delta y} \times$$

$$\left[E_{zx}^{n+1/2}(i+1/2, j+1/2) - E_{zx}^{n+1/2}(i+1/2, j-1/2) + \right.$$

$$\left. E_{zy}^{n+1/2}(i+1/2, j+1/2) - E_{zy}^{n+1/2}(i+1/2, j-1/2) \right] \quad (29)$$

$$H_y^{n+1}(i, j+1/2) = e^{-\sigma_x^*(i)\Delta t / \mu_0} H_y^n(i, j+1/2) + \frac{(1 - e^{-\sigma_x^*(i)\Delta t / \mu_0})}{\sigma_x^*(i)\Delta x} \times$$

$$\left[E_{zx}^{n+1/2}(i+1/2, j+1/2) - E_{zx}^{n+1/2}(i-1/2, j+1/2) + \right.$$

$$\left. E_{zy}^{n+1/2}(i+1/2, j+1/2) - E_{zy}^{n+1/2}(i-1/2, j+1/2) \right] \quad (30)$$

$$E_{zx}^{n+1/2}(i+1/2, j+1/2) = e^{-\sigma_x(i+1/2)\Delta t / \varepsilon_0} E_{zx}^{n-1/2}(i+1/2, j+1/2) +$$

$$\frac{(1 - e^{-\sigma_x(i+1/2)\Delta t / \varepsilon_0})}{\sigma_x(i+1/2)\Delta x} \times [H_y^n(i+1, j+1/2) - H_y^n(i, j+1/2)] \quad (31)$$

$$E_{zy}^{n+1/2}(i+1/2, j+1/2) = e^{-\sigma_y(i+1/2)\Delta t / \varepsilon_0} E_{zy}^{n-1/2}(i+1/2, j+1/2) -$$

$$\frac{(1 - e^{-\sigma_y(i+1/2)\Delta t / \varepsilon_0})}{\sigma_y(i+1/2)\Delta y} \times [H_x^n(i+1/2, j+1) - H_x^n(i+1/2, j)] \quad (32)$$

where $\sigma_n(0) = \frac{\varepsilon_0 c \ln R(0)}{2^{n+2} \Delta x N^{n+1}}$,

$$\sigma_n(L > 0) = \sigma_n(0) [(2L+1)^{n+1} - (2L-1)^{n+1}],$$

where $\sigma_n(0)$ is the conductivity value at the interface of the PML medium, $R(0)$ is the reflection factor expressed as a percentage, n is the profile of the PML medium (1 for linear, 2 for parabolic), N is the number of cells of the PML medium and L is the index of a particular cell in the PML medium with respect to the interface (see figure 3-3).

Chapter 3 Finite Difference Time Domain (FDTD) Method

Figure 3-3 Diagram of PML medium surrounding the simulation domain and the definitions of the terms used in the update equations.

Figure 3-4 shows the simulation after applying Berenger’s PML. None of the radiating electromagnetic field impinging on the boundary space has been reflected, thus effectively simulating the waves radiating into open space.

Figure 3-4 Sequence showing the radiated wave being absorbed by Berenger’s perfectly matched layer at the boundary of the simulation domain.

Chapter 3 Finite Difference Time Domain (FDTD) Method

3.8. *Summary*

The finite difference time domain (FDTD) method for electromagnetic wave simulation was presented. The update equations for implementing the FDTD method was also shown. Waves reflecting off the boundaries of the FDTD simulation domain was an inherent problem of the FDTD method and the use of Berenger's perfectly matched layer (PML) was highlighted to solve the problem.

Chapter 4

FDTD Simulation using Matlab

4.1. Introduction

Using the finite difference formulation of Maxwell's equations and Berenger's PML, a two-dimensional simulation model was designed and coded using the Matlab programming software. The simulation modeled the use of a single transmitter to emit an ultra wideband (UWB) pulse through a wall. The reflected pulse from an object behind the wall was recorded using a receiver array. The recorded data was then saved for post-processing.

4.2. Setup Parameters

Figure 4-1 Layout of the simulation domain

Chapter 4 FDTD Simulation using Matlab

The set up parameters for the two-dimensional FDTD simulation are shown in figure 4-1. The simulation domain consists of a 800-by-450 cell FDTD grid with a cell size of 0.0037 m. A gypsum wall ($\epsilon_r=2.8$, $\sigma=0.15$, $\mu_r=1.0$) and iron object ($\epsilon_r=1.0$, $\sigma=1 \times 10^7$, $\mu_r=1.0$) were used for this simulation. For display purposes, the receivers have been placed at every cell position (i.e. there are 800 receivers along the face of the wall). The Berenger's PML surrounding the simulation domain consists of an 8-cell thick conductive medium. Finally, the update equations were calculated with a time-step, Δt , of 6.25 ps for a total of 2000 time-steps.

4.3. UWB Input Pulse

The transmitted UWB pulse is given by the following equation:

$$x(t) = \cos(2\pi f_c t) \exp\left(-\frac{t^2}{u^2}\right) \quad (33)$$

where f_c is the center frequency and u determines the bandwidth of the pulse. For this simulation, a pulse with center frequency 4 GHz and bandwidth of 2 GHz was used. Figure 4-2 illustrates the transmitted ultra wideband pulse used in the simulation and its corresponding frequency spectrum.

Figure 4-2 Transmitted UWB pulse and its frequency spectrum.

Chapter 4 FDTD Simulation using Matlab

4.4. *Simulation Run*

In figure 4-3, sequences of snapshots were taken at various time instances of the simulation program. The two horizontal dotted lines define the wall while the rectangular enclosure defines a metallic object. Using the simulation program, the reflected signal can be sampled at any point within the simulation domain.

A single simulation run took approximately 10 minutes using a Pentium III 2.4 GHz computer with 512 Mb of RAM, running on Matlab version 6.5. It should be noted that the simulation program was not designed for speed and therefore, the simulation time could be significantly reduced through code optimization or by using another programming language such as C++.

The amount of CPU time needed is approximately proportional to the size of the simulation domain (i.e. the number of cells used). For example, in the scenario given above, a simulation domain of 3 m by 1.7 m took about 10 mins. Therefore, a simulation domain of 6 m by 1.7 m would take twice the amount of time (i.e. 20 mins) and a simulation domain of 6 m by 3.4 m would take 40 mins. Although a 3-D FDTD simulation was beyond the scope of this thesis, a similar reasoning can be applied. Namely, every doubling of the simulation domain size in the third dimension would result in twice the amount of CPU time taken.

Chapter 4 FDTD Simulation using Matlab

Figure 4-3 Simulation of electromagnetic field passing through a wall and interacting with an object

4.5. Received Signal

The UWB pulse was transmitted through the wall and the receivers recorded its reflected signal from the object. Figure 4-4 shows the received signal of a single receiver over 2000 time-steps. The magnified portion is the time-gated portion which is the reflected signal from the object only. Identification and classification of the signal will be applied only to the time-gated signal.

Chapter 4 FDTD Simulation using Matlab

Figure 4-4 Received signal from a single receiver as well as the time-gated portion which is the result of reflecting off the object

The signal from multiple receivers can be gathered together to form an image of the wall as well as the object behind the wall. Figure 4-5 below shows an image formed by gathering all the signals from the receivers together. The transmitter was positioned at the center of the simulation domain. The approximate position of the object can be seen from the strong signal at the center. However, the exact width of the object cannot be accurately determined. Also, a distinct hyperbolic shape can be seen. This is a result of varying time-delays of the transmitted pulse to the various receivers.

Chapter 4 FDTD Simulation using Matlab

Figure 4-5 Image formed from signals of many receivers.

4.6. *Summary*

The FDTD simulation model was implemented using Matlab. The received signals were recorded and saved for post-processing. The reading from a single receiver could be used for signal classification and identification (chapter 7) whereas combining the signals from the receiver array allowed for image reconstruction, which will be presented in the next chapter.

Chapter 5

Image Reconstruction

5.1. Introduction

A simple radar imaging system consists of a single transmitter and receiver located at the same position (figure 5-1). A single pulse is transmitted and the reflected signal is recorded by the single receiver. The transmitter and receiver pair is then shifted spatially and the signal acquisition process is repeated until there are sufficient readings to construct a coherent image of the surrounding environment.

Figure 5-1 Signal acquisition process using a single transmitter and receiver pair

An inherent drawback is that the transmitter and receiver pair needs to be shifted many times in order to cover a sufficient spatial area. Also, multiple readings may need to be taken to improve the signal to noise ratio (SNR). All these add to the time taken for the signal acquisition process.

As a result, many modern radar imaging systems consist of a single transmitter along with an array of receivers (figure 5-2). This process is more practical and efficient as compared to a single transmitter and receiver pair, which can only acquire a single

Chapter 5 Image Reconstruction

reading per cycle. However, with the use of multiple receivers, there is a difference in time-delay between the transmitted pulse and the received signals from the multiple receivers which are at varying distances away from the transmitter (figure 5-3). This is known as normal moveout (NMO) [6]. NMO correction will then be needed to time-align the received signals before they can be combined to form a coherent image of the wall and the objects behind the wall.

Figure 5-2 Single transmitter and array of receivers used in radar imaging system

Figure 5-3 Time-delay difference due to varying distance of different receivers from the transmitter

In the current scenario, where the object is behind a wall, NMO correction cannot be directly applied to the object due to the presence of the wall. Instead, the material properties of the wall will first have to be estimated via a “velocity search”. Following

Chapter 5 Image Reconstruction

this, Dix's formula [19] will have to be applied before NMO correction can be implemented accurately.

Finally, the transmitter and receiver array will be shifted spatially to obtain multiple readings. These multiple NMO-corrected signals will be summed together or "stacked" to filter out spurious reflections and noise, thus improving the SNR and the coherence of the reconstructed image.

5.2. NMO Correction

NMO [20] correction needs to be applied to the received signal of each individual receiver to correct for the time-delay. The time-delay for a particular receiver can be calculated based on the following:

$$T_x^2 = T_0^2 + \frac{x^2}{\bar{v}^2} \quad (34)$$

where T_x is the two-way time for offset distance x , T_0 is the two-way travel time at zero-offset distance, x is the distance between the transmitter and receiver and $\bar{v}$ is the root-mean-square (RMS) velocity (definition of $\bar{v}$ will be discussed in detail in section 5.4). Figure 5-4 shows the definition of some of the terms described. Also, the ability to accurately determine the travel times will depend on the detection of the transmitted pulse. This will be discussed in section 5.6.

Chapter 5 Image Reconstruction

Figure 5-4 Definition of terms used in NMO equation for a multi-layered medium

The time-delay was thus obtained using

$$T_{delay} = T_x - T_0 \tag{35}$$

and the received signal for a particular receiver was time corrected as shown in figure 5-5.

Figure 5-5 NMO correction applied on receiver signal

5.3. Velocity Search

Based on equation (34) above, it can be seen that the velocity of the medium, $\bar{v}$, must be known in order for NMO correction to be applied. If we were applying NMO correction to the reflected signals from the first layer of wall (see figure 5-6), $\bar{v}$ would simply be the velocity in air, V_{air} . However, if we were applying NMO correction to the reflected signal

Chapter 5 Image Reconstruction

from the second layer of the wall, then $\bar{v}$ would be more complex as it involves the velocity of the wall, V_{wall} , as well.

Figure 5-6 Velocities of the different mediums

A problem thus arises as the velocity of the wall, V_{wall} , is unknown. Without the wall velocity, NMO correction cannot be accurately applied to the object to determine its position. However, by assuming that the two wall surfaces are flat and parallel, we can estimate the wall velocity by performing a velocity search based on certain coherence measures. The measure which was implemented was the sum of the squared amplitudes of the received signals. Another possible coherence measure is cross-correlation.

Figure 5-7 shows how the velocity search was performed. At the correct velocity estimate, the coherence measure will have the highest value due to the superposition of the signal amplitudes. Under-compensation and over-compensation will result in a much lower coherence value. Thus, the velocity value which returned the highest coherence value was taken as the velocity of the wall.

