DEEP LEARNING PROCESSING AND INTERPRETATION OF GROUND PENETRATING RADAR DATA USING A NUMERICAL EQUIVALENT OF A REAL GPR TRANSDUCER

OURANIA PATSIA

Doctor of Philosophy

School of Engineering

The University of Edinburgh

2022

Ourania Patsia: *Deep learning processing and interpretation of ground penetrating radar data using a numerical equivalent of a real GPR transducer*, Doctor of Philosophy, ©2022

Στους γονείς μου, Μαρία και Σταύρο
Στην αδερφή μου, Ολυμπία
Στον παππού μου, Γιώργο

The more that you read, the more things you will know. The more that you learn, the more places you'll go.
- Dr. Seuss

DECLARATION

I hereby declare that this thesis and the work reported herein was composed and originated entirely by myself, under the supervision of Dr. Antonios Giannopoulos in the School of Engineering at The University of Edinburgh.

The following conference publications are as a result of the research conducted for this thesis, which has not been submitted for any other degree or professional qualification.

Conferences

- Patsia, O., Giannopoulos, A., Giannakis, I. (2021). "A Deep Learning framework for Ground Penetrating Radar." In: Proceedings of the 11th International Workshop on Advanced Ground Penetrating Radar (IWAGPR), Valletta, Malta.
- Patsia, O., Giannopoulos, A., Giannakis, I. (2021). "A digital twin of the GSSI 2000 MHz palm antenna developed using multi-parametric optimisation." In: Proceedings of the 11th International Workshop on Advanced Ground Penetrating Radar (IWAGPR), Valletta, Malta.
- Patsia, O., Giannopoulos, A., Giannakis, I. (2021). "Full Waveform Inversion of common offset GPR data using a fast deep learning based forward solver." In: Proceedings of the 11th International Workshop on Advanced Ground Penetrating Radar (IWAGPR), Valletta, Malta.

Awards

- **Best Young Scientist Award/Oral Presentation.** Awarded to the best paper submitted by a PhD or post-doc in the 11th International Workshop on Advanced Ground Penetrating Radar 2021.

Edinburgh, 2022

Ourania Patsia

ABSTRACT

Ground-Penetrating Radar (GPR) is a popular non-destructive electromagnetic (EM) technique that is used in diverse applications across different fields, most commonly geophysics and civil engineering. One of the most common applications of GPR is concrete scanning, where it is used to detect structural elements and support the assessment of its condition. However, in any GPR application, the data have no resemblance to the characteristics of targets of interest and a means of extracting information from the data regarding the targets is required.

Interpreting the GPR data, to infer key properties of the subsurface and to locate the targets is a difficult and challenging task and is highly dependent on the processing of the data and the experience of the user. Traditional processing techniques have some drawbacks, which can lead to misinterpretations of the data in addition to the interpretation being subjective to the user. Machine learning (ML) has proven its ability to solve a variety of problems and map complex relationships and in recent years, is becoming an increasingly attractive option for solving GPR and other EM problems regarding processing and interpretation. Numerical modelling has been extensively used to understand the EM wave propagation and assist in the interpretation of GPR responses. If ML is combined with numerical modelling, efficient solutions to GPR problems can be acquired.

This research focuses on developing a numerical equivalent of a commercial GPR transducer and utilising this model to produce realistic synthetic training data sets for deep learning applications. The numerical model is based on the high-frequency 2000 MHz “palm” antenna from Geophysical Survey Systems, Inc. (GSSI). This GPR system is mainly used for concrete scanning, where the targets are located close to the surface. Unknown antenna parameters were found using global optimisation by minimising the mismatch between synthetic and real responses. A very good match was achieved, demonstrating that the model can accurately replicate the behaviour of the real antenna which was further validated using a number of laboratory experiments. Real data were acquired using the GSSI transducer over a sandbox and reinforced concrete slabs and the same scenarios were replicated in the simulations using the antenna model, showing excellent agreement.

The developed antenna model was used to generate synthetic data, which are similar to the true data, for two deep learning applications, trained entirely using synthetic data. The first deep learning application suggested in the present thesis is background response and properties prediction. Two coupled neural networks are trained to predict the background response given as input total

GPR responses, perform background removal and subsequently use the predicted background response to predict its dielectric properties. The suggested scheme not only performs the background removal processing step, but also enables the velocity calculation of the EM wave propagating in a medium using the predicted permittivity value. The ML algorithm is evaluated using a number of synthetic and measured data demonstrating its efficiency and higher accuracy compared to traditional methods. Predicting a permittivity value per A-scan included in a B-scan results in a permittivity distribution, which is used along with background removal to perform reverse-time migration (RTM). The proposed RTM scheme proved to be superior when compared with the commonly used RTM schemes.

The second application was a deep learning-based forward solver, which is used as part of a full-waveform inversion (FWI) framework. A neural network is trained to predict entire B-scans given certain model parameters as input for reinforced concrete slab scenarios. The network makes predictions in real time, reducing by orders of magnitude the computational time of FWI, which is usually coupled with an FDTD forward solver. Therefore, making FWI applicable to commercial computers without the need of high-performance computing (HPC). The results clearly illustrate that ML schemes can be implemented to solve GPR problems and highlight the importance of having a digital representation of a real transducer in the simulations.

ACKNOWLEDGMENTS

I would like to express my gratitude towards my supervisor Prof. Antonios Giannopoulos for all the support, knowledge and constant encouragement he provided during my PhD. I could not thank him enough for all his guidance, his interest for the subject and excitement for this research that he transferred to me. I am grateful for working with him and all the opportunities that he provided me with through the years. Great supervisor and person!

I would also like to thank Prof. Panagiotis Tsourlos from Aristotle University of Thessaloniki, my undergraduate and master's supervisor. His encouragement and support from my undergraduate years up to this day were invaluable. I would not have reached this far without him and all his support! I honestly cannot imagine better (and funnier!) supervisors than Prof. Antonios Giannopoulos and Prof. Panagiotis Tsourlos during my research journey and I have been very lucky for having them.

I also want to thank Associate Prof. George Vargemezis from Aristotle University of Thessaloniki for all his help and knowledge I gained from him from the beginning of my studies. A big thanks to Dr. Iraklis Giannakis for his help, great ideas and collaboration the past years.

I would like to thank James Quinn for his constant support and being there for me whenever I needed. A big thanks goes also to my friends Kostis Chatziioannou and John Hartley that shared the office with for all the times we had in and out of the office and all the daily chat and breaks. Special thanks goes also to the girls I shared the office with for half of my PhD, Anastasia, Benedicta, Desi and Yasmin, for all our breaks and chats together. Many thanks to my friends Danijela and Apostolis for all the good times we had together during our PhDs in Edinburgh until today. I would also like to thank my friends Vasia Tzanakopoulou and Robin Delbart for all the fun we had together. Writing the PhD thesis was made a lot easier with you guys!

A big thanks goes to all my friends from Greece (especially Venetia, Iliana, Dimitra, Fay and Christos) for all their patience, support and love all these years. Thanks also to my friend from master studies Prodromos Louvaris for listening to me complaining. Special thanks goes to my cool aunt Eleni for all her support these years.

Finally, I would like to thank my parents, Maria and Stauros and my beautiful sister, Oly for their constant support and love all these years and for always being there for me in tough times. I would not have made this PhD without them. Love you all!

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ABC	Absorbing Boundary Condition
Adam	Adaptive Moment Estimation
AE	Autoencoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
CFS-PML	Complex Frequency Shifted Perfectly Matched Layer
CO	Common Offset
CMP	Common Mid-Point Reflection
CNN	Convolutional Neural Network
CPML	Convolutional Perfectly Matched Layer
CPU	Central Processing Unit
FD	Finite-Differences
FDTD	Finite-Difference Time-Domain
FEM	Finite Element Method
FFT	Fast Fourier Transform
FVTD	Finite-Volume Time-Domain
FWI	Full-Waveform Inversion
GA	Genetic Algorithm
GCI	Grid Convergence Index
GPML	General Perfectly Matched Layer
GPR	Ground-Penetrating Radar
GPU	Graphics Processing Unit
HPC	High Performance Computing
HT	Hilbert Transform

ICA	Independent Component Analysis
IIR	Infinite Impulse Response
MCA	Morphological Component Analysis
ML	Machine Learning
MoM	Method of Moments
NDT	Non-Destructive Testing
NN	Neural Networks
PCA	Principal Components Analysis
PCB	Printed Circuit Board
PEC	Perfect Electric Conductor
PMC	Perfect Magnetic Conductor
PML	Perfectly Matched Layer
PSO	Particle Swarm Optimisation
ReLU	Rectified Linear Unit
RF	Radio Frequency
RNN	Recurrent Neural Network
RPCA	Robust Principal Components Analysis
RTM	Reverse-Time Migration
SGD	Stochastic Gradient Descent
SFCW	Stepped Frequency Carrier Wave
SNR	Signal to Noise Ratio
SVD	Singular Value Decomposition
TEM	Transient electromagnetic
TE	Transverse Electric
TM	Transverse Magnetic
TLM	Transmission Line Method
UPML	Uniaxial Perfectly Matched Layer

UWB	Ultra Wide Band
VNA	Vector Network Analyzer
WARR	Wide-Angle Reflection-Refraction

1

INTRODUCTION

1.1 MOTIVATION AND AIMS OF THE THESIS

Ground Penetrating Radar (GPR) is a widely used non-destructive geophysical technique with a broad spectrum of applications, ranging from engineering and geophysical applications to planetary applications. GPR data are collected over complex inhomogeneous environments, which can be natural or man-made with a number of buried objects. These data can vastly vary depending on the environment conditions encountered in a specific application and require an experienced user for accurate interpretation. Characterising the environment along with inferring the location and properties of targets by interpreting the GPR data is a challenging problem, for which, understanding the behavior of electromagnetic waves (EM) in materials is needed. To acquire knowledge on EM wave propagation, numerical modelling is utilised.

Numerical modelling has been used for several years in many disciplines for modelling real systems and their behavior in order to gain insight on the characteristics of the real system used in practice, improve a system or develop a new one and interpret output data among other reasons. A typical GPR simulation should include a model of the subsurface or material of interest, models of the targets and an accurate representation of an antenna system. However, most GPR numerical simulations include simple theoretical sources instead of realistic models of real transducers, which cannot capture the interactions of the real antenna with different materials and objects and therefore cannot reproduce accurately the real responses. Nevertheless, even in cases where an accurate antenna replica is included in the models, which are used to assist in the interpretation of real data, the interpretation is still a difficult task and is always relied on some form of processing, experience and user subjectivity. Therefore, automated ways of processing and interpreting the data are needed.

The aim of this thesis was to develop a numerical model of a real GPR transducer and use it to generate training data for machine learning (ML) applications in order to automate the processing and enhance the interpretation of the GPR data. This research focuses on ML schemes for civil engineering applications using a high-frequency GPR system, but can be applied to other cases, as well.

The key objectives of the present thesis are summarised as follows:

- Create an accurate 3D Finite-Difference Time-Domain (FDTD) model of a widely-used commercial GPR antenna.
- Analysis of the antenna model and validation using data from the real GPR transducer.
- Generate training sets for machine learning (ML) purposes using the antenna model in the simulations in order to obtain realistic responses.
- Develop a deep learning scheme that predicts the background response and its dielectric properties.
- Develop an ML-based forward solver that predicts GPR responses for reinforced concrete slab scenarios.
- Evaluate the performance of both ML schemes using real data.

1.2 OVERVIEW OF THE THESIS

A brief overview of the basic structure of the thesis is as follows:

Chapter 2: The aim of this chapter is to introduce the basic principles of GPR and the propagation mechanisms of electromagnetic waves in materials. Data gathering and processing techniques are also presented along with a description of the GPR equipment.

Chapter 3: In this chapter, Maxwell's equation and the FDTD method, which is commonly used to solve these equation and model GPR responses, are discussed. Moreover, both dispersion in real world and numerical dispersion in FDTD are described.

Chapter 4: The development of the numerical model of a commercial GPR system is described in this chapter. The geometry used and the optimisation process used to obtain estimates of unknown key properties of the antenna are presented. In addition, the antenna radiation patterns in different materials are presented. Validation of the antenna model is performed using real data from different scenarios.

Chapter 5: A basic background on ML is provided with emphasis given on deep learning and particularly artificial neural networks (ANNs). The structure and development of an ANN is described along with an explanation of the backpropagation algorithm, and the training process of an ANN.

Chapter 6: In this chapter, a deep learning framework is proposed, that predicts the background response and its dielectric properties. For the training set, simulations were generated that include the developed model of the real transducer. The scheme is evaluated with both synthetic and real data. In addition, it is used to perform reverse-time migration (RTM) with a permittivity distribution along a GPR profile.

Chapter 7: An ML-based forward solver is described in this chapter, that is used to predict GPR responses for reinforced concrete slab scenarios in real time and generate entire B-scans. The algorithm is validated with both modelled and real data and is used as part of a full-waveform inversion (FWI) scheme. Furthermore, comparison is made between the suggested scheme and the commonly-used FDTD solver.

Chapter 8: Finally, this chapter summarises the main findings and provides suggestions for future work.

2

PRINCIPLES OF GROUND PENETRATING RADAR

In this chapter, the basic principles of GPR are introduced. First, a brief introduction of the method and its applications is given. The basic principles of electromagnetic wave propagation, the signal paths and the dielectric properties of materials are discussed. In addition, the GPR data acquisition process, the equipment used to acquire the data and key processing methods are described.

2.1 INTRODUCTION

GPR is a non-destructive technique that belongs to the general group of geophysical methods. Geophysical techniques (Electrical Resistivity Method, GPR, Seismic Refraction-Reflection, Potential Field Methods, Magnetic methods etc.) are used to map the distribution of the subsurface's physical properties (electrical resistivity, dielectric properties, elastic wave velocity, density etc.) in order to obtain information on the nature of the earth materials and buried targets. In the GPR case, the distribution of the dielectric properties (electric permittivity, electric conductivity and magnetic permeability) is sought. These properties are not directly measured but can be obtained through advanced processing of the GPR data.

The GPR method is an "active" method, meaning that the field which is used, is created artificially, in contrast to "passive" methods, which utilize existing fields. GPR uses a transmitting antenna to emit ultra wide-band electromagnetic (EM) pulses into the ground. When a contrast in the dielectric properties exists in the subsurface, part of the EM energy is reflected and received by a receiving antenna. This received signal can be processed and interpreted to obtain information regarding the structure of the survey area.

GPR is widely used for a diverse range of applications in different disciplines such as engineering, geology and geophysics. Applications are not limited only to geological materials, but GPR is also used for man-made materials such as concrete, asphalt or other construction materials. Depending on the survey, the depth of penetration, the size, type and properties of targets being sought can vary, spanning from near-surface surveys for locating small targets, such as in civil engineering, to geological surveys up to hundreds of meters deep. The most common GPR applications are the following:

- Concrete scanning: Location of reinforcing bars (rebars), post-tension cables, conduits and concrete assessment (Wai-Lok Lai et al., 2018; Kim et al., 2014; Barrile and Pucinotti, 2005; González-Drigo et al., 2008)
- Bridge and road inspection: Find layer thickness and structure assessment (Alani et al., 2013; Benedetto, 2013; Zhao and Al-Qadi, 2016; Diamanti and Redman, 2012; Varela-González et al., 2014; Solla et al., 2013)
- Buried utilities: Detection of metallic or non-metallic pipes, conduits, cables and voids (Li et al., 2015; Jaw and Hashim, 2013; Sagnard et al., 2016)
- Geology and geophysics: Water table mapping, bedrock depth etc. (Benedetto et al., 2016)
- Military: Detection of unexploded ordnance (Daniels, 2005)
- Forensics: Locate buried evidence (Daniels, 2005)
- Archaeological: Map archaeological features such as buried buildings and infrastructure (Conyers and Goodman, 1997; Tsokas et al., 1994)
- Ice profiling (Bohleber et al., 2017)
- Mineral exploration (Daniels, 2005)
- Planetary exploration (Pettinelli et al., 2022)

2.2 WAVE PROPAGATION

Although GPR shares the same principles with conventional radars, which are mostly used to locate larger, above-ground or above-sea level targets with air as a surrounding material, the ground in which GPR is used affects greatly the propagation of the transmitted EM pulses. The transmitter sends EM pulses, that travel into a medium at a certain velocity which is governed by the medium dielectric properties in contrast to conventional radars, where the waves propagate with the speed of light ($c \approx 3 \cdot 10^8 \mathrm{m/s}$). For GPR, permittivity and conductivity hold the most important role. When there is a contrast in these properties, part of the EM energy is scattered and/or reflected, while the rest of the energy continues travelling through the medium. Although the materials that are encountered with GPR are complex, most of them are non-magnetic, having a permeability close to free space. If the materials have significant magnetic content, such as in the presence of magnetite, maghemite or hematite, then the permeability should be taken into account.

GPR can be used to investigate lossless dielectrics but most of the materials encountered are lossy dielectrics. The velocity of the EM wave in a lossy material is given by Balanis (2012)

$$u = \frac{c}{\sqrt{\frac{\mu_r \epsilon_r}{2}(\sqrt{1 + (\frac{\sigma}{\omega})^2} + 1)}} \tag{2.1}$$

where c is the speed of light in free space, μ_r is the relative permeability, ϵ_r is the relative permittivity, σ is the electric conductivity (S/m) and ω is the angular frequency (rad/s). In the case of lossless and non-magnetic material, this simplifies to

$$u = \frac{c}{\sqrt{\epsilon_r}} \tag{2.2}$$

where the velocity is computed using only the permittivity and decreases as the permittivity is increased. A lossy material has a considerable conductivity, that introduces losses and attenuates the EM waves as they are travelling through the medium, dissipating the energy as heat. Therefore, limiting the penetration depth of EM energy before being absorbed. The attenuation accounts only for one component of the total path loss encountered by an EM wave propagating in the ground, as shown in Figure 2.1. The wave travelling outwards into the medium introduces the spreading losses combined with scattering losses from small features and reflected energy from the ground surface which further reduce the energy propagating into the ground. Other contributors are the dielectric and conduction losses associated with the antenna itself as well as losses due to impedance mismatches between the elements of the transducer.

2.3 MATERIAL PROPERTIES

The dielectric properties describe the behavior of materials under the influence of electric and magnetic fields. The term dielectric is used for poor conducting materials that can be polarised under the influence of an applied electric field with most materials encountered in the subsurface being described as dielectrics. A material holding only bound electrical charges is a perfect dielectric but is never encountered in the subsurface, where all the materials always contain free charges as well. The materials are characterised by the dielectric properties, which are the electric permittivity ϵ, the electric conductivity σ and the magnetic permeability μ.

The electric permittivity is a property of matter which expresses the degree of polarization experienced by bounded charges in a material under the influence of an external electric field. Therefore, expressing the amount of electrical energy

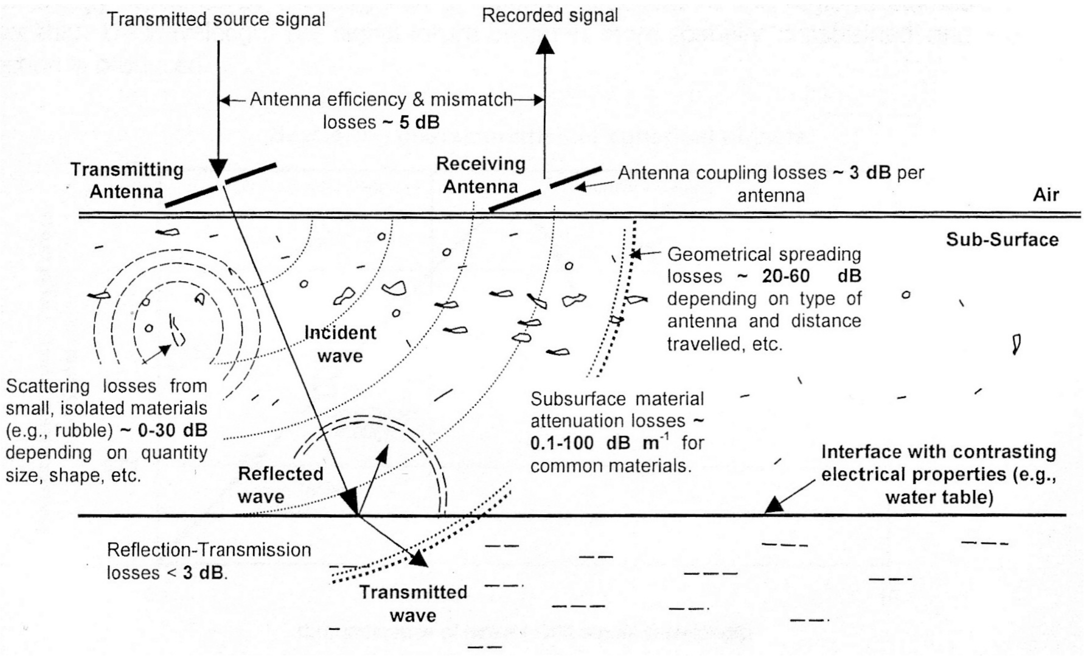


Figure 2.1: GPR signal loss mechanisms (Cassidy, 2008).

that is stored in the material. The permittivity is usually described with respect to the permittivity of free-space $\epsilon_0 = 8.854 \times 10^{-12}(F/m)$ as the relative permittivity ϵ_r given by

$$\epsilon_r = \frac{\epsilon}{\epsilon_0} \tag{2.3}$$

which is a dimensionless quantity. The permittivity of materials commonly investigated by GPR can vary significantly, especially in the presence of water in the medium. It is actually a complex and frequency dependent quantity with a real part that represents the storage and an imaginary part that represents the loss of energy but is usually simplified to a constant static value. Similarly to permittivity, the magnetic permeability expresses how a material responds to an applied magnetic field. It is usually represented as a relative magnetic permeability μ_r with respect to the free-space permeability μ_0

$$\mu_r = \frac{\mu}{\mu_0} \tag{2.4}$$

where $\mu_0 = 4\pi \times 10^{-7}(H/m)$. The electric conductivity describes how easily a material allows electric current to flow through it under the influence of an applied electric field. It explains the ease at which free electrons move in a medium. Materials with high conductivity are known as conductors, while materials with very low conductivity are known as insulators.

The materials that a GPR survey usually comes across are rarely homogeneous and usually are a combination of different materials with each material representing a certain fraction of the total volume of the area of interest. Therefore, when

Material	ϵ_r	**σ (mS/m)**	**u (m/ns)**
Air	1	0	0.3
Clay(dry)	2-20	1-100	0.07-0.21
Clay(wet)	15-40	100-1000	0.05-0.08
Concrete (dry)	4-10	1-10	0.09-0.15
Concrete (wet)	10-20	10-100	0.07-0.09
Fresh water	81	0.1-10	0.03
Fresh water ice	3-4	1	0.15-0.17
Granite (dry)	5-8	0.001-0.00001	0.11-0.13
Granite (wet)	5-15	1-10	0.08-0.13
Limestone (dry)	4-8	0.001-0.0000001	0.11-0.15
Limestone (wet)	6-15	10-100	0.08-0.12
Sand (dry)	4-6	0.001-1	0.12-0.15
Sand (wet)	10-30	0.1-10	0.05-0.09
Sea water	81	4000	0.03
Sea water ice	4-8	10-100	0.11-0.15
Soil (average)	16	5	0.08

Table 2.1: Material properties (Cassidy, 2008).

characterising such a material at a macroscopic level by its dielectric properties, the bulk properties are used that represent the total volume and not each individual component. The bulk properties can be thought as a weighted average of the properties of the materials that synthesise the mixture. Table 2.1 illustrates the properties of commonly encountered materials, as obtained by Cassidy (2008).

2.4 SIGNAL PATHS

The EM waves emitted by a transmitting antenna travel along a number of paths. A brief description of the most important paths is given in this section with the paths illustrated in Figure 2.2.

The direct air wave is the wave that travels directly from the transmitter to the receiver. The direct ground wave refers to the wave that travels along the air-ground interface and then reaches the receiver. Both the direct air and direct ground wave are represented with black arrows in Figure 2.2a. These two waves usually appear as a single response in the recorded GPR data, the direct wave. The direct wave is the first signal recorded and is usually the largest in amplitude.

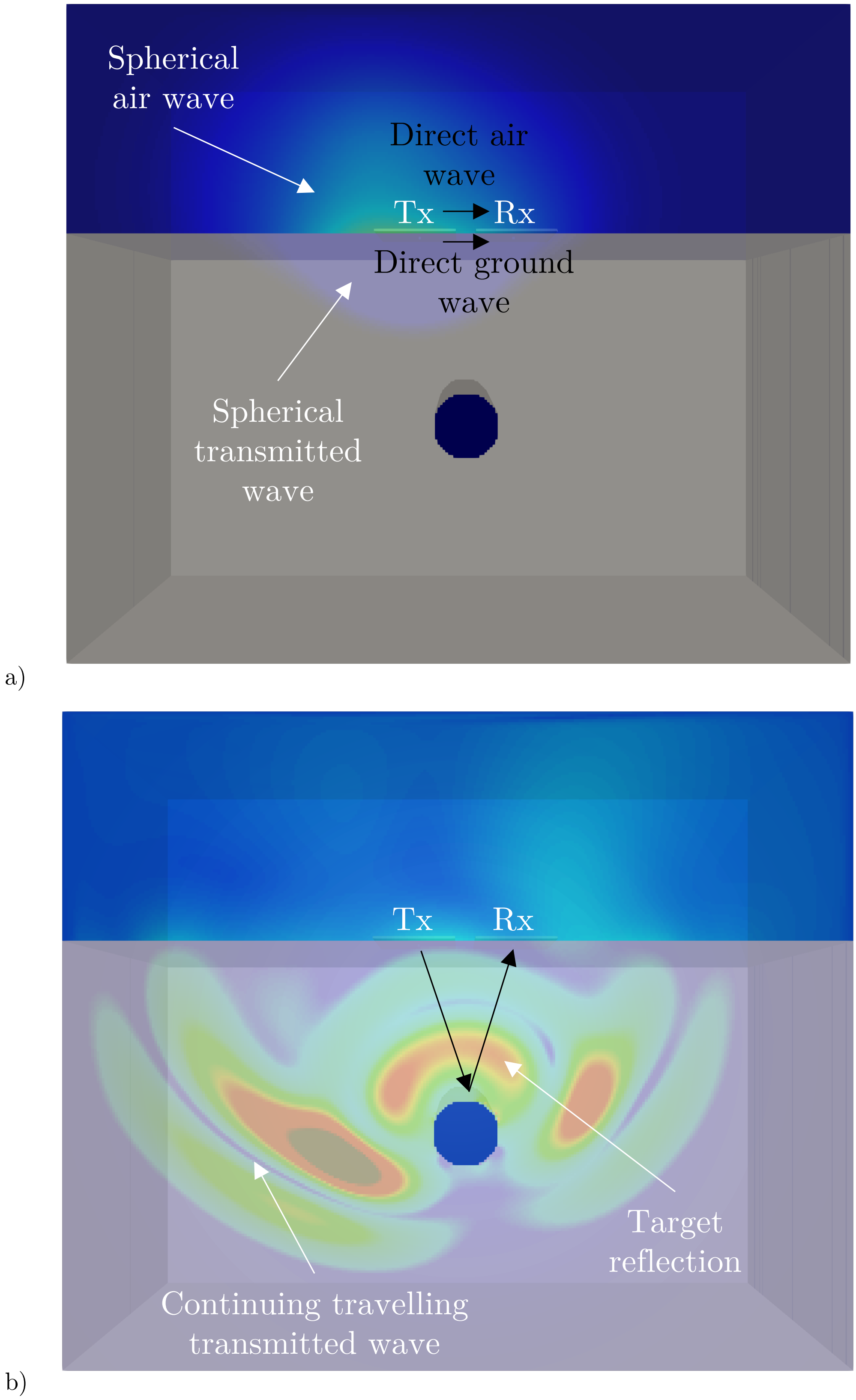


Figure 2.2: Wave propagation and signal paths.

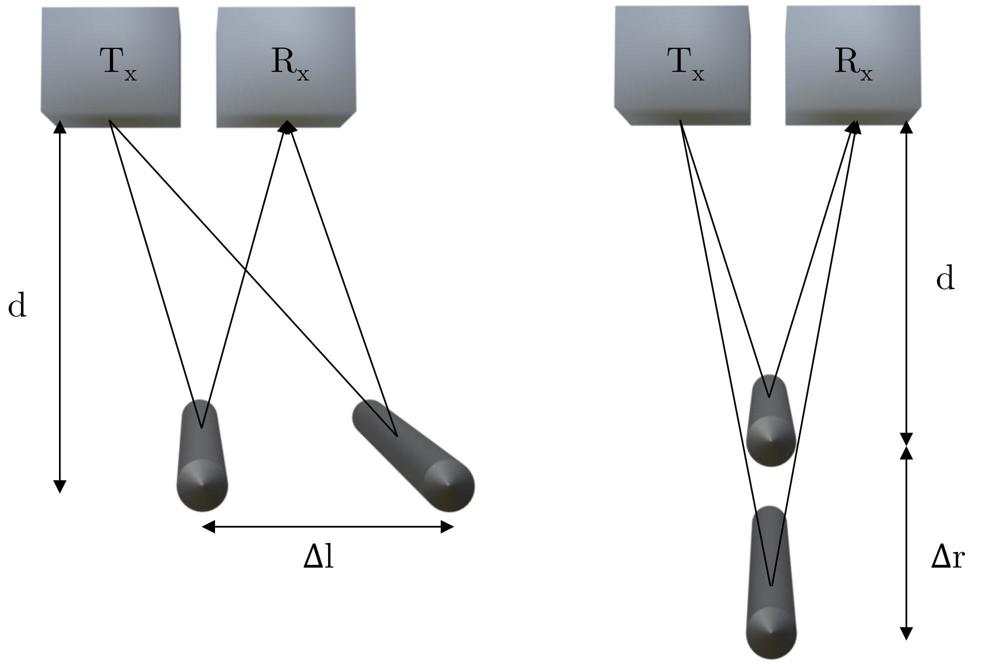


Figure 2.3: Horizontal and vertical resolution by considering two cylindrical targets (Adapted by Annan (2005)).

In case the transducers are lifted at a height above ground or the separation between the transmitter and the receiver is large, the two waves will be recorded as separate events.

Energy also propagates into the ground and when a target is encountered, part of the energy is reflected and travels upwards, where it is recorded by the receiver. The reflected waves from a cylindrical target are illustrated in Figure 2.2b, whereas the two black arrows represent the transmission path from the transmitter towards the target and the reflection path from the target to the receiver. In reality, antennas radiate energy in a volume following a 3D pattern that depends on their type and surroundings as also shown by the “spherical waves” depicted in the Figure and therefore, there are infinitely many paths that the waves can propagate.