Chapter 5 Image Reconstruction

Figure 5-7 Velocity search

5.4. Dix Formula

After obtaining a velocity estimate of the wall, NMO correction can be applied to the second layer of the wall and subsequently, on the object behind the wall as well. Knowing the velocity of the wall, V_{wall} , and the velocity of air, V_{air} , the RMS velocity, $\bar{v}$, can be obtained by using Dix's formula [19]:

$$\bar{v}_n^2 = \frac{\sum_{i=1}^n v_i^2 t_i}{\sum_{i=1}^n t_i} \quad (36)$$

where $\bar{v}_n$ is the RMS velocity of the n^{th} layer, v_i is the velocity of the i^{th} layer and t_i is the one-way zero-offset travel time in the i^{th} layer. Figure. 5-8 below illustrates the terms defined in Dix's formula.

Chapter 5 Image Reconstruction

Figure 5-8 Illustration of terms used in Dix's formula.

Figure 5-9 (a) Common mid-points for 3 different transmitter positions (b) common mid-point gather

Chapter 5 Image Reconstruction

5.5. *Stacking*

By changing the transmitter position, multiple time-delayed signals can be obtained for a single point (known as the common midpoint) on the wall or object (figure 5-9(a)). These signals (corresponding to the same mid-point) can then be accumulated together to form the common mid-point gather (figure 5-9(b)).

NMO correction can then be applied to the signals of the common mid-point gather and the signals added together. This process is known as *stacking*. Some of the signals in the common mid-point gather will contain spurious reflected signals from other objects as well as noise from the surrounding environment. By performing *stacking*, only the reflected signal from the object will be coherently added, while the spurious reflections will be cancelled out, thus improving the SNR of the reconstructed image.

5.6. *Pulse Detection*

To apply NMO correction, there is a need to accurately determine the time taken for the transmitted pulse to travel through the wall, reflect from the object and back to the receiver. The detection of the transmitted pulse was done via cross-correlation. Hilbert transform was then used to obtain the envelope of the cross-correlation result. Finally, simple peak detection was used to determine the point where the cross-correlation had the highest value, thus obtaining the travel time of the transmitted pulse.

5.6.1. **Cross-correlation**

Assuming two real signals, x and y , both of length N . Then the cross-correlation sequence, R_{xy} , of the two signals is given as:

$$R_{xy}(m) = \sum_{n=0}^{N-m-1} x_{n-m} y_n \quad (37)$$

Chapter 5 Image Reconstruction

Figure 5-10 shows the received signal, the original transmitted pulse as well as the cross-correlation result of the two signals. The peaks of the cross-correlation result represent the points where the received signal closely resembles the transmitted pulse.

Figure 5-10 (a) Signal from receiver (b) transmitted pulse and (c) cross-correlation result.

Chapter 5 Image Reconstruction

5.6.2. Hilbert Transform

The correlation results contains a rapidly oscillating component. By using the Hilbert transform [33], the rapid oscillations were removed to produce the envelope of the correlation results. The frequency domain approach to implementing the Hilbert transform involves the following steps:

- 1) Assume the correlation results, $x[n]$, consists of N real data samples.
- 2) The Discrete Fourier Transform was then applied on the signal to obtain $X[N]$.
- 3) Form the transform domain analytic signal as follows

$$Z[M] = \begin{cases} X[0], & \text{for } M = 0 \\ 2X[M], & \text{for } 1 \leq M \leq \frac{N}{2} - 1 \\ X\left[\frac{N}{2}\right], & \text{for } M = \frac{N}{2} \\ 0, & \text{for } \frac{N}{2} + 1 \leq M \leq N - 1 \end{cases} \quad (38)$$

- 4) By taking the Inverse Fourier Transform of $Z[M]$, the time domain analytic signal, $z[m]$, was obtained. It should be noted that $z[m]$ is a complex-valued signal.
- 5) Finally, the envelope was obtained by taking the magnitude of $z[m]$.

Figure 5-11 Envelop detection using Hilbert transform.

Chapter 5 Image Reconstruction

Figure 5-11 shows the envelope of the correlation results by applying the Hilbert transform. Thus the time instance whereby the pulse was detected can be determined by simply detecting the peaks of the envelopes detected using the Hilbert transform.

5.6.3. Result of Pulse Detection

Having performed cross-correlation and Hilbert transform, the time-step in which the transmitted pulse was detected can thus be determined. Figure 5-12 shows the results. The red markers represent the time-step of the detected pulse. The presence of the three pulses have all been detected and their corresponding time-steps are indicative of the start of the pulse.

Figure 5-12 Time-step in which transmitted pulse was detected in the received signal.

5.7. Summary

The use of normal moveout (NMO) correction in image reconstruction was discussed in this chapter. The use of a velocity search was also highlighted in solving the problem of the unknown wall velocity when applying the NMO equation. Finally, the signals from the multiple receivers were coherently summed together or *stacked* to form the final

Chapter 5 Image Reconstruction

reconstructed image. In addition, a combination of cross-correlation, Hilbert transform and peak detection was used to detect the time-step of the transmitted pulse.

Chapter 6

Results for Image Reconstruction

6.1. Introduction

In this chapter, normal moveout (NMO) and stacking will be applied on the received signals (generated using the FDTD simulation model) from the receiver array. Effects of noise on stacking will also be studied. Furthermore, a Gaussian smoothing filter will be used to compensate for the effects of noise. Different shapes of objects will be used during image reconstruction. These include a flat object, a cylindrical object and two partially overlapping flat objects.

6.2. Application of NMO Correction

Figure 6-1 shows the received signals from the receiver array for a single transmitter position. The effects of NMO can be seen clearly from the hyperbolic curves caused by the difference in distance between the transmitter and the array of receivers.

Figure 6-1 Received signals for a single transmitter position

Chapter 6 Results for Image Reconstruction

In the following sub-sections, NMO correction will be applied on the first and second layer of the wall as well as the first layer of the object.

6.2.1. First Layer of Wall

For the first layer of the wall, application of equation (34) is simple as $\bar{v}$ is simply the velocity in air. Thus, the velocity search and the application of Dix's formula need not be applied. The results after NMO correction on the first layer of the wall can be seen in figure 6-2. NMO correction has time-aligned the signals such that they now accurately represent the actual surface of the first layer of the wall.

Figure 6-2 NMO correction applied to first layer of wall

6.2.2. Second Layer of Wall

For the second layer of the wall, the RMS velocity, $\bar{v}$, will involve the velocity of the wall. Thus, there is a need for a velocity search and the application of Dix's formula. The results after NMO correction on the second layer of the wall can be seen in figure 6-3.

Chapter 6 Results for Image Reconstruction

Figure 6-3 NMO correction applied to second layer of wall

6.2.3. First Layer of Object

Based on the wall velocity calculated from the previous section, Dix's formula can then be applied to determine $\bar{v}$ by assuming that the material medium between the object and the wall is air. Figure 6-4 shows the results after NMO correction. A comparison with the image before NMO correction (figure 6-1) will show that the signals nearer the left and right edges have been time-shifted significantly to match the shape of the flat iron object.

Figure 6-4 NMO correction applied to first layer of object

Chapter 6 Results for Image Reconstruction

6.3. Application of Stacking

After recording a set of readings for a particular transmitter position, the transmitter and receiver array were re-positioned to the adjacent cell in the simulation domain, and the signal acquisition process was repeated. A total of 200 recordings were taken at different transmitter positions. NMO correction was then applied to all the received signals and they were stacked appropriately.

Figure 6-5 (a) Image from a single transmitter and receiver pair (b) Improved image using an array of receivers and applying stacking. Note: Amplitude has been normalized to 1.

Chapter 6 Results for Image Reconstruction

Figure 6-6 Single receiver reading from (a) single transmitter and receiver pair and (b) array of receivers with stacking applied. Note: Amplitude has been normalized to 1.

Figure 6-5 (b) shows the results after stacking. For comparison, a single transmitter and receiver pair (see figure 5-1) was used to obtain single readings at the same 200 transmitter positions. The results from the single transmitter and receiver pair are shown in figure 6-5 (a) for comparison.

Chapter 6 Results for Image Reconstruction

As a result of stacking, the received signal amplitude has increased as can be seen from the clearer image of the first layer of the object (figure 6-5 (b)). Furthermore, some of the spurious reflections have been eliminated, resulting in a “cleaner” image of the object. Also, the actual width of the object (shown as dotted lines) shows good correlation with the final reconstructed object image.

In addition, notice the improved signal-to-noise ratio from the increase in signal amplitude of the object. This is illustrated more clearly in figure 6-6, which is a plot of the signal at the center receiver position. With stacking, the amplitude of the reflected signal (from the object) is approximately 8 times larger than the signal without stacking.

Chapter 6 Results for Image Reconstruction

Figure 6-7 Stacked signals (a) without noise and (b) noise with SNR 20 dB

6.4. Effect of Noise on Stacking

White Gaussian noise was added to the received signals before stacking was applied. The purpose was to investigate the effects of noise on the amplitude of the reflected signal after stacking. Figure 6-7 (a) shows the reconstructed image after stacking with no noise added. Figure 6-7 (b) shows the reconstructed image of the original signal with SNR of 20 dB. Figure 6-8 shows the corresponding plots of a single stacked signal. The stacked

Chapter 6 Results for Image Reconstruction

signal with 20 dB noise is approximately 35% smaller in amplitude than the stacked signal without any noise added.

Figure 6-8 Single stacked signal (a) without noise and (b) noise with SNR 20 dB

The effect of noise on stacking can be explained as follows. The ability to apply NMO correction is dependent on the ability to accurately detect the time-step of the reflected pulse. With the addition of noise, the time-step of some of the reflected pulses will contain errors, and subsequent application of NMO correction will produce inaccurate time shifts. As a result, during stacking, some of the signals will not be

Chapter 6 Results for Image Reconstruction

coherently added together and may even cancel each other out, thus producing a stacked signal of smaller amplitude.

6.5. Application of Noise Filter on Stacking

To improve stacking results due to the introduction of noise, a 2-D Gaussian smoothing filter was applied. The function of the 2-D Gaussian filter is of the form

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{39}$$

where σ is the standard deviation. An example of a 5x5 Gaussian filter is shown below.

0.004	0.015	0.026	0.015	0.004
0.015	0.059	0.095	0.059	0.015
0.026	0.095	0.150	0.095	0.026
0.015	0.059	0.095	0.059	0.015
0.004	0.015	0.026	0.015	0.004

The filter was then applied onto the image by convolving the filter with the image pixels. In the actual implementation, a 5x5 Gaussian smoothing filter was applied to the noisy reconstructed image prior to NMO correction and stacking. Figure 6-9 shows the gain in amplitude after applying the Gaussian smoothing filter of different sizes. Ideally, a filter size of 11x11 should be chosen as it provides the highest amplitude gain. However, the 5x5 filter was chosen instead as it reduced computational time due to its small size while still maintaining a reasonable amplitude gain (dotted line represents the *mean* of the amplitude gain).

Chapter 6 Results for Image Reconstruction

Figure 6-9 Effect of filter size on amplitude gain.

Figure 6-10 (a) and (b) show the results without and with filtering, respectively. Figure 6-11 shows the corresponding single receiver signal result. The amplitude of the reflected pulse with filtering applied is approximately 20% larger than that without filter.

Chapter 6 Results for Image Reconstruction

Figure 6-10 Stacked signals (20 dB noise added) (a) without filtering and (b) with filtering applied.

Chapter 6 Results for Image Reconstruction

Figure 6-11 Single stacked signal (20 dB noise added) (a) without filtering and (b) with filtering applied.

As discussed in the previous section, with the use of a noise filter, the reflected pulses can be more accurately detected. As a result, the multiple signals will be aligned accurately during NMO correction and subsequent stacking will add the signals coherently, thus increasing the signal amplitude.