2.5 FREQUENCY OF OPERATION AND RESOLUTION

GPR operates over a frequency range of approximately ∼1 MHz to 5 GHz in the microwave frequency band of the electromagnetic spectrum. At lower frequencies the EM fields do not act as waves but become diffusive in character, whereas at higher frequencies substantial losses limit the penetration. The GPR transducers are ultra wide-band systems and emit a range of frequencies and not just a single

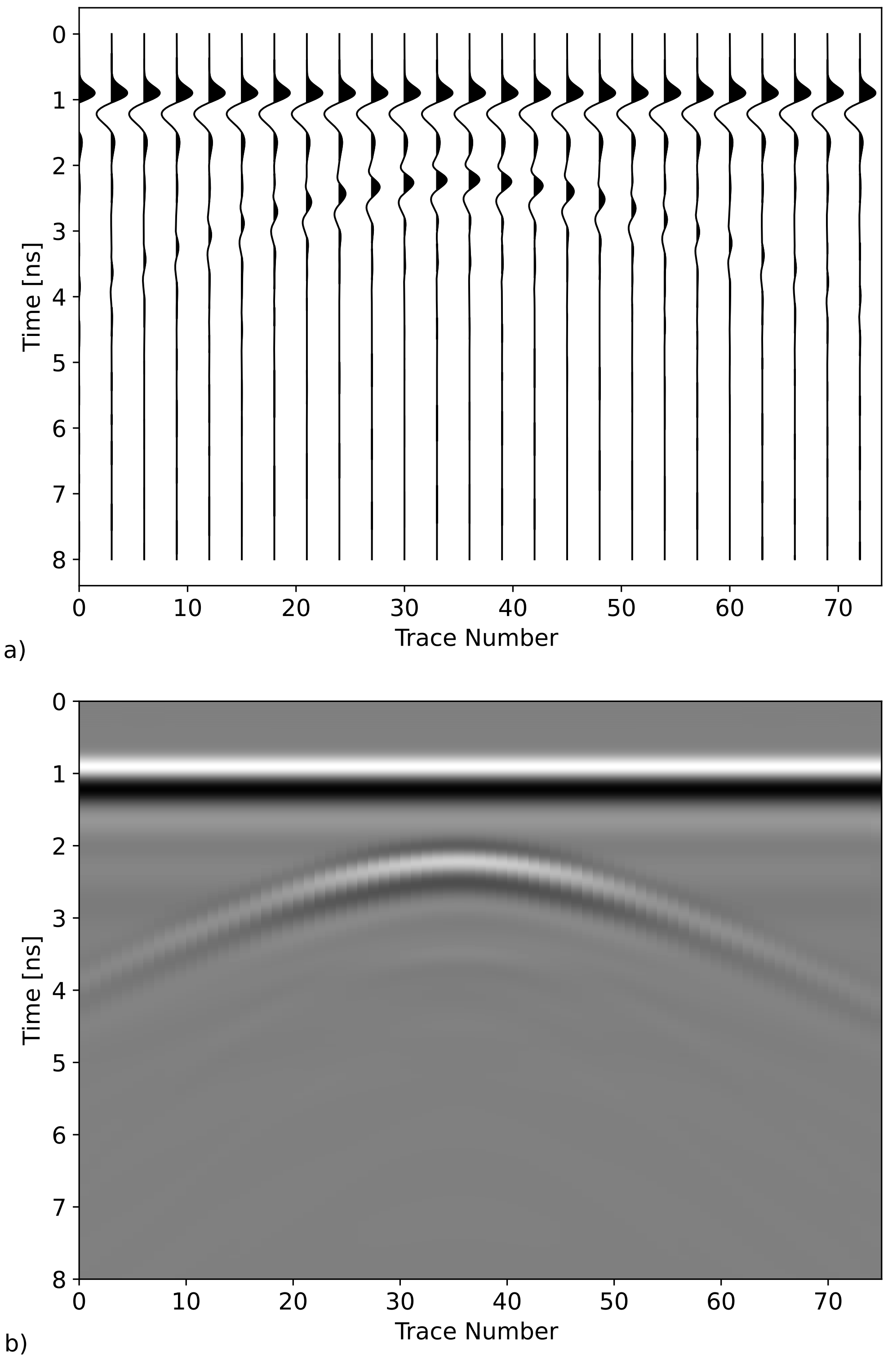


Figure 2.4: a) Plot of multiple A-scans, b) B-scan formed by the A-scans in a) as an image.

frequency. This frequency range is called the bandwidth, B and the GPR systems are usually characterized by the center frequency, f_c of this range. Thus, when referring to a 1000 MHz GPR transducer, means that the center frequency of this system is 1000 MHz.

The frequency of the system controls the trade-off between resolution and depth of penetration. High-frequency antennas have a high resolution, meaning they can resolve small targets but with the cost of limited penetration depth, while low-frequency antennas can "see" deeper but with a low resolution. Therefore, the GPR system that will be used is chosen based on the survey and size of targets being sought. For a given system, resolution determines the size of targets that can be detected and how closely two targets can be in order to appear as two separate events in the received signal. Two pulses from two identical targets must be separated in time by at least half their pulse width. The pulse width, W, is defined as the width at half amplitude and is directly related to the bandwidth as (Annan, 2005)

$$W = \frac{1}{B} = \frac{1}{f_c} \tag{2.5}$$

Translating this into the spatial domain, for the horizontal resolution, the two targets must have a separation distance of

$$\Delta l \geq \sqrt{\frac{udW}{2}} \tag{2.6}$$

where d is the depth of the targets, while for the vertical resolution they must be separated by

$$\Delta r \geq \frac{uW}{4} \tag{2.7}$$

with both depending on the pulse width and the velocity (Annan, 2005). Considering two identical cylindrical targets, the horizontal and the vertical resolution are shown in Figure 2.3.

2.6 DATA COLLECTION

The received GPR signal is a waveform of voltage versus time. This is called an A-scan or a trace or a wiggle and provides information locally, at the vicinity of the measurement point. Collecting A-scans at different measurement points along a line produces a B-scan, which is usually presented as a 2D image in time and space. Each pixel of this image has a color associated with the magnitude of the received signal. A visual representation of A-scans and a B-scan is presented in Figure 2.4, where in a) multiple A-scans are shown and b) illustrates the B-scan formed by these A-scans as an image. A number of parallel B-scans produces a 3D image in the spatial x,y and time t or depth z coordinates. A time/depth slice produces a 2D image in the x,y dimensions and at a certain time/depth, known as a C-scan.

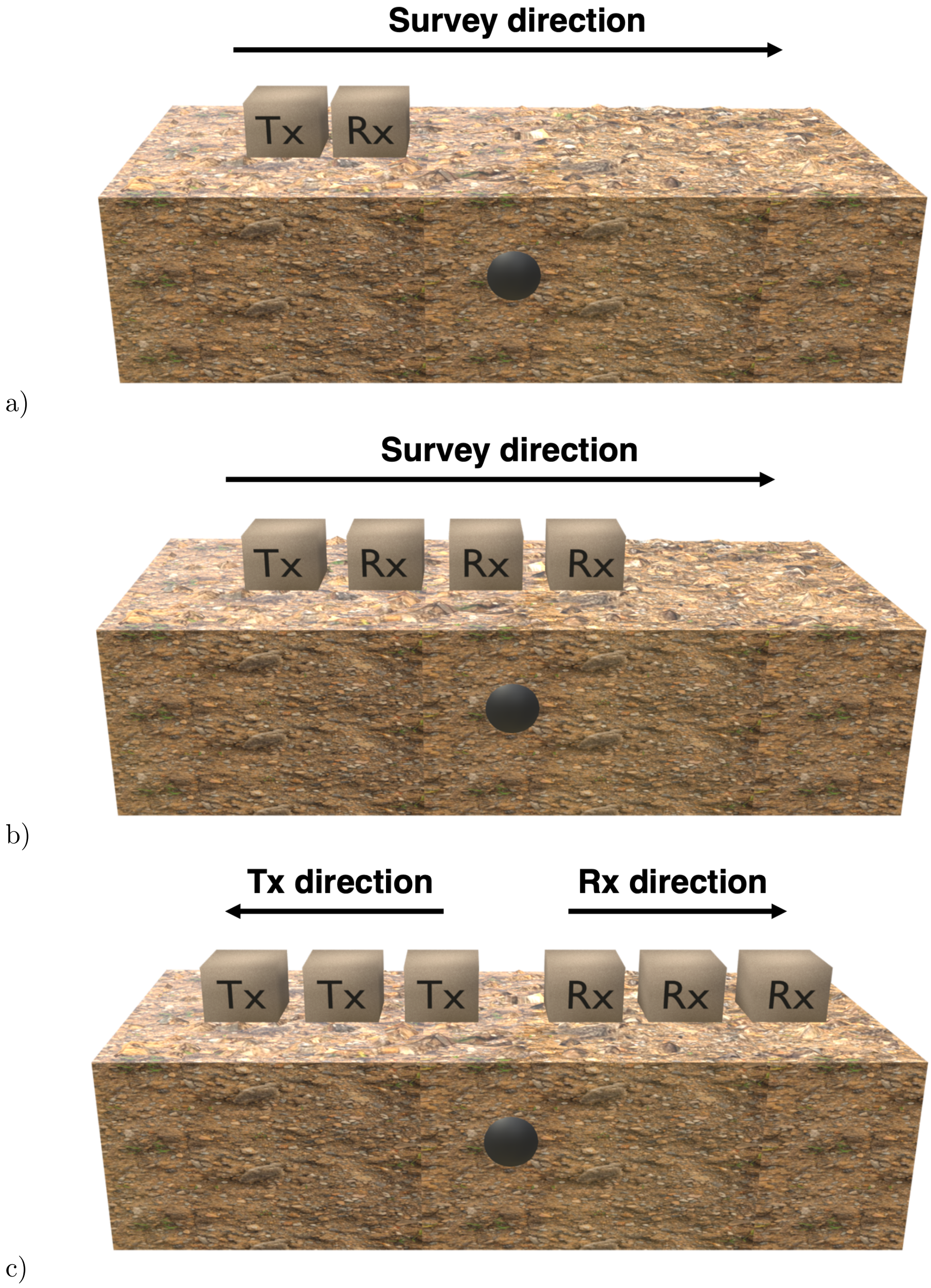


Figure 2.5: a) Common offset method (CO), b) Wide Angle Reflection-Refraction method (WARR) and c) Common mid-point method (CMP).

There are three different modes for data collection: the Common Offset (CO), the Common Mid-Point Reflection (CMP) and the Wide-Angle Reflection-Refraction (WARR) method. Given that in most GPR systems, the transmitting and receiving antenna are in the same enclosure, the CO method is the main choice for most GPR surveys. CO uses a single transmitter and a receiver at a fixed separation distance, which are being moved together along the survey line, as shown in Figure 2.5a. The CMP method, illustrated in Figure 2.5c, employs a transmitter and a receiver having their separation distance gradually increased about a common mid-point and is mainly used to obtain velocity profiles. WARR uses a transmitter placed at a fixed location, while a receiver is moved across the survey line, as illustrated in Figure 2.5b.

In most surveys the data are acquired with the transducers placed directly on the ground in order to maximize energy going into the ground. However, in certain cases such as in landmine detection, the transducers have to be raised off-ground at a certain height from the surface. Furthermore, in case where the transmitter and the receiver are separate, different orientations can be utilised. The three most common orientations are displayed in Figure 2.6, which depend on the polarisation of the transmitting antenna compared to the polarisation of the receiving antenna and with respect to the survey direction. The polarisation of an antenna refers to the direction of the E-field, which is the long axis of a bowtie antenna. In the perpendicular broadside orientation, the antennas are placed next to each other, with the antenna long axis being perpendicular to the survey direction, whereas in the parallel broadside orientation, the antenna long axis is parallel to the scanning direction. In the cross polarised orientation, the antennas are perpendicular to each other. Each orientation performs better in detecting responses from certain types of targets, whereas there is no optimal orientation for equidimensional targets. The most commonly used orientation in GPR surveys is the perpendicular broadside (Annan, 2005).

2.7 GPR EQUIPMENT

A GPR system usually consists of a transmitter to emit the EM energy, a receiver to receive EM energy, along with the control and display unit. The transmitter consists of a source, which is used to generate time varying voltages and an antenna. Antennas transform the electrical signals running in cables to EM waves propagating in space and vice versa. The transmitting antenna translates the excitation electrical signals, which flow on the metallic antenna structure, to EM waves, while a receiving antenna translates the detected EM waves to electrical signals. The transmitter and the receiver are usually identical and can be stored in the same enclosure or as separate units. The antennas are usually, but not always, placed inside a container, called the shield, which is used in order to maximise the energy going into the ground and minimise the energy propagating in the air but also to provide considerable protection from interference. Shielding is more common to high-frequency antennas, while lower-frequency antennas usually come with no shielding due to portability issues rising from the large

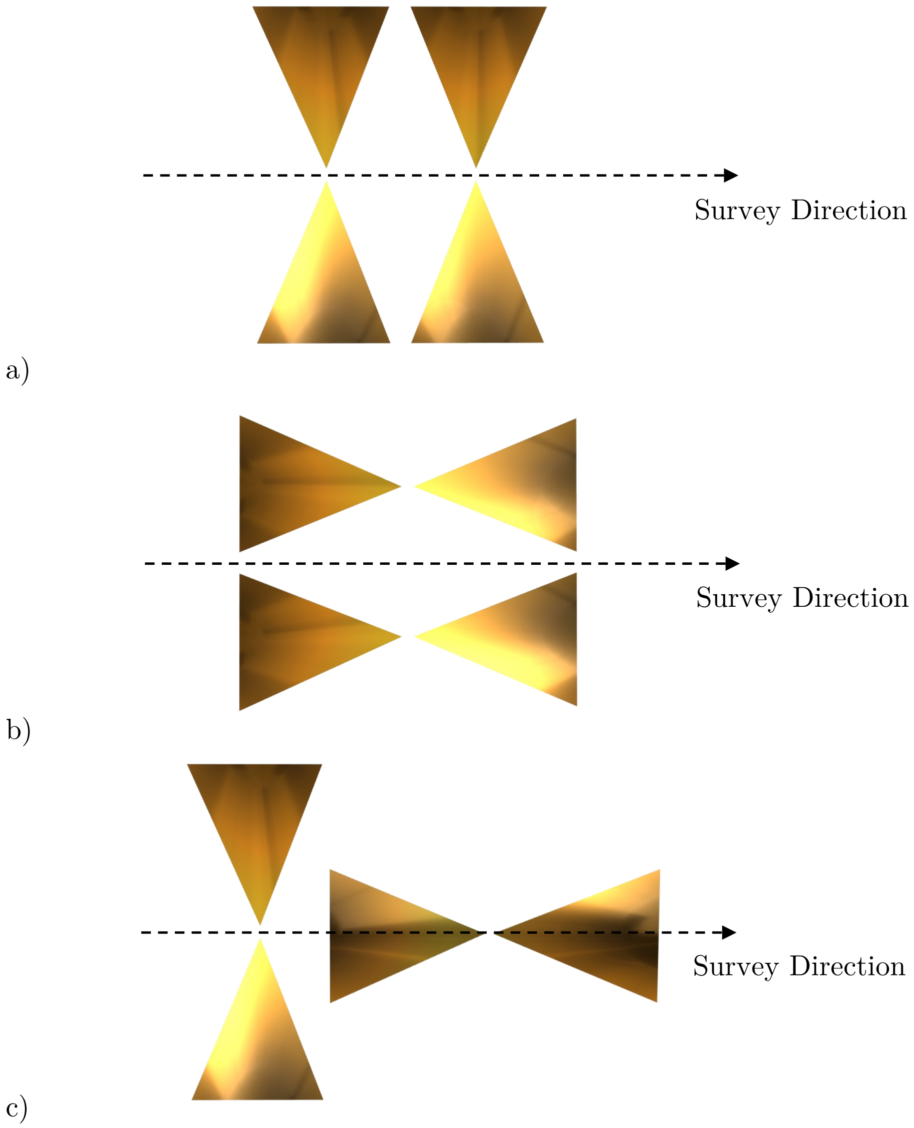


Figure 2.6: Antenna orientations: a) Perpendicular Broadside, b) Parallel Broadside and c) Cross polarised.

transducer size. Even when an antenna is shielded, a portion of the energy can still escape into the air.

The majority of the GPR systems operate in the time domain, where a very short UWB pulse is transmitted. This type of radars are known as impulse. The shape of the pulse is usually a derivative of a Gaussian pulse characterised by a given centre frequency. Frequency domain GPR, such as the Stepped Frequency Carrier Wave (SFCW), have also been developed. A SFCW radar transmits a series of distinct frequencies one at a time. The data are collected in the frequency domain and after collection can be combined and converted to time domain through a Fourier transform.

Since GPR requires a broadband transmission, bowtie antennas which are efficient for broadband applications, are the most common antennas used for GPR. This antenna type, which will be described in more detail at a later chapter, is more broadband than linear dipoles (Balanis, 2015), when unloaded types of antennas are compared. While bowties are used for ground-coupled surveys, horn antennas have also been used for air-coupled applications.

2.8 DATA PROCESSING

To enhance the responses from the targets, remove unwanted noise and clutter and assist in the interpretation, different approaches have been suggested, most of which have been originally developed for seismic data Annan (2003); Yilmaz (2001). In this section, a brief description of the processing methods, that will be implemented in latter chapters, is presented. In addition, to show visually the effect of the processes on the GPR data, processing of a B-scan over a single target, which is shown in Figure 2.7, is performed.

2.8.1 *Time zero correction*

Time zero is the time when the transmitter starts off the signal transmission. To make an accurate interpretation of the targets' depth from the GPR data, knowledge of the time zero is required. However, due to time delays associated with the instrumentation, it is not possible to determine the exact time the transmitter emits (Annan, 2003). Therefore, to compensate for this, a time-zero correction should be applied to shift the response so that the time-zero corresponds to the surface reflection.

Choosing the optimal time zero is one of the problems encountered when processing GPR data and is dependent on the system used and the antenna separation. Commonly, it is performed by measuring the first break, which is the time when the receiver starts detecting the direct signal sent by the transmitter and is detectable in the GPR responses, and then infer time zero by knowing the separation distance between transmitter and receiver. The first break, the positive peak, the negative peak, the zero amplitude point between the first positive and negative peak and the midpoint between the first positive and negative peak of a response have also been suggested as points for time zero correction (Yelf, 2004). Even though the first break is used in many cases for time zero correction, it is different from time zero, except when the transmitter and receiver are at zero offset. Figure 2.8 shows the B-scan data after performing time zero correction by shifting to the first break point.

2.8.2 *Background Removal*

Background removal is one of the main GPR data processing steps, that is used to remove the direct air and the direct ground wave, to suppress coherent system noise or/and remove horizontal events in GPR data. Background reduction is usually performed by subtracting the mean trace of all the traces in a GPR line from every trace in the line (Nobes, 1999) as

$$A(x,1:T) = A(x,1:T) - \frac{1}{M}\sum_{i=1}^{M} A(i,1:T) \tag{2.8}$$

where A is the A-scan, T is the number of time steps and M is the number of collected A-scans. After time zero correction, mean subtraction was performed, as shown in Figure 2.9.

A number of variations of the background average subtraction method exist, including the moving-average background subtraction. In this process, firstly a window of traces is selected, a weighted average of this window is calculated and the resultant trace is subsequently removed from the trace in the middle of the window. Afterwards, this window is shifted by one trace and the process is repeated until the end of the profile is reached. Therefore, each trace is replaced by the weighted average trace of itself and its neighbour traces. This method apart from direct wave removal, can be useful for suppressing localised flat-lying responses and enhancing other localised responses like hyperbolic responses from point targets.

Throughout the present thesis, background removal is performed with the goal of suppressing the direct air and direct ground wave. Frequently in GPR images, the target signatures are masked by the direct air and ground wave responses, due to having a stronger signal. Therefore, the background response should be removed from the total responses in order to enhance other events.

2.8.3 *Time-varying gain*

As the EM waves get attenuated while they are travelling through the subsurface, the returned signal from a target is weaker in amplitude compared to the one from the surface. To compensate for that, time-varying gain can be applied to amplify the responses arriving later at time in order to make them clearly visible (Annan, 2005). This is a non-linear operation implemented with a function that increases the amplification with time in order to compensate for the weaker signals. Being a non-linear function, when applied to data, time-varying gain alters the frequency content and phase of the and therefore the shape and amplitudes of the responses. Figure 2.10 illustrates the B-scan image after an exponential gain was applied on the data, where by comparison with Figure 2.9, it is obvious that gain has enhanced significantly the target response.

Being a non-linear process, time-varying gain if applied on data can have a significant impact on ML. If an ML algorithm is trained on raw data with no gain and a gained trace is given as an input to test the ML model's performance, then the gained input will most likely be misinterpreted as a raw response that was part of the training set, which is similar to the gained trace but corresponds to a different earth model or in worst case as a response that follows a different pattern that then one learned by the ML algorithm. In both cases, the ML algorithm will predict incorrect and maybe unrealistic results. The same applies also in case an ML model is trained with gained data and predictions need to be made on raw responses.

In case an ML model is to be trained using gained responses, the same gain should be applied to the whole training set. Using different gaining functions for each GPR trace in a training set will again degrade the ML performance, since gaining two responses which correspond to different events with a different gain function each can result in similar responses and therefore resulting in the same ML output.

Training with raw data has advantages over training with gained data. Each GPR response, comprised of certain events, requires a different gain function from other responses with different events, making hard to finding a single gain function that can satisfy all responses. In addition, apart from enhancing weaker target responses, gain can also introduce artifacts in the data if applied incorrectly. Amplitude information and relative amplitudes between events in a response are preserved when using raw data, whilst by applying gain these are altered. Furthermore, since with gain a non-linear function is applied on the data, most likely a more complex ML model will be required to capture the relationship between input and output training data, meaning that more parameters would be required resulting in longer computational times.

2.8.4 *Frequency filtering*

Frequency filtering is the most common signal processing technique, where a high-pass filter is utilised to attenuate low-frequency components and a low-pass filter is implemented to remove the high-frequency noise that may be present in the data. Applying both results in a band-pass filter. Real data always contain some form of noise, which in some cases might be extensive and thus, frequency filtering is required. Frequency filtering can be implemented in time domain, such

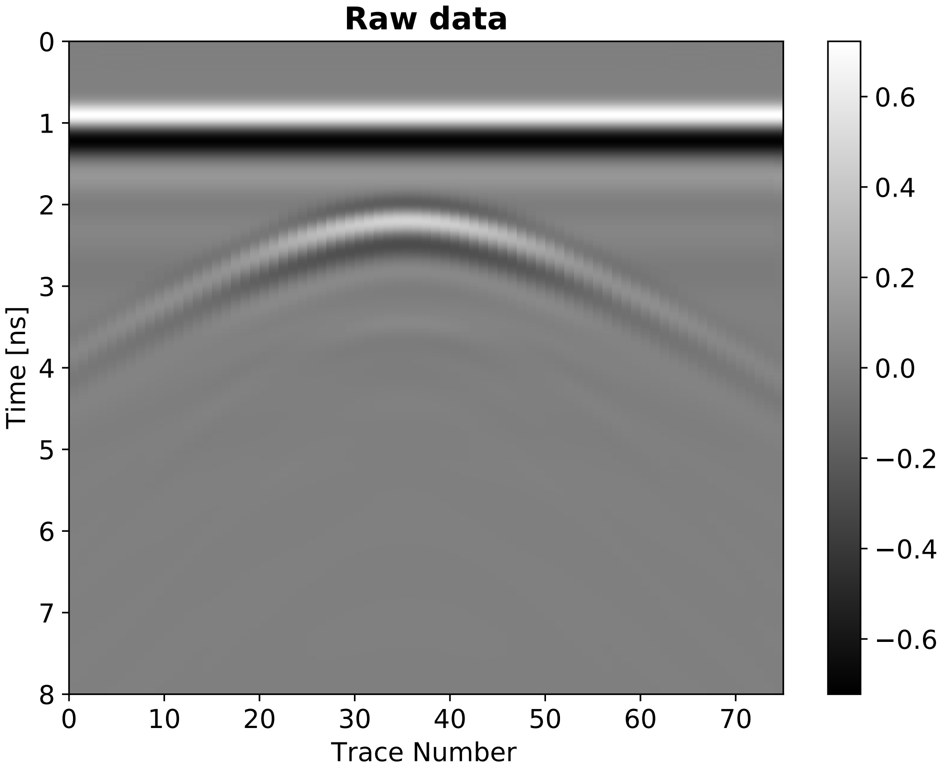


Figure 2.7: Example of a raw GPR B-scan with one target response.

as a moving average smoothing filter but is more commonly implemented in the frequency domain by a Fourier transform

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{i\omega t}dt \tag{2.9}$$

where $f(t)$ represents a signal in the time domain, t is time, ω is the angular frequency and i represents the imaginary unit. Undesired frequencies can be dampen in the frequency domain and the signal can be converted back to time domain using an inverse Fourier transform

$$f^*(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F^*(\omega)e^{-i\omega t}d\omega \tag{2.10}$$

where $f^*(t)$ is the filtered signal in time domain.

2.8.5 *Migration*

In a typical GPR B-scan image, which is collected with the antenna moving along a survey direction, a target appears as a hyperbola due to the different two-way propagation times of the EM wave for each antenna position. While

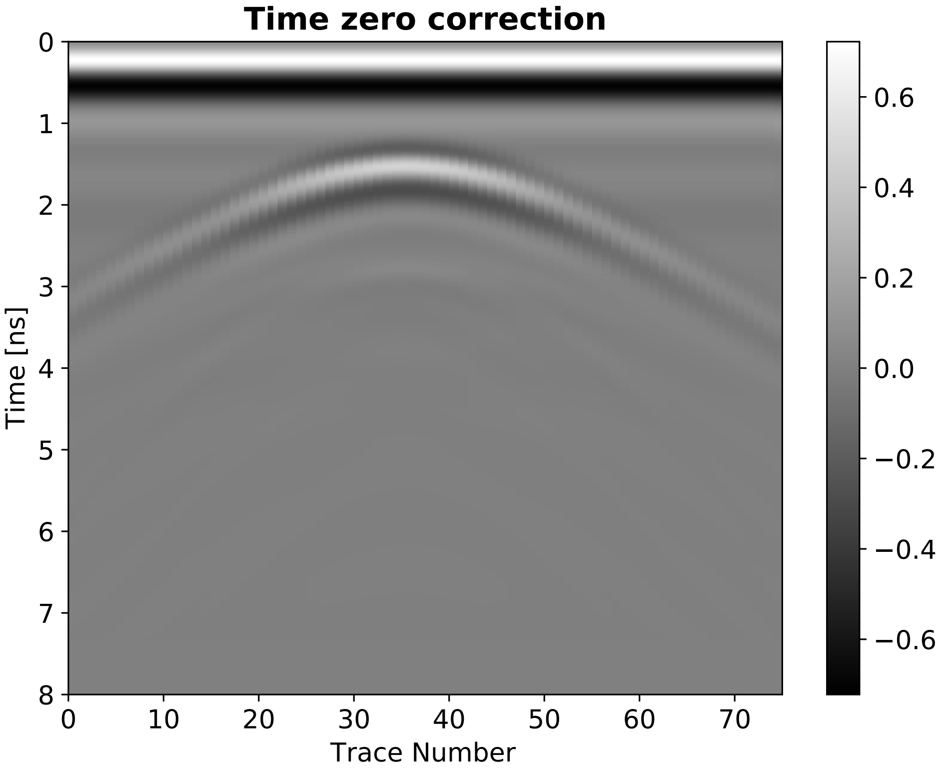


Figure 2.8: Time zero correction of GPR data.

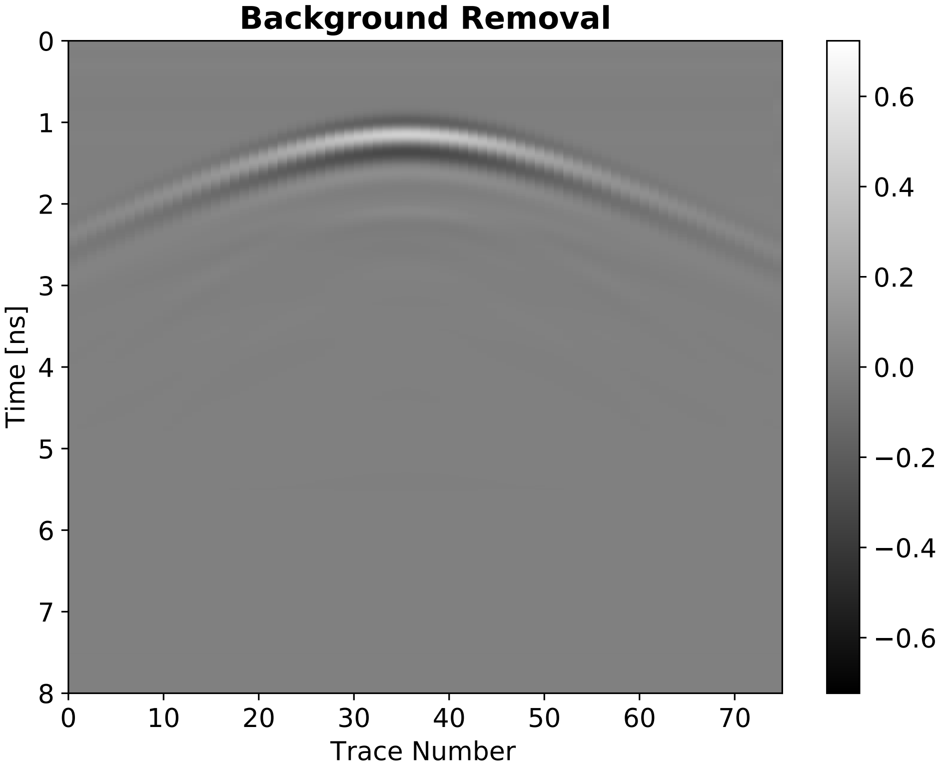


Figure 2.9: GPR data after background removal.

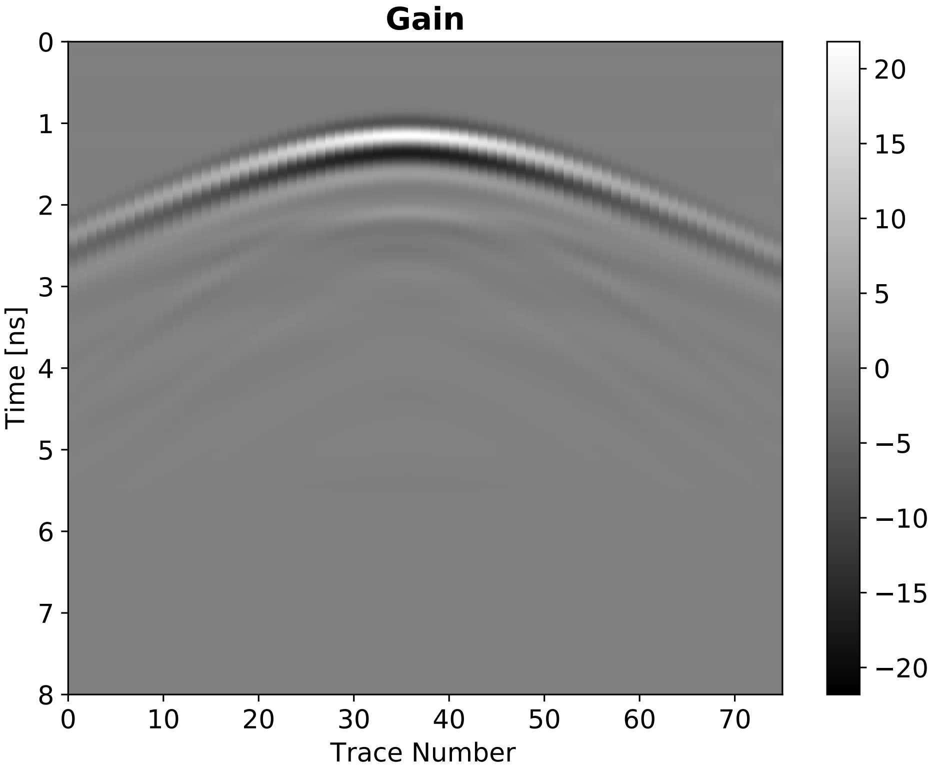


Figure 2.10: Time varying gain of GPR data.

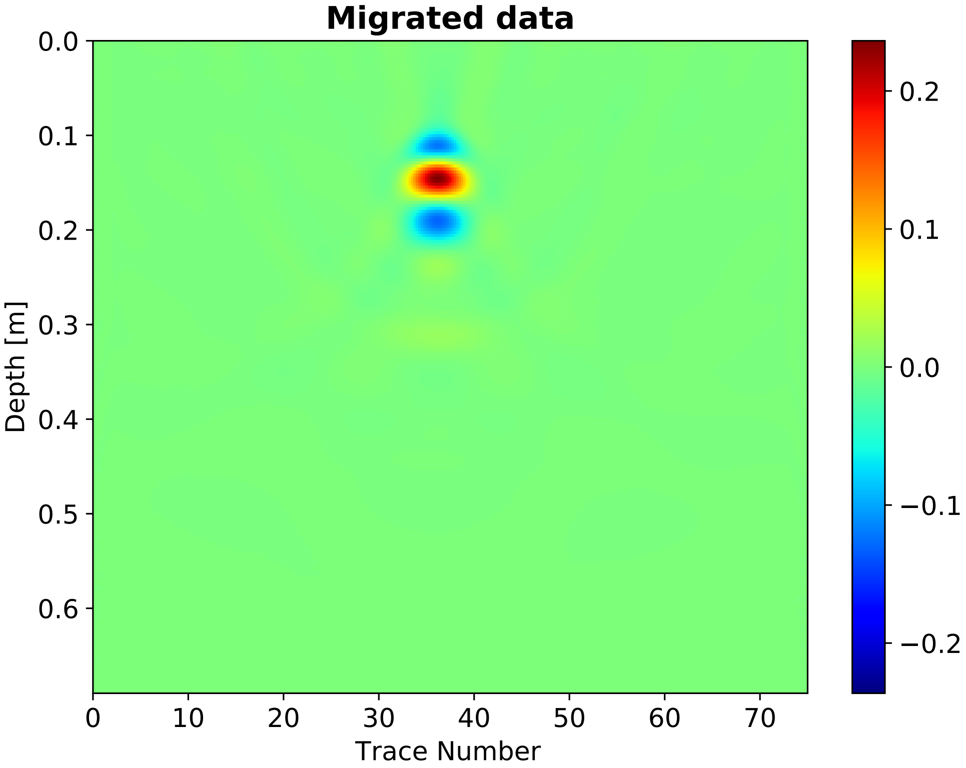


Figure 2.11: GPR data after performing migration.

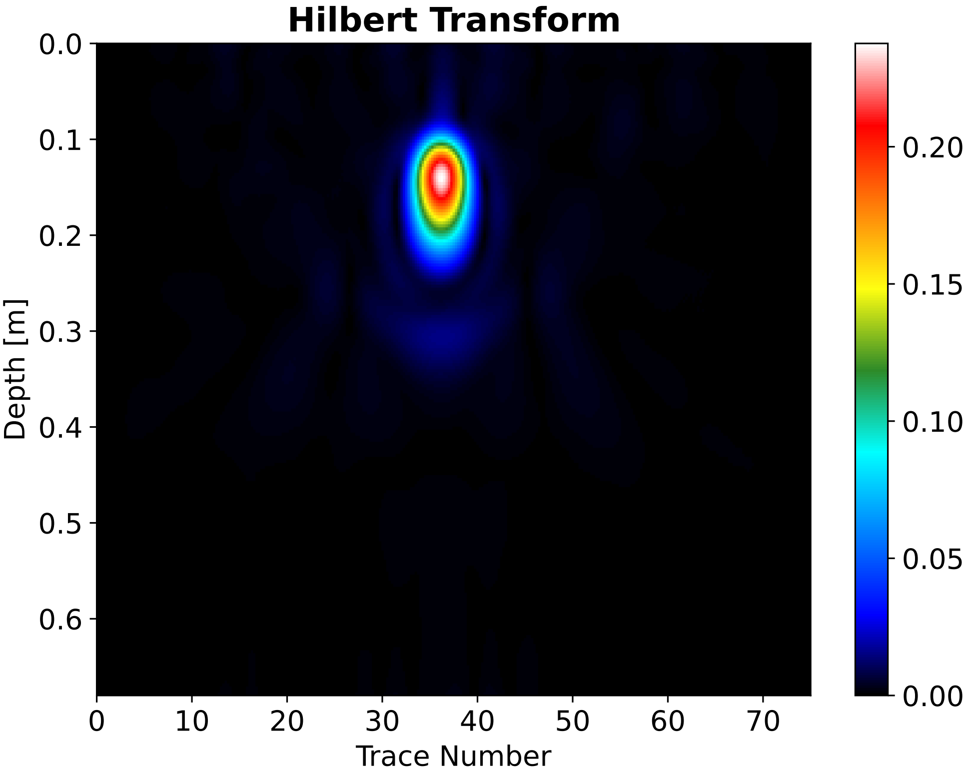


Figure 2.12: GPR data after performing hilbert transform.

processing the GPR data, it is common to correct this by transforming the unfocused B-scan image to a focused one. Migration is an advanced imaging technique used to transform the GPR image to a form more representative of the subsurface structure and more easily interpretable to the human eye. Assuming the EM wave velocity in the medium is known, migration collapses the hyperbolic responses of targets and places them to their correct spatial location. The final reconstructed image of the targets will resemble their true geometrical characteristics. A migrated section of the data is shown in Figure 2.11, where the hyperbolic response from the target has collapsed to its apex. Migration will be described at a later chapter.

2.8.6 *Hilbert Transform*

After migration, a commonly applied processing step is Hilbert transform (HT). With HT essentially, the envelope of a signal is found, where a wavelet with both positive and negative parts is converted to a wavelet with only positive values,

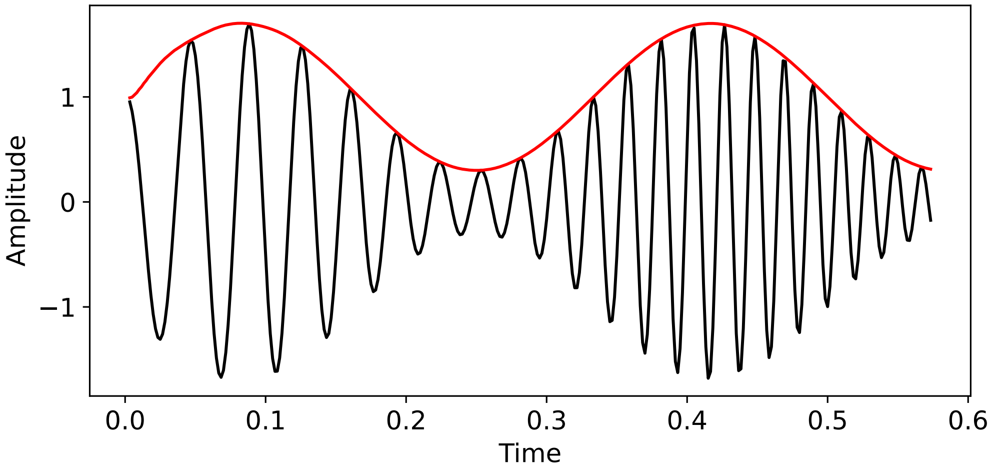


Figure 2.13: A signal plotted in black along with its envelope plotted in red.

which makes the interpretation of the data easier. A complex signal $s(t)$ with real part $a(t)$ and imaginary part $b(t)$

$$s(t) = a(t) + b(t)j \tag{2.11}$$

has its trace envelope or instantaneous amplitude calculated as

$$h(t) = \sqrt{a^2(t) + b^2(t)} \tag{2.12}$$

This generates the envelope and removes the signal oscillations. An example of a signal along with its envelope is illustrated in Figure 2.13. Using HT, the instantaneous phase and instantaneous frequency are also calculated as

$$\phi(t) = atan\left(\frac{b(t)}{a(t)}\right) \tag{2.13}$$

$$\omega(t) = \left(\frac{1}{2\pi}\right)\frac{d\phi}{dt} \tag{2.14}$$

The Hilbert transform of the migrated GPR data of Figure 2.11 is shown in Figure 2.12.

3

MAXWELL'S EQUATIONS AND THE FINITE-DIFFERENCE TIME-DOMAIN (FDTD) METHOD

This chapter introduces Maxwell's equations, which are a set of laws that govern the electromagnetic wave propagation. The effect of the material properties on wave propagation is discussed with emphasis given on dispersive material. The FDTD method along with Yee's algorithm are described, which are used to solve numerically Maxwell's equations. In addition, numerical errors with focus on numerical dispersion and stability and absorbing boundary conditions for FDTD are presented.

3.1 MAXWELL'S EQUATIONS

The electromagnetic wave propagation is governed by a set of four partial differential equations, that were brought together by physicist and mathematician James Clerk Maxwell. These equations are a generalisation of Gauss's law for magnetic and electric fields, Ampere's law and Faraday's law. Maxwell's equations can be written either in integral form or more commonly in their differential form, which is the basis for the derivation of the FDTD method. Here the macroscopic formulation of Maxwell's equations is presented, also known as Maxwell's equations in matter.

3.1.1 *Gauss's Law for electric fields*

Gauss's law for electric fields states that the electric flux through any closed surface S is equal to the total electric charge enclosed by this surface. This is expressed in integral and differential form, respectively, as

$$\oint\!\!\oint \overrightarrow{D} \cdot \overrightarrow{ds} = \iiint \rho dv = Q \tag{3.1}$$

$$\nabla \cdot \overrightarrow{D} = \rho \tag{3.2}$$

where $\overrightarrow{D}$ is the electric flux density (C/m^2), ρ is the charge density (C/m^3) and Q is the total charge (C) surrounded by the closed surface.

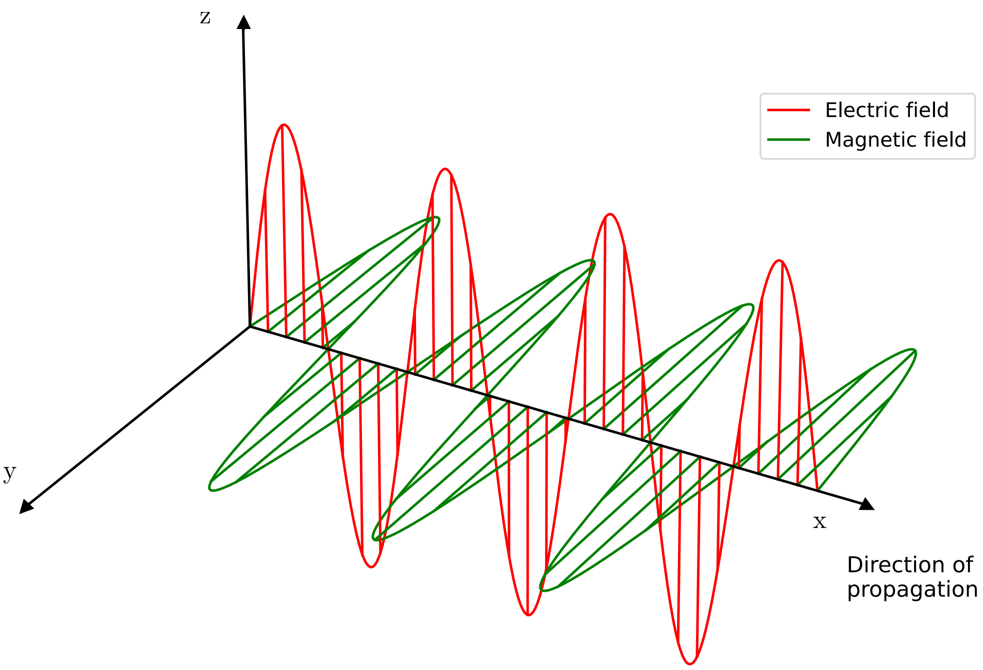


Figure 3.1: An electromagnetic wave propagating. The electric field is shown with red, the magnetic field with green and the direction of propagation lies on the x axis.

3.1.2 *Gauss's Law for magnetic fields*

Gauss's law for magnetic fields states that magnetic monopoles do not exist; magnetic charges appear only in pairs. This is expressed in integral form as

$$\oiint \overrightarrow{B} \cdot \overrightarrow{ds} = 0 \tag{3.3}$$

where B is the magnetic flux density (Wb/m^2), while its differential form is given by

$$\nabla \cdot \overrightarrow{B} = 0 \tag{3.4}$$

3.1.3 *Faraday's law*

Faraday's law describes how an electromotive force (emf) is induced around a loop with changes in the magnetic flux through the loop, expressed by

$$\oint \overrightarrow{E} \cdot \overrightarrow{dl} = -\iint \frac{\partial \overrightarrow{B}}{\partial t} \cdot \overrightarrow{n}\, ds \tag{3.5}$$

$$\nabla \times \overrightarrow{E} = -\frac{\partial \overrightarrow{B}}{\partial t} \tag{3.6}$$

where $\overrightarrow{E}$ is the electric field (V/m). It is very common to add to Faraday's law an equivalent magnetic current density M_c (Taflove and Hagness, 2000)

$$\begin{aligned} \nabla \times \overrightarrow{E} &= -\frac{\partial \overrightarrow{B}}{\partial t} - M_c && (3.7) \\ &= -\frac{\partial \overrightarrow{B}}{\partial t} - M_S - \sigma^* H_S && (3.8) \end{aligned}$$

where M_S is the magnetic impressed source and σ^* is the magnetic loss (Ω/m).

3.1.4 *Ampere's Law*

Ampere's law states that the line integral of the magnetic field around a closed loop is equal to the electric current passing through this loop

$$\begin{aligned} \oint \overrightarrow{H} \cdot \overrightarrow{dl} &= I \\ &= \iint \overrightarrow{J} \cdot \overrightarrow{n} ds \\ &= \iint (\overrightarrow{J_s} + \overrightarrow{J_c} + \overrightarrow{J_d}) \cdot \overrightarrow{n} ds \\ &= \iint (\overrightarrow{J_s} + \sigma \overrightarrow{E} + \frac{\partial \overrightarrow{D}}{\partial t}) \cdot \overrightarrow{n} ds \end{aligned}$$

where I is the electric current intensity (A), $\overrightarrow{H}$ is the magnetic field (A/m) and $\overrightarrow{J}$ is the total current density (A/m^2). The current density $\overrightarrow{J}$ can be divided to the impressed source electric current density $\overrightarrow{J_s}$, the electric conduction current density $\overrightarrow{J_c}$ and the electric displacement current density $\overrightarrow{J_d}$. The source current density results from an artificially added into the system source, the conduction current density accounts for the flow of electrons, while the displacement current density accounts for the displacement of charges. Using Stoke's theorem, the differential form is derived as

$$\nabla \times \overrightarrow{H} = \overrightarrow{J_s} + \sigma \overrightarrow{E} + \frac{\partial \overrightarrow{D}}{\partial t} \tag{3.9}$$

where σ is the conductivity (S/m). Without an external source, this equation reduces to

$$\nabla \times \overrightarrow{H} = \sigma \overrightarrow{E} + \frac{\partial \overrightarrow{D}}{\partial t} \tag{3.10}$$

Changing electric fields create magnetic fields, which in turn create electric fields and this continuing succession results in fields propagating through a material. In open space propagation, the electric field is perpendicular to the magnetic field, while both are perpendicular to the direction of propagation as shown in Figure 3.1, where a propagating EM wave is plotted.

3.2 CONSTITUTIVE EQUATIONS

The constitutive equations are a set of three equations that are used to integrate the dielectric properties of a given material under influence into Maxwell's equations. The three dielectric properties, described previously, are known as constitutive parameters and are used to derive the constitutive equations as

$$\overrightarrow{D} = \overline{\overline{\epsilon}} * \overrightarrow{E} \tag{3.11}$$

$$\overrightarrow{J_c} = \overline{\overline{\sigma}} * \overrightarrow{E} \tag{3.12}$$

$$\overrightarrow{B} = \overline{\overline{\mu}} * \overrightarrow{H} \tag{3.13}$$

where * denotes the convolution operator and $\overline{\overline{\ }}$ denotes a tensor quantity. For many GPR applications, the constitutive parameters are assumed to be scalar constants and the equations are reduced to

$$\overrightarrow{D} = \epsilon \overrightarrow{E} \tag{3.14}$$

$$\overrightarrow{J_c} = \sigma \overrightarrow{E} \tag{3.15}$$

$$\overrightarrow{B} = \mu \overrightarrow{H} \tag{3.16}$$

In general, the dielectric properties represent tensor quantities and are a function homogeneity, isotropy, linearity and dispersivity. Homogeneity describes how a property varies with different position in a material, with homogeneous materials having the same property throughout their extent. Isotropy refers to the situation where the material responds differently with changes in the direction of the applied field. Linearity describes how one property varies with the strength of the applied field. Lastly, dispersivity refers to how a dielectric property changes with frequency. In case the dielectric properties are frequency-independent, the convolution operations are transformed to multiplications.

To understand how the dielectric properties affect the propagation, the wave equation for the electric field is derived. Taking the curl of Equation 3.6

$$\nabla \times \nabla \times \overrightarrow{E} = -\frac{\partial(\nabla \times \overrightarrow{B})}{\partial t} \tag{3.17}$$

and substituting Equation 3.10 and the constitutive relations to the former yields

$$\nabla \times \nabla \times \overrightarrow{E} = -\mu\epsilon \frac{\partial^2 \overrightarrow{E}}{\partial t^2} - \mu\sigma \frac{\partial \overrightarrow{E}}{\partial t} \tag{3.18}$$

Using the vector identity $\nabla \times \nabla \times F = \nabla(\nabla \cdot F) - \nabla^2 F$ and assuming there is no free charge in the region ($\nabla \cdot \overrightarrow{E} = 0$) gives the wave equation for the electric field

$$\nabla^2 \overrightarrow{E} = \mu\epsilon \frac{\partial^2 \overrightarrow{E}}{\partial t^2} + \mu\sigma \frac{\partial \overrightarrow{E}}{\partial t} \tag{3.19}$$

Assuming a time-harmonic field, the wave equation is given by

$$\nabla^2 \overrightarrow{E} - \gamma^2 \overrightarrow{E} = 0 \tag{3.20}$$

which is known as the Helmholtz equation. The constant γ is known as the wavenumber or propagation constant and is a complex value that takes the form of

$$\begin{aligned} \gamma = \alpha + j\beta &= \sqrt{j\omega\mu(\sigma + j\omega\epsilon)} && (3.21) \\ &= \sqrt{-\omega^2\mu\epsilon + j\omega\mu\sigma} && (3.22) \end{aligned}$$

The real α and imaginary β parts are the attenuation and phase constant, respectively. Expanding γ, these can be obtained as

$$\alpha = \omega \sqrt{\frac{\mu\epsilon}{2}\left(\sqrt{1 + \left(\frac{\sigma}{\omega\epsilon}\right)^2} - 1\right)} \tag{3.23}$$

$$\beta = \omega \sqrt{\frac{\mu\epsilon}{2}\left(\sqrt{1 + \left(\frac{\sigma}{\omega\epsilon}\right)^2} + 1\right)} \tag{3.24}$$

The attenuation constant is given in Nepers per meter (Np/m), whereas the phase constant in rad/m. However, the attenuation constant is often described in decibels per meter (dB/m) and can be converted as

$$|\alpha(dB/m)| = 8.68|\alpha(Np/m)| \tag{3.25}$$

The distance the wave needs to travel to reduce its amplitude by a factor of 1/e is defined as the skin depth δ

$$\delta = \frac{1}{\alpha} \tag{3.26}$$

For a time-harmonic wave propagating in the +z-direction, a solution to Equation 3.20 is given as

$$\overrightarrow{E}(z) = \overrightarrow{E}_0 e^{-\alpha z} e^{j(\omega t - \beta z)} \tag{3.27}$$

with a constant phase point of this wave travelling with a velocity of

$$u = \frac{\omega}{\beta} \tag{3.28}$$

which when substituting β results in Equation 2.1, as discussed in the previous chapter.

3.3 THE FINITE-DIFFERENCE TIME-DOMAIN (FDTD) METHOD

To get a better understanding of the GPR responses and acquire information regarding targets, modelling of responses can be employed. Understanding the EM wave propagation can enhance the interpretation of GPR data. Modelling GPR responses in complex realistic media requires the solution of Maxwell's equations, which govern the propagation of EM waves. Solving these equations analytically presents a difficult task and analytical solutions were inferred only for simple scenarios. Therefore, numerical techniques are implemented to acquire approximate solutions to Maxwell's equations. These techniques are referred to as the EM forward solvers. FDTD is a commonly-used forward solver amongst the group of numerical methods used for solving Maxwell's equations. These methods can be classified into two categories, depending on whether they are used to solve the differential or integral form of Maxwell's equations and can be implemented either in time or frequency domain. The differential equations solvers include the FDTD, the Finite Element Method (FEM), the Finite-Volume Time-Domain (FVTD) and the Transmission Line Method (TLM).

FDTD has been widely used to model GPR responses and develop antenna models, as well. A detailed review of FDTD used to create antenna models will be given in Chapter 4.

Before describing the FDTD method, an introduction on finite difference is required. Finite difference approximates a certain function at a point utilising information around that point. To approximate a function, Taylor series expansion is used, where the function $f(x)$ is expanded about the point x with Δx increment as

$$f\left(x + \frac{\Delta x}{2}\right) = f(x) + \frac{\Delta x}{2} f'(x) + \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2 f''(x) + \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3 f'''(x) + ... \tag{3.29}$$

$$f\left(x - \frac{\Delta x}{2}\right) = f(x) - \frac{\Delta x}{2} f'(x) + \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2 f''(x) - \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3 f'''(x) + ... \tag{3.30}$$

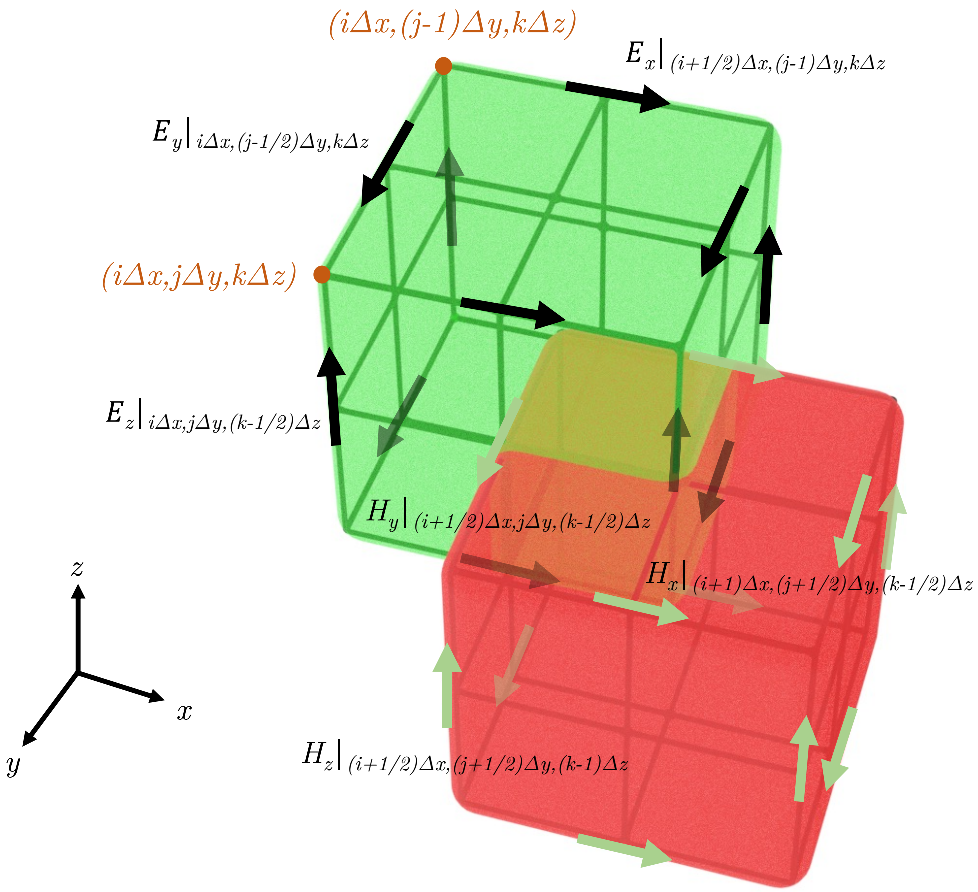


Figure 3.2: The 3D Yee cell with the electric and magnetic field components.

which are Euler's forward and backward series, respectively. Subtracting the second equation from the first and dividing by Δx yields

$$\frac{f\left(x+\frac{\Delta x}{2}\right)-f\left(x-\frac{\Delta x}{2}\right)}{\Delta x}=f'(x)+\frac{1}{3!}\frac{\Delta x^2}{2^2}f'''(x)+\ldots \tag{3.31}$$

Rearranging this equation produces

$$\frac{\partial f(x)}{\partial x}=\frac{f\left(x+\frac{\Delta x}{2}\right)-f\left(x-\frac{\Delta x}{2}\right)}{\Delta x}+O(\Delta x^2) \tag{3.32}$$

where the big $O(\Delta x^2)$ represents the higher-order terms. If Δx is sufficiently small, the derivative can be approximated by discarding the higher-order terms as

$$\frac{\partial f(x)}{\partial x}\approx\frac{f\left(x+\frac{\Delta x}{2}\right)-f\left(x-\frac{\Delta x}{2}\right)}{\Delta x} \tag{3.33}$$

which is the central difference approximation. The big $O(\Delta x^2)$ determines the order of accuracy of the method, which in this case has a second-order accuracy. In the limit as Δx goes to zero, the derivative approximation becomes exact. Using more terms, a higher-order accurate central difference can be constructed. Apart from the central difference, forward or backward difference can be also used for approximating the derivative as

$$\frac{\partial f(x)}{\partial x} \approx \frac{f\left(x + \frac{\Delta x}{2}\right) - f\left(x\right)}{\Delta x} \tag{3.34}$$

$$\frac{\partial f(x)}{\partial x} \approx \frac{f\left(x\right) - f\left(x - \frac{\Delta x}{2}\right)}{\Delta x} \tag{3.35}$$

The FDTD method was proposed in 1966 by Kane Yee (Yee, 1966), as a technique to approximate both the spatial and temporal derivatives of the E-field and the H-field that appear in Maxwell's equations using finite differences. Yee's algorithm solves for both fields, instead of solving for the electric or the magnetic separately using the wave equation. Therefore, resulting in a more robust solution. In his original paper, Yee implemented second-order accurate, central difference approximations. In FDTD, both space and time are discretised in order to solve for the E-field and the H-field at different time-space points. The 3D space is discretised in a grid with a number of cubic cells, which is known as the Yee lattice. The E-field and the H-field are staggered in both space and time. In space, these are staggered by half a cell as shown by a Yee cell in Figure 3.2, where an electric component is surrounded by four circulating magnetic components. Similarly, a magnetic component is surrounded by four electric components. Plugging the finite differences and rearranging results in the update equations for the E-field and H-field, which by solving provide the future fields in terms of the past known fields. The update of these equations is performed using the leapfrog time-stepping method, where the E-field is updated first at all space points using the previously calculated H-field and then the H-field components are updated using the previously calculated E-field. The time is advanced and the above process is repeated until the fields are obtained for a specified duration.

Substituting the constitutive equations into Ampere's and Faraday's Laws yields the following six coupled partial differential equations in 3D

$$\frac{\partial E_x}{\partial t} = \frac{1}{\epsilon}\left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - J_{Sx} - \sigma E_x\right) \tag{3.36}$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\epsilon}\left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - J_{Sy} - \sigma E_y\right) \tag{3.37}$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\epsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - J_{Sz} - \sigma E_z\right) \tag{3.38}$$

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - M_{Sx} - \sigma^* H_x\right) \tag{3.39}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - M_{Sy} - \sigma^* H_y\right) \tag{3.40}$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - M_{Sz} - \sigma^* H_z\right) \tag{3.41}$$

For simplicity, the update equations in 2D will be derived assuming there are also no magnetic losses or magnetic impressed sources. In 2D, two modes exist, the transverse magnetic (TM) and the transverse electric (TE) mode which are decoupled, meaning that their equations contain no common field components and thus, can be set up as two different problems. To demonstrate how the update equations are derived, Maxwell's equation for TM^z mode are used, in which it is assumed that no variation exists in the z-direction with the structure being modelled extending to infinity

$$\frac{\partial H_x}{\partial t} = -\frac{1}{\mu}\frac{\partial E_z}{\partial y} \tag{3.42}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\frac{\partial E_z}{\partial x} \tag{3.43}$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\epsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z - J_z\right) \tag{3.44}$$

Using a second order central difference, the following update equation is derived for E_z

$$\begin{aligned} E_z\Big|_{i,j}^{q+1/2} = E_z\Big|_{i,j}^{q-1/2} + \frac{\Delta t}{\epsilon}\Bigg(& \frac{H_y\Big|_{i+1/2,j}^{q} - H_y\Big|_{i-1/2,j}^{q}}{\Delta x} \\ & - \frac{H_x\Big|_{i,j+1/2}^{q} - H_x\Big|_{i,j-1/2}^{q}}{\Delta y} \\ & - J_z\Big|_{i,j}^{q} \\ & - \sigma E_z\Big|_{i,j}^{q}\Bigg) \end{aligned} \tag{3.45}$$

where $\Delta x, \Delta y, \Delta z$ are the spatial increments and Δt is the time step. A problem with this equation is that the right hand side requires for E_z term in the conduction current to be evaluated at time q, which does not exist in the FDTD grid as the values of E_z are only computed at q+1/2 multiples of Δt. To approximate the correct field value, the values at $(q+1/2)\Delta t$ and $(q-1/2)\Delta t$ are averaged as

$$E_z\Big|_{i,j}^{q} = \frac{E_z\Big|_{i,j}^{q+1/2} - E_z\Big|_{i,j}^{q-1/2}}{2} \tag{3.46}$$

This is known as the a semi-implicit approximation (Taflove and Hagness, 2000). Substituting this equation to equation (3.45) yields the final update equation for E_z

$$\begin{aligned} E_z\Big|_{i,j}^{q+1/2} = \frac{1-\frac{\sigma\Delta t}{2\epsilon}}{1+\frac{\sigma\Delta t}{2\epsilon}} E_z\Big|_{i,j}^{q-1/2} + \frac{\frac{\Delta t}{\epsilon}}{1+\frac{\sigma\Delta t}{2\epsilon}} \Bigg(& \frac{H_y\Big|_{i+1/2,j}^{q} - H_y\Big|_{i-1/2,j}^{q}}{\Delta x} \\ & - \frac{H_x\Big|_{i,j+1/2}^{q} - H_x\Big|_{i,j-1/2}^{q}}{\Delta y} \\ & - J_z\Big|_{i,j}^{q} \Bigg) \end{aligned} \tag{3.47}$$

which can be written also as

$$\begin{aligned} E_z\Big|_{i,j}^{q+1/2} = C_\alpha E_z\Big|_{i,j}^{q-1/2} + C_b \Bigg(& \frac{H_y\Big|_{i+1/2,j}^{q} - H_y\Big|_{i-1/2,j}^{q}}{\Delta x} \\ & - \frac{H_x\Big|_{i,j+1/2}^{q} - H_x\Big|_{i,j-1/2}^{q}}{\Delta y} \\ & - J_z\Big|_{i,j}^{q} \Bigg) \end{aligned} \tag{3.48}$$

$$C_\alpha = \frac{1-\frac{\sigma\Delta t}{2\epsilon}}{1+\frac{\sigma\Delta t}{2\epsilon}} \tag{3.49}$$

$$C_b = \frac{\frac{\Delta t}{\epsilon}}{1+\frac{\sigma\Delta t}{2\epsilon}} \tag{3.50}$$

The arrangement of the electric and magnetic field components in the TMz mode for a single 2D Yee cell is presented in Figure 3.3, whereas the leapfrog scheme for time marching on a TMz grid is illustrated in Figure 3.4. The Figure shows the electric field E_z component, represented with red circles, at times steps t = q+1/2, q+3/2, which is circulated by the magnetic field components half a

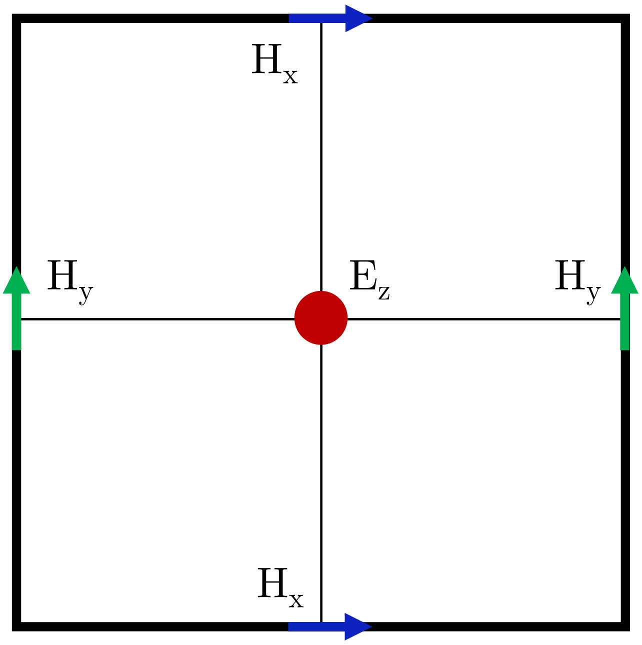


Figure 3.3: Arrangement of electric and magnetic field components for 2D TMz mode.