Varying amounts of noise (SNR of 10 to 45 dB) were added and stacking was performed for both cases of with and without filter. The results are shown in figure 6-12. There was an increase in the received signal amplitude when the noise filter was

Chapter 6 Results for Image Reconstruction

implemented for noise levels up to SNR 30 dB. Above SNR 30 dB, the application of the noise filter no longer showed significant improvement.

Figure 6-12 Improvement in the received signal amplitude with the application of filter for various noise levels.

6.6. Image Reconstruction on Objects of Different Shapes

Sections 6.2 and 6.3 have shown the application of NMO correction and stacking on image reconstruction for a flat iron object. NMO correction and stacking was then applied on objects of different shapes and position. Two other configurations were used. They were a cylindrical object and two flat objects partially overlapping each other. In both configurations, the objects were made of iron.

6.6.1. Cylindrical Object

Figure 6-13 shows the simulation layout where a cylindrical object was placed behind the wall.

Chapter 6 Results for Image Reconstruction

Figure 6-13 Layout of simulation domain with cylindrical object.

Figure 6-14 Received signal from a single transmitter position.

Figure 6-14 above shows the received signals from the receiver array for a single transmitter position. Notice that the curvature of the first layer of the cylindrical object is much greater as compared to that for a flat object (see figure 6-1). This is expected as the cylindrical object acts like a point scatterer and reflects the transmitted electromagnetic wave in all directions. On the other hand, the flat object only reflects the transmitted wave back in the direction perpendicular to its surface.

Chapter 6 Results for Image Reconstruction

Normal moveout correction was performed and the results are shown in figure 6-15. The curved shape of the cylindrical object is visible.

Figure 6-15 NMO correction applied to first layer of cylindrical object.

Stacking was then applied to the numerous received signals for different transmitter positions. The results after stacking are shown in figure 6-16 (b). As a comparison, the reconstructed image using a single transmitter and receiver pair is shown in figure 6-16 (a). The curved shape of the cylindrical object can be seen in the stacked image. However, the exact shape of the cylindrical object could not be obtained. The problem lies in the fact that when the transmitted signal impinges on a round object, the object acts as point source and scatters the signal in all directions. As a result, the curvature of the object in the reconstructed image will always be greater than that of the actual object. In the field of seismology, the same result is obtained due to the presence of an *anticline* and a technique known as *migration* [34] is used to correct the problem. However, migration is beyond the scope of this research and will not be discussed.

Chapter 6 Results for Image Reconstruction

Figure 6-16 (a) Image from a single transmitter and receiver pair (b) Improved image using an array of receivers and applying stacking.

In addition, stacking has improved the clarity of the reconstructed image. This is more clearly shown in figure 6-17, where the amplitude of the received signal for the object with stacking applied is approximately 7 times larger than that without stacking.

Chapter 6 Results for Image Reconstruction

Figure 6-17 Single receiver reading from (a) single transmitter and receiver pair and (b) array of receivers with stacking applied. Note: Amplitude has been normalized to 1.

6.6.2. Two Partially Overlapped Flat Objects

Figure 6-18 shows the simulation layout and the position of the two partially overlapped flat objects behind the wall. Figure 6-19 shows the received signals for a single transmitter position. Two distinct reflected signals can be seen, which corresponds to the two overlapping objects respectively.

Chapter 6 Results for Image Reconstruction

Figure 6-18 Layout of simulation domain with two partially overlapped flat objects.

Figure 6-19 Received signal from a single transmitter position.

NMO correction was then applied to the received signals and the results are shown in figure 6-20 below. Comparing figure 6-20 with figure 6-19, it can be seen that the left and right edges of each of the two objects have been time-shifted, thus changing the shape of the object from hyperbolic (caused by the time-delay) to a flat one.

Chapter 6 Results for Image Reconstruction

Figure 6-20 NMO correction applied to first layer of the two partially overlapped flat objects.

Figure 6-21 (a) Image from a single transmitter and receiver pair (b) Improved image using an array of receivers and applying stacking.

Chapter 6 Results for Image Reconstruction

The results after stacking was applied are shown in figure 6-21 (b). The white dotted lines represent the actual width of the two partially overlapped objects. From the figure, it can be seen that the reconstructed image correlates well with the actual positions of the objects. Figure 6-21 (a) shows the receiver signals from a single transmitter and receiver pair. Comparing figures 6-21 (a) and (b), it can be seen that stacking has improved the image reconstruction significantly.

Figure 6-22 Single receiver reading from (a) single transmitter and receiver pair and (b) array of receivers with stacking applied. Note: Amplitude has been normalized to 1.

Chapter 6 Results for Image Reconstruction

Figure 6-22 shows the corresponding signals from a single receiver. The amplitude of the received signal for the object with stacking applied is approximately 7 times larger than that without stacking.

6.7. *Summary*

Normal moveout (NMO) was applied to the first layer of the wall, second layer of the wall and the first layer of the object to illustrate the effects of NMO correction on the time-delayed received signals. Then, stacking was performed on the NMO corrected signals and it was shown that stacking actually increased the received signal amplitude by 8 times and spurious reflections were also reduced in the process.

Noise (of SNR 10 to 45 dB) was added to study its effects on stacking. Subsequently, a Gaussian smoothing filter was applied to reduce the effects of noise on the performance of stacking. It was shown that below SNR of 30 dB, the noise filter helped increase the signal amplitude by approximately 20%. However, above SNR 30 dB, the noise filter did not improve the signal amplitude significantly.

Finally, a cylindrical object and two partially overlapping objects were used to investigate the effectiveness of the image reconstruction. The shape of the cylindrical object could not be reconstructed due to the fact that the cylindrical object acted as a point scatterer. On the other hand, the reconstructed image of the two partially overlapping objects showed good correlation with the actual size and shape of the actual objects.

Chapter 7

Signal Classification and Identification

7.1. Introduction

This chapter investigates the classification and identification of objects behind a wall based on their material composition. The materials are defined based on their permittivity, permeability and conductivity and include wood, glass and iron.

Prony's method [23-25] was used to decompose the transient signal into a sum of decaying sinusoids. These sinusoids are represented by their poles and residues. Analysis of the poles and residues was then carried out with the aim of classifying them based on the material medium of the object.

It is known that the pole positions tend to become unstable and will shift when there is noise. Therefore, an investigation was also carried out on the effect of noise (with SNR from 20 to 40 dB) on the pole positions. Singular value decomposition [26] was also used to enhance the pole stability in the presence of noise.

Lastly, Fisher's Linear Discriminant (FLD) was used to classify the dominant poles based on their frequency and residue components.

7.2. Prony's Method

Prony's method decomposes a time signal into decaying exponential functions. By Prony's method, the signal $f(t)$ can be estimated as follows,

$$f(t) = \sum_{v=1}^N R_v e^{S_v t} \quad (40)$$

where N is the number of poles and residues, R_v are the residues and S_v are the poles.

Chapter 7 Signal Classification and Identification

It can be shown that $f(t)$ satisfies the difference equation

$$\sum_{p=0}^n a_p f_{p+k} = 0, \quad k = 0, \dots, N-1 \quad (41)$$

Since $a_n = 1$, equation (41) can be written as

$$\begin{bmatrix} f_0 & f_1 & \cdots & f_{n-1} \\ f_1 & f_2 & \cdots & f_n \\ \vdots & \vdots & & \vdots \\ f_{N-1} & f_{N-2} & \cdots & f_{N-1+n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = - \begin{bmatrix} f_n \\ f_{n+1} \\ \vdots \\ f_{N-1+n} \end{bmatrix} \quad (42)$$

or more compactly as

$$\mathbf{B}\mathbf{a} = \mathbf{C} \quad (43)$$

Note that the size of the above matrix equation will depend on the length of the sampled signal as well as the number of poles and residues used. For example, if the signal consisted of 1000 discrete values and 200 poles and residues were used, then the matrix will be of size 500 by 200.

From equation (39), a_0 to a_n will be known where a_p are the coefficients of the algebraic equation, expressed as follows:

$$\sum_{p=0}^N a_p z^p = 0 \quad (44)$$

Solving the polynomial equation (41), the poles can thus be obtained by

$$S_v = \frac{1}{\Delta t} \ln z_v \quad (45)$$

Thus, R_v can be found from equation (37).

A more detailed derivation [35] of the above equations can be found in Appendix A.

Chapter 7 Signal Classification and Identification

7.3. *Singular Value Decomposition (SVD)*

Application of Prony's algorithm requires prior knowledge of the number of poles in the sampled data [24]. If lesser poles are used to estimate the data than the exact number, then the resulting poles will probably be inaccurate. If more poles are used than are contained in the sampled data, then the extra poles will cause the residues of the true poles to be inaccurate and also result in excess in computation time.

Furthermore, when noise is added, the matrix $\mathbf{B}$ in equation (43) will take on its full rank regardless of the actual number of poles it contains. This results in the inaccurate estimation of the true poles.

SVD [26] can be used to find the best lower rank approximation to the matrix $\mathbf{B}$, thus reducing the effects of noise on the shift in the pole positions.

Using SVD, the matrix $\mathbf{B}$ can be decomposed as

$$\mathbf{B} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^* \quad (46)$$

where $\mathbf{\Sigma}$ is a diagonal matrix with non-negative values and $\mathbf{U}$ and $\mathbf{V}^*$ are unitary matrices.

The diagonal entries of $\mathbf{\Sigma}$ are called the singular values of $\mathbf{B}$. In the noiseless case, there will be N number of non-negative values corresponding to the N number of true poles and the rest of the singular values will be zero. With the addition of noise, the previously zero values will now have small non-zero values, thus increasing the rank of $\mathbf{B}$.

Normally, the small non-zero singular values caused by the noise are generally a few magnitudes smaller than the other N singular values due to the actual poles. Thus, to correct for the addition of noise, simply replace $\mathbf{B}$ with

$$\hat{\mathbf{B}} = \mathbf{U} \hat{\mathbf{\Sigma}} \mathbf{V}^* \quad (47)$$

Chapter 7 Signal Classification and Identification

where $\hat{\Sigma}$ is obtained from Σ by setting to zero all singular values which are several magnitudes below the largest value.

7.4. Fisher Linear Discriminant (FLD)

Given two classes of data (in two-dimensional space), FLD [36] will find a line, such that the data samples, when projected onto this line, are well separated.

Let $x_{11}, x_{12}, \dots, x_{1n}$ and $x_{21}, x_{22}, \dots, x_{2n}$ be samples of two-dimensional data belonging to the first and second class respectively. Let the projection of the mean of the first class onto a direction vector, $\mathbf{v}$, be defined as

$$\tilde{\mu}_1 = \mathbf{v}^T \boldsymbol{\mu}_1 = \mathbf{v}^T \frac{1}{n} \sum_{i=1}^n \mathbf{x}_{1i} \quad (48)$$

Let the projection of the sample variance of the first class onto a direction vector, $\mathbf{v}$, be defined as

$$\tilde{s}_1^2 = \sum_{i=1}^n (\mathbf{v}^T \mathbf{x}_{1i} - \tilde{\mu}_1)^2 \quad (49)$$

The projection of the mean and variance of the second class are defined similarly as

$$\tilde{\mu}_2 = \mathbf{v}^T \boldsymbol{\mu}_2 = \mathbf{v}^T \frac{1}{n} \sum_{i=1}^n \mathbf{x}_{2i} \quad (50)$$

$$\tilde{s}_2^2 = \sum_{i=1}^n (\mathbf{v}^T \mathbf{x}_{2i} - \tilde{\mu}_2)^2 \quad (51)$$

Thus, FLD finds a line with direction vector, $\mathbf{v}$, which maximises the function

$$J(\mathbf{v}) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \quad (52)$$

Chapter 7 Signal Classification and Identification

In essence, FLD determines a projection line which maximizes separation between projected class means and at the same time, minimizes the variance within each class. Finally, the projection vector, $\mathbf{v}$, can be found by solving the generalized eigenvalue problem

$$\mathbf{S}_B \mathbf{v} = \lambda \mathbf{S}_W \mathbf{v} \quad (53)$$

where

$$\mathbf{S}_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \quad (54)$$

$$\mathbf{S}_W = \sum_{i=1}^n (\mathbf{x}_{1i} - \boldsymbol{\mu}_1)(\mathbf{x}_{1i} - \boldsymbol{\mu}_1)' + \sum_{i=1}^n (\mathbf{x}_{2i} - \boldsymbol{\mu}_2)(\mathbf{x}_{2i} - \boldsymbol{\mu}_2)' \quad (55)$$

Thus, it has been shown how data from two classes can be effectively separated and classified by projecting them only a line. The use of FLD to aid in the classification of the dominant poles of two different materials, will be highlighted in the next chapter.