time step and half a space step away. The H_y is shown with a green circle and the H_x component with a blue circle at time steps t = q, q+1, q+2. In 2D, the following notation

$$E_z\Big|_{i,j}^{q} = E_z(i\Delta x, j\Delta y, q\Delta t) = E_z^q(i,j) \tag{3.51}$$

represents the electric field at time-space point at grid position $i\Delta x, j\Delta y, k\Delta z$ and time step $n\Delta t$. Similarly for 3D

$$E_z\Big|_{i,j,k}^{q} = E_z(i\Delta x, j\Delta y, k\Delta z, q\Delta t) = E_z^q(i,j,k) \tag{3.52}$$

Figure 3.5 shows the same arrangement for the TMz model in the way it is implemented by a computer program, where the coordinates have changed to indices. Although $H_y[i,j], H_x[i,j]$ have the same indices as $E_z[i,j]$, they actually represent the magnetic field components existing on the left of $E_z[i,j]$ in the y and x direction, respectively. The reason for virtually using the same indices is that nodes are stored in the form of arrays in a computer, where the array elements are accessed with integer indices. The magnetic and electric field components are

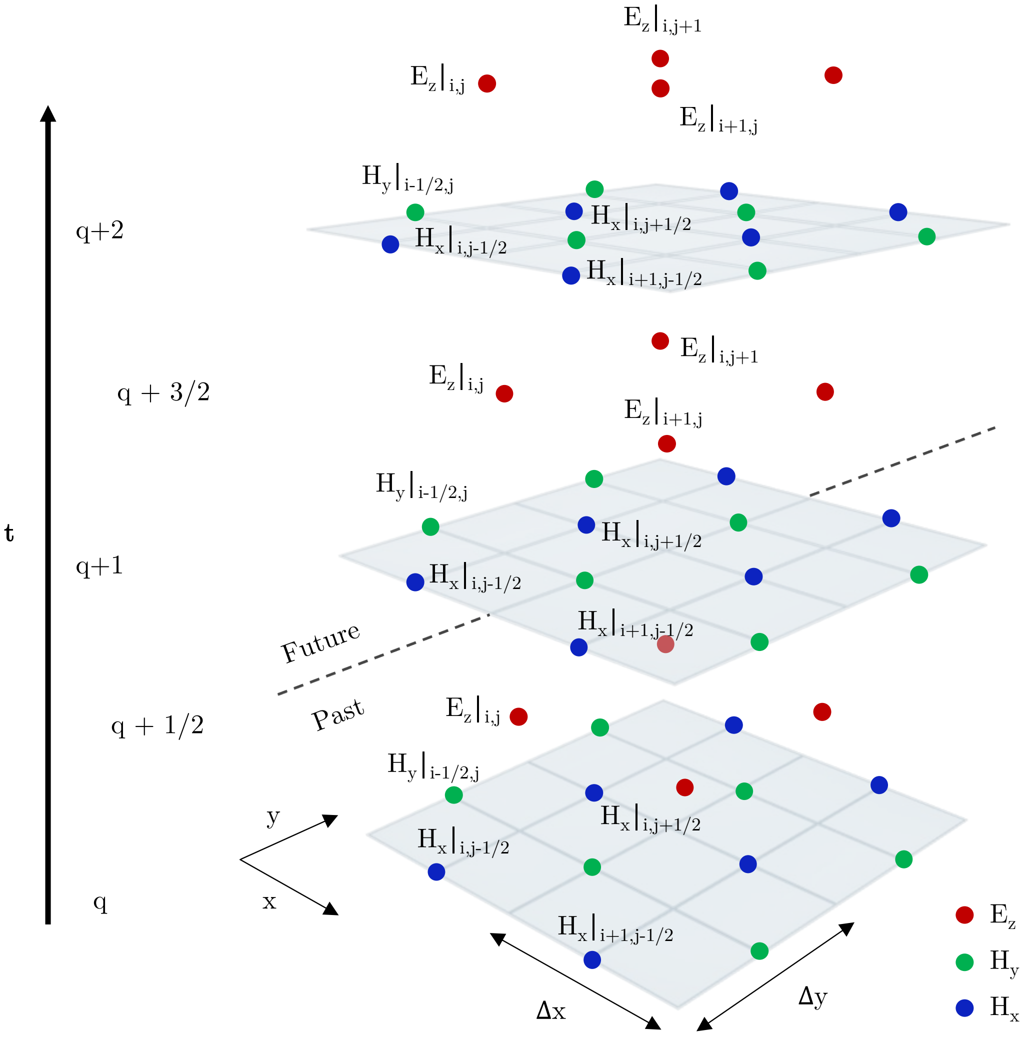


Figure 3.4: Leapfrog scheme for the electric and magnetic field components for TMz mode.

updated at the same iteration, the one after the other. In the same manner, the update equations for the magnetic components are obtained

$$H_x\Big|_{i,j+1/2}^{q+1} = H_x\Big|_{i,j+1/2}^{q} + \frac{\Delta t}{\mu}\left(\frac{E_z\Big|_{i,j}^{q+1/2} - E_z\Big|_{i,j+1}^{q+1/2}}{\Delta y}\right) \tag{3.53}$$

$$H_y\Big|_{i+1/2,j}^{q+1} = H_y\Big|_{i+1/2,j}^{q} + \frac{\Delta t}{\mu}\left(\frac{E_z\Big|_{i+1,j}^{q+1/2} - E_z\Big|_{i,j}^{q+1/2}}{\Delta y}\right) \tag{3.54}$$

The parameters ϵ, σ, μ and σ_m are defined at the point with (i,j) spatial coordinates as $\epsilon = \epsilon_{i,j}$, $\sigma = \sigma_{i,j}$, $\mu = \mu_{i,j}$, $\sigma^* = \sigma^*_{i,j}$. Therefore, different materials can be incorporated in the grid by assigning different values of these parameters in the grid at their corresponding locations.

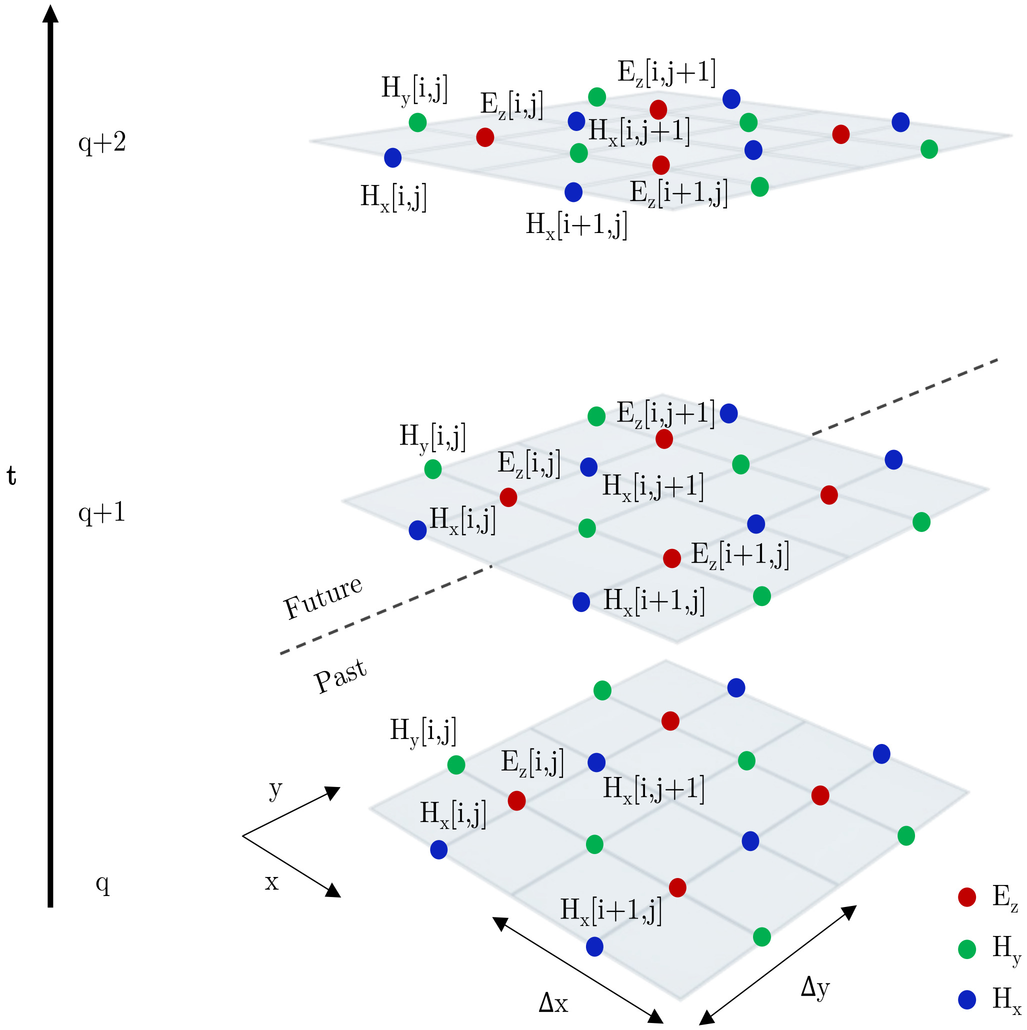


Figure 3.5: Electric and magnetic field components for TMz mode as specified in a computer program.

From the equations presented above, it is evident that when the electric field is updated at time step $q\Delta t$, the magnetic field components are updated at $(q + 1/2)\Delta t$ half a time step away. These equations illustrate that the method provides an explicit solution of Maxwell's equation, where the fields are updated using previously known quantities without requiring to solve equations simultaneously. The scheme for deriving the update equations can be expanded to 3D to obtain the full 3D FDTD update equations (see Appendix A).

3.4 DISPERSIVE MATERIALS

As mentioned earlier, the constitutive parameters of materials are tensor quantities and affected by a number of factors. However, in many problems, it is assumed that the material properties are constant scalar quantities. Having a constant scalar permittivity means that the change in polarisation under the influence of an applied field happens instantly and in perfect proportion with the field. It is also assumed that the permittivity does not change with frequency, direction, position or field strength. However all these assumptions are not true in reality and should be made only in cases where the dependence is small and the effect on the properties insignificant. Most materials that are encountered with GPR can be assumed to be linear and isotropic, since negligible changes in the properties are observed with changes in the direction and strength of the applied field and therefore the constitutive parameters reduce to scalars. Many materials investigated are assumed to be homogeneous as well but that is not always the case, especially in soils. However, for most materials, properties depend and change with frequency, especially with increasing frequency. These materials are characterised as dispersive and the dispersion effect cannot be neglected. This dispersion is a real phenomenon observed in materials and is different from the numerical dispersion discussed at a later section, which is a numerical artifact. According to Taflove and Hagness (2000), there are four types of materials at a macroscopic level based on their behaviour with changing frequency and field strength:

1. Linear materials: Their dielectric properties do not change with frequency of field amplitude.
2. Linear dispersive materials: These are frequency-dependent materials.
3. Non-linear materials: Their dielectric properties do not change with respect to frequency, but only with changes in the amplitude of the field.
4. Non-linear dispersive materials: These are materials which change their behavior with both frequency and field amplitude.

The linear dispersive type of materials is of most interest for GPR applications. Dispersive behaviour of materials investigated by GPR is usually related to the water content of the materials. Although a dry material might not exhibit a dispersive character at GPR frequencies, adding water will lead to a frequency dependent behaviour. A number of permittivity and loss measurements made by Von Hippel (1954) on soils with different water contents showed that the soils were demonstrating dispersive behavior, which was undoubtedly due to the dielectric relaxation of water contained in the soils. Similar measurements were taken by Campbell and Ulrichs (1969) on dry rocks and minerals at two frequencies, 450 MHz and 35 GHz, showed that dry materials have do not exhibit dispersive behavior. The above among other studies demonstrate the effect of the water on materials' behaviour.

Different equations exist that capture the behaviour of the linear dispersive materials with the most popular being the Debye, Lorentz, Havriliak-Negami, Drude and the Jonsher equation. Here only the Debye relaxation is described, which is used throughout the thesis. The Debye model (Debye, 1929) is used to describe the dispersive behaviour of many different materials and especially water. To understand the Debye model , first the electric displacement field or electric flux is presented in a time-harmonic form as

$$\hat{\mathbf{D}}(\omega) = \hat{\epsilon}(\omega)\hat{\mathbf{E}}(\omega) \tag{3.55}$$

$$= \epsilon_0\hat{\mathbf{E}}(\omega) + \hat{\mathbf{P}}(\omega) \tag{3.56}$$

while in time-domain this multiplication is equivalent to convolution, as shown earlier. The polarisation vector $\hat{\mathbf{P}}$ takes into account the displacement of the bound charges in a material. The permittivity and conductivity of dispersive materials are complex quantities, which in frequency domain are given by

$$\hat{\epsilon}_r(\omega) = \epsilon_r^{'}(\omega) - j\epsilon_r^{''}(\omega) \tag{3.57}$$

$$\hat{\sigma}(\omega) = \sigma^{'}(\omega) - j\sigma^{''}(\omega) \tag{3.58}$$

where $\epsilon_r^{'}(\omega), \sigma^{'}$ is the real parts and $\epsilon_r^{''}, \sigma^{''}$ are the imaginary parts of the permittivity and the conductivity, respectively. In a Debye model, the real part of the complex permittivity, which reflects the ability of the material to store electric energy, is given by

$$\epsilon_r^{'}(\omega) = \epsilon_\infty + \frac{\Delta\epsilon}{1+\omega^2\tau^2} \tag{3.59}$$

whereas the imaginary part that accounts for the losses due to relaxation is written as

$$\epsilon_r^{''} = \frac{\omega\tau\Delta\epsilon}{1+\omega^2\tau^2} \tag{3.60}$$

with the total equation for the complex relative permittivity given as

$$\hat{\epsilon}_r(\omega) = \epsilon_\infty + \frac{\Delta\epsilon}{1+\omega^2\tau^2} - j\frac{\omega\tau\Delta\epsilon}{1+\omega^2\tau^2} \tag{3.61}$$

$$= \epsilon_\infty + \frac{\Delta\epsilon}{1+j\omega\tau} \tag{3.62}$$

$$= \epsilon_\infty + \hat{\chi}_e \tag{3.63}$$

where $\Delta\epsilon = \epsilon_s - \epsilon_\infty$ is the difference between the real part of the relative permittivity at zero frequency ϵ_s (static, low frequency permittivity) and at infinite frequency ϵ_∞ (high frequency permittivity) and τ is the relaxation frequency. The term $\hat{\chi}_e(\omega)$ is known as the electric susceptibility function and represents the dispersive part of the relative permittivity. Taking the inverse transform of $\hat{\chi}_e(\omega)$, its time domain form is obtained

$$\chi_e(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{\chi}_e(\omega)e^{j\omega t}d\omega \tag{3.64}$$

$$= \frac{\Delta\epsilon}{\tau}e^{-\frac{t}{\tau}} \tag{3.65}$$

which is used by the time-domain solvers. From the relaxation time, the relaxation frequency can be obtained as

$$f_0 = \frac{1}{2\pi\tau} \tag{3.66}$$

which is the frequency at which the imaginary part of the relative permittivity is at its maximum value. To consider explicitly the losses due to conductivity, a conductive term is added to the Debye model yielding

$$\hat{\epsilon}_r(\omega) = \epsilon_\infty + \frac{\Delta\epsilon}{1+j\omega\tau} + \frac{\hat{\sigma}_{DC}}{j\omega\epsilon_0} \tag{3.67}$$

where σ_{DC} is the dc conductivity. This model is known as the extended Debye model. The Debye model described uses a single Debye pole. For some problems, one Debye pole might not be sufficient to capture the dispersive properties of a material and therefore multi-Debye model is used

$$\epsilon = \epsilon_\infty + \sum_{i=1}^{P}\frac{\Delta\epsilon_i}{1+j\omega t_{0,i}} + \frac{\sigma}{j\omega\epsilon_0} \tag{3.68}$$

where P represents the number of poles used. The behaviour of water is usually represented using a single Debye pole. The real and imaginary parts of the Debye equation for pure water are shown in Figure 3.6. It is evident that for the frequencies that GPR operates in, the real part of the permittivity is almost constant, whereas the imaginary part increases with frequency. The relaxation frequency, for which the imaginary part for pure water reaches the maximum value, is also highlighted in the graph.

3.5 NUMERICAL ERRORS

As with any numerical analysis method, a number of errors are introduced in the solution of an FDTD solver, which should always be taken into account before running an FDTD simulation. Numerical errors are a consequence of the inherent discrete approximations in numerical algorithms executed in a computer to simulate real world continuous problems. In this section, the sources of errors in FDTD are discussed.

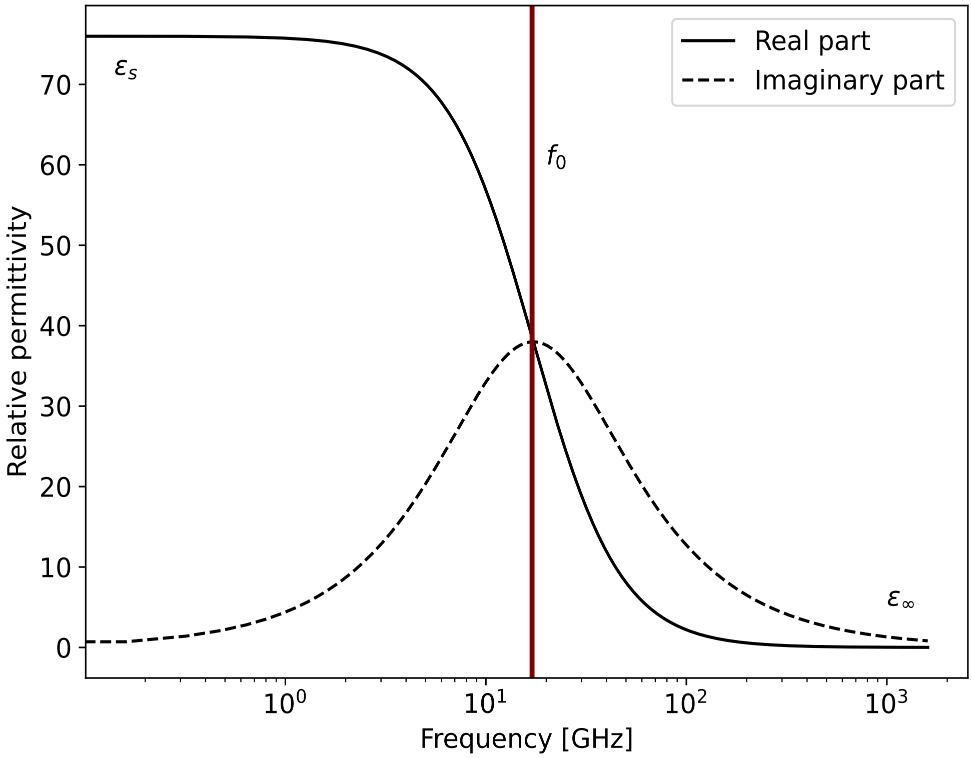


Figure 3.6: Real and imaginary part of a Debye model for pure water.

3.5.1 *Round-off error*

Every computer hardware has a finite amount of bits that it can use to store a number, which is determined by its precision. The system's precision determines which is the smallest and the greatest number that a computer can store and also the number of digits that can be used to represent a number. The difference between the exact number and the finite-precision, rounded number represented by a computer is called a round-off error. The round-off error is a characteristic of a certain computer hardware and will be different for different hardware. With increasingly more calculations performed between numbers, the round-off errors get accumulated, and therefore increase the total round-off error in the solution.

3.5.2 *Truncation error*

Truncation error is the difference between the true solution to the analytical equation and the solution obtained by the numerical FDTD approximation. The truncation error is associated with the order of the method used. The most commonly used FDTD solver is a second-order central-difference scheme, where the error is in the order of the squared step size $O(\Delta x^2)$. This means that the truncation error is affected by the step size used and by decreasing the step size, the error is reduced. A greater accuracy is also achieved by using more terms in the approximation resulting in a higher-order scheme.

3.5.3 *Staircasing error*

The FDTD lattice is based on a uniform rectangular grid. The staircase error arises when a target's interface does not align with the one of the discrete grid. In this case, the target will be modelled with a staircased approximation of its true boundary. For PEC surfaces, this results in delays in the propagating waves as shown by Cangellaris and Wright (1991). The use of an unstructured mesh, although not easy to implement in FDTD, or the use of a finer grid, which will require more computational resources are possible solutions for reducing the staircasing effect.

3.5.4 *Numerical dispersion and stability*

The most important source of error in FDTD is the numerical dispersion. In the previous chapter, the dispersion phenomenon of materials in the continuous world was described. However, numerical dispersion also exists, which is a non-physical dispersion that arises from numerical errors in an FDTD scheme. The numerical dispersion equation in FDTD is found by assuming plane wave solutions in the frequency domain

$$E_z = E_0 e^{j(\omega q \Delta t - \tilde{k}\Delta x)} \tag{3.69}$$

$$H_y = H_0 e^{j(\omega q \Delta t - \tilde{k}\Delta x)} \tag{3.70}$$

where $\tilde{k}$ is the numerical wavenumber, which is different from the true wavenumber. Substituting these equations to the discretised Faraday's law equation yields

$$j\mu \frac{2}{\Delta_t} \sin\left(\frac{\omega \Delta_t}{2}\right) e^{-j\omega\Delta_x/2} H_y = -j\frac{2}{\Delta_x} \sin\left(\frac{\tilde{k}\Delta_x}{2}\right) e^{-j\omega\Delta_x/2} E_z \tag{3.71}$$

Cancelling terms and rearranging gives

$$\frac{E_0}{H_0} = -\frac{\mu\Delta_x}{\Delta_t} \frac{\sin(\frac{\omega\Delta_t}{2})}{\sin(\frac{\tilde{k}\Delta_x}{2})} \tag{3.72}$$

Similarly for Ampere's law, an equation of the ratio of the electric and magnetic field is obtained

$$\frac{E_0}{H_0} = -\frac{\Delta_t}{\epsilon\Delta_x} \frac{\sin(\frac{\tilde{k}\Delta_x}{2})}{\sin(\frac{\omega\Delta_t}{2})} \tag{3.73}$$

Equating equations 3.72 and 3.73 yields

$$\sin^2\left(\frac{\omega\Delta_t}{2}\right) = c^2\frac{\Delta_t^2}{\Delta_x^2}\sin^2\left(\frac{\tilde{k}\Delta_x}{2}\right) \tag{3.74}$$

$$\sin\left(\frac{\omega\Delta_t}{2}\right) = c\frac{\Delta_t}{\Delta_x}\sin\left(\frac{\tilde{k}\Delta_x}{2}\right) \tag{3.75}$$

which is the 1D numerical dispersion equation. Extending to 3D

$$\sin^2\left(\frac{\omega\Delta_t}{2}\right) = c^2\Delta_t^2\left[\frac{1}{\Delta_x^2}\sin^2\left(\frac{\tilde{k}_x\Delta_x}{2}\right) + \frac{1}{\Delta_y^2}\sin^2\left(\frac{\tilde{k}_y\Delta_y}{2}\right) + \frac{1}{\Delta_z^2}\sin^2\left(\frac{\tilde{k}_z\Delta_z}{2}\right)\right] \tag{3.76}$$

Therefore, it is possible to have numerical instability in FDTD, which is the case where the results of an explicit numerical solver spuriously increase as time advances. The choice of the spatial discretisation steps $\Delta x, \Delta y, \Delta z$ and the time step Δt affect the propagation velocity of the numerical EM waves and consequently the numerical error and thus, they cannot be chosen independently. To guarantee numerical stability, Δt must be bounded by the Courant-Friedichs-Lewy (CFL) condition, given in 3D by

$$\Delta t \leq \frac{c}{\sqrt{(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2})}} \tag{3.77}$$

The CFL condition simply states that the maximum acceptable time step in order to ensure stability is governed by the choice of the spatial discretisation step. Physically, this means that the numerical wave cannot propagate more than a single cell per time step in any direction. In 2D, this equation becomes

$$\Delta t \leq \frac{c}{\sqrt{(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2})}} \tag{3.78}$$

and similarly in 1D

$$\Delta t \leq \frac{c}{\sqrt{(\frac{1}{\Delta x^2})}} \tag{3.79}$$

An example of numerical dispersion in 1D is presented in Figure 3.7 based on Schneider (2010). The left plot in the Figure represents a Gaussian wavelet propagating with a discretisation of 20 points per wavelength, $N_p = 20$ and Courant number $S_c = 1$, where the Courant number is defined as the ratio

$$S_c = \frac{c\Delta t}{\Delta x} \tag{3.80}$$

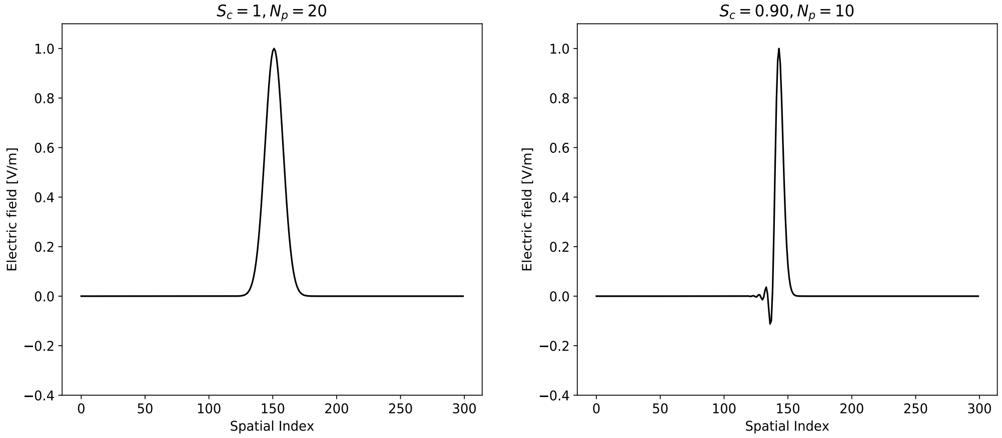


Figure 3.7: Example of electric field propagating with different Courant numbers and discretisations: a) $S_c = 1, N_p = 20$ and b) $S_c = 0.9, N_p = 10$.

In 1D, the numerical dispersion error is reduced with increasing S_c, with $S_c = 1$ being the largest possible value. When S_c is equal to unity, there is no numerical dispersion and the resulting time step is referred to as the magic time step. Therefore, there is no numerical dispersion in Figure 3.7a. In contrast, on the right plot with $N_p = 10$ and $S_c = 0.9$ there is numerical dispersion as shown by the trailing edge of the wavelet.

3.6 ABSORBING BOUNDARY CONDITIONS

Although the subsurface that the EM waves travel into can be considered of infinite extent where the waves travel at great distances until full absorption takes place, the FDTD method requires for Maxwell's equations to be solved in a discretised and of limited size domain. Therefore, certain boundary conditions, called absorbing boundary conditions (ABCs) are required to truncate artificially the computational domain and make it appear as if the FDTD simulation was performed in an infinite extent domain. The simplest method is to set a zero field value at the boundaries by assuming a PEC or a PMC. However, truncating the domain this way, results in unwanted reflections from the boundaries travelling back to the computational domain, which do not naturally occur in the real open space. One way to handle this problem would be to increase the domain size in order for the unwanted boundary reflections to be delayed and not captured within the specified window. However, a large domain size requires substantial computational resources, making this solution inefficient. Analytical absorbing boundary conditions have been suggested for truncating the domain by Mur (1981); Liao et al. (1984); Higdon (1986). The analytical solutions try to predict the fields at the boundaries utilising the field values from the main grid.

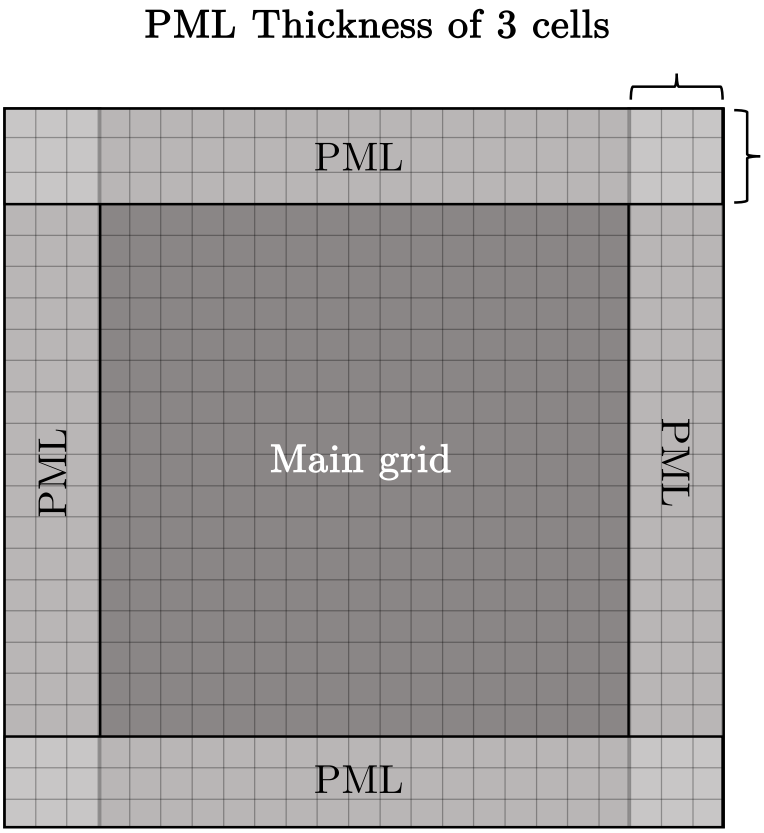


Figure 3.8: Illustration of a PML in 2D with thickness of 3 cells.

The perfectly matched layer (PML) ABC, introduced by Berenger (1994), is currently the most efficient technique for truncating the domain. The PML introduces a non-physical anisotropic lossy material that surrounds the computational domain and is used to absorb the fields incoming to the boundaries. In the direction tangential to the interface between the PML and non-PML region there is no loss, whereas in the direction normal to the interface there is. In his initial implementation, known as split-field PML, Bérenger split each field component into two parts in order to terminate the reflections at the interface between a PML and non-PML region. Around the same time, another approach was suggested by Chew and Weedon (1994) which uses complex coordinate-stretching method along the three cartesian coordinate axis and is considered the most efficient approach for implementing PML. A third approach for employing PML is the uniaxial perfectly matched layer (UPML), which was proposed by Gedney (1996) and is based on a lossy uniaxial medium. Compared to Bérenger's implementation, UPML is based on Maxwell's equations and not a modified set of equations.

A limitation of the above formulations of PML was that they were not able to absorb efficiently evanescent waves. Kuzuoglu and Mittra (1996) introduced the complex frequency shifted PML (CFS-PML), which proved to be more effective in reducing the reflections of evanescent waves (Tong et al., 1999) and also the late time reflections (Abarbanel et al., 2002). An efficient way of implementing CFS-PML was presented by Roden and Gedney (2000), which is based on a complex coordinate-stretching function and a recursive convolution approach approach, and is known as the convolutional perfectly matched layer (CPML). An different implementation of a CFS-PML was proposed by Drossaert and Giannopoulos (2007) for FDTD modelling of elastic waves, which is based on a recursive integration approach. This method is an unsplit implementation, in contrast with conventional split-field PML, and is referred to as recursive integration PML (RPML). The RPML approach was tuned to electromagnetic waves by Giannopoulos (2008), where also a comparison with the CPML was given.

PMLs with different thicknesses (number of cells) can be used, where increasing thickness leads to a better performance but this means that the main grid gets smaller or a larger domain is required which will require more computational resources. An example of a PML along with the main grid in 2D is shown in Figure 3.8, for a PML with a thickness of 3 cells. Furthermore, using a product of different individual stretching functions results in a higher order PML (Giannopoulos, 2012), while the summation of stretching functions lead to the multipole perfectly matched layer (MPML) (Giannopoulos, 2018).

Throughout this work, the PML presented in Giannopoulos (2012) is used to truncate the FDTD computational domain for all models. The geometry and all of the simulations that were used throughout this thesis were conducted using gprMax (Giannopoulos, 2005; Warren et al., 2016), an open-source FDTD solver for electromagnetic (EM) wave propagation that is tuned for GPR applications. The PML presented in Giannopoulos (2012) is also embedded into gprMax.

3.7 SCALE OF THINGS AND SUBGRIDDING

The choice of the spatial and temporal discretisation steps in an FDTD grid is very important. The smaller the steps used, the closer the FDTD model can represent the real problem. Finer grids decrease the overall numerical errors of the method but increase significantly the overall computational requirements and result in longer execution times. Coarser grids require less computational resources but come with a cost in resolution in addition to introducing numerical errors in the FDTD responses. Consequently, the discretization step chosen for a specific model should balance between accuracy and efficiency.