7.5. Summary

Prony's method was used to decompose a signal into its poles and residues. In addition, the use of singular value decomposition (SVD) was highlighted to reduce the effects of noise on the poles and residues of the signal. Fisher's linear discriminant (FLD) was used to separate two classes of data.

Chapter 8

Results for Signal Classification and Identification

8.1. *Introduction*

In this chapter, Prony's method will be applied to the received signals (generated using the FDTD simulation) in order to classify the signals based on their poles and residues. An investigation will be carried out on the factors which affect the accuracy of the poles and residues obtained using Prony's method. These factors include the initial estimate on the number of poles and residues as well as noise. Singular value decomposition (SVD) will also be used to reduce the effects of noise on the accuracy of the poles and residues.

Invariant or dominant poles will first be found for an object made of iron. Classification and identification of the iron object will be based on its dominant poles. Subsequently, classification and identification of other object materials (such as glass and wood) will also be studied.

8.2. *Number of Poles and Residues used to estimate a Signal*

In applying the Prony's method, the number of sampled data points must be equal to or greater than twice the number of poles and residues (or parameters) needed to represent the signal. Often, the number of sampled data points exceeds the number of parameters required. However, it is not always possible to know in advance how many parameters are needed to represent the signal.

If an excessive number of initial parameters are used, the original signal can be precisely reconstructed with minimum error. However, with a large number of parameters, the computational time required also increases dramatically. On the other

Chapter 8 Results for Signal Classification and Identification

hand, underestimating the number of parameters will result in greater errors when reconstructing the original signal.

Figure 8-1 (a) shows the mean-squared error (MSE) of the reconstructed signal when different numbers of initial parameters are used. It can be seen that if more than 150 parameters are used, the MSE is small, and thus the parameters accurately represent the original signal. As a result, the original signal can be decomposed with just 150 parameters. If less than 150 are used, then the MSE increases, indicating that the parameters obtained do not accurately represent the parameters of the original signal.

Figure 8-1 (b) shows the original signal used, which is a time-gated portion of the reflected signal from an iron object which was behind a gypsum wall (see figure 4-1). Figure 8-1 (c) shows the reconstructed signal when 170 parameters are used. It can be seen that the original signal has been perfectly reconstructed. Figure 8-1 (d) shows the reconstructed signal when only 130 parameters are used. Due to the insufficient number of poles and residues, the original signal cannot be perfectly reconstructed, thus resulting in the high MSE.

When noise is added to the signal, (i.e. low SNR), the number of poles and residues needed increases. As a result, a higher initial parameter estimate needs to be used. This is shown in figure 8-2, where different quality measures were used and are defined as

$$\text{Mean Square Error} = \frac{\sum_{i=1}^n (x_i - y_i)^2}{n}$$

$$\text{Mean Absolute Error} = \frac{\sum_{i=1}^n |x_i - y_i|}{n}$$

Chapter 8 Results for Signal Classification and Identification

Figure 8-1 (a) Mean squared error of reconstructed signal using different initial number of pole and residue estimates (b) Time-gated portion of the reflected pulse from the object. (c) Reconstructed signal based on 170 poles and residues. (d) Reconstructed signal based on 130 poles and residues

Chapter 8 Results for Signal Classification and Identification

Figure 8-2 (a) Mean squared error (b) mean absolute error and (c) mean absolute percentage error, versus different initial pole and residue estimates for different noise levels

Chapter 8 Results for Signal Classification and Identification

$$\text{Mean Absolute Percentage Error} = \frac{\sum_{i=1}^n |(x_i - y_i) / x_i|}{n}$$

In figure 8-2(a), for a signal with SNR of 40 dB, approximately 200 or more poles and residues are needed to accurately reconstruct the signal, and for SNR of 20 dB, approximately 230 parameters are needed. Using different quality measures, as shown in figure 8-2(b) and 8-2(c), a similar observation can be made.

8.3. *Effect of Noise on Residue Amplitude*

The previous section showed how the initial choice on the number of parameters used will affect the accuracy of the poles and residues. Also, the addition of noise increases the number of parameters needed to represent the signal.

Alternatively, this can be viewed from the residue (or magnitude) of the poles. Prony's method was applied to the signal (figure 8-1 (b)) using an initial estimate of 300 parameters. In general, the maximum number of poles and residues was used. For example, given a signal consisting of 600 discrete samples, the maximum number of poles and residues is half the number of signal samples (i.e. 300).

The poles were sorted based on the amplitude of their residues and 80 of the residues with the highest amplitude have been plotted in figure 8-3 (a) below. It is evident that the first 20 poles have residues which are several magnitudes greater than the rest. These constitute the more significant poles of the signal.

In figure 8-3 (b), noise was added to the signal and the top 80 residues were plotted. It is no longer possible to distinguish the significant poles easily. The addition of noise has increased the residue amplitude of many of the poles, which accounts for the increased number of parameters needed to reconstruct the signal with minimum error.

Chapter 8 Results for Signal Classification and Identification

Figure 8-3 Residue values of 80 poles from a signal (a) without noise (b) with SNR of 20 dB

8.4. Effect of Noise on Reconstructed Signal

Given a signal as illustrated in figure 8-4 (a), the objective is to decompose the signal using Prony's method into its constituent poles and residues such that the reconstructed signal will have minimum error from the original signal. However, as illustrated in figure 8-4 (b), the original signal will contain some amount of noise. Prony's method was applied to this noisy signal to obtain 300 poles and residues and the reconstructed signal based on these 300 parameters is shown in figure 8-4 (c).

The 300 parameters were obtained by decomposing the noisy signal and thus, the

Chapter 8 Results for Signal Classification and Identification

reconstructed signal will naturally be an estimate of the noisy signal rather than the original signal. From section 8.3, it can be seen that more poles and residues are needed as a result of the presence of noise and that the noise has “masked out” the significant poles of the original signal.

Figure 8-4 (a) Signal with no noise (b) Signal with SNR 20 dB (c) Reconstructed signal based on 300 poles and residues.

Chapter 8 Results for Signal Classification and Identification

Figure 8-5 (a) Mean-squared error (b) Mean absolute error and (c) Mean absolute percentage error between reconstructed signal and original signal (with no noise) with varying amounts of noise added.

Chapter 8 Results for Signal Classification and Identification

Figure 8-5(a) shows the mean-square error between the original signal and the reconstructed signal with increasing amounts of noise present. The mean absolute error and mean absolute percentage error between the original signal and reconstructed signal were also calculated and shown in figures 8-5(b) and 8-5(c) respectively. In all the three quality measures, it can be seen that with more noise, there exists a greater error between the reconstructed signal and the original signal. The next section will illustrate the use of singular value decomposition (SVD) in reducing this error.

8.5. *Improvement using SVD*

As stated in section 8.3, given an original signal (with no noise), the number of non-zero singular values indicates the number of true poles and residues of a signal. Addition of noise to the original signal will cause the originally non-zero singular values to take on small amplitudes. This can be seen in figure 8-6 and figure 8-7 below.

The left diagram in figure 8-6 shows the singular values and the diagram on the right is a magnified view of its amplitude. Only the first 40 singular values have non-zero amplitude values. In other words, the original signal consists of about 40 true poles and residues.

Figure 8-7 shows the magnitude of the singular values for the signal with SNR of 20 dB. The previously zero values have now attained small amplitudes due to the addition of noise.

Chapter 8 Results for Signal Classification and Identification

Figure 8-6 Singular values of signal with no noise.

Figure 8-7 Singular values of signal with SNR of 20 dB.

By keeping the singular values with the highest amplitudes and setting the rest to zero, the mean-squared error between the reconstructed signal and the original signal can be reduced. Figure 8-8 (a) shows the reconstructed signal without applying SVD whereas figure 8-8 (b) shows the reconstructed signal after applying SVD. With the use of SVD, the effects of noise on the reconstructed signal have been reduced and thus the reconstructed signal more closely resembles the original signal.

Chapter 8 Results for Signal Classification and Identification

Figure 8-8 Reconstructed signal (a) without and (b) with the application of SVD.

Figure 8-9(a) shows the improvement in the mean-square error between the reconstructed signal and the original signal (without noise) with varying amounts of noise added. At SNR of 20 dB, there is a 72% improvement in the mean-square error and at SNR of 30 dB, there is a 37% improvement. Above SNR of 34 dB, the improvement in mean-square error is no longer significant and therefore, use of SVD will not improve the signal estimation. In figures 8-9(b) and 8-9(c), where the mean absolute error and mean absolute percentage error were measured, a similar trend can be observed. At low SNR, use of SVD will improve the reconstructed signal, however, at higher SNR, use of SVD will no longer be effective in improving the reconstructed signal.

Chapter 8 Results for Signal Classification and Identification

Figure 8-9 Improvement in the (a) mean-squared error (b) mean absolute error and (c) mean absolute percentage error, with the use of SVD for varying amounts of noise

Chapter 8 Results for Signal Classification and Identification

8.6. *Dominant Poles*

The poles and residues represent the characteristics or the “signature” which uniquely identifies the given signal. As such, it is desirable that these characteristics remain unchanged. Unfortunately, when the signal has been corrupted by certain amounts of noise, it is inevitable that the residue amplitudes and the pole values will vary from those of the original.

However, for a given signal, there are poles which do not vary drastically in position even when the signal has been corrupted by noise. Such poles are known as dominant poles. Classification of a signal will thus be based on such dominant poles.

8.6.1. **Residue vs. Frequency**

The parameters (poles and residues) were extracted from the original received signal of an iron object (figure 8-4 (a)). Subsequently, noise of SNR 40 dB and 20 dB was added to the original signal and the parameters were also extracted from these signals. A plot of 15 of the poles with the highest residues from each signal is shown in figure 8-10 (a).

Out of the 15 poles with the highest residues, only 4 can be considered as dominant. These are shown in figure 8-10 (b). The frequencies of these dominant poles remain approximately the same despite the corruption of the signal by noise. The dominant poles are selected by first determining the frequency interval as

$$\text{freq_interval} = \frac{\text{MAX}(\text{freq}) - \text{MIN}(\text{freq})}{1000} \quad (56)$$

where MAX(freq) and MIN(freq) represents the maximum and minimum frequency values of all the poles of the three signals. Only if the frequency of the poles from the three signals (SNR 40, SNR 20 and No noise) all fall within the *freq_interval* range, will the poles be considered dominant.

Chapter 8 Results for Signal Classification and Identification

Figure 8-10 (a) Position of poles with high residues and (b) dominant poles.

8.6.2. Attenuation vs. Frequency

The previous section showed how a signal can be classified by their dominant poles which had high residue and similar frequency components. However, the attenuation component of the poles also plays an important role in the classification of the signal. In this section, we will look at the classification of a signal according to its dominant poles based on their attenuation and frequency components.

The poles and residues of the original signal, signal with SNR 40 dB and signal with SNR 20 dB were extracted using Prony’s method. The frequency and attenuation

Chapter 8 Results for Signal Classification and Identification

components of the poles are plotted and shown in figure 8-11 (a) below. A corresponding magnified plot of the figure is shown in figure 8-11 (b).

Figure 8-11 (a) Pole positions of signals with varying amounts of noise (b) magnified view.

Dominant poles were then found having attenuation and frequency values very close to each other. As such, the frequency and attenuation values did not shift despite the addition of noise to the signal. The results are shown in figure 8-12 (a) and magnified in figure 8-12 (b). Notice that the frequency of the dominant poles found fall in the range of 4-5 GHz. This can be expected as the original transmitted pulse was of 4 GHz center frequency.

Chapter 8 Results for Signal Classification and Identification

Figure 8-12 (a) Dominant poles (b) magnified view.

8.7. Dominant Poles of Objects of Different Material

The dominant poles obtained in section 8.6 were for the case whereby the object behind the wall was made of iron. Received signals from objects of different materials will now be used to obtain their dominant poles.

Figure 8-13 shows the received signals for 3 objects of different materials behind a gypsum wall. The object materials are iron, wood and glass and their material properties are shown in table 8-1. The signals shown in figure 8-13 have been time-gated and thus correspond to the portion where the transmitted pulse reflected from the object.

Chapter 8 Results for Signal Classification and Identification

From figure 8-13, the received signal for the metal object has the highest amplitude due to the high reflectivity of metal. For the glass object, there is a second reflected signal. This is due to the transmitted pulse passing through the first surface of the object and then reflecting off the second surface of the object.

	permittivity, ϵ	conductivity, σ
iron	1.0	2e9
glass	6	1e-9
wood	2.5	1.18

Table 8-1 Material properties of the different objects used

Figure 8-13 Received signal for different object materials

Section 8.6.1 showed how the dominant poles of an iron object were extracted based on their residue and frequency components. In figure 8-14, the dominant poles for the iron object is compared with those of the glass and wood object.

Chapter 8 Results for Signal Classification and Identification

Figure 8-14 Dominant poles of received signal from objects of different material.

All the dominant poles fall within the 3 - 5 GHz range of frequencies. However, there is distinction in terms of their spread across the frequency range as well as their residue amplitudes. The dominant poles of the metal object are spread evenly over the 3-5 GHz range and their residue amplitudes are higher as compared to the dominant poles for wood and glass.

The dominant poles of the glass object have low residue amplitudes and they fall within the 3.5 – 4.5 GHz frequency range. Finally, for the wood object, the dominant poles fall within the 4 – 5 GHz frequency range and its residue amplitudes fall within the intermediate range.

Classification of dominant poles based on their attenuation and frequency components was shown in section 8.6.2. Figure 8-15 shows a comparison of the dominant poles for the 3 different object materials, iron, glass and wood.

The dominant poles of the iron object are well clustered together and clearly distinct from the dominant poles of the glass and wood object. As such, the identification of an iron object can be made without ambiguity.

Chapter 8 Results for Signal Classification and Identification

Figure 8-15 Dominant poles of received signal from objects of different material.

One of the dominant poles of the glass object falls within the same location of the dominant poles of the wood object. Therefore, identification of the wood object can no longer be made distinctly. However, classification is still possible with the use of Fisher's linear discriminant (FLD) as described in section 7.4. Figure 8-16 shows the projection of the classes onto a line which aids in separating the classes.

Figure 8-16 Use of Fisher linear discriminant to classify the different materials.

Chapter 8 Results for Signal Classification and Identification

8.8. *Summary*

Prony's method was applied to decompose the received signal and represent it as poles and residues. It was shown that the initial estimate of the number of poles and residues used in Prony's method was an important factor. Care must be taken in selecting these values as underestimating them would result in error in the final reconstructed signal.

Addition of noise caused the reconstructed signal to have high mean-square error (MSE) from the original signal. Singular value decomposition (SVD) was used to reduce the MSE. It was shown that with the use of SVD, a 37% reduction in the MSE could be obtained for a signal with SNR of 30 dB.

Finally, dominant poles were obtained for signals of three types of materials, namely, wood, glass and iron. Classification of the signals was based on the pole positions as well as the amplitude of the residues. It was shown that the dominant poles of the iron object were unique from those of wood and glass object. As such, an iron object can be distinctly identified from a glass or wood object. On the other hand, two of the dominant poles of the wood and glass object overlapped each other. However, identification and classification was still possible based on the remaining non-intersecting poles. Use of Fisher linear discriminant also aided in separating the classes more distinctly, resulting in better classification.

Chapter 9

Conclusion and Future Direction

The aim of this research was to investigate through-wall imaging and classification. A simulation model was designed and implemented based on the finite difference time domain (FDTD) method. The model was then used to simulate an UWB signal transmitting through a gypsum wall and reflecting off an object. The received signals were then recorded for post-processing purposes.

Image reconstruction was implemented by first time-aligning the recorded signals using normal moveout (NMO) correction. The problem of applying the NMO equation due to the unknown velocity of the wall was highlighted. A *velocity search* was performed to overcome this problem. Then, different sets of recordings were taken for different transmitter positions. These recordings were *stacked* to improve the signal to noise ratio of the final reconstructed image. It was shown that with stacking, the signal amplitude improved by 20 times when 200 sets of signals were stacked. However, addition of noise of 20 dB was shown to reduce the stacked signal amplitude by 35%. A Gaussian smoothing filter was used to compensate for this effect.

Through-wall signal classification and identification was investigated by using Prony's method to decompose the reflected UWB signal into poles and residues. The choice on the initial number of poles and residues used to estimate the original signal was shown to be an important factor in the accuracy of the poles and residues. Addition of noise caused the reconstructed signal to have high mean-squared error (MSE). However, with the use of singular value decomposition (SVD), it was shown that a reduction of 37% in the MSE was possible for a SNR of 30 dB.

Chapter 9 Conclusion and Future Direction

Dominant poles and residues were then used to classify objects of different materials (iron, wood and glass). When using the residues and the frequency component of the poles, it was shown that the dominant characteristics of the 3 signals were separable and thus identification of the 3 different materials was possible. However, when using the attenuation and frequency components of the poles, some of the dominant characteristics of the wood and glass signals overlapped. As a result, the wood and glass material could not be uniquely identified.

9.1. *Future Work*

Currently, only a single UWB pulse was transmitted and received in the simulation model. A stream of time-modulated pulses might be used instead. By accurately correlating the transmitted and received time-modulated pulses, additional signal characteristics can be obtained.

As highlighted in section 6.6.1, the reconstructed image of a cylindrical object does not accurately represent the actual shape of the cylindrical object. This is because the cylindrical object acted as a point scatterer which scattered the electromagnetic waves in all direction. In the future, a technique known as migration will have to be applied in order to accurately reconstruct the image for a cylindrical object.

The effective use of singular value decomposition (SVD) is dependent on which singular-values are set to zero values. In most cases, the singular values attributed to noise are several magnitudes smaller than those corresponding to the true signal and thus, they can be identified easily. However, with the increase in noise level, this distinction might not be so obvious. In the future, an adaptive thresholding scheme will be developed to better identify the singular values attributed to noise.

The pole and residue values for all possible material types will have to be generated and stored in a database. The database will act as a look-up table to identify the

Chapter 9 Conclusion and Future Direction

type of material detected. When a signal is received, it can be decomposed and its poles and residues can then be searched through the database to find a possible match.

Lastly, the current 2D simulation can be extended to a 3D simulation. A 3D simulation will be extremely computational intensive (see section 4.4). However, the problem of computational cost can be dismissed, especially with ever increasing CPU speeds and efficient usage of computer processors using techniques such as grid computing. In the current 2D simulation, a problem lies in differentiating the direction of the reflected signal from the object during image reconstruction. This problem was partially alleviated with the use of stacking (section 5.5). With a 3D simulation, the problem will be made worse due to the ambiguity in the third dimension. Therefore, more readings at different transmitter positions will have to be taken to improve stacking.

Author's Publications

Author's Publications

- [1] Ho, A. T. S., **Tham, W. H.**, Low, K. S., "Improving Classification Accuracy in Through-Wall Radar Imaging Using Hybrid Prony's and Singular Value Decomposition Method," IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2005, 25-29 July 2005, Seoul, South Korea.
- [2] Ho, A. T. S., **Tham, W. H.**, Low, K. S., "Through-Wall Radar Image Reconstruction Based on Time-Domain Transient Signals in the Presence of Noise," IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2005, 25-29 July 2005, Seoul, South Korea.

References

References

- [1] Frazier, L. M., "Surveillance through walls and other opaque materials," IEEE Aerospace and Electronic Systems Magazine, vol. 11, pp. 6-9, 1996.
- [2] McMillan, R. W., Currie, N. C., Ferris, D. D., Jr. and Wicks, M. C., "Concealed weapon detection using microwave and millimeter wave sensors," 1998 International Conference on Microwave and Millimeter Wave Technology Proceedings, ICMMT '98. 18-20 Aug. 1998.
- [3] Nag, S., Fluhler, H. and Barnes, M., "Preliminary interferometric images of moving targets obtained using a time-modulated ultra-wide band through-wall penetration radar," Proceedings of the 2001 IEEE Radar Conference, pp. 64-69, 1-3 May 2001.
- [4] Nag, S. and Barnes, M., "A moving target detection filter for an ultra-wideband radar," Proceedings of the 2003 IEEE Radar Conference, pp. 147-153, 5-8 May 2003.
- [5] Hunt, A.R., "Image formation through walls using a distributed radar sensor array," Proceedings of 32nd Applied Imagery Pattern Recognition Workshop, 2003., pp. 232-237, 15-17 Oct. 2003.
- [6] Falconer, D. G., Ficklin, R.W. and Konolige, K. G., "Robot-mounted through-wall radar for detecting, locating, and identifying building occupants," IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA '00. vol. 2, pp.1868-1875, 24-28 April 2000.
- [7] Stratis, G. and Demetriou, D., "Numerical study of reflection and transmission coefficients for different inhomogeneous walls," IEEE Antennas and Propagation Society International Symposium, 1999. vol. 1, pp. 590-593, 11-16 July 1999.
- [8] Suzuki, H., "Accurate and efficient prediction of coverage map in an office environment using frustum ray tracing and in-situ penetration loss measurement," The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring. vol. 1, pp. 236-240, 22-25 April 2003.
- [9] de Coster, I., Madariaga, G.A., Souto, B.P., Van Lil, E. and Perez-Fontan, F., "An extended propagation software package for indoor communication systems," Tenth International Conference on (Conf. Publ. No. 436) Antennas and Propagation, vol. 2, pp.298-301, 14-17 April 1997.
- [10] Savov, S. V. and Herben, M. H. A. J., "Modal transmission-line modelling of radio wave propagation through periodic inhomogeneous walls," The Fourth International Conference on Computation in Electromagnetics, 2002. CEM 2002. 8-11 April 2002.
- [11] Vertiy, A. A., Gavrilov, S. P., Stepanyuk, V. N. and Voynovskyy, I. V., "Through-wall and wall microwave tomography imaging," IEEE Antennas and Propagation Society International Symposium, 2004. vol. 3, pp. 3087-3090, 20-25 June 2004.
- [12] Aryanfar, F. and Sarabandi, K., "Through wall imaging at microwave frequencies using space-time focusing," IEEE Antennas and Propagation Society International Symposium, 2004. vol. 3, pp. 3063-3066, 20-25 June 2004.
- [13] Attiya, A. M., Bayram, A., Safaai-Jazi, A. and Riad, S.M., "UWB applications for through-wall detection," IEEE Antennas and Propagation Society International Symposium, 2004. vol. 3, pp. 3079-3082, 20-25 June 2004.