The spatial discretisation step in a model is usually dictated by the smallest feature that is included in the model. The smaller the feature, the finer will be the grid required to adequately model this feature. Furthermore, the near-field requires a finer resolution than the far-field because it has higher spatial variability compared to the far-field. In addition, as mentioned previously, finer grids can minimise errors caused by staircased approximations of feature boundaries. A model usually consists of many different parts (e.g layers, targets) some of which can be accurately resolved using a coarse grid, whereas others require a finer grid. Using a uniform small spatial step to calculate the FDTD solution for the whole model leads to longer computational times. Instead, subgridding techniques have been devised, where smaller grids with a finer spatial step are introduced into a coarser grid, with the subgrids being placed in parts of the FDTD model, that require a finer mesh to be resolved. Since a subgrid has a different spatial step from the main grid, it needs to have its own time step, which will be different from the time step of the main grid, since the subgrid and the main grid need to satisfy a different stability condition. Another challenging issue that subgridding faces is the treatment of the interface between the main grid and the sub-grid, which is typically handled using interpolation methods to interpolate the EM fields that exist in the vicinity of the interface (Kim and Hoefer, 1990; Zivanovic et al., 1991; Prescott and Shuley, 1992).

4

DEVELOPING A NUMERICAL MODEL OF A REAL GPR TRANSDUCER

In this chapter, the development, optimisation and validation of a digital equivalent of a true commercial GPR system is presented. A grid convergence analysis is performed and the antenna radiation patterns are given. Finally different units from the same transducer are modelled and comparison is made between the two. Before describing the antenna modelling process, a review on antenna modelling is provided.

4.1 REVIEW OF ANTENNA MODELLING

A typical GPR case study consists of the background medium, the targets and the antenna system itself. Therefore, accurate representations of these components should be included in the simulations. Even though the transducer plays an important part, nonetheless, it is a common practice amongst GPR practitioners to use a theoretical source in most GPR models instead of an accurate digital representation of the actual GPR system. The use of simplistic theoretical sources , such as infinitesimal dipoles, is helpful for simulations that reproduce overall patterns in GPR scans or for investigating the far-field of the antenna but the resulting A-Scans using such simple sources differs significantly in a number of key aspects from what is being observed in the real GPR measurements.

Until the early 2000s, most of the antenna models used in the simulations were either theoretical sources, simple or custom-built antennas. Simple theoretical antennas, such as monopoles and dipoles, correspond to a well-defined theoretical EM problem that has been extensively analysed (Balanis, 2015; King, 1956). The FDTD method was also used to analyze a cylindrical and a conical monopole by Maloney et al. (1990) where the antennas were driven through an image plane from a coaxial transmission line. The modelled FDTD results showed a very good agreement with experimental measured data. A 2-D FDTD modelling with a theoretical source was used in Oristaglio and Hohmann (1984) to simulate transient electromagnetic (TEM) surveys. A hybrid method was presented by Monorchio et al. (2004) that utilises FDTD, FE and MoM methods to model thin-wire antennas used to investigate inhomogeneous dielectric targets with arbitrary shapes. Each technique was used to model different parts of the problem, while some late-time instabilities of the scheme were acknowledged.

Many researchers have developed and numerically modelled simple custom-made antennas. Shlager et al. (1994) used FDTD modelling to optimise a bow-tie antenna and improve its radiation. The 3D structure was attached to a 1D transmission line feed and used a differentiated Gaussian pulse as an excitation. Continuous resistive loading was used to optimise the bow-tie antenna. A custom-built antenna was created based on the optimised design and its measurements showed a good agreement with the FDTD results. FDTD was also employed by Lestari et al. (2004) to assist in the designing of a circular-end bow-tie GPR antenna with improved radiation efficiency. Similarly, loading is used for higher efficiency with a combination of a constant resistive loading and a linearly increasing capacitive loading is used. An FDTD model of s spiral GPR antenna system was developed by McFadden and Scott (2009) and compared with a prototype system. The transmitter and the receiver of the system included an Eccosorb AN-79 RF absorber, where a Debye model was fitted to acquire its dielectric properties and approximate its behavior in the simulations. The FDTD results were compared with the measured to determine the best-fit Debye model. The FDTD responses for free space, a half-space and over a known-target corresponded well to the real signals. Following the work presented in Lestari et al. (2004), Lestari et al. (2010) optimised a wire-structured loaded bow-tie antenna to exhibit higher amplitude and lower level of late-time ringing. A numerical equivalent of the antenna was developed and was shown that the computed transmitted waveform matches the measured one. In addition, the analytical expression which approximates the time-harmonic current distribution of the proposed antenna is derived.

The first fully 3D FDTD realistic representation of a one-third size scale model of an actual GPR system was presented by Bourgeois and Smith (1996). The model included resistively loaded bowtie transmitting and receiving antennas shielded in a metallic case. Feeding of the antennas was implemented using parallel-wire transmission lines. Synthetic responses were compared with measurements for signals from different types of buried pipes, showing an agreement between the two. The authors acknowledged that their results are valid only for the early times, since a perfect electric conductor (PEC) wall was used to truncate the domain, which introduced unwanted reflections in the data. Roberts and Daniels (1997) used 3D FDTD to approximate coaxial cables and a GPR antenna. The feed was modelled as two square coaxial cables and the antenna model was a 300 MHz unshielded bowtie antenna. The radiation patterns of this antenna model were compared with measured data from Wenskink et al. (1990) above water and from Wenskink et al. (1991), where pipes of different composition were submerged in water. The results showed some agreement, however no comparison of amplitude data was made.

A similar antenna model to Bourgeois and Smith (1996) was analysed by Nishioka et al. (1999) which used FDTD to investigate the characteristics of the system. In this case, the inner walls of the rectangular metallic cavities were coated with a frequency dependent ferrite absorber and Berenger's perfectly matched layer (PML) was implemented to truncate the computational space. Lampe and Holliger (2001) employed 3D FDTD with a general perfectly matched layer GPML to numerically model a similar antenna system. The algorithm allows also for the use of subgrids. The input impedance of the synthetic data was compared with laboratory measurements of corresponding antennas, demonstrating a high level of agreement. Furthermore, the antenna radiation patterns for lossless and lossy half-spaces were provided for both shielded and unshielded antenna cases. This antenna model was further improved by Holliger et al. (2003), Lampe et al. (2003), where damping was introduced not only using resistive loading but also with the use of absorbing material in the cavity. However, the simulated results were not compared with responses from an equivalent real system.

Lambot et al. (2003, 2004) proposed a SFCW system with a TEM horn antenna tuned to be used for off-ground applications in monostatic mode, which was designed using a vector network analyzer (VNA). Forward modelling was performed based on linear system transfer functions and the exact solution of the 3D Maxwell's equations in a medium composed of arbitrary number of different horizontal homogeneous layers. Measurements and simulations were compared in air and above a metal sheet, validating the model.

An UWB horn-fed bowtie antenna was modelled and designed by Lee et al. (2004), where modelling was used to optimise certain antenna parameters and validate the performance of the antenna design. The model includes realistic features such as feeding cables, dielectric loading and tapered resistive loading. The effects of the ground properties in impedance were investigated, since the antenna is designed for GPR applications. A 3D FDTD simulation, which included a realistic representation of an antenna system was described by Uduwawala et al. (2005). The transmitter and the receiver were enclosed in rectangular conducting cavities and two lump resistors were connected at each the ends of each bow-tie arm. Simulations were performed for two realistic soil types, lossy and dispersive using a Debye model, however the FDTD results were not compared with responses from a real system. Chen and Liu (2010) modelled and developed an experimental 900 MHz antenna system. Edge cutting bow-tie antennas were enclosed in a metallic case with absorbing material inserted in the back-cavity. Measured data using the real system were provided but they have not been not compared with simulated data.

Over the last years, more sophisticated antenna models have been developed due to the advancements of algorithms and computers that made it possible and reduced the overall computational times. In order to understand how the electromagnetic waves propagate in materials of interest using numerical modelling, a model of a real GPR transducer, that is used in practice, must be included in the simulations. Modelling of a commercial GPR transducer was first attempted by Klysz et al. (2006), where a 3D FDTD simplified model of the GSSI 1.5 GHz antenna was developed. The model consists of the transmitter antenna only and does not include the receiver. In order to create the model comparisons between synthetic and measured responses were made in free space at four points around the transducer. Values for the dielectric layers in the cavities of the box and the resistance of the voltage source at the feed point were determined through trial-and-error. Two concrete slabs of different water content were used to validate the antenna model. Comparing the results showed that both the free space and the concrete slab real responses were approximated with a good accuracy by the synthetic responses.

Warren and Giannopoulos (2011) developed 3D FDTD models of two commercial antennas, based on a 1.2 GHz antenna from MALÅ Geoscience and a 1.5 GHz antenna from GSSI. The geometrical features of the transducers were considered known, while values for the unknown properties of the materials in the transducers were obtained using Taguchi's optimisation method. A Gaussian-shaped pulse was used an an excitation. To evaluate the performance of the optimisation and update the parameters of interest, the error between the synthetic and the real free space response was used as a criterion. The optimisation resulted in a good agreement between the modelled and the real responses, while a series of oil-in-water emulsions were used to further validate the models resulting again in a good match. In Giannakis et al. (2019a) the model of the GSSI 1.5 GHz was further improved using a linear/non-linear full-waveform inversion (FWI) scheme, which updates simultaneously the dielectric properties of the antenna elements and in addition, derives an optimised excitation pulse. This optimised pulse can be regarded as a correction term that accounts for the features that could not be modelled, either due to lack of information, such as the unknown true excitation pulse, or due to general limitations of the scheme. Two different complex scenarios were used to assess the accuracy of the scheme, showing that the model was able to replicate the behaviour of the real transducer successfully.

An antenna model based on a 400 MHz GPR transducer from GSSI is presented in Stadler and Igel (2018). The system consists of shielded bowtie antennas, where estimates of unknown key antenna properties obtained using Taguchi's optimisation. In this study, GPR waves were guided in boreholes, where the antenna system was located next to a borehole with a metallic cylinder inserted inside.

4.2 ANTENNA THEORY

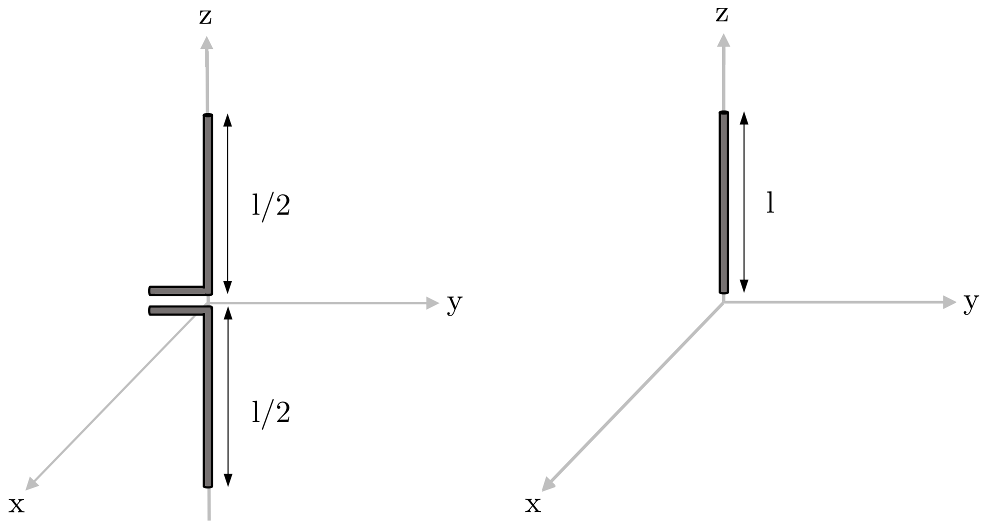
The basic elements of a GPR transducer is the antenna. Dipoles and monopoles are the simplest types of antennas, which belong to the category of wire antennas. A dipole antenna has two symmetrical radiating arms, which are linear metallic wires or rods with a feed point at the center. A monopole antenna has a single radiating arm and is most often used above a ground plane. Figure 4.1 illustrates examples of dipole and monopole antennas. An infinitesimal dipole or hertzian dipole is a theoretical dipole whose length is far less than the wavelength ($L << \lambda$), so that is is assumed that the current I is constant along its length. For the infinitesimal dipole, the radiation is maximum in the x-y plane normal to the dipole axis and zero for directions along the length of the dipole. Although the Hertzian dipole is used in many simulations, it is not realisable in practice. The most commonly used dipole is the center-fed half-wave dipole. This type of antennas have two quarter-wavelength elements, resulting in a total length of approximately $L = \lambda/2$, where λ is the wavelength.

When designing antennas for a desired operating frequency, the length of the dipole has to be considered. Antennas with larger size resonate at lower frequencies, whereas antennas with a small size resonate at high frequencies. This means that for lower frequencies, an antenna with a considerable size is required, which is not practical. To tackle this problem, techniques to achieve a desired frequency with a smaller antenna size have been implemented, such as the resistive loading of an antenna. When it comes to GPR antenna design, the goal is to minimise the resonance in order to achieve greater bandwidth and less sensitivity tuning in the GPR system.

Biconical antennas are broadband dipoles which are formed using two roughly conical shaped elements of finite length. A bowtie antenna is a type of biconical antenna, which can be considered as a 2D version of the a biconical antenna. Figure 4.2 illustrates both a biconical and a bowtie antenna fed by a transmission line. Bowtie uses two identical conductive triangular elements, instead of linear wires or rods, as its antenna elements. Together the triangular elements resemble a bow tie, as the name suggests. The bowtie antenna is characterised by the flare angle of its triangular elements. As the flare angle goes to zero, the bowtie reduces to a simple linear dipole. The input impedance of a bowtie antenna varies with wavelength and flare angle, having a relatively flat input impedance response for wide flare angles.

The space around the antenna is usually divided into three regions, as shown in Figure 4.3, which are:

1. Reactive near-field: First is the region that immediately surrounds the antenna and in which the reactive field dominates. The theoretical boundary



(a) Dipole antenna

(b) Mononopole antenna

Figure 4.1: Figure showing a) a dipole antenna, which has two radiating arms and b) a monopole antenna with a single arm.

of this region is taken to be at a distance of $R < 0.62\sqrt{D^3/\lambda}$ from the antenna, where D is the largest dimension of the antenna and λ is the wavelength.

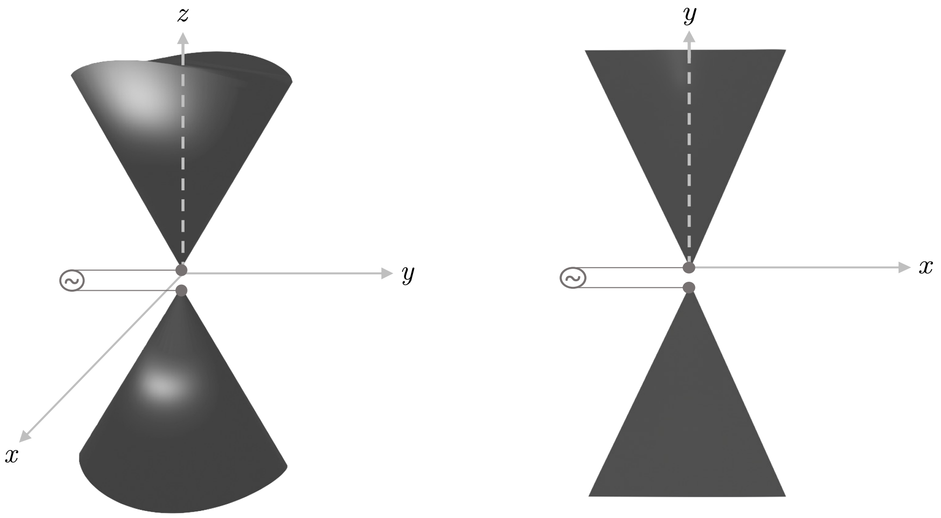
2. Radiating near-field (Fresnel) region: This is the area between the reactive region and the far-field region, in which the radiation fields dominate and the angular field distribution depends on the distance from the antenna. If the biggest dimension of the antenna is small compared to the wavelength, the Fresnel region may not exist. This area exists between $R \geq 0.62\sqrt{D^3/\lambda}$ and $R < 2D^2/\lambda$

3. Far-field (Fraunhofer) region: In the far-field region, the angular field distribution is independent of the distance from the antenna. The inner boundary of the region is a distance $R = 2D^2/\lambda$ and extends to infinity.

The amplitude pattern of an antenna changes with increasing observation distance from the near field to the far field region, since the fields themselves are changing in magnitude and phase (Balanis, 2015).

4.3 DESCRIPTION OF THE GPR TRANSDUCER

The antenna that was chosen to be modelled is the 2000 MHz “palm” antenna from Geophysical Survey Systems, Inc. (GSSI). In the GPR field, this is considered a high-frequency antenna, meaning that it has a high-resolution and can resolve



(a) Biconical antenna

(b) Bowtie antenna

Figure 4.2: a) Biconical antenna and b) Bowtie antenna. Both antennas are connected to a transmission line, which in turn is connected to a source.

small targets more easily. Due to the trade-off between resolution and penetration depth, high-resolution comes at the expense of a small penetration depth, which is approximately ∼0.3 m for the specific GPR system, depending on the conditions. High-frequency antennas are most commonly used for the assessment of concrete slabs: pinpointing structural elements such as reinforcing bars (rebars), post-tension cables and conduits or measuring the thickness of the slab. Its small size allows for scanning in previously not easily accessible areas, such as corners, between obstacles and in walls.

Figure 4.4 illustrates an image of the GSSI transducer, which was used along with the GSSI SIR-4000 data acquisition system. The transducer integrates a survey wheel and a “deadman” switch operator for turning off the transmitter when it is not used.

4.4 GEOMETRY OF THE ANTENNA MODEL

In order to simulate an antenna transducer, knowledge of the geometry and the dielectric properties of its elements is required, as well as the excitation pulse. Creating a numerical model of a real GPR transducer is more challenging than modelling a custom-built one. Although the geometrical features can be acquired by observation, most of the values of the dielectric properties of the materials that were used to construct the antenna and the excitation pulse are unknown, most often due to commercial sensitivity in addition to genuine lack of accurate characterisation of the wide-band dielectric properties for certain materials.

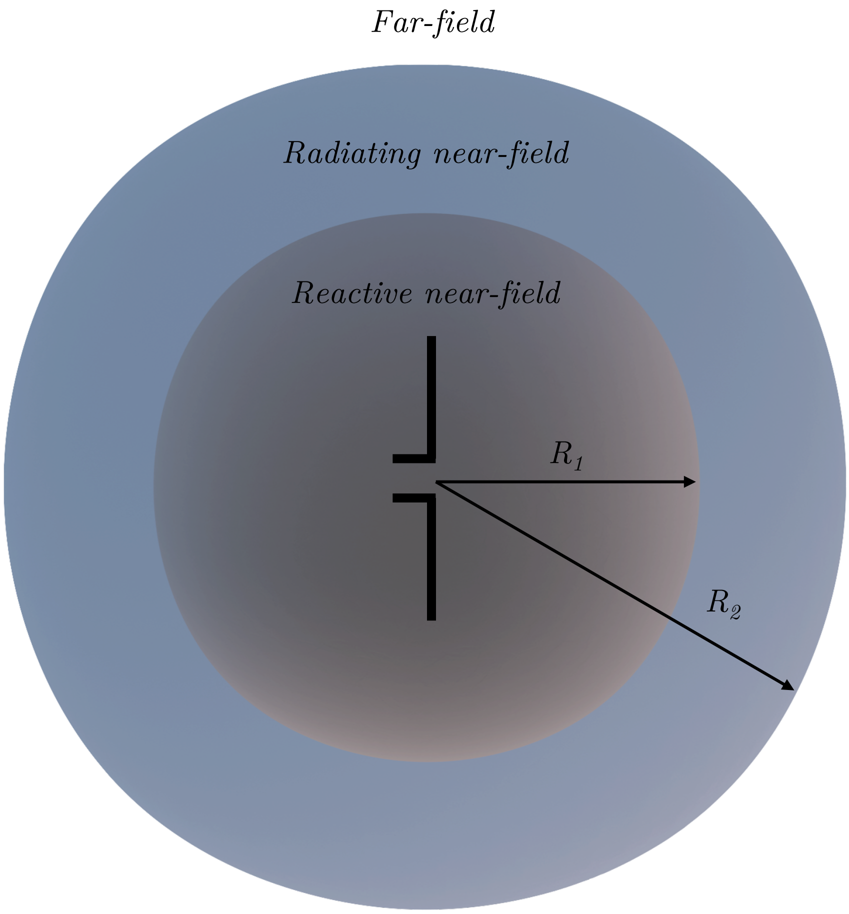


Figure 4.3: Field regions around an antenna.

The first step when modelling a GPR transducer is to build the geometry of the antenna. Inspecting the geometry of the real GPR system, the most important components are chosen to be included in the model. The key design hypothesis in developing such a model is to include as best as possible the parts that directly affect and shape the EM radiation and reception and avoid the detailed modelling of the circuit components, which have a negligible effect. Moreover, due to the small size of these components, an extremely fine grid, that would greatly increase the computational cost, would be required.

Figures 4.5-4.12 illustrate the simplified FDTD geometry of the different parts of the antenna model being investigated. The model includes two planar surface bowties, one that acts as the transmitting antenna and one as the receiving antenna, with additional rectangular extensions at the ends of both bowties, reducing the antenna dimensions and also increasing the wide-band characteristics. The Tx and Rx bowtie are separated by a distance of 4.3 cm, as displayed in Figure 4.11, where the fine geometry of the bowties is presented. The bowties are etched from copper onto printed circuit boards (PCBs), which are displayed in

Figure 4.4: The 2000 MHz “palm” antenna from Geophysical Survey Systems, Inc. (GSSI).

Figure 4.8. Underneath the PCBs, two different types of electromagnetic absorber foams are inserted in order to reduce the back-cavity radiation (see Figures 4.7 and 4.10). In the middle of the two absorbers, a gap is present for both the Tx and the Rx, where a small plastic case is inserted. Furthermore, EMI shielding gaskets are placed on the bottom side of the second absorber in the vicinity of each bowtie’s midpoint, as shown in the backside geometry of the antenna model in Figure 4.12. The bowties are placed in the same enclosure, which is modelled as a square metal box and acts as a shield, shown in Figure 4.6. The shielding is utilized in order for the antenna to serve as a directional device. GPR antennas are required to maximise the radiation energy transmitted in the ground and suppress it in all other directions, hence act as a directional device. In the middle of the case, as shown in the side view of the antenna model in Figure 4.10, there is a plate that separates the transmitter from the receiver, which are centered in each section of the box. The divider between Tx and Rx includes two gaps, which are visible in the Figure. The box is open on the side of the bowties and is further enclosed in an outer plastic case as shown in Figure 4.5. The skid plate is also included, as shown in the full model in the same Figure, resulting to an overall size of 86 × 86 × 68 mm.

The real bowtie antennas are actually connected to electronic circuits, which were not included in the model and therefore, for the excitation, a voltage source is used in a single-cell gap between the two arms of the transmitter bowtie

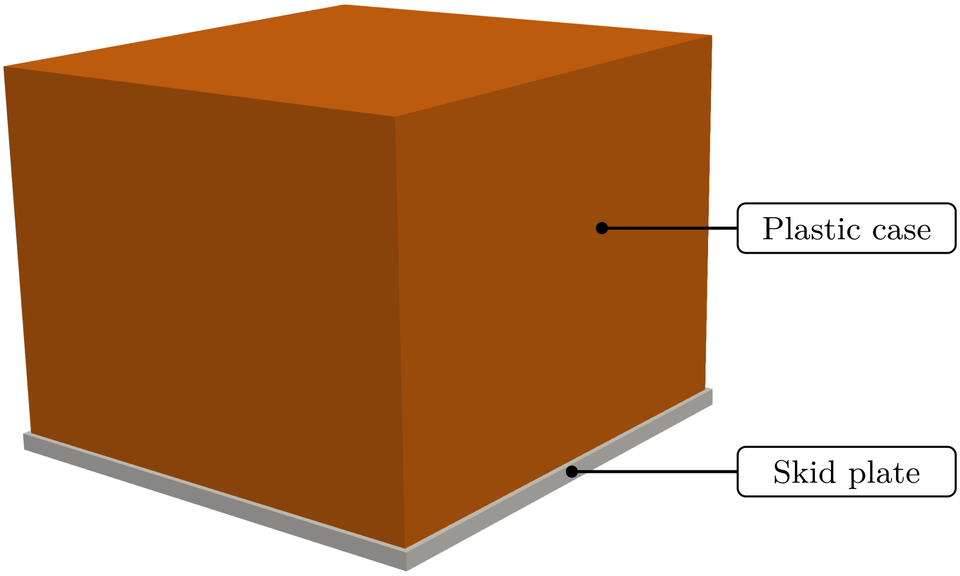


Figure 4.5: Modelled geometry of the 2000 MHz “palm” antenna from GSSI: Plastic enclosure and skid plate.

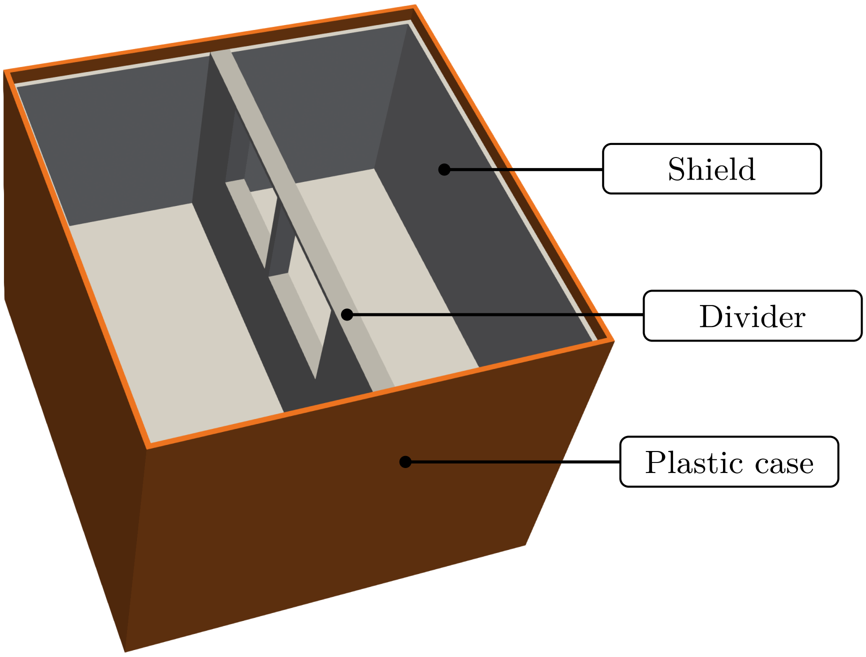


Figure 4.6: Modelled geometry of the 2000 MHz “palm” antenna from GSSI: Inner geometry showing the shield and the plate between Tx and Rx.

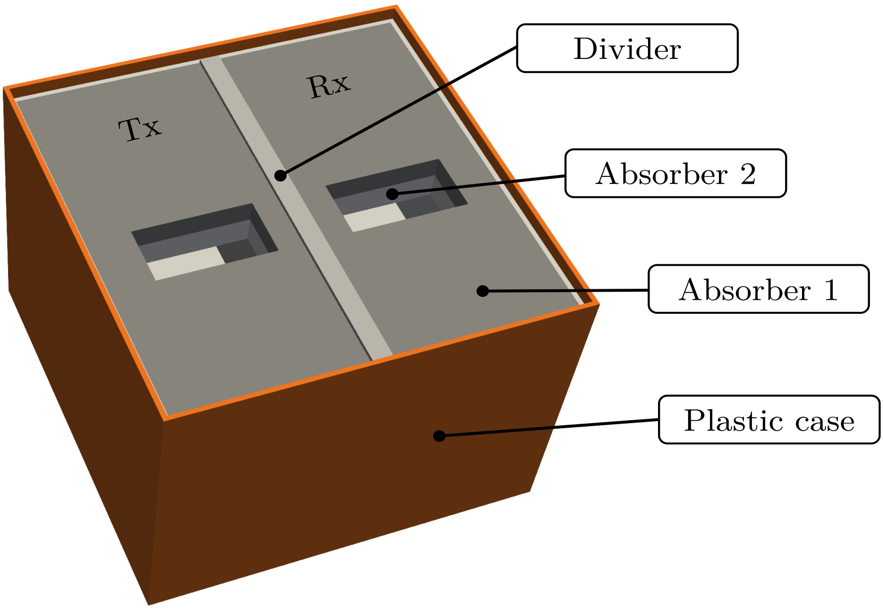


Figure 4.7: Modelled geometry of the 2000 MHz "palm" antenna from GSSI: Inner geometry showing the two layers of EM absorber foams.

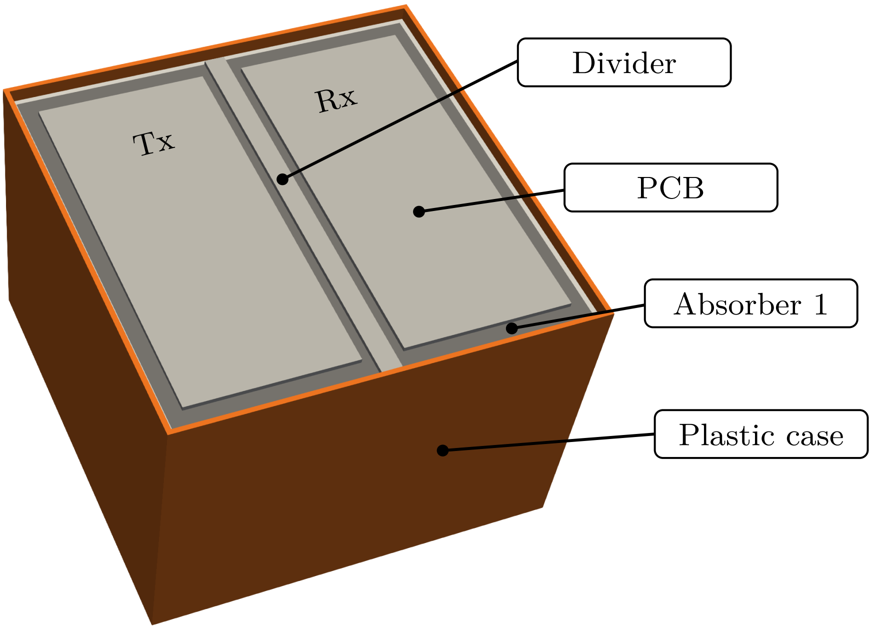


Figure 4.8: Modelled geometry of the 2000 MHz "palm" antenna from GSSI: Inner geometry showing the PCB.

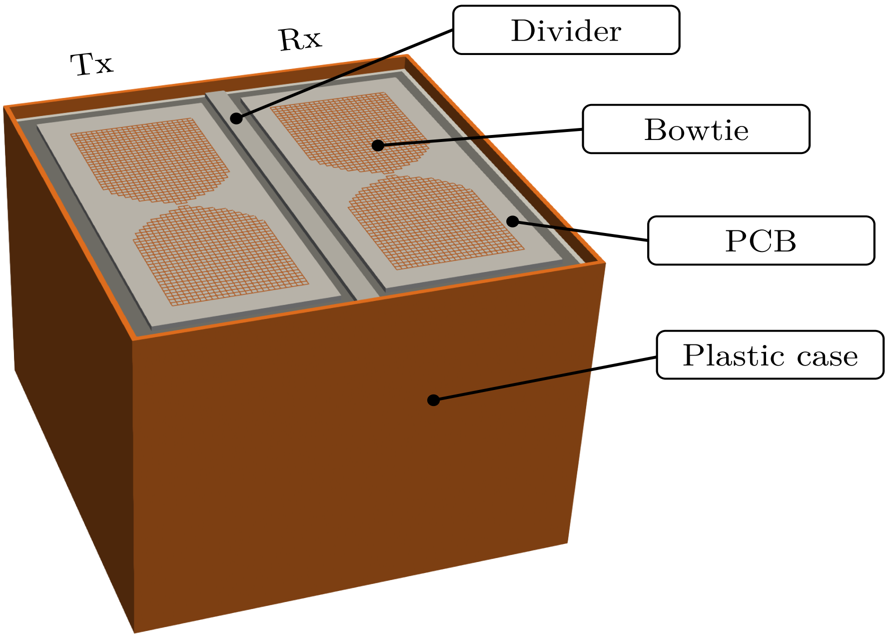


Figure 4.9: Modelled geometry of the 2000 MHz "palm" antenna from GSSI: Inner geometry showing the Tx and Rx bowties placed on top of the PCB.