References

- [14] Benjamin, R, Craddock, I. J., McCutcheon, E. and Nilavalan, R., "Through-wall imaging using real-aperture radar," Proc. URSI General Assembly, Aug. 2002.
- [15] Barrie, G., "Ultra-wideband synthetic aperture data and image processing," DRDC Ottawa, Technical Memorandum, 2003.
- [16] Bugaev, A. S., Chapursky, V. V., Ivashov, S. I., Razevig, V. V., Sheyko, A. P. and Vasilyev, I. A., "Through wall sensing of human breathing and heart beating by monochromatic radar," Proceedings of the Tenth International Conference on Ground Penetrating Radar, 2004. GPR 2004. vol 1, pp. 291-294, 2004.
- [17] Yee, K. S., "Numerical solutions of initial boundary value problems involving Maxwell's equations in isotropic media," IEEE Transactions on Antennas and Propagation, vol. 14, pp. 310-317, 1966.
- [18] Taflove, A. and Hagness, C., *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 2nd Edition, Artech House, 2000.
- [19] Montalbetti, J. F., "Computer determination of seismic velocities – a review," Journal of the Canadian Society of Exploration Geophysicists, vol. 7, pp. 32-45, 1971.
- [20] Gajewski, D. and Vanelle, C. , "Revisiting NMO again ?," EAGE 64th Conference and Technical Exhibition, 2002.
- [21] Kessinger, W., "Curved-ray time migration can improve seismic imaging," Oil and Gas Journal, Oct. 2002.
- [22] Boyce, J. F., "Seismic processing and image analysis," Journal of Physics D: Applied Physics, vol. 19, pp. 397-415, 1986.
- [23] Poggio, A., VanBlaricum, M., Miller, E. and Mittra, R., "Evaluation of a processing technique for transient data," IEEE Transactions on Antennas and Propagation, vol. 26, pp. 165-173, 1978.
- [24] Kumaresan, R. and Tufts, D., "Estimating the parameters of exponentially damped sinusoids and pole-zero modeling in noise," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 30, pp. 833-840, 1982.
- [25] VanBlaricum, M. and Mittra, R., "Problems and Solutions Associated with Prony's Method for Processing Transient Data", IEEE Transactions on Antennas and Propagation, vol. 26, pp. 174-182, 1978.
- [26] Tufts, D. and Kumaresan, R., "Singular Value Decomposition and Improved Frequency Estimation Using Linear Prediction", IEEE Transactions on Acoustics, Speech and Signal Processing, vol 30, pp. 671-675, 1982.
- [27] Taflove, A. and Brodwin, M. E., "Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations," IEEE Transactions on Microwave Theory and Techniques, vol. 23, pp. 623-630, 1975.
- [28] Luebbers, R. J., Kunz, K. S. and Chamberlain, K. A., "An interactive demonstration of electromagnetic wave propagation using time-domain finite differences," IEEE Transactions on Education, vol. 33, pp. 60-68, 1990.
- [29] Berenger, J. P., "A Perfectly Matched Layer for the Absorption of Electromagnetic Waves," Journal of Computational Physics, vol. 114, pp. 185-200, 1994.
- [30] Berenger, J. P., "Perfectly matched layer for the FDTD solution of wave-structure interaction problems," IEEE Transactions on Antennas and Propagation, vol. 44,

References

- pp. 110-117, 1996.
- [31] Mur, G., "Absorbing boundary conditions for finite-difference approximation of the time-domain electromagnetic-field equations," IEEE Transactions on Electromagnetic Comptability, vol. 23, pp. 1073-1077, 1981.
 - [32] Hoffmann, G., "Pattern recognition of multiple signals from ground penetrating radar for metal and plastic objects," IEE Proceedings on Radar, Sonar and Navigation, vol. 151, pp. 91-99, 2004.
 - [33] Marple, S. L., "Computing the discrete-time analytic signal via FFT," IEEE Transactions on Signal Processing, vol. 47, pp. 2600-2603, 1999.
 - [34] Gazdag, J., Sguazzero, P., "Migration of seismic data," Proceedings of IEEE, vol. 72, pp. 1302-1315, 1984.
 - [35] Weiss, L., McDonough, R. N., "Prony's method, z-transforms and Pade approximation," SIAM Review, vol. 5, pp. 145-149, 1963.
 - [36] Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.R., "Fisher discriminant analysis with kernels," Proceedings of the 1999 IEEE Signal Processing Society Workshop, pp. 41-48, 1999.

Appendix A: Derivation of Prony's Equations

Appendix A: Derivation of Prony's Equations

As shown in equation (40) and reiterated here for clarity, the analog signal $f(t)$ can be estimated as follows,

$$f(t) = \sum_{j=1}^N R_j e^{S_j t} \quad (57)$$

where N is the number of poles and residues.

We define the equivalent digital signal as

$$f_i = f(iT) \quad (58)$$

where T is the time sampled at a constant interval.

Next, we define

$$z_j = e^{S_j T} \quad (59)$$

and the relation

$$\sum_{i=0}^n \alpha_i z^i = \prod_{j=1}^n (z - z_j) , \quad \alpha_n = 1 \quad (60)$$

where the sampled signal consists of $2n$ points.

Using equations (57), (58) and (59), we obtain

$$f_i = \sum_{j=1}^N R_j z_j^i \quad (61)$$

Using equations (60) and (60), we form

Appendix A: Derivation of Prony's Equations

$$\begin{aligned}
 \sum_{i=0}^n f_{k+i} \alpha_i &= \sum_{i=0}^n \left\{ \sum_{j=1}^n R_j z_j^{k+i} \right\} \alpha_i \\
 &= \sum_{j=1}^n R_j z_j^k \left\{ \sum_{i=0}^n \alpha_i z_j^i \right\} \\
 &= 0, \quad (k = 0, \dots, N-1)
 \end{aligned} \tag{62}$$

which proves equation (41). Expanding the above equation in matrix form,

$$\begin{bmatrix} f_0 & f_1 & \cdots & f_n \\ f_1 & f_2 & \cdots & f_{n+1} \\ \vdots & \vdots & & \vdots \\ f_{N-1} & f_{N-2} & \cdots & f_{N-1+n} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = 0 \tag{63}$$

and since $a_n = 1$, we can obtain

$$\begin{bmatrix} f_0 & f_1 & \cdots & f_{n-1} \\ f_1 & f_2 & \cdots & f_n \\ \vdots & \vdots & & \vdots \\ f_{N-1} & f_{N-2} & \cdots & f_{N-1+n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = - \begin{bmatrix} f_n \\ f_{n+1} \\ \vdots \\ f_{N-1+n} \end{bmatrix} \tag{64}$$

From equation (63) above, a_0 to a_n can be obtained by performing a matrix inversion.

Note that a_0 to a_n are the coefficients of the polynomial equation (60). Thus from equation (60), the exponentials z_j can be obtained by factorization. The poles can then be obtained using the following

$$S_j = \frac{1}{T} \ln z_j \tag{65}$$

Finally, R_j can be found from equation (61) given the values of z_j .

Appendix B: Matlab Code for FDTD Simulation

Appendix B: Matlab Code for FDTD Simulation

```
function Fdtd2d(nx,ny,nmax,pointSource,eps,sig,plotOutput,saveFileName)
% E.g. :
% Fdtd2d(450,800,2000,[5 400],[1.0 2.8 1.0],[0.0 0.15 1e7],1,'test')
%
% Flat iron object

%*****
% Constants
%*****
c0 = 2.99792458e8;           %speed of light
mu0 = 4.0*pi*1.0e-7;        %permeability
eps0 = 1.0/(c0*c0*mu0);     %permittivity

%*****
% User-defined Variables
%*****
nPML = 8;                   %thickness of PML region
R0 = 0.00001;               %reflectivity of PML in terms of percentage
order = 2;

freq = 4.0e+9;              %center frequency of source excitation
dx = c0/freq/20;            %space increment of square lattice
                               (sampled at 1/20 of wavelength)

mu = [1.0 1.0 1.0];
sigS = [0.0 0.0 0.0];

%*****
% Other parameters
%*****
lambda = c0/freq;           %center wavelength of source excitation
omega = 2.0*pi*freq;

dt = dx/(2.0*c0);          %time step

nxPlus1 = nx+1;
nyPlus1 = ny+1;
nPMLPlus1 = nPML+1;

%*****
% Create Field Arrays
%*****
ex = zeros(nx, nyPlus1);    %fields in Main grid
ey = zeros(nxPlus1, ny);
hz = zeros(nx, ny);

exFront = zeros(nx+2*nPML, nPML);    %fields in Front PML region
eyFront = zeros(nx+2*nPML+1, nPML);
hzxFront = zeros(nx+2*nPML, nPML);
hzyFront = zeros(nx+2*nPML, nPML);

exBack = zeros(nx+2*nPML, nPMLPlus1); %fields in Back PML region
eyBack = zeros(nx+2*nPML+1, nPML);
hzxBack = zeros(nx+2*nPML, nPML);
hzyBack = zeros(nx+2*nPML, nPML);

exLeft = zeros(nPML, nyPlus1);        %fields in Left PML region
eyLeft = zeros(nPML, ny);
hzxLeft = zeros(nPML, ny);
```

Appendix B: Matlab Code for FDTD Simulation

```

hzyLeft = zeros(nPML, ny);

exRight = zeros(nPML,nyPlus1);           %fields in Right PML region
eyRight = zeros(nPMLPlus1,ny);
hzxRight = zeros(nPML,ny);
hzyRight = zeros(nPML,ny);

%*****
% Updating Coefficients for Main Grid
%*****
c1EField = (2*eps*eps0 - sig*dt) ./ (2*eps*eps0 + sig*dt);
    %coefficient 1 for E-field
c2EField = 2*dt ./ (dx*(2*eps*eps0+sig*dt));
    %coefficient 2 for E-field
c1HField = (2*mu*mu0 - sigS*dt) ./ (2*mu*mu0 + sigS*dt);
    %coefficient 1 for H-field
c2HField = 2*dt ./ (dx*(2*mu*mu0+sigS*dt));
    %coefficient 2 for H-field

% Initialize entire main grid to free space
c1Ex(1:nx,1:nyPlus1) = c1EField(1);
c2Ex(1:nx,1:nyPlus1) = c2EField(1);
c1Ey(1:nxPlus1,1:ny) = c1EField(1);
c2Ey(1:nxPlus1,1:ny) = c2EField(1);
c1Hz(1:nx,1:ny)      = c1HField(1);
c2Hz(1:nx,1:ny)      = c2HField(1);

%*****
% Define Object
%*****
% Wall
wallX = 60:100;      %approx. 10cm with dx=0.0037
wallY = 1:ny;
c1Ex(wallX, wallY) = c1EField(2);
c2Ex(wallX, wallY) = c2EField(2);
c1Ey(wallX, wallY) = c1EField(2);
c2Ey(wallX, wallY) = c2EField(2);

% Flat Object
objectX = 360:374;
objectY = 550:650;
c1Ex(objectX, objectY) = c1EField(3);
c2Ex(objectX, objectY) = c2EField(3);
c1Ey(objectX, objectY) = c1EField(3);
c2Ey(objectX, objectY) = c2EField(3);

% % Sphere Object
% diam = 40;           % diameter of cylinder: 6 cm
% rad = diam/2.0;      % radius of cylinder: 3 cm
% icenter = 367;       % i-coordinate of cylinder's center
% jcenter = ny/2;      % j-coordinate of cylinder's center
%
% for i=1:nx
%     for j=1:ny
%         dist2=(i+0.5-icenter)^2 + (j-jcenter)^2;
%         if dist2 <= rad^2
%             c1Ex(i,j)=c1EField(3);
%             c2Ex(i,j)=c2EField(3);
%         end
%         dist2=(i-icenter)^2 + (j+0.5-jcenter)^2;
%         if dist2 <= rad^2
%             c1Ey(i,j)=c1EField(3);
%             c2Ey(i,j)=c2EField(3);
%         end
%     end

```


Appendix B: Matlab Code for FDTD Simulation

```

% end
% end

%*****
% Updating Coefficients for PML Layers
%*****
delta = nPML*dx;
sigmaM = -log(R0/100)*eps0*c0*(order+1)/(2*delta);
sigmaN0 = sigmaM / ((order+1)*(2^(order+1))*(nPML^order));

% Front region -----

c1ExFront(1:nx+2*nPML, 1) = 1;      %R1 : outer most layer (which is set
    to a perfect reflector)
c2ExFront(1:nx+2*nPML, 1) = 0;

for j = 2:nPML                      %R2 : for the different layers of Ex in front
    PML region
    L = nPML-j+1;
    sigmaY = sigmaN0*((2*L+1)^(order+1) - (2*L-1)^(order+1));
    c1ExFront(1:nx+2*nPML, j) = exp(-sigmaY*dt/eps0);
    c2ExFront(1:nx+2*nPML, j) = (1-exp(-sigmaY*dt/eps0)) / (sigmaY*dx);
end

for j = 1:nPML                      %R6 : for the different layers of Hzy in front
    PML region
    L = nPML-j+0.5;
    sigmaY = sigmaN0*((2*L+1)^(order+1) - (2*L-1)^(order+1));
    sigmaYS = sigmaY*mu0/eps0;      %change to
    magnetic equivalent
    c1HzyFront(1:nx+2*nPML, j) = exp(-sigmaYS*dt/mu0);
    c2HzyFront(1:nx+2*nPML, j) = (1-exp(-sigmaYS*dt/mu0)) / (sigmaYS*dx);
end

sigmaY = sigmaN0;                  %becos L=0      %for the interface between
    Left&Front, Main&Front, Right&Front
c1Ex(1:nx,1) = exp(-sigmaY*dt/eps0);      %R3
c2Ex(1:nx,1) = (1-exp(-sigmaY*dt/eps0)) / (sigmaY*dx);
c1ExLeft(1:nPML,1) = exp(-sigmaY*dt/eps0);      %R4
c2ExLeft(1:nPML,1) = (1-exp(-sigmaY*dt/eps0)) / (sigmaY*dx);
c1ExRight(1:nPML,1) = exp(-sigmaY*dt/eps0);      %R5
c2ExRight(1:nPML,1) = (1-exp(-sigmaY*dt/eps0)) / (sigmaY*dx);

for j = 1:nPML                      %for the rest of the fields in front region, use
    normal coefficients which do not vary conductivities
    c1EyFront(1:nx+2*nPML+1, j) = c1EField(1);      %R7
    c2EyFront(1:nx+2*nPML+1, j) = c2EField(1);
    c1HzxFront(1:nx+2*nPML, j) = c1HField(1);      %R6
    c2HzxFront(1:nx+2*nPML, j) = c2HField(1);
end

% Back region -----

c1ExBack(1:nx+2*nPML, nPMLPlus1) = 1;      %R8 : for the outer most
    layer (which is set to a perfect reflector)
c2ExBack(1:nx+2*nPML, nPMLPlus1) = 0;

for j = 2:nPML                      %for the different layers of Ex in
    back PML region
    L = j-1;
    sigmaY = sigmaN0*((2*L+1)^(order+1) - (2*L-1)^(order+1));
    c1ExBack(1:nx+2*nPML, j) = exp(-sigmaY*dt/eps0);      %R9
    c2ExBack(1:nx+2*nPML, j) = (1-exp(-sigmaY*dt/eps0)) / (sigmaY*dx);
end

```

Appendix B: Matlab Code for FDTD Simulation

```

for j = 1:nPML                                %for the different layers of Hzy in
    back PML region
    L = j-0.5;
    sigmaY = sigmaN0*((2*L+1)^(order+1) - (2*L-1)^(order+1));
    sigmaYS = sigmaY*mu0/eps0;                %change to
    magnetic equivalent
    c1HzyBack(1:nx+2*nPML, j) = exp(-sigmaYS*dt/mu0);          %R13
    c2HzyBack(1:nx+2*nPML, j) = (1-exp(-sigmaYS*dt/mu0)) / (sigmaYS*dx);
end

sigmaY = sigmaN0;          %becos L=0          %for the interface between
    Left&Back, Main&Back, Right&Back
c1Ex(1:nx,nyPlus1) = exp(-sigmaY*dt/eps0);          %R10
c2Ex(1:nx,nyPlus1) = (1-exp(-sigmaY*dt/eps0)) / (sigmaY*dx);
c1ExLeft(1:nPML,nyPlus1) = exp(-sigmaY*dt/eps0);          %R11
c2ExLeft(1:nPML,nyPlus1) = (1-exp(-sigmaY*dt/eps0)) / (sigmaY*dx);
c1ExRight(1:nPML,nyPlus1) = exp(-sigmaY*dt/eps0);          %R12
c2ExRight(1:nPML,nyPlus1) = (1-exp(-sigmaY*dt/eps0)) / (sigmaY*dx);

for j = 1:nPML          %for the rest of the fields in back region, use
    normal coefficients which do not vary conductivities
    c1EyBack(1:nx+2*nPML+1, j) = c1EField(1);          %R14
    c2EyBack(1:nx+2*nPML+1, j) = c2EField(1);
    c1HxzBack(1:nx+2*nPML, j) = c1HField(1);          %R13
    c2HxzBack(1:nx+2*nPML, j) = c2HField(1);
end

% Left region -----

c1EyLeft(1, 1:ny) = 1;          %R15 : outer most layer (which is set to a
    perfect reflector)
c2EyLeft(1, 1:ny) = 0;

for i = 2:nPML          %for the different layers of Ey in left
    PML region
    L = nPML-i+1;
    sigmaX = sigmaN0*((2*L+1)^(order+1) - (2*L-1)^(order+1));
    c1EyLeft(i, 1:ny) = exp(-sigmaX*dt/eps0);          %R16
    c2EyLeft(i, 1:ny) = (1-exp(-sigmaX*dt/eps0)) / (sigmaX*dx);
    c1EyFront(i,1:nPML) = exp(-sigmaX*dt/eps0);          %R17
    c2EyFront(i,1:nPML) = (1-exp(-sigmaX*dt/eps0)) / (sigmaX*dx);
    c1EyBack(i,1:nPML) = exp(-sigmaX*dt/eps0);          %R18
    c2EyBack(i,1:nPML) = (1-exp(-sigmaX*dt/eps0)) / (sigmaX*dx);
end

for i = 1:nPML          %for the different layers of Hzx in
    front PML region
    L = nPML-i+0.5;
    sigmaX = sigmaN0*((2*L+1)^(order+1) - (2*L-1)^(order+1));
    sigmaXS = sigmaX*mu0/eps0;                %change to
    magnetic equivalent
    c1HzxLeft(i, 1:ny) = exp(-sigmaXS*dt/mu0);          %R22
    c2HzxLeft(i, 1:ny) = (1-exp(-sigmaXS*dt/mu0)) / (sigmaXS*dx);
    c1HzxFront(i,1:nPML) = exp(-sigmaXS*dt/mu0);          %R23
    c2HzxFront(i,1:nPML) = (1-exp(-sigmaXS*dt/mu0)) / (sigmaXS*dx);
    c1HzxBack(i,1:nPML) = exp(-sigmaXS*dt/mu0);          %R24
    c2HzxBack(i,1:nPML) = (1-exp(-sigmaXS*dt/mu0)) / (sigmaXS*dx);
end

sigmaX = sigmaN0;          %becos L=0          %for the interface between
    Front&Left, Main&Left, Back&Left
c1Ey(1,1:ny) = exp(-sigmaX*dt/eps0);          %R19
c2Ey(1,1:ny) = (1-exp(-sigmaX*dt/eps0)) / (sigmaX*dx);

```


Appendix B: Matlab Code for FDTD Simulation

```

c1EyFront(nPMLPlus1,1:nPML) = exp(-sigmaX*dt/eps0); %R20
c2EyFront(nPMLPlus1,1:nPML) = (1-exp(-sigmaX*dt/eps0)) / (sigmaX*dx);
c1EyBack(nPMLPlus1,1:nPML) = exp(-sigmaX*dt/eps0); %R21
c2EyBack(nPMLPlus1,1:nPML) = (1-exp(-sigmaX*dt/eps0)) / (sigmaX*dx);

for i = 1:nPML %for the rest of the fields in left region, use
    normal coefficients which do not vary conductivities
    c1ExLeft(i, 2:ny) = c1EField(1); %R25
    c2ExLeft(i, 2:ny) = c2EField(1);
    c1HzyLeft(i, 1:ny) = c1HField(1); %R22
    c2HzyLeft(i, 1:ny) = c2HField(1);
end

% Right region -----
-----

c1EyRight(nPMLPlus1, 1:ny) = 1; %R23 : outer most layer (which is
    set to a perfect reflector)
c2EyRight(nPMLPlus1, 1:ny) = 0;

for i = 2:nPML %for the different layers of Ey in Right PML region
    L = i-1;
    sigmaX = sigmaN0*((2*L+1)^(order+1) - (2*L-1)^(order+1));
    c1EyRight(i, 1:ny) = exp(-sigmaX*dt/eps0); %R24
    c2EyRight(i, 1:ny) = (1-exp(-sigmaX*dt/eps0)) / (sigmaX*dx);
    c1EyFront(i+nPML+nx,1:nPML) = exp(-sigmaX*dt/eps0); %R25
    c2EyFront(i+nPML+nx,1:nPML) = (1-exp(-sigmaX*dt/eps0)) / (sigmaX*dx);
    c1EyBack(i+nPML+nx,1:nPML) = exp(-sigmaX*dt/eps0); %R26
    c2EyBack(i+nPML+nx,1:nPML) = (1-exp(-sigmaX*dt/eps0)) / (sigmaX*dx);
end

for i = 1:nPML %for the different layers of Hzx in
    front PML region
    L = i-0.5;
    sigmaX = sigmaN0*((2*L+1)^(order+1) - (2*L-1)^(order+1));
    sigmaXS = sigmaX*mu0/eps0; %change to magnetic equivalent
    c1HzxRight(i, 1:ny) = exp(-sigmaXS*dt/mu0); %R30
    c2HzxRight(i, 1:ny) = (1-exp(-sigmaXS*dt/mu0)) / (sigmaXS*dx);
    c1HzxFront(i+nx+nPML,1:nPML) = exp(-sigmaXS*dt/mu0); %R31
    c2HzxFront(i+nx+nPML,1:nPML) = (1-exp(-sigmaXS*dt/mu0)) /
        (sigmaXS*dx);
    c1HzxBack(i+nx+nPML,1:nPML) = exp(-sigmaXS*dt/mu0); %R32
    c2HzxBack(i+nx+nPML,1:nPML) = (1-exp(-sigmaXS*dt/mu0)) / (sigmaXS*dx);
end

sigmaX = sigmaN0; %becos L=0 %for the interface between
    Front&Right, Main&Right, Back&Right
c1Ey(nxPlus1, 1:ny) = exp(-sigmaX*dt/eps0); %R27
c2Ey(nxPlus1, 1:ny) = (1-exp(-sigmaX*dt/eps0)) / (sigmaX*dx);
c1EyFront(nPML+nxPlus1, 1:nPML) = exp(-sigmaX*dt/eps0); %R28
c2EyFront(nPML+nxPlus1, 1:nPML) = (1-exp(-sigmaX*dt/eps0)) /
    (sigmaX*dx);
c1EyBack(nPML+nxPlus1, 1:nPML) = exp(-sigmaX*dt/eps0); %R29
c2EyBack(nPML+nxPlus1, 1:nPML) = (1-exp(-sigmaX*dt/eps0)) / (sigmaX*dx);

for i = 1:nPML %for the rest of the fields in Right region, use
    normal coefficients which do not vary conductivities
    c1ExRight(i, 2:ny) = c1EField(1); %R33
    c2ExRight(i, 2:ny) = c2EField(1);
    c1HzyRight(i, 1:ny) = c1HField(1); %R30
    c2HzyRight(i, 1:ny) = c2HField(1);
end

%*****