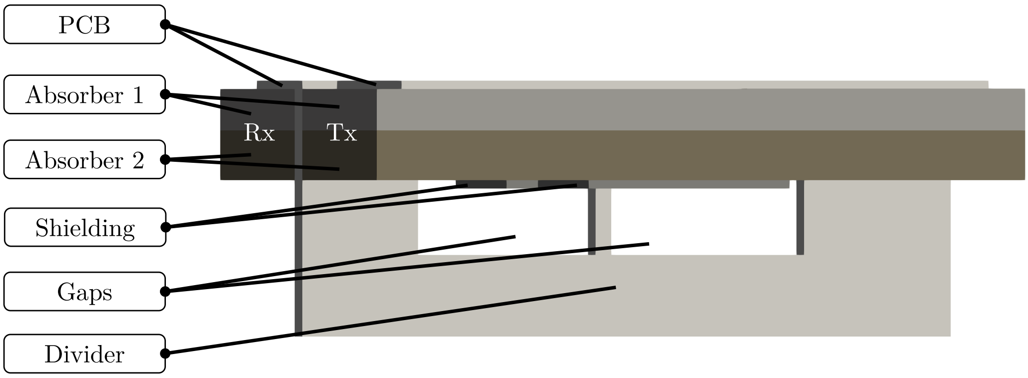


Figure 4.10: Modelled geometry of the 2000 MHz "palm" antenna from GSSI: Side view.

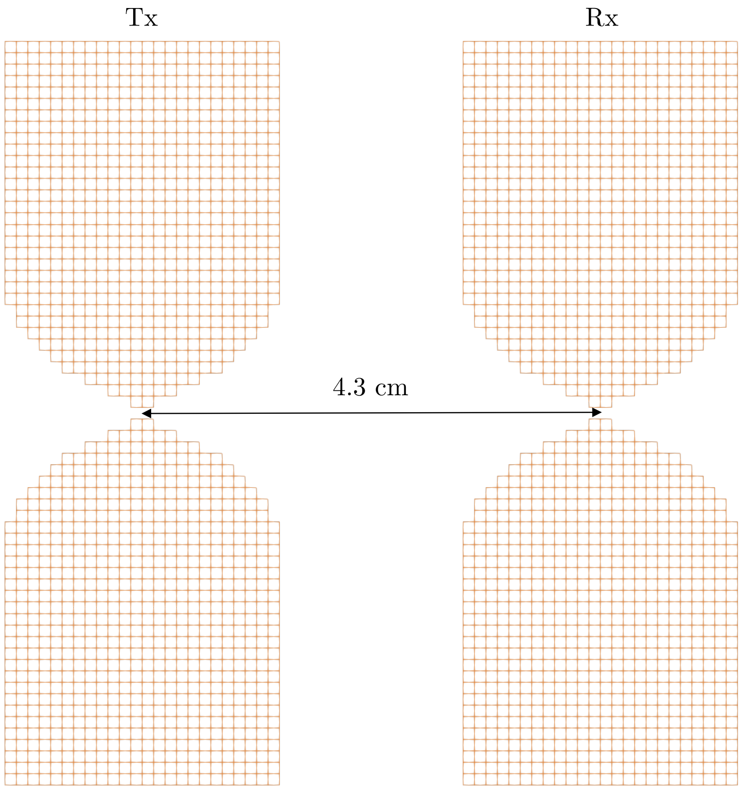


Figure 4.11: Modelled geometry of the 2000 MHz "palm" antenna from GSSI: Fine geometry of Tx and Rx bowties.

(feed point). At the feed point the tangential E-field component is related to an impressed voltage

$$V(t) = -\int_0^l \overrightarrow{E} \cdot \overrightarrow{dl} \tag{4.1}$$

where l is the gap width. The impressed voltage for the voltage source is constrained to follow a certain pulse and can include an internal resistance or not. In this case, a resistance was included, which was determined from optimisation. Since the shape of the true pulse is unknown, it was derived via trial and error that a Gaussian-shaped pulse produces the best results for the specific model. The Gaussian pulse is given by

$$P(t) = e^{-2\pi^2 f^2 (t - \frac{1}{f})} \tag{4.2}$$

where f is the peak frequency of the pulse and t is time. A time delay of 1/f is introduced to the pulse to ensure a smooth start. The center frequency was not chosen via trial and error but was determined by the optimisation process, resulting in a value close to the frequency specified for the real transducer. The Gaussian pulse used along with its power spectrum is illustrated in Figure 4.13. Using the approach presented in Giannakis et al. (2019a), and considering the

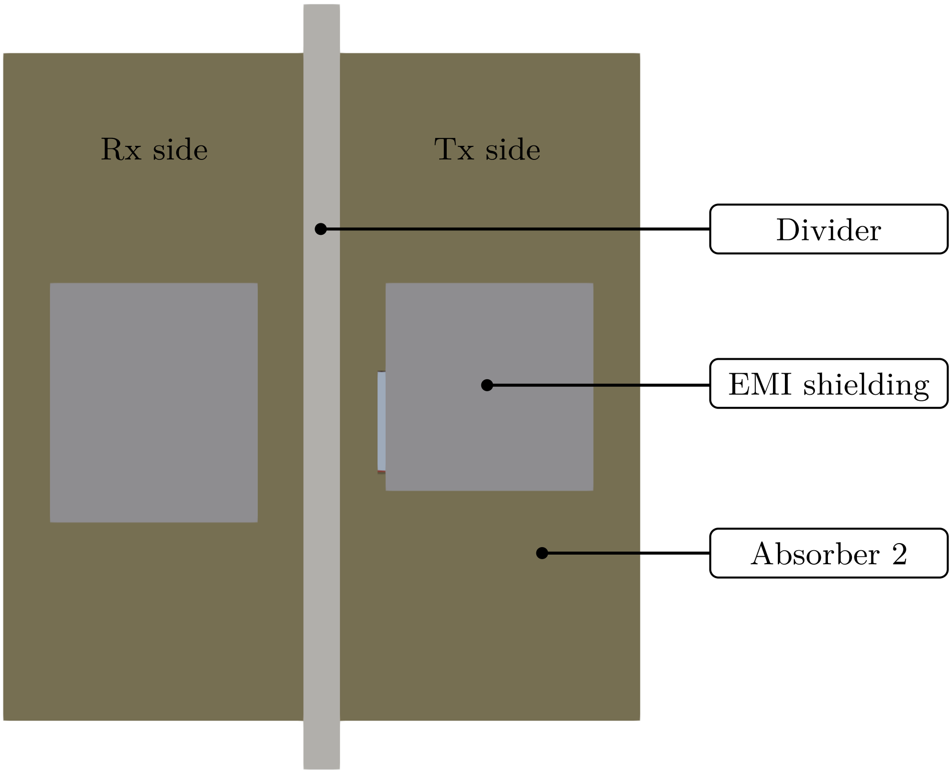


Figure 4.12: Modelled geometry of the 2000 MHz “palm” antenna from GSSI: Backside view of the GSSI antenna model.

pulse shape also as an unknown, did not affect the results, supporting the premise that a Gaussian-shaped pulse is a good approximation of the real pulse. The receiver was modelled using a single-cell gap between the arms of the receiver bowtie with an edge of unknown conductivity. The conductivity of the receiver as well as the dielectric properties of certain components of the antenna are to be estimated using optimisation after having the geometry fixed.

4.5 OPTIMISATION OF THE ANTENNA

In order to acquire estimates of the key antenna properties, optimisation techniques were used. Optimisation tries to find a set of parameters that solves as best as possible a certain problem. It is widely used in many disciplines such as engineering, mathematics, statistics and others where a large number of optimisation problems arise. The goal in an optimisation problem is to acquire a set of values for the variables for which a certain function, called objective function, is maximized (fitness function) or minimized (loss or cost function). The objective function, which is a measure of “goodness”, can take many different forms but most commonly it is an error function to be minimized. The general form of an

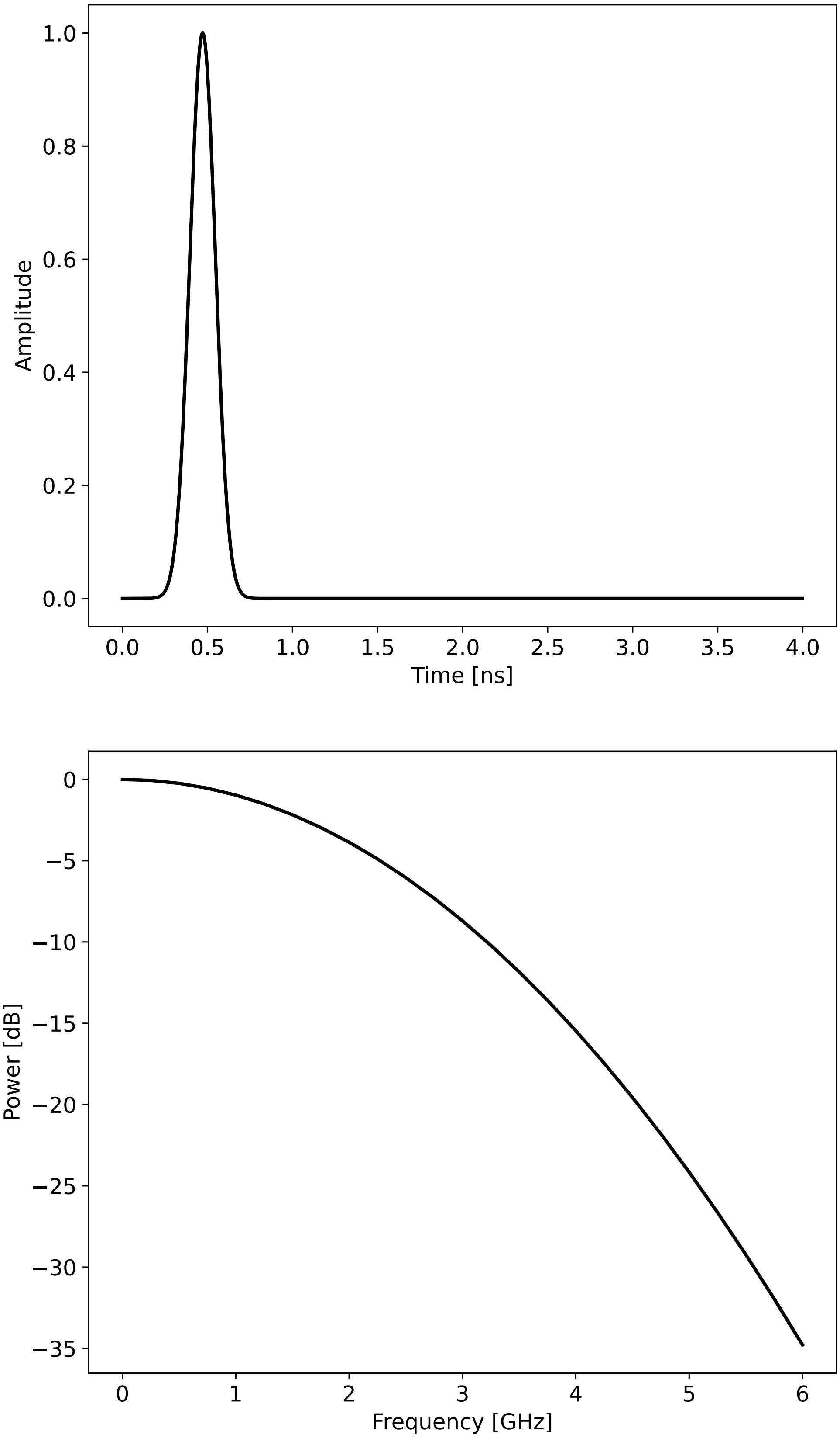


Figure 4.13: 2.12 GHz Gaussian shaped pulse: a) Time domain waveform, b) Power spectrum.

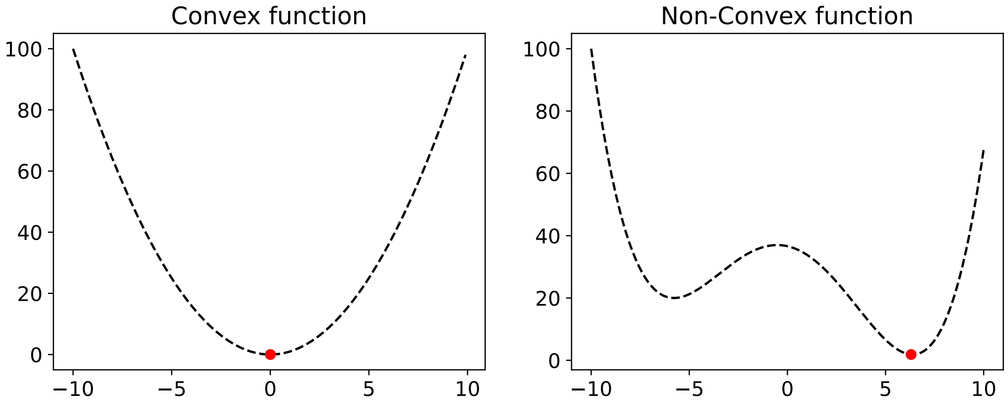


Figure 4.14: a) A convex function and b) A non-convex function. Their global minima are annotated with a red circle.

optimisation problem can be expressed as:

$$\begin{aligned} &\underset{x \in R^n}{\text{minimize}}\, f(\mathbf{x}) \\ \textit{subject to } &c_i(\mathbf{x}) = 0, \quad i = 1, 2, 3, ..., k \\ &c_i(\mathbf{x}) \geq 0, \quad i = k+1, ..., q \end{aligned}$$

where $f(x)$ is the objective function, which in this case is a loss function since it is a minimisation problem, x is the n-dimensional vector with the parameters to be optimised and $c_i(x)$ are the constraints of the problem. The variables k and q represent the number of equality and inequality constraints. These constraints are restrictions that determine the acceptable values of the parameters. To the same optimisation problem, there might be many different sets of parameters that produce the same outcome but some of them might not be realistic solutions, e.g. obtaining a negative value for permittivity. For this reason constraints are used to set the lower and upper limits of the values that the parameters can take in order to be realistic. The constraints imposed on the variables are considered a priori information, since the range of values of the most common materials used in antenna designing is known. Thus, one optimisation problem can be considered as a minimization or maximisation problem of a non-linear function subject to upper and lower bounds on the parameters.

Based on the problem, multiple objective functions can be used, where the total objective function must satisfy more than one goal. For instance, to maximise the gain of the antenna and at the same time to minimise the unwanted reflections. This type of problem is known as multiple-objective optimisation but in the present work only single-objective optimisation is considered.

There are many ways to classify optimisation algorithms, with a broad classification being either traditional local techniques or advanced global approaches.

Most local optimisation techniques are gradient-based, meaning that they utilize gradients to reach to an optimal solution. Algorithms that use gradient information have been widely used to solve a number of optimisation problems due to their efficiency. But they have the drawback that the algorithm, when reaching a local optimum, cannot escape and search the optimisation space for a better solution. Thus, the algorithm provided the local optimum as the optimal solution and fails to locate the global optimum. Therefore, gradient-based techniques are efficient in cases when dealing with convex functions, which are functions that have only one local minimum or maximum which is also the global optimum. Figure 4.14a illustrates a convex function with the characteristic bowl shape and its global minimum annotated in red. When dealing with problems with multiple local minima or maxima, different approaches are required to acquire an accurate solution. Figure 4.14b displays a function with a local minimum and a global minimum, which is annotated in red.

Global optimisers have been extensively used for the numerical modelling of real systems in a broad spectrum of applications. Their success relies on their ability to handle the non-convexity and non-linearity of many problems that require a global search approach. The most common are the stochastic methods, which introduce randomness in the optimisation, such as generating and using random variables. Genetic algorithms (GA) and particle swarm optimisation (PSO) fall in this category and were used to optimise the transducer, described below.

Optimisation has been mostly used for new antenna design purposes by optimising different sets of antenna parameters. Haupt (1994) employed GAs in order to find an optimal thinned array that produces the smallest maximum relative sidelobe level. Optimisation was performed for a 200-element linear array and a 200-element planar array, but could be extended to other arrays.

GAs were also implemented in the work by Altshuler and Linden (1997) to design four different antennas, two modified Yagi-Uda antennas, monopole loaded with a modified folded dipole and a seven-wire antenna. These antenna were also built and tested with the measured data agreeing well with the computational results. Although the antennas were optimised for efficiency at a single frequency, the built designs proved to be highly broadband. To design an efficient Yagi-Uda antenna, Pareto GA was used by Kuwahara (2005), where more than one objective functions are involved, which are optimized simultaneously. Specifically, the antenna was optimised to minimise the sidelobe level, maximise the gain, and obtain an input impedance of approximately 50Ω. To demonstrate that the proposed optimisation scheme is superior to the current at the time design techniques, it was applied on an 8-element and a 12-element Yagi–Uda antenna, but it was also employed to antennas with less number of elements.

A parallel PSO/FDTD scheme is presented by Jin and Rahmat-Samii (2005), used to design multi-band and wide-band patch antennas, where parameters regarding the geometry of the antennas are selected to be optimised. The algorithm was tested on a rectangular patch antenna and an E-shaped patch antenna, where the length and width of the patch and the slot, the slot position and the feed position were optimised. Prototype antennas were built based on the optimised antennas to validate the accuracy of the scheme with the S_{11} curves showing a good agreement with the simulated results.

GA was applied by van Coevorden et al. (2006) to design a resistively loaded thin-wire bowtie GPR antenna. The resistive loads and the number of wires and the angular distances between these wires were chosen to be optimised. The performance of the optimised antenna is compared to that of the straight thin-wire Wu–King dipole antenna, where the proposed antenna proved to be more efficient. In the work presented by Pantoja et al. (2007), a PSO scheme was utilised to optimise several parameters of a log-periodic dipole array in order to obtain desired radiation characteristics. The log-periodic dipole array is a coplanar linear array that consists of a number of unequally spaced dipole elements with gradually increasing length. A 10-element array was used with 11 design parameters to be optimised, where most of the parameters are related to the length and positions of the dipole elements.

Each of the above proposed schemes are used to optimise certain properties with the aim of achieving different radiation characteristics. This proves the applicability and performance of global optimisers to a variety of problems related to antenna designing. Although optimisation was utilised to design new antennas, only a few attempts to acquire estimates of the properties of an existing commercial system can be found in the literature.

The goal of optimising a real unknown antenna is to acquire estimates of the dielectric properties of the components of the antenna, which when used in numerical modelling will replicate the behavior of the real system. For this purpose, a measurement of error is required in order to determine how well the model approximates the real behavior. This measurement of error can be the differences between real and simulated GPR responses. One of the most common functions, which was the one that was chosen to be minimized, is the residual sum of the square differences (RSS) between the synthetic and real data

$$\sum_{i=1}^{N}(X_i - Y_i)^2 \tag{4.3}$$

where X and Y are the vectors containing the simulated and real data, respectively and N is the total number of the investigated GPR scenarios. If only one scenario is used, the replica of the transducer will reproduce accurately the behavior of the real system only for that specific or similar settings and fail to respond reliably to different scenarios in the future. In optimisation, this is known as the overfitting problem, where the algorithm fits too closely to the data that learns not only the signal but also the noise contained in the real responses, making it unable to generalize to new cases. To tackle this problem, two A-scan responses from two distinct scenarios were used. The first was the free-space direct coupling and the second was the response of the transducer over a metallic plate. The specific scenarios were chosen due to their simplicity and ease of implementation. Therefore, the optimisation searches for the model parameters that will reproduce the behavior of the real transducer for these two scenarios as close as possible.

The real A-scans were collected using the GSSI 2 GHz antenna along with the SIR-3000 system for a time window of 8e-9s with 512 time samples. The only filters applied to the real data were stacking and a 10 MHz vertical high-pass infinite impulse response (IIR) filter, with vertical meaning that the filter operates in the time domain. These filters are used to reduce the noise in the data without modifying the responses significantly, whereas other filters or excessive filtering can alter the responses significantly. In the latter case, an optimisation algorithm would fit filtered responses, resulting in a model that can be thought of as a model of a modified version of the real antenna, and will not capture the behavior of the actual antenna. Since the true excitation pulse of the real transducer is unknown, the real data are normalised to the maximum amplitude value of the free space response as:

$$
\begin{aligned}
max_{fs} &= max(|Y_{fs}|) \\
Y_{fs\_norm} &= Y_{fs}/max_{fs} \\
Y_{pec\_norm} &= Y_{pec}/max_{fs}
\end{aligned}
$$

where Y_{fs} is the real free space response and Y_{pec} is the A-scan received from the metal plate, whereas Y_{fs_norm} and Y_{pec_norm} are the normalised responses for free space and from the metal plate, respectively. In the simulations, the size of the domain is set to 250 × 200 × 170 mm with a step size of 1 mm for both simulations. The time step is set to $\Delta t = 1.92$ ps as calculated by the Courant-Friedrichs-Lewy (CFL) condition (Taflove and Hagness, 2000), resulting in 4156 time steps for a time window of 8e-9s, while the metallic plate was modelled as a perfect electric conductor (PEC), which is an idealised material assuming to have an infinite electrical conductivity and is modelled by setting the electric field to zero. To simulate the second scenario, the antenna model is placed exactly on top of the PEC plate. In order for the synthetic and the real data to be comparable for the objective function, the following steps are performed at each iteration before evaluating the RSS:

1. Downsample the synthetic data from the FDTD time steps to 512 in order to match the number of samples of the real data. The downsampling was performed using the Fourier method.
2. Normalise the synthetic data to the maximum absolute amplitude of the simulated free space response.
3. Align the synthetic with the real data by their negative peak by shifting the synthetic data the required number of samples. Instead of the negative peak, the positive peak could have been chosen to align the data as well.

4.5.1 *Genetic Algorithm*

Genetic algorithm (GA) is based on natural selection imitating biological evolution. GA simulates the process of natural selection, where the species that adapt to changes are able to survive and reproduce to form the next generation. In other words, they simulate “survival of the fittest” among individuals of a generation. GA has several advantages over traditional optimisation techniques, such as that it does not require any derivative information, it works even with extremely complex cost surfaces and is suitable for parallel computing. In addition to that, GA provides a set of optimum parameters and not just a single solution and works well with large number of parameters.

GA is an iterative process that starts by creating an initial population of candidate solutions based on the set of constrains. At each iteration the population forms a generation. Assuming an individual solution is

$$\mathbf{p} = [p_1, p_2, ..., p_n] \tag{4.4}$$

with i corresponding to the i-th solution, the population matrix can be written as:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{bmatrix} \tag{4.5}$$

where n is the number of parameters to be optimised and m is the size of the population. Thus, each row of the matrix forms an individual candidate solution or in other words, a point in the search space of a given problem. The initial population is randomly generated

$$\mathbf{P} = random(m, n) \tag{4.6}$$

where usually samples are drawn from the uniform distribution over the interval [0,1) adjusted to the constraints. Each solution from the population is evaluated based on an objective function to calculate the cost in this case,

$$cost_1 = f(p_{11}, p_{12}, ..., p_{1n}) \tag{4.7}$$

$$cost_2 = f(p_{21}, p_{22}, ..., p_{2n}) \tag{4.8}$$

$$\vdots$$

$$cost_m = f(p_{m1}, p_{m2}, ..., p_{mn}) \tag{4.9}$$

and the members with the best fit to this function, called “parents”, are chosen. Since natural selection is invoked, only the healthiest members, the parents, survive and the solutions corresponding to the rest of the members are discarded. There are two common ways to decide how many solutions are kept. The first is to define the number of solutions kept, while the second is to use thresholding and keep only the solutions that had a cost value less than a threshold. There are three types of “children” that are created, which form the population of solutions for the next generation:

1. Elite are the individuals with lowest cost value in the current generation. Therefore, the parents automatically survive to the next generation.
2. Crossover (combining) are the children created by combining the solution vectors of the parents.
3. Mutation children are a result of introducing random changes in a single parent.

The way a child is generated from parents with the three processes is shown in Figure 4.15 using both schematic diagrams and solution vectors. In the case of the solution vectors, a 6-dimensional solution was used as an example with $p_i \in \mathbb{R}^6$. The child was produced by randomly combining parameter values between two parents, but there are other approaches that can be used, as well. For crossover, there are many selection methods in order to choose which parents will be paired, with the most common being pairing top to bottom, random pairing and weighted random pairing, where each parent is assigned a probability of being chosen, which is inversely proportional to the value of its cost. Assuming, m_{keep} is the number of parents kept, pairing continues until $m - m_{keep}$ children are generated to displace the discarded solutions. From the new generation, ”parents” are again selected to produce the children for the next generation and this process continues until it eventually converges to a single optimal solution which has the best fitness value of the objective function. The flow chart of this algorithm is presented in Figure 4.16.

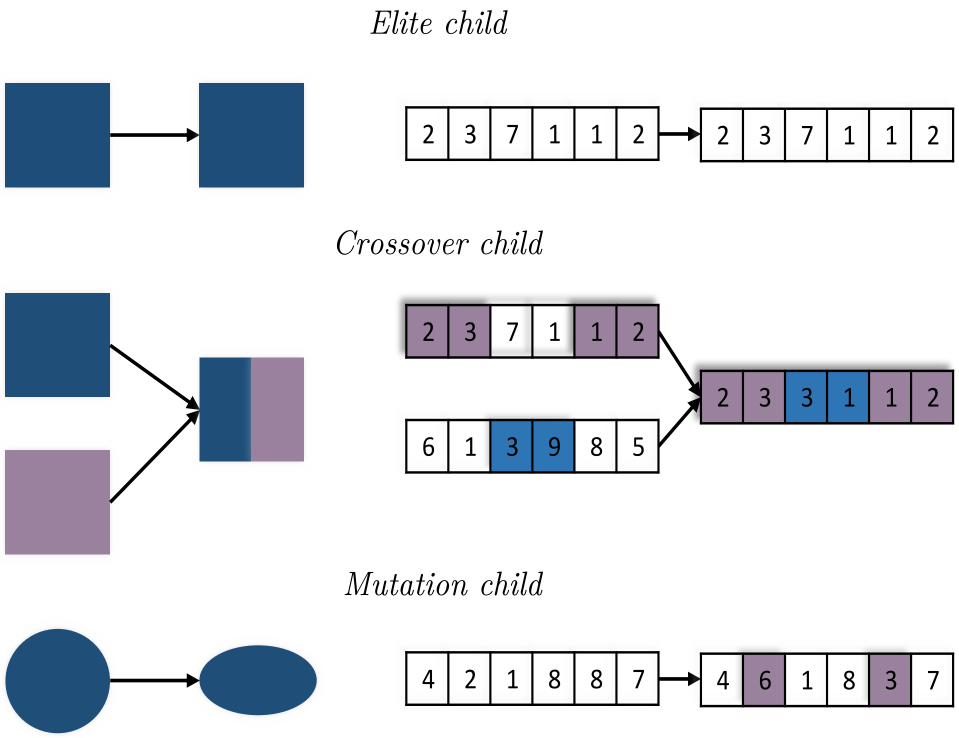


Figure 4.15: The three ways children are formed by parents in a Genetic Algorithm, illustrated using schematic diagrams (left) and solution vectors (right).

As mentioned above, the number of variables n that are chosen is 15, the maximum number of iterations is set equal to 40, while the population size m is set to 80 based on the number of variables in the problem and their variability. Thus the population matrix has a size of 80×15. These values provided the best results after testing different sets of options to tune the solver. Using more iterations or increasing the population size, did not improve the solution and resulted in larger computational times. The stopping criteria are to obtain an RSS value of 1% or to reach the maximum number of iterations.

4.5.2 *Particle Swarm Optimisation*

PSO, which was proposed by Kennedy and Eberhart (1995) is similar to the GA method by means of having a population, called swarm, of candidate solutions, called particles. It begins by creating particles with initial random locations and velocities

$$\mathbf{P} = random(m, n) \tag{4.10}$$

$$\mathbf{V} = random(m, n) \tag{4.11}$$

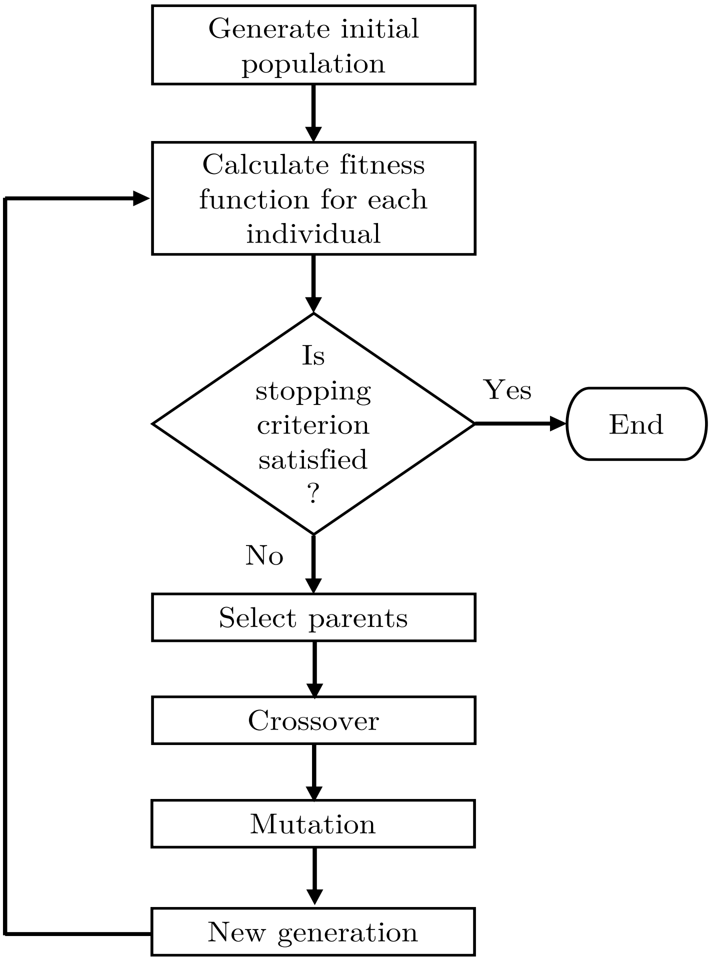


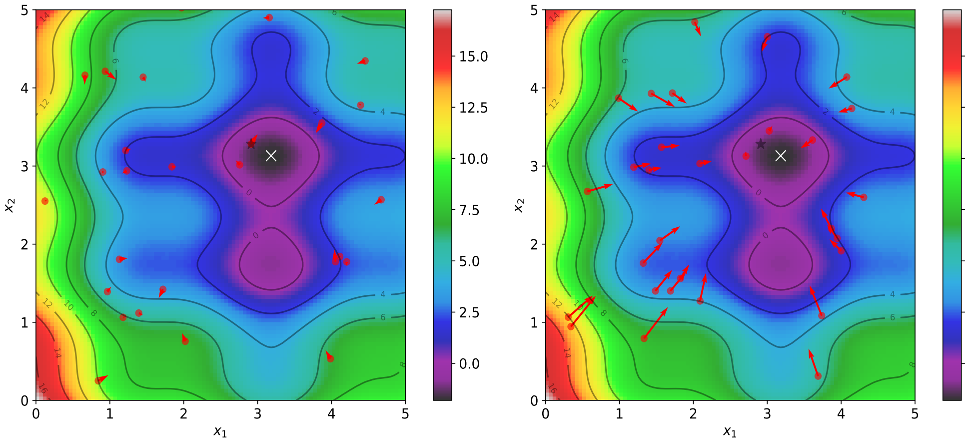
Figure 4.16: Flow chart illustrating the basic process of the Genetic Algorithm.

Therefore, two matrices are considered in this case, one containing the locations of the particles **P** and one for their velocities **V**. The velocity determines towards which direction and what distance each particle travels in the optimisation space at each iteration and essentially drives the optimisation process. At each iteration, the objective function is calculated for each particle and the best fit of the swarm and the corresponding position of this particle is determined. All the particles update their velocities based on their current velocities $\mathbf{v}^{(i)}(t)$, their best known positions $\mathbf{pbest}^{(i)}$ and the location of the particle with the best fit of the entire swarm so far **gbest** and then use these velocities to find their new positions

$$\mathbf{v}^{(i)}(t+1) = w\mathbf{v}^{(i)}(t) + c_1 r_1(\mathbf{pbest}^{(i)} - \mathbf{p}^{(i)}(t)) + c_2 r_2(\mathbf{gbest} - \mathbf{p}^{(i)}(t)) \quad (4.12)$$

$$\mathbf{p}^{(i)}(t+1) = \mathbf{p}^{(i)}(t) + \mathbf{v}^{(i)}(t+1) \quad (4.13)$$

where $w, c-1$ and c_2 are constant parameters of the PSO algorithm and r_1, r_2 are random numbers $\in [0,1]$. Parameter $w \in [0,1]$ is called the inertia weight constant and controls how much the previous velocity affects the particle in the current update. Parameters c_1 and c_2 are known as the cognitive and social coefficients, respectively. These are weighting the particle's best position and the swarm's best position, respectively, when adjusting the velocity. The vectors $\mathbf{pbest}^{(i)}$ and $\mathbf{gbest}$ are also updated in each iteration to acquire the best ones found so far. This way the entire swarm moves towards the direction of the best solution at each iteration and eventually reaching to an optimal location. Similarly to GA,


(a) Iteration 1 (b) Iteration 4

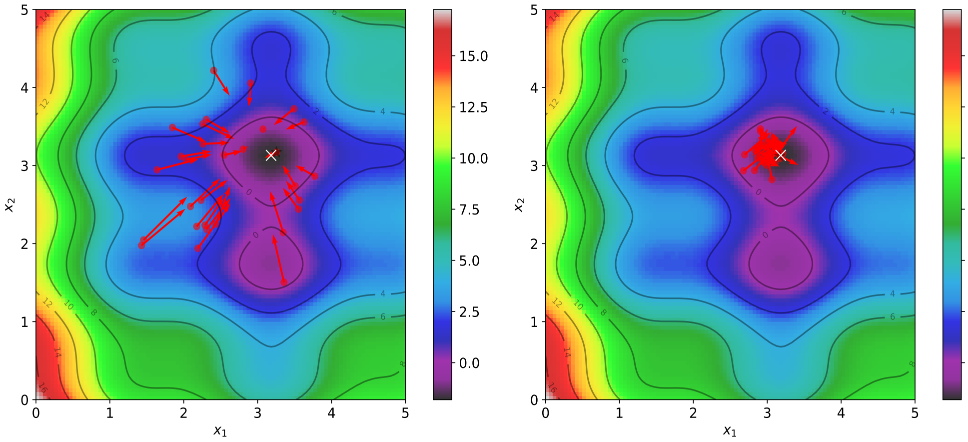

(c) Iteration 6 (d) Iteration 10

Figure 4.17: Geometrical representation of Particle Swarm Optimisation for a two-dimensional search space. The dots represent the positions of each particle, while the vectors are the velocities showing the step and direction each particle should take to reach the optimal solution. The white cross represents the minimum of the cost function.

PSO does not require the gradient of the objective function. Figure 4.17 shows a geometrical representation of the PSO process for a two-dimensional optimization space, starting from a random generated swarm until convergence. The particle with the best fit of the objective function in this location is chosen as the final

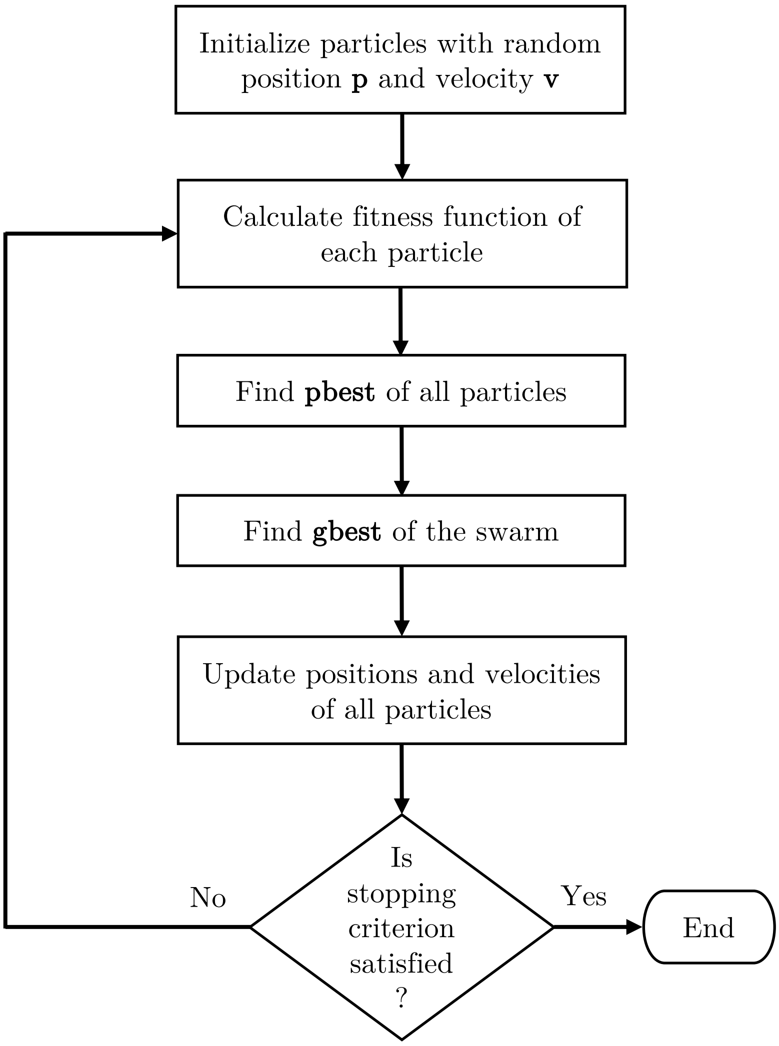


Figure 4.18: Flow chart showing the steps of the Particle Swarm Optimization.

estimate of the global optimum. Figure 4.18 illustrates the steps followed through this process with a flow chart.

The number of particles in the swarm are selected to be 40. Increasing the particles did not seem to improve the solution. The stopping criteria and the maximum number of iterations are the same with the GA method. Choosing a small number of iterations may terminate the optimisation search before converging to an optimal model, while using too many iterations can unnecessarily increase the computational costs. Therefore, these settings were again obtained through tuning the PSO solver. Both GA and PSO schemes were executed multiple times, generating each time a different initial random population and initial random swarm in order to acquire the optimal models. Both GA and PSO were performed using the global optimisation toolbox in MATLAB.

4.5.3 *Optimisation domain*

After considering different sets of parameters to be optimised and running multiple optimisation models, certain electrical properties of the materials that affect the radiation of the transducer the most were selected. The 15-dimensional vector that was chosen includes the following parameters:

1) Absorber 1 relative permittivity (ϵ_{a1})
2) Absorber 2 relative permittivity (ϵ_{a2})
3) PCB relative permittivity (ϵ_{pcb})
4) Skid relative permittivity (ϵ_{s})
5) Plastic Case relative permittivity (ϵ_{p})
6) Shield relative permittivity (ϵ_{sh})
7) Absorber 1 conductivity (σ_{a1})
8) Absorber 2 conductivity (σ_{a2})
9) PCB conductivity (σ_{pcb})
10) Skid conductivity (σ_{s})
11) Plastic Case conductivity (σ_{p})
12) Shield conductivity (σ_{sh})
13) Source impedance (R_s)
14) Receiver conductivity (σ_R)
15) Center frequency (f)

The above dielectric properties are assumed to be constant and frequency independent. Remaining properties of the antenna elements as well as the geometry of the model are kept fixed. The metallic parts of the antenna are modelled as PECs, whereas all the elements of the antenna are considered non-magnetic, having relative permeability $\mu_r = 1$ and zero magnetic loss $\sigma_\mu = 0$. To limit the size of the optimisation domain and ensure positivity of the values obtained, constraints were imposed. The constraints used for each parameter are given in Table 4.1.

Optimised Parameters	**Lower limit**	**Upper limit**
Absorber 1 ϵ_{a1}	1	7
Absorber 2 ϵ_{a2}	1	7
PCB ϵ_{pcb}	1	6
Skid ϵ_{s}	1	6
Plastic Case ϵ_{p}	1	7
Shield ϵ_{sh}	1	3
Absorber 1 σ_{a1} (S/m)	0	2.5
Absorber 2 σ_{a2} (S/m)	0	2.5
PCB σ_{pcb} (S/m)	0	0.05
Skid σ_{s} (S/m)	0	0.01
Plastic Case σ_{p} (S/m)	0	0.01
Shield σ_{sh} (S/m)	0	10^9
$R_s(\Omega)$	1	700
$\sigma_R(\Omega^{-1})$	0	1
f (GHz)	1.8	2.2

Table 4.1: Upper and lower limits set for the parameters used in optimisation.

4.6 RESULTS

Figures 4.19 and 4.21 demonstrate the GA results for the free space and the PEC, respectively compared with the real traces, whereas Figures 4.20 and 4.22 display the PSO results for free space and PEC, respectively. It is obvious that in all cases, the modelled and real traces are in very good agreement for both scenarios. The optimised models can accurately reproduce the real patterns and predict the amplitudes and arrivals of the responses. Small differences exist in the amplitudes between the real and simulated responses which can be attributed to the Gaussian pulse used in the simulations or due to the numerical representation of the antenna, which being only a model does not include all of the elements used to construct the real complex antenna but only the essential components. Nevertheless, the similarity between modelled and real data is eminent.

The unknown antenna parameters that were optimised are given in Table 4.2 along with the values that were obtained from the GA and the PSO scheme. By comparing the values, it is obvious that both optimisation techniques converged to similar values. The greatest difference between the resultant values of the two schemes exists in the impedance of the source with the GA value being

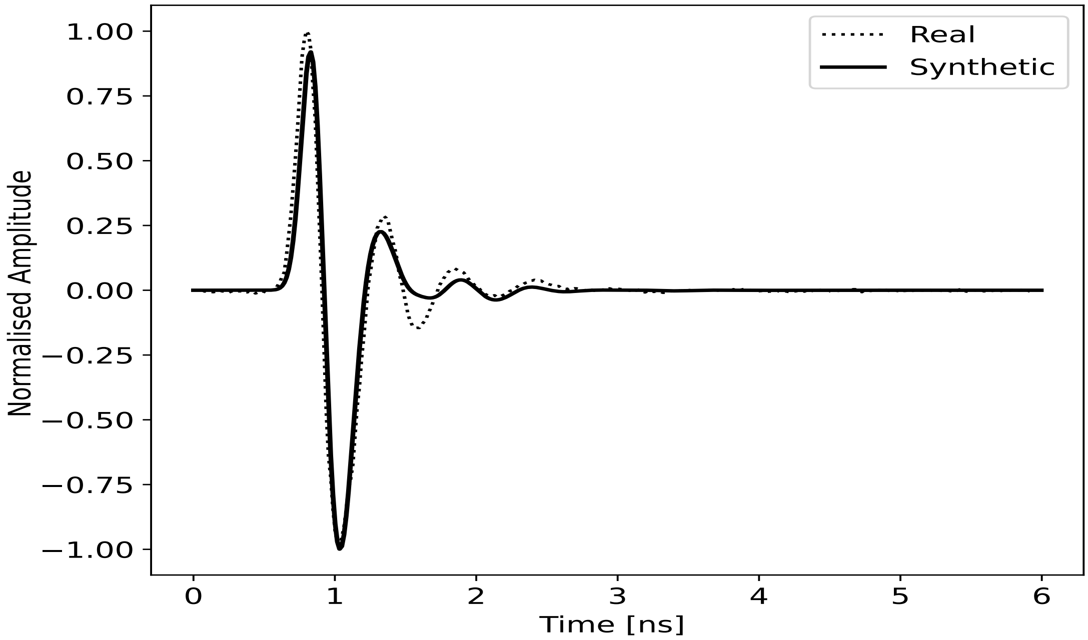


Figure 4.19: Real versus modelled responses from free-space using optimised values from the GA scheme.

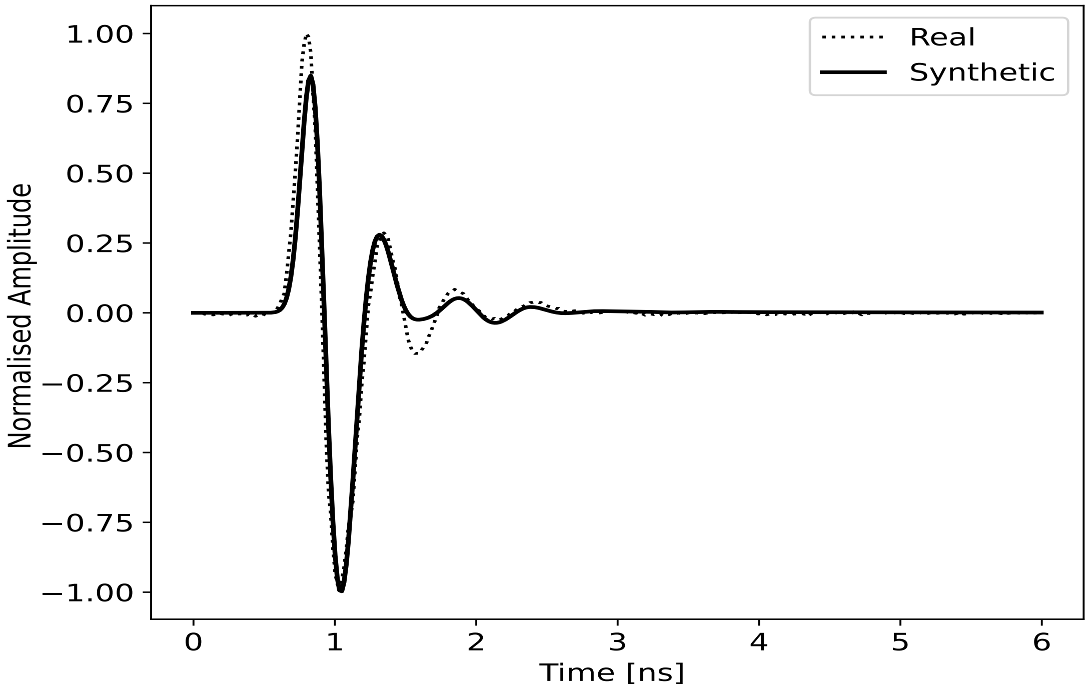


Figure 4.20: Real versus modelled responses from free-space using optimised values from the PSO scheme.

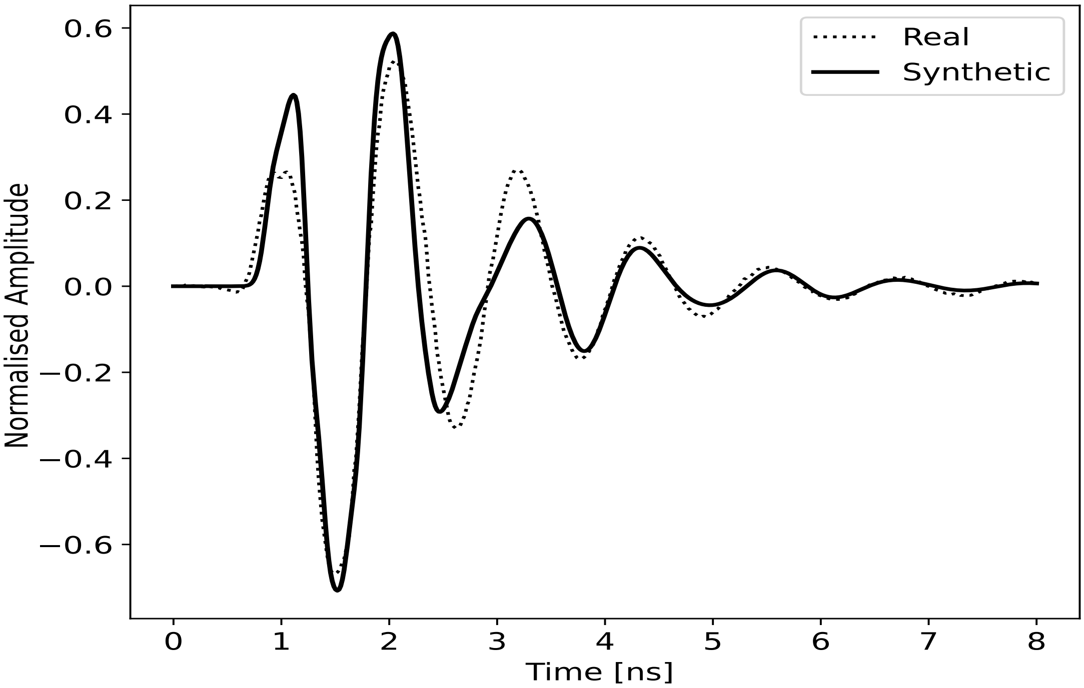


Figure 4.21: Real versus modelled responses from the metal plate using optimised values from the GA scheme.

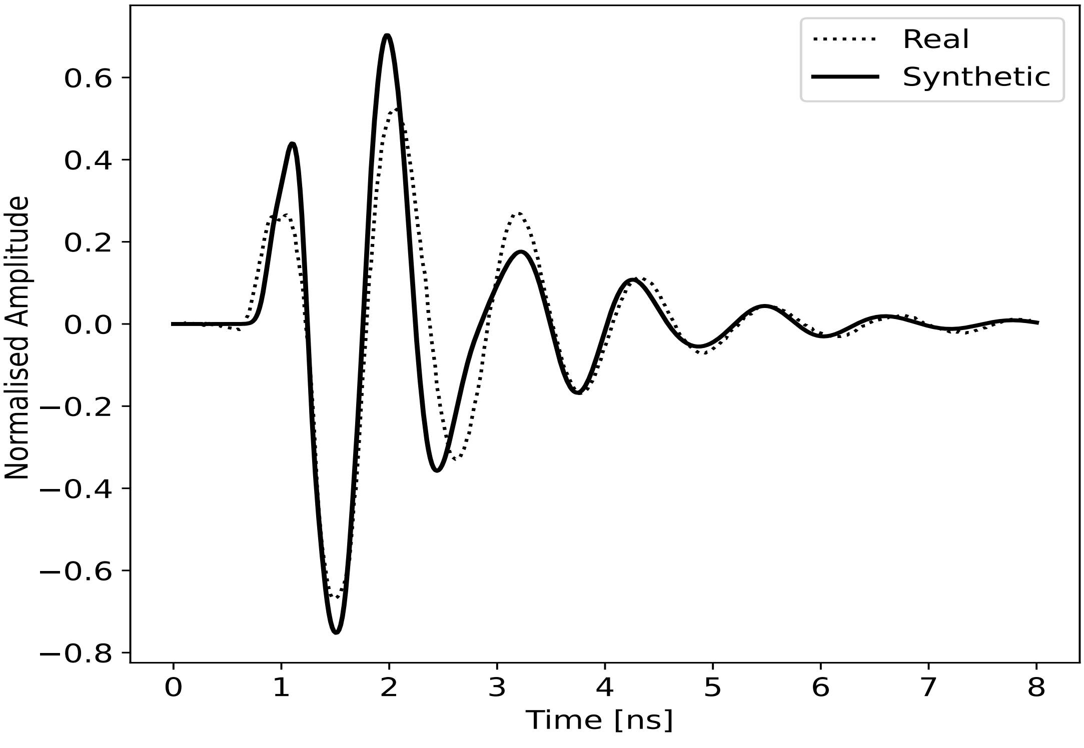
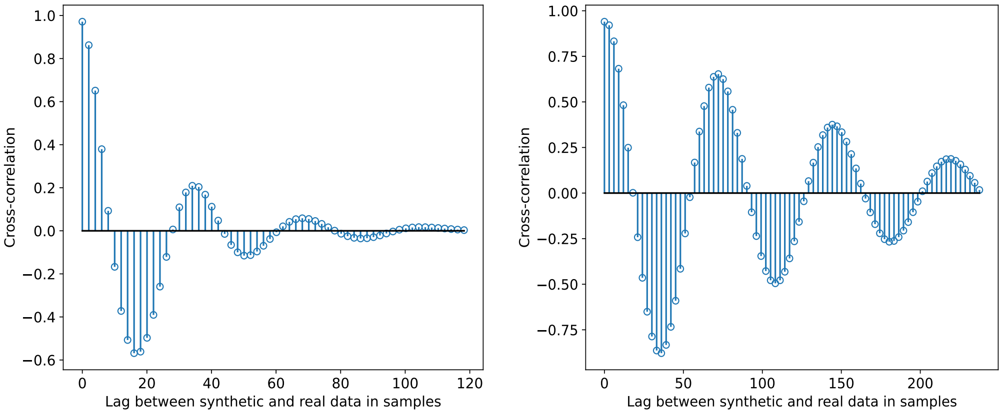
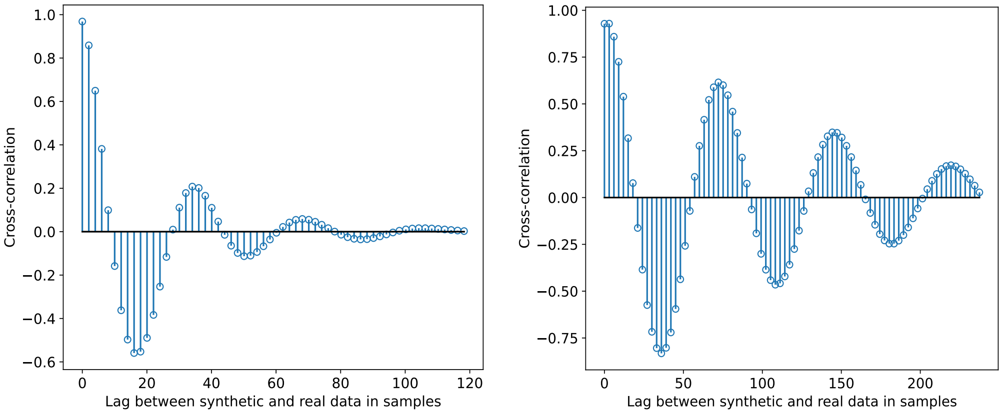


Figure 4.22: Real versus modelled responses from the metal plate using optimised values from the PSO (bottom) scheme.



(a) GA free-space

(b) GA pec



(c) PSO free-space

(d) PSO pec

Figure 4.23: Cross-correlation between real and synthetic data for GA (top) and PSO (bottom).

100 Ω larger than the one obtained from the PSO and in the conductivities of the absorbers, considering also the ranges of values each parameter can take. The values obtained demonstrate how the two optimisation approaches arrive in different combinations of parameters as realistic solutions to the same problem.

Both optimisation schemes terminated after the maximum number of iterations was reached. The solutions remained unchanged for the last few iterations meaning that the algorithms achieved the smallest possible RSS error they could converge to. To determine which is the best model between the two, the root mean square (RMS) error is chosen as a criterion. RMS is calculated using both the responses from the free space and the PEC. The GA model resulted to an RMS value of 5.57, while the RMS for the PSO is 6.02 and therefore, the GA output is chosen as the optimal solution. To further evaluate the goodness of the fit, the cross-correlation between real and synthetic responses was calculated and the cross-correlation plots are given in Figure 4.23 for both GA and PSO. Cross-correlation quantifies the similarity between two signals by shifting (lagging in

Optimised Parameters	**GA values**	**PSO values**
Absorber 1 ϵ_{a1}	1.10	1.60
Absorber 2 ϵ_{a2}	1.07	1.75
PCB ϵ_{pcb}	1.52	2.06
Skid ϵ_{s}	2.67	2.13
Plastic Case ϵ_{p}	6.1	5.2
Shield ϵ_{sh}	1	1
Absorber 1 σ_{a1} (S/m)	1.08	0.79
Absorber 2 σ_{a2} (S/m)	1.26	2
PCB σ_{pcb} (S/m)	0.023	0.0006
Skid σ_{s} (S/m)	0.005	0.001
Plastic Case σ_{p} (S/m)	0.0029	0.0014
Shield σ_{sh} (S/m)	4.5×10^{7}	5.6×10^{8}
$R_s(\Omega)$	560	445
$\sigma_R(\Omega^{-1})$	0.005	0.01
f (GHz)	2.12	2.07

Table 4.2: Optimised antenna parameters.

time) the one signal across the other and calculating the correlation coefficient at each lag. The correlation coefficient can take values between [-1, 1], with -1 meaning a perfect negative (inverse) correlation, 1 meaning a perfect positive (exact match) correlation and 0 meaning no correlation between the two signals. The lag that results in the highest correlation coefficient has the best match between the signals. The responses computed using the GA parameters resulted in the highest correlation of 97% for the free space and 95% for the pec, whereas for the PSO the highest correlations were 96% and 94% for the free-space and the pec, respectively. Thus, showing the very good match between synthetic and real data.

The GA algorithm required an overall execution time of 8,5 days on an Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz CPU, whereas the PSO required a runtime of 4.5 days on the same system. The execution times are presented for the algorithms executed with the final settings that resulted after tuning the solvers, which were described above.

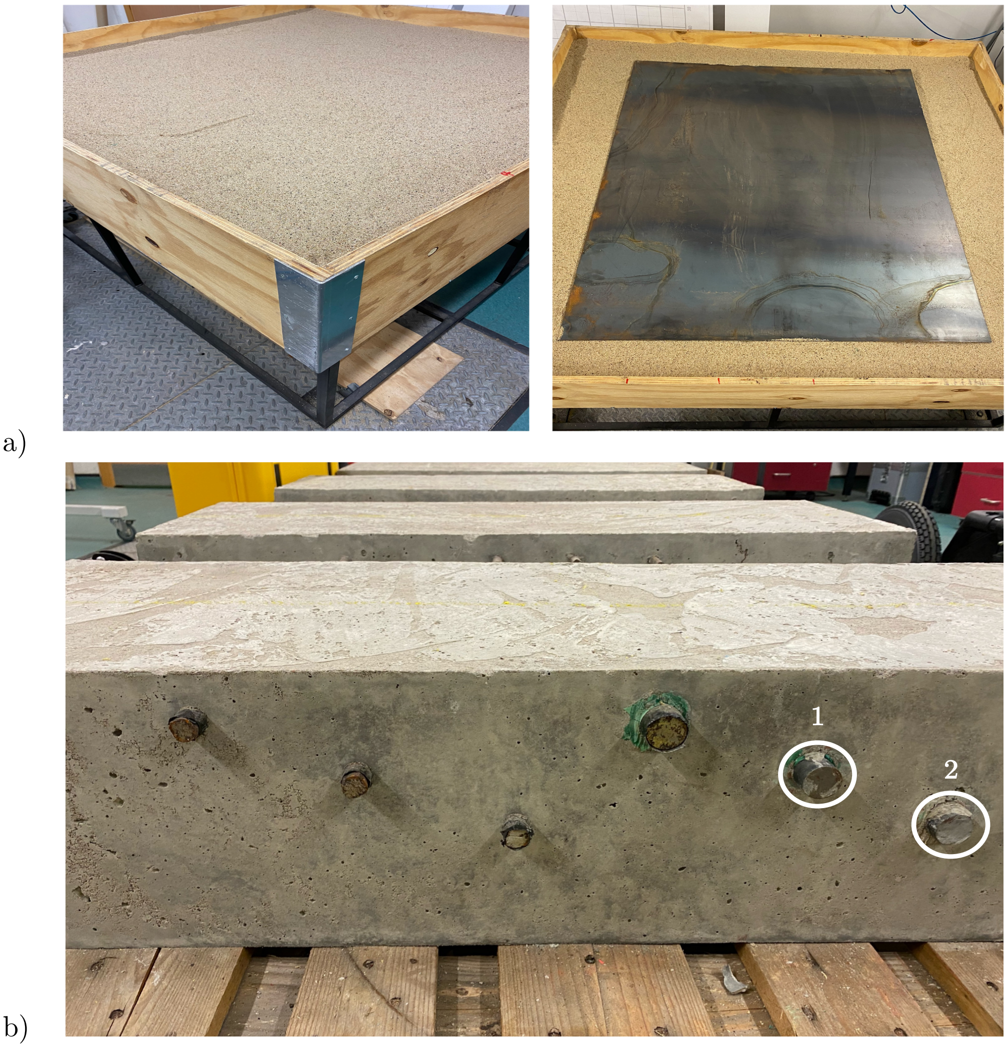


Figure 4.24: Real setup of the validation scenarios: a) Sandbox scenario. b) Concrete slab scenario.

4.7 VALIDATION OF THE ANTENNA MODEL

In order to demonstrate the accuracy of the numerical model and investigate how it responds to divergent scenarios, two experiments were conducted and GPR data were acquired using the real GSSI 2 GHz transducer. Both case studies were replicated in the simulations and compared with the real responses. For the first experiment, a metallic plate was buried at a depth $d = 19$ cm in a wooden box which contained dry sand as a background medium. Both the sand box and the metallic plate are shown in Figure 4.24a, while the modelled geometry of this scenario is illustrated in Figure 4.25a. The sand was modelled as a non-dispersive, homogeneous material with $\epsilon_r = 3$ and $\sigma = 0.01 S/m$, while the metallic plate is modelled as a PEC. The second scenario was from a reinforced concrete slab, visible in Figure 4.24b. The dispersive behavior of concrete was modelled using

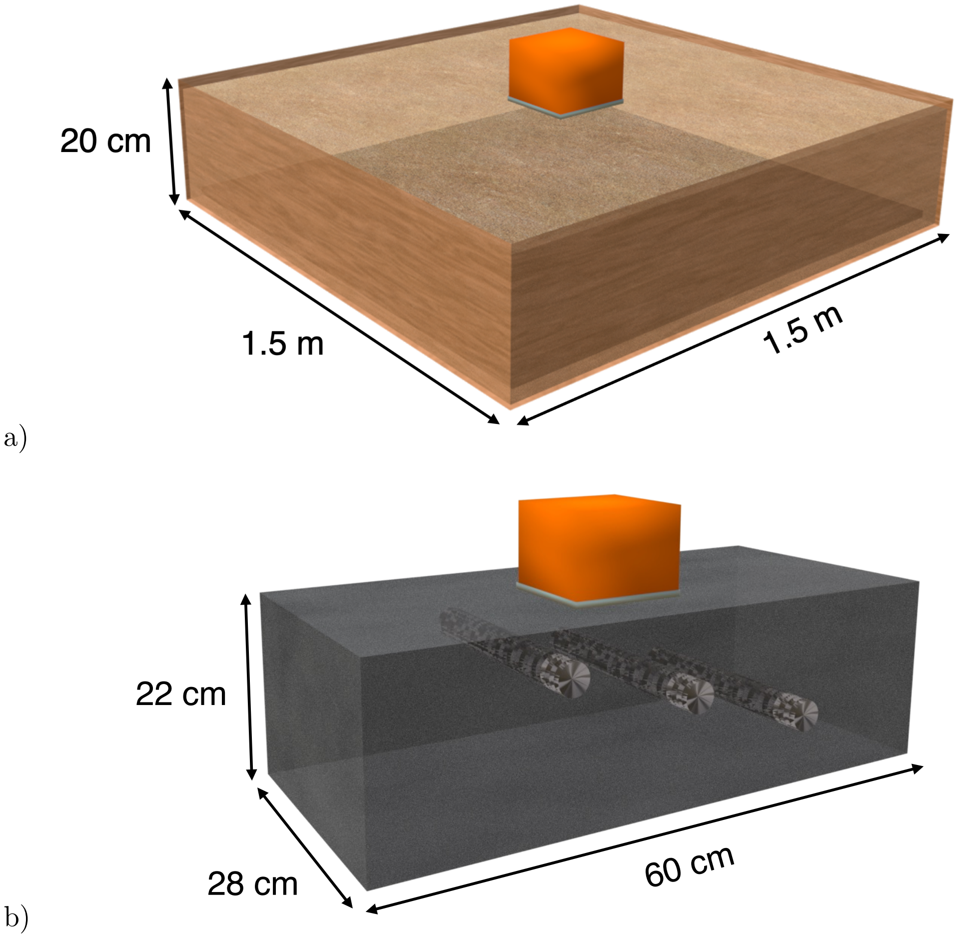


Figure 4.25: Modelled geometry of the validation scenarios: a) Sandbox scenario. b) Concrete slab scenario.

a single Debye pole with a relative permittivity at infinite frequency $\epsilon_\infty = 7.3$, zero-frequency relative permittivity $\epsilon_p = 12.2$, relaxation time of $t_p = 0.62 \times 10^{-9}$ s and a conductivity of $\sigma = 0.05$ (S/m). Responses from the two reinforcing bars (rebars), annotated in 4.24b, were investigated. The first rebar is buried at $d = 6.3$ cm and the second rebar at a depth $d = 10cm$, while both rebars have a radius of $r = 1.25cm$. Both rebars are steel rebars and were simulated as a PEC. The modelled geometry for the slab is illustrated in Figure 4.25b. To replicate accurately the real concrete slab, the third rebar in the vicinity of the two investigated rebars was also included in the model, as illustrated in the Figure.

The parameters used for both scenarios are only approximates of the true dielectric properties of the materials. The domain size was set to $280 \times 240 \times 360$ mm for the first scenario and to $600 \times 280 \times 330$ mm for the second, considering the dimensions of the transducer and the real setup of each case. In the simulations, all materials were considered homogeneous and of infinite extent, which is not

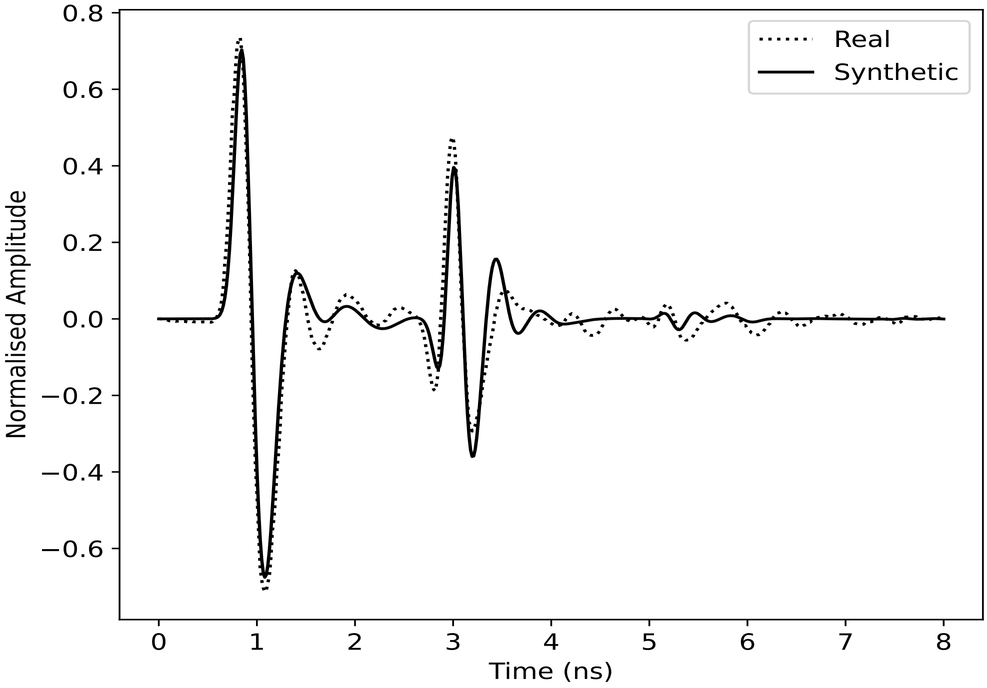


Figure 4.26: Real versus modelled A-scans from the metal plate buried in sand.

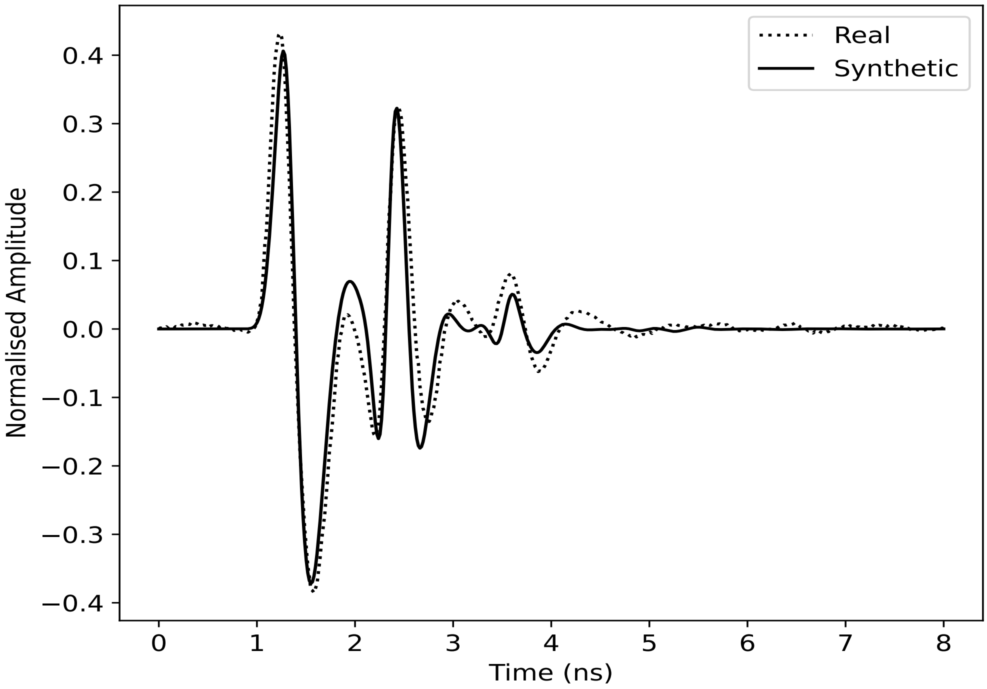


Figure 4.27: Real versus modelled A-scans from the reinforced concrete slab for rebar 1.

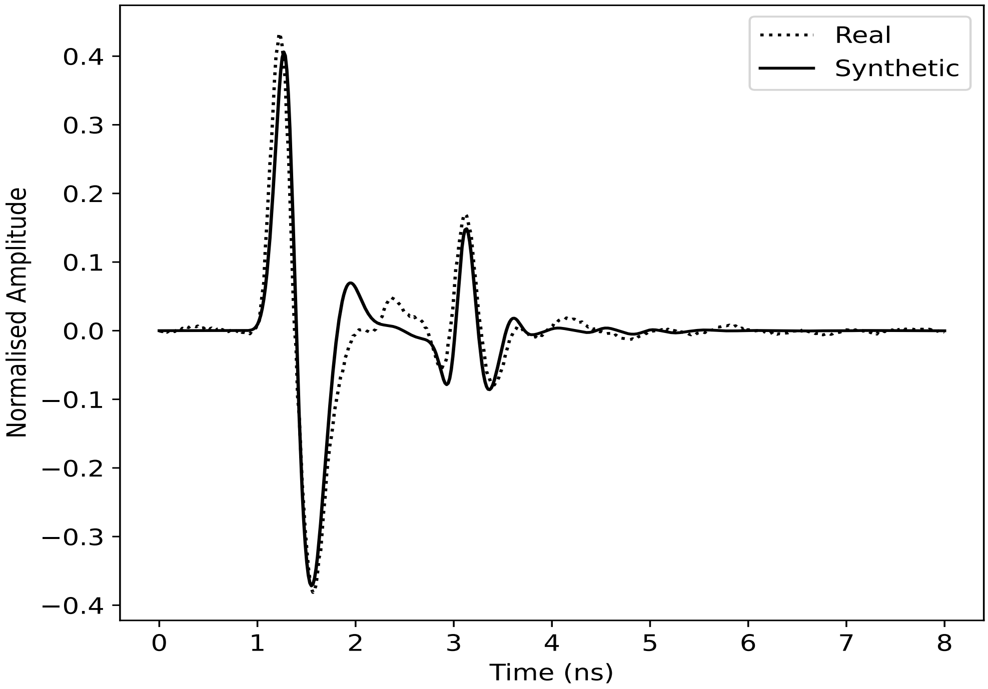
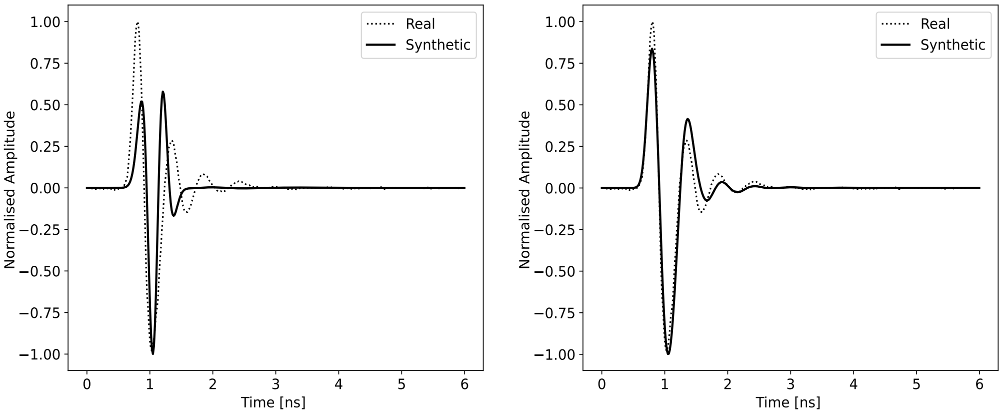


Figure 4.28: Real versus modelled A-scans from the reinforced concrete slab for rebar 2.

true for the real finite slab and sandbox. Furthermore, unwanted responses from the bottom of the slab and other sources of noise that are present in the real data, are not included in the simulations.

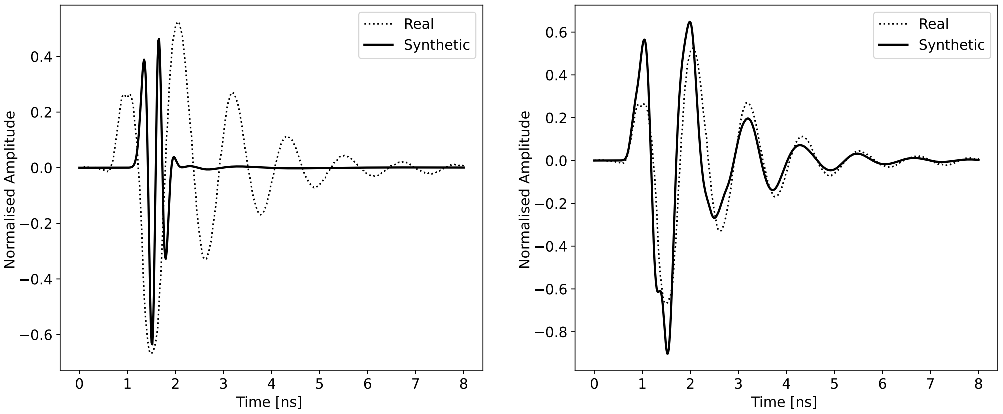
The modelled A-scan responses versus the true ones for the sand are plotted in Figure 4.26, while for the first and second rebar of the concrete slab in Figures 4.27 and 4.28, respectively. The highest cross-correlation between the synthetic and real validation data was 96% for the sand and 94% for both the rebars of the concrete slab. Given all the above, it is obvious that the modelled data follow closely the pattern of the real responses, predicting accurately the arrivals of the responses, the amplitudes and shape in all cases. This demonstrates that the antenna model can replicate accurately the behavior of the real system.

To further demonstrate the importance of including a numerical model of a real GPR transducer in the simulations and also the need for optimisation of the antenna parameters, responses from free-space, pec and the sandbox scenario were generated using a) a Hertzian dipole and b) an uncalibrated model of the GSSI 2GHz antenna in the simulations. The uncalibrated model has the exact geometry as the optimised model presented above but the antenna parameters were not chosen though optimisation; instead the average values were used for each of the parameters, which were calculated based on the constrained ranges presented in Table 4.1. For the excitation, a voltage source with a Gaussian-shaped pulse of 2GHz center frequency was used for the uncalibrated model, whereas for the Hertzian dipole a Ricker waveform with the same frequency



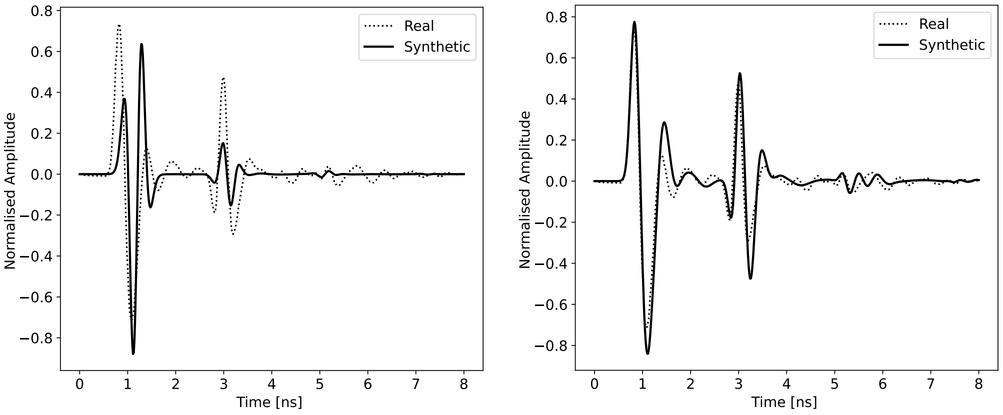
Hertzian dipole - Free space

Uncalibrated - Free space



Hertzian dipole - PEC

Uncalibrated - PEC



Hertzian dipole - Sand

Uncalibrated - Sand

Figure 4.29: Real versus modelled responses using a hertzian dipole (left) and an uncalibrated antenna model (right) for free-space, pec and sand cases.

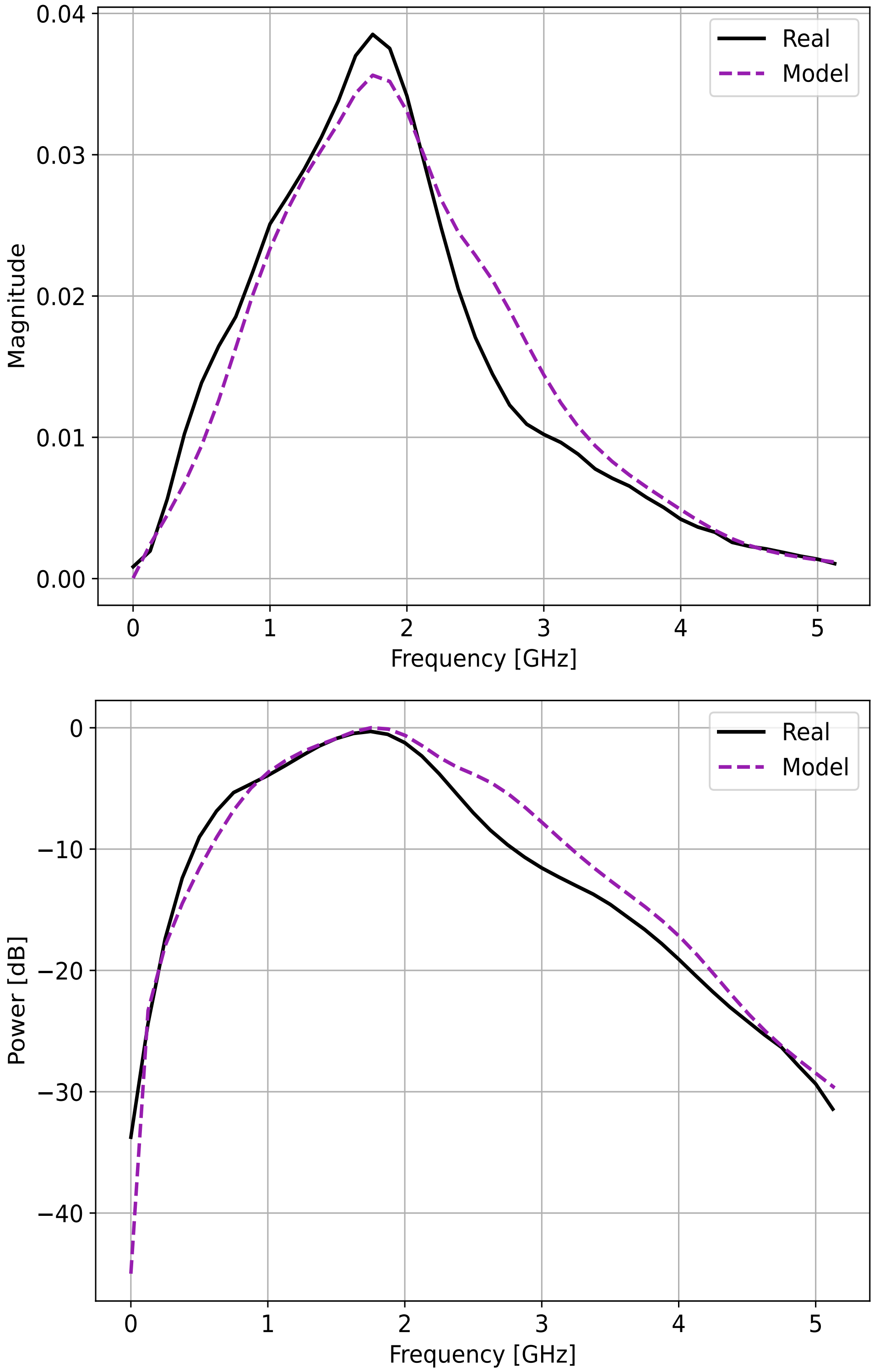


Figure 4.30: a) Magnitude spectrum in free space, b) Power spectrum in free space.

was used. Figure 4.29 shows the real versus the synthetic responses using a hertzian dipole (left) and an uncalibrated antenna model (right). Observing the results using a theoretical source, it is obvious that the synthetic responses have significant differences with the real traces and cannot be used to assist in the interpretation of real data, supporting the need for realistic antenna models in the simulations. Using an uncalibrated antenna model, the synthetic responses approximate better the true data, with the overall patterns being similar for all cases but still having considerable differences. By optimising the antenna parameters, these dissimilarities were reduced, yielding responses very close to the true ones.

In addition, the frequency spectrum for the free-space responses were investigated. It is assumed that the bandwidth of the antenna is the same as the center frequency and so, the antenna behaves efficiently at the range 1 GHz to 3 GHz. Both the normalised magnitude and the power spectrum of the real and the modelled free-space response are displayed in Figure 4.30 for frequency range 0-5 GHz, showing similar patterns with the peak energy being also around the same frequency for both signals. The above demonstrate that global optimisers can be successfully employed for creating models of commercial antennas with unknown characteristics and having limited knowledge.

4.8 RADIATION PATTERNS OF THE ANTENNA

An antenna radiation pattern as defined in the IEEE Standard Definitions of Terms for Antennas (IEEE Std 145, 2013) is

> The spatial distribution of a quantity that characterizes the electromagnetic field generated by an antenna

The radiation pattern is usually presented as a graphical representation of a property as a function of space. These properties can be the field strength, power, directivity or phase among others. When the spatial distribution of the electric (or magnetic) field at a constant radius is presented, it is called a field pattern, whereas the spatial variation of the power along a constant radius is called a power pattern. Usually, the patterns are normalised to their maximum value and presented in a logarithmic scale or decibels (dB) in order to amplify parts with very low values. In practice, the 3D antenna patterns are presented as a series of 2D patterns of the E-plane or the H-plane. The E-plane is the plane that contains the E-field and the direction of maximum radiation, whereas the H-plane contains the H-field and the direction of maximum radiation (Balanis, 2015).

The different parts of a radiation pattern are known as lobes and can be classified as major, minor, side or back lobes. As major lobe is defined the radiation lobe which contains the direction of the maximum radiation, whereas a minor lobe is any lobe except the major lobe. A side lobe includes any other direction rather the intended direction. A back lobe is a radiation lobe which is in the direction exactly opposite of the major lobe with his axis forming an $\sim 180°$ angle with respect to the antenna beam. The minor lobes represent radiation in undesired directions, which should be minimised (Balanis, 2015).

Usually, the radiation patterns are presented for a single frequency. However, for UWB systems such as a GPR antenna, it is more useful to construct a radiation pattern of the total emitted energy in a certain angular direction, instead of plotting a pattern at a single frequency. A measure of the total emitted energy $\mathcal{E}_{tot}$ given by Diamanti and Annan (2013) and adapted from Warren and Giannopoulos (2017) is:

$$\mathcal{E}_{tot} = \sum_{t=0}^{T} E(r, \theta)^2 \tag{4.14}$$

where E is the electric field, r and θ are the observation radius and angle, respectively and T is the time window.

Theoretical radiation patterns of theoretical antennas, such as an infinitesimal dipole, have been made in free space (King, 1956). Some efforts of studying the patterns of infinitesimal dipoles over lossless and low-loss materials can also be found in literature. To measure the radiation pattern of a real transducer, a second antenna is required. Using a second unit, patterns have been created for both simple and commercial GPR antennas in free space (Klysz et al., 2006; Pérez-Gracia et al., 2009), and over low-loss homogeneous (Annan et al., 1975) materials. However, measuring radiation of real antennas in realistic lossy heterogeneous environments faces many difficulties that make it hard to implement in practice. Therefore, numerical simulations that include digital representations of real antennas can be used to approximate the real antenna patterns. Diamanti and Annan (2013) characterised the energy distribution of a shielded dipole antenna over a range of lossless half-spaces using 3D FDTD simulations. Warren (2009) developed radiation patterns of a 3D FDTD accurate model of a GSSI 1.5 GHz antenna and a model of a 1.2 GHz antenna from MALA in lossless homogeneous and lossy environments, while the same group studied the radiation patterns of the GSSI model over lossy heterogeneous materials (Warren and Giannopoulos, 2017).

Radiation patterns of the GSSI 2GHz antenna model were developed for the free space and over lossless and lossy homogeneous and inhomogeneous media. The E-plane and H-plane radiation patterns were acquired by measuring the E-field and the H-field, respectively, for observation distances between 0.11-0.35m of the center of the antenna with 0.015m step. For each observation distance, measurements were taken at every degree of a circle centred on the transmitter bowties of each antenna, with radius equal to the observation distance. Since this is a high frequency antenna with small penetration depth, for the center frequency used most of the targets that can be detected are at a depth <=0.35m depending on the ground conditions. The theoretical boundary where the far-field region starts for this antenna in free space is at 0.054 m. However, this is only theoretical and can be different from the true distance that the far-field effects start to take occur (Wenskink et al., 1990; Valle et al., 2001). The patterns are plotted using a logarithmic scale and are normalised to the maximum value of all distances. No compensation for distance was made for plotting the patterns.

The radiations patterns for the free-space are depicted in Figure 4.31. Although the free space patterns are insignificant for GPR, since the GPR antennas are not designated for use in free space, they are presented for comparison with the half-space patterns. From the Figure, it is observed that the E-plane exhibits a broad main lobe in the downward direction, whereas energy going in the upward direction is reduced due to the shielding of the antenna. Regarding the H-plane, its broad pattern is similar to that of an infinitesimal dipole in free-space.

Figure 4.32 illustrates the radiation patterns of the antenna model over a lossless half-space with $\epsilon_r = 5$, while Figure 4.33 displays the patterns over a lossless half-space with $\epsilon_r = 10$ for both the E-plane and H-plane. The presence of a main lobe is visible in all patterns with most of the energy going into the subsurface. The maximum energy of the E-plane is at an angle of 180°, perpendicular to the antenna, while for the H-plane occurs at 160°, due to the transmitter and receiver being offset from each other. The existing backlobes are energy going into the air, which is smaller since the bowties are shielded. As it is expected, with increasing relative permittivity of the half-space, the main lobe becomes narrower and more directive. In addition, the energy escaping in the air is minimised with increasing permittivity since more energy gets drawn into the subsurface.

The radiation patterns for the same previous lossless cases are demonstrated in Figure 4.34 and Figure 4.35 for $\epsilon_r = 5$ and $\epsilon_r = 10$, respectively, but in this case the transducer is lifted 1cm above ground. As expected for both cases, the minor back lobes are expanded with more energy going into the air, whereas the main lobes have become more directive. Again, with increasing permittivity the directivity of the main lobes is increasing.

For the half-space with $\epsilon_r = 10$, a conductivity of $\sigma = 0.05$ S/m was introduced and the radiation patterns over the lossy homogeneous medium were calculated and shown in Figure 4.36. Comparing the radiation patterns in Figures 4.36 and 4.33, the attenuation introduced by the conductivity in the fields with increasing distance from the antenna becomes obvious.

A lossy inhomogeneous space was also investigated. To build a realistic soil model, a mixing model suggested by Peplinski et al. (1995) was used to generate a set of materials with different dispersive properties. The soil created has a sand fraction $S = 0.4$, a clay fraction $C = 0.6$, bulk density of 2 g/cm^3, sand particle density of 2.66 g/cm^3, and a volumetric water fraction ranging between 0.005 - 0.2. A total of 60 materials with different dispersive properties were used to capture this range. Figure 4.37 displays the patterns over the lossy inhomogeneous space. Due to the inhomogeneities introduced in the subsurface, significant variations in the energy and the pattern shape can be observed. With increasing observation distances, attenuation in the fields is also observed due to the lossy nature of the soil.

Finally, since the GSSI 2 GHz transducer is mostly used for concrete scanning, the radiation patterns over two concrete slab with different dispersive properties were investigated. The first case represents a dry concrete slab simulated using a single Debye pole with a relative permittivity at infinite frequency $\epsilon_\infty = 4.5$, zero-frequency relative permittivity $\epsilon_p = 4.82$, relaxation time of $t_p = 0.83 \times 10^{-9}$ s and a conductivity of $\sigma = 0.0006$ (S/m). In the second case, concrete was modelled having a relative permittivity at infinite frequency $\epsilon_\infty = 7.3$, zero-frequency relative permittivity $\epsilon_p = 12.2$, relaxation time of $t_p = 0.62 \times 10^{-9}$ s and a conductivity of $\sigma = 0.05$ (S/m), representing a concrete with higher water content. The radiation patterns for the first and second case are shown in Figures 4.38 and 4.39, respectively. The greater attenuation for the concrete model with higher water content can be clearly seen by comparing the two Figures.

4.9 SNAPSHOTS OF THE FIELDS

Figures 4.40 and 4.41 illustrate snapshots of the magnitude of the E-field over a lossy half-space with a buried pec cylinder for different times between 0.1-1.9 ns. The half-space was simulated with an $\epsilon_r = 4$ and a $\sigma = 0.001 S/m$, while the cylinder was simulated with a radius of 1.25 cm placed at a depth 8 cm. It is obvious that most of the EM energy goes into the subsurface and a portion of the transmitted energy gets reflected by the target as shown in the snapshots for $t \geq 1.0 ns$. The rest of the energy transmitted in the ground continues to propagate in the subsurface where attenuation reduces its amplitudes, whereas some energy wraps around the metallic target and reaches below it. Since no shield is perfect, part of the energy escapes into the air and travels upwards.

In Figure 4.42, snapshots of the current distribution on the bowties are displayed for different time steps between 0.26-0.75 ns. The current flows on the transmitting element, starting from the feed-point and travelling along the arms of the bowtie

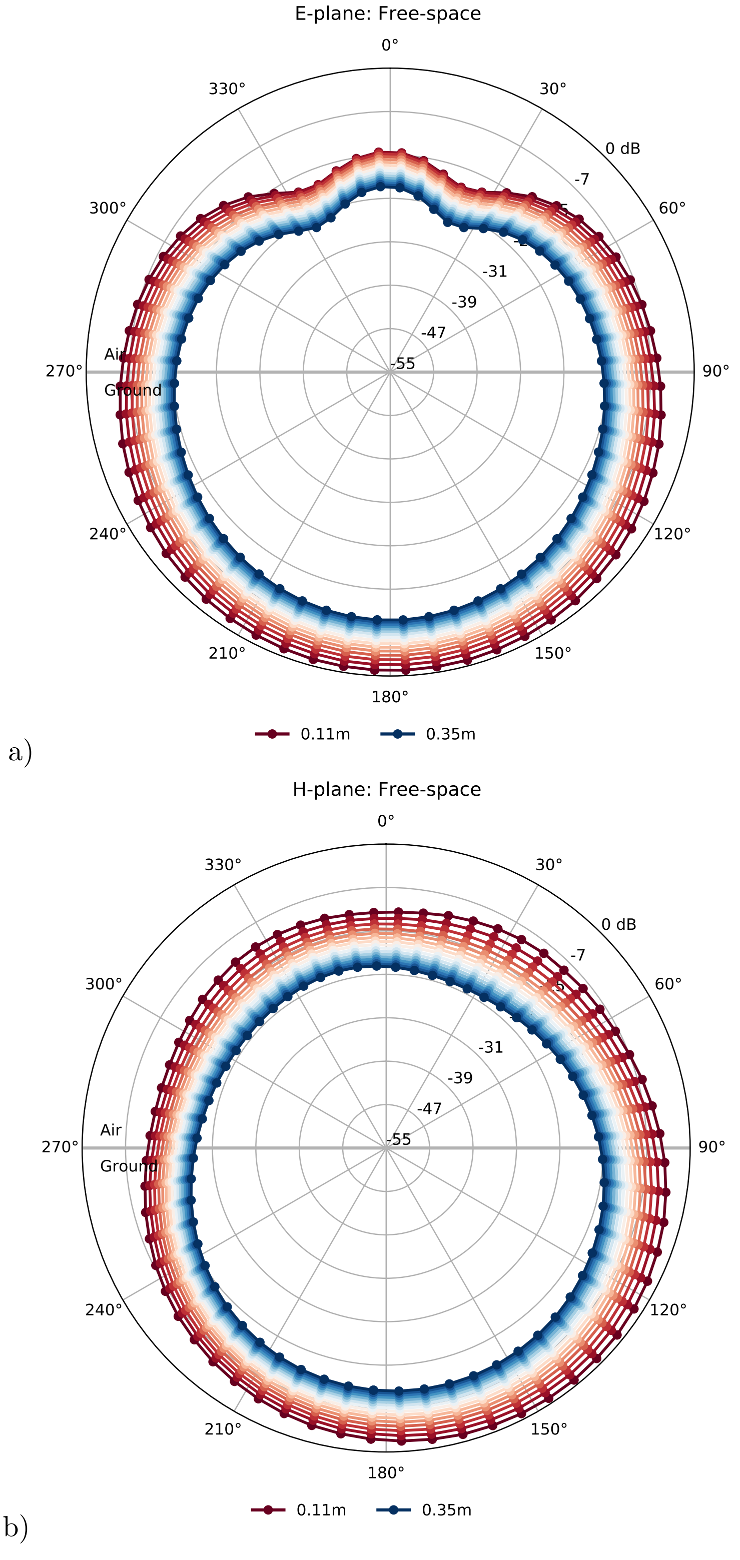
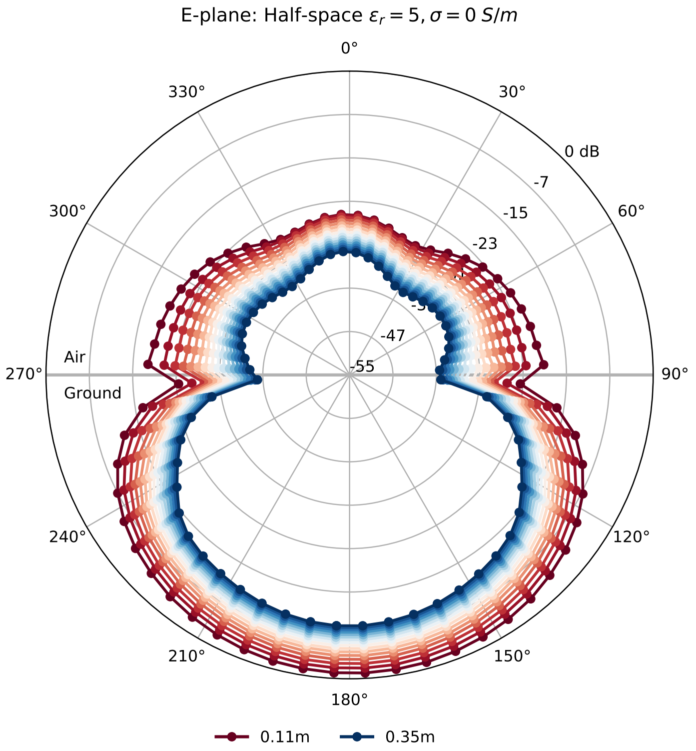