```

Appendix B: Matlab Code for FDTD Simulation

```
% Wave Excitation
%*****
bw = 1e9;           %bandwidth in Hz
u = 1/(2*pi*bw*sqrt(log10(exp(1))));
t = 0:dt:(nmax-1)*dt;
delay = 150*dt;

source = cos(2*pi*freq.*(t-delay)) .* exp(-0.5*(t-delay).^2/u^2);
% figure; plot(t, source, '-');

%*****
% Main Loop
%*****
terminate = 0;
h_waitbar = waitbar(0,'Processing ...');    %create handle for waitbar
        so that can be closed later
sampleData = [];
if (plotOutput == 1)    %if need to plot output, then create the figure
    h_fig =
        figure('numbertitle','off','menubar','none','buttondownfcn','terminate=1;');
end
for n = 1:nmax

    % Update E-fields in Main grid
    x = 1:nx;
    y = 2:ny;
    ex(x,y) = c1Ex(x,y).*ex(x,y) + c2Ex(x,y).*(hz(x,y)-hz(x,y-1));

    x = 2:nx;
    y = 1:ny;
    ey(x,y) = c1Ey(x,y).*ey(x,y) - c2Ey(x,y).*(hz(x,y)-hz(x-1,y));

    % Update Ex in the PML region
    % Front
    x = 1:nx+2*nPML;
    y = 2:nPML;
    exFront(x,y) = c1ExFront(x,y).*exFront(x,y) + ...
        c2ExFront(x,y).*(hzxFront(x,y)-hzxFront(x,y-1)+hzyFront(x,y)-
        hzyFront(x,y-1));
    x = 1:nx;
    y = 1;
    ex(x,y) = c1Ex(x,y).*ex(x,y) + ...
        c2Ex(x,y).*(hz(x,y)-hzxFront(nPMLPlus1:nPML+nx, nPML)-
        hzyFront(nPMLPlus1:nPML+nx, nPML));
    % Back
    x = 1:nx+2*nPML;
    y = 2:nPML;
    exBack(x,y) = c1ExBack(x,y).*exBack(x,y) + ...
        c2ExBack(x,y).*(hzxBack(x,y)-hzxBack(x,y-1)+hzyBack(x,y)-
        hzyBack(x,y-1));
    x = 1:nx;
    y = nyPlus1;
    ex(x,y) = c1Ex(x,y).*ex(x,y) + ...
        c2Ex(x,y).*(hzxBack(nPMLPlus1:nPML+nx,
        1)+hzyBack(nPMLPlus1:nPML+nx, 1)-hz(x,y-1));
    % Left
    x = 1:nPML;
    y = 2:ny;
    exLeft(x,y) = c1ExLeft(x,y).*exLeft(x,y) + ...
        c2ExLeft(x,y).*(hzxLeft(x,y)-hzxLeft(x,y-1)+hzyLeft(x,y)-
        hzyLeft(x,y-1));
    x = 1:nPML;
    y = 1;
```


Appendix B: Matlab Code for FDTD Simulation

```

    exLeft(x,y) = c1ExLeft(x,y).*exLeft(x,y) + ...
        c2ExLeft(x,y).*(hzxLeft(x,y)-hzxFront(x, nPML)+hzyLeft(x,y)-
hzyFront(x, nPML));
    x = 1:nPML;
    y = nyPlus1;
    exLeft(x,y) = c1ExLeft(x,y).*exLeft(x,y) + ...
        c2ExLeft(x,y).*(hzxBack(x,1)-hzxLeft(x,ny)+hzyBack(x,1)-
hzyLeft(x,ny));
    % Right
    x = 1:nPML;
    y = 2:ny;
    exRight(x,y) = c1ExRight(x,y).*exRight(x,y) + ...
        c2ExRight(x,y).*(hzxRight(x,y)-hzxRight(x,y-1)+hzyRight(x,y)-
hzyRight(x,y-1));
    x = 1:nPML;
    y = 1;
    exRight(x,y) = c1ExRight(x,y).*exRight(x,y) + ...
        c2ExRight(x,y).*(hzxRight(x,y)-hzxFront(nPML+nx+1:nx+2*nPML,
nPML)+hzyRight(x,y)-hzyFront(nPML+nx+1:nx+2*nPML, nPML));
    x = 1:nPML;
    y = nyPlus1;
    exRight(x,y) = c1ExRight(x,y).*exRight(x,y) + ...
        c2ExRight(x,y).*(hzxBack(nPML+nx+1:nx+2*nPML, 1)-
hzxRight(x,ny)+hzyBack(nPML+nx+1:nx+2*nPML, 1)-hzyRight(x,ny));

% Update Ey in the PML region
% Front
    x = 2:nx+2*nPML;
    y = 1:nPML;
    eyFront(x,y) = c1EyFront(x,y).*eyFront(x,y) - ...
        c2EyFront(x,y).*(hzxFront(x,y)-hzxFront(x-1,y)+hzyFront(x,y)-
hzyFront(x-1,y));
% Back
    x = 2:nx+2*nPML;
    y = 1:nPML;
    eyBack(x,y) = c1EyBack(x,y).*eyBack(x,y) - ...
        c2EyBack(x,y).*(hzxBack(x,y)-hzxBack(x-1,y)+hzyBack(x,y)-
hzyBack(x-1,y));
% Left
    x = 2:nPML;
    y = 1:ny;
    eyLeft(x,y) = c1EyLeft(x,y).*eyLeft(x,y) - ...
        c2EyLeft(x,y).*(hzxLeft(x,y)-hzxLeft(x-1,y)+hzyLeft(x,y)-
hzyLeft(x-1,y));
    x = 1;
    y = 1:ny;
    ey(x,y) = c1Ey(x,y).*ey(x,y) - ...
        c2Ey(x,y).*(hz(x,y)-hzxLeft(nPML,y)-hzyLeft(nPML,y));
% Right
    x = 2:nPML;
    y = 1:ny;
    eyRight(x,y) = c1EyRight(x,y).*eyRight(x,y) - ...
        c2EyRight(x,y).*(hzxRight(x,y)-hzxRight(x-1,y)+hzyRight(x,y)-
hzyRight(x-1,y));
    x = nxPlus1;
    y = 1:ny;
    ey(x,y) = c1Ey(x,y).*ey(x,y) - ...
        c2Ey(x,y).*(hzxRight(1,y)+hzyRight(1,y)-hz(nx,y));

% Update H-fields in Main grid
x = 1:nx;
y = 1:ny;
hz(x,y) = c1Hz(x,y).*hz(x,y) + ...
    c2Hz.*(ex(x,y+1)-ex(x,y)+ey(x,y)-ey(x+1,y));

```

Appendix B: Matlab Code for FDTD Simulation

```

hz(pointSource(1), pointSource(2)) = hz(pointSource(1),
    pointSource(2)) + source(n);

% Update Hxz in the PML region
% Front
x = 1:nx+2*nPML;
y = 1:nPML;
hzxFront(x,y) = c1HxzFront(x,y).*hzxFront(x,y) - ...
    c2HxzFront(x,y).*(eyFront(x+1,y)-eyFront(x,y));

% Back
x = 1:nx+2*nPML;
y = 1:nPML;
hzxBack(x,y) = c1HxzBack(x,y).*hzxBack(x,y) - ...
    c2HxzBack(x,y).*(eyBack(x+1,y)-eyBack(x,y));

% Left
x = 1:nPML-1;
y = 1:ny;
hzxLeft(x,y) = c1HxzLeft(x,y).*hzxLeft(x,y) - ...
    c2HxzLeft(x,y).*(eyLeft(x+1,y)-eyLeft(x,y));
x = nPML;
y = 1:ny;
hzxLeft(x,y) = c1HxzLeft(x,y).*hzxLeft(x,y) - ...
    c2HxzLeft(x,y).*(ey(1,y)-eyLeft(x,y));

% Right
x = 2:nPML;
y = 1:ny;
hzxRight(x,y) = c1HxzRight(x,y).*hzxRight(x,y) - ...
    c2HxzRight(x,y).*(eyRight(x+1,y)-eyRight(x,y));
x = 1;
y = 1:ny;
hzxRight(x,y) = c1HxzRight(x,y).*hzxRight(x,y) - ...
    c2HxzRight(x,y).*(eyRight(x+1,y)-ey(nxPlus1,y));

% Update Hzy in the PML region
% Front
x = 1:nx+2*nPML;
y = 1:nPML-1;
hzyFront(x,y) = c1HzyFront(x,y).*hzyFront(x,y) + ...
    c2HzyFront(x,y).*(exFront(x,y+1)-exFront(x,y));
x = 1:nPML;
y = nPML;
hzyFront(x,y) = c1HzyFront(x,y).*hzyFront(x,y) + ...
    c2HzyFront(x,y).*(exLeft(x,1)-exFront(x,y));
x = nPML+1:nPML+nx;
y = nPML;
hzyFront(x,y) = c1HzyFront(x,y).*hzyFront(x,y) + ...
    c2HzyFront(x,y).*(ex(1:nx,1)-exFront(x,y));
x = nPML+nx+1:nx+2*nPML;
y = nPML;
hzyFront(x,y) = c1HzyFront(x,y).*hzyFront(x,y) + ...
    c2HzyFront(x,y).*(exRight(1:nPML,1)-exFront(x,y));

% Back
x = 1:nx+2*nPML;
y = 2:nPML;
hzyBack(x,y) = c1HzyBack(x,y).*hzyBack(x,y) + ...
    c2HzyBack(x,y).*(exBack(x,y+1)-exBack(x,y));
x = 1:nPML;

```

Appendix B: Matlab Code for FDTD Simulation

```

y = 1;
hzyBack(x,y) = c1HzyBack(x,y).*hzyBack(x,y) + ...
    c2HzyBack(x,y).*(exBack(x,y+1)-exLeft(x,nyPlus1));
x = nPML+1:nPML+nx;
y = 1;
hzyBack(x,y) = c1HzyBack(x,y).*hzyBack(x,y) + ...
    c2HzyBack(x,y).*(exBack(x,y+1)-ex(1:nx,nyPlus1));
x = nPML+nx+1:nx+2*nPML;
y = 1;
hzyBack(x,y) = c1HzyBack(x,y).*hzyBack(x,y) + ...
    c2HzyBack(x,y).*(exBack(x,y+1)-exRight(1:nPML,nyPlus1));

% Left
x = 1:nPML;
y = 1:ny;
hzyLeft(x,y) = c1HzyLeft(x,y).*hzyLeft(x,y) + ...
    c2HzyLeft(x,y).*(exLeft(x,y+1)-exLeft(x,y));

% Right
x = 1:nPML;
y = 1:ny;
hzyRight(x,y) = c1HzyRight(x,y).*hzyRight(x,y) + ...
    c2HzyRight(x,y).*(exRight(x,y+1)-exRight(x,y));

sampleData = [sampleData; hz(5,:)];
if (n == 500)
    save tempDat1 sampleData;
    sampleData = [];
elseif (n == 1000)
    save tempDat2 sampleData;
    sampleData = [];
elseif (n == 1500)
    save tempDat3 sampleData;
    sampleData = [];
elseif (n == 2000)
    save tempDat4 sampleData;
    sampleData = [];
end

if (plotOutput == 1)
    if (mod(n,10)==0)
        subplot(2,1,1);
        imagesc(hz, [-0.005,0.005]);
        axis('equal');
        rectangle('position',[1 min(wallX) ny (max(wallX)-min(wallX))]);
        %draw outline of wall
        rectangle      %draw outline of object

        subplot(3,1,3);
        plot(hz(:,round(ny/2)));
        axis([0 nx -0.005 0.005]);
    end
    set(h_fig,'name',[num2str(n) '/' num2str(nmax)]);
    drawnow;
end

waitbar(n/nmax);                %update the waitbar
if (terminate == 1)
    break;
end
end      %end of Main Loop

close(h_waitbar);                %close the waitbar

```

Appendix B: Matlab Code for FDTD Simulation

```

%*****
% Combine data, Save and Clean-up
%*****
load tempDat1;    temp = sampleData;           %concatenate the data
               into 'temp'
load tempDat2;    temp = [temp; sampleData];
load tempDat3;    temp = [temp; sampleData];
load tempDat4;    temp = [temp; sampleData];
sampleData = temp;           %assign data back to the variable
               'sampleData'
save(saveFileName,'sampleData','nx','ny','nmax','freq','dx','wallX','wal
    lY','objectX','objectY','pointSource');    %save to the file name
               specified by user
delete('tempDat1.mat','tempDat2.mat','tempDat3.mat','tempDat4.mat');
               %delete unnecessary temp data files

```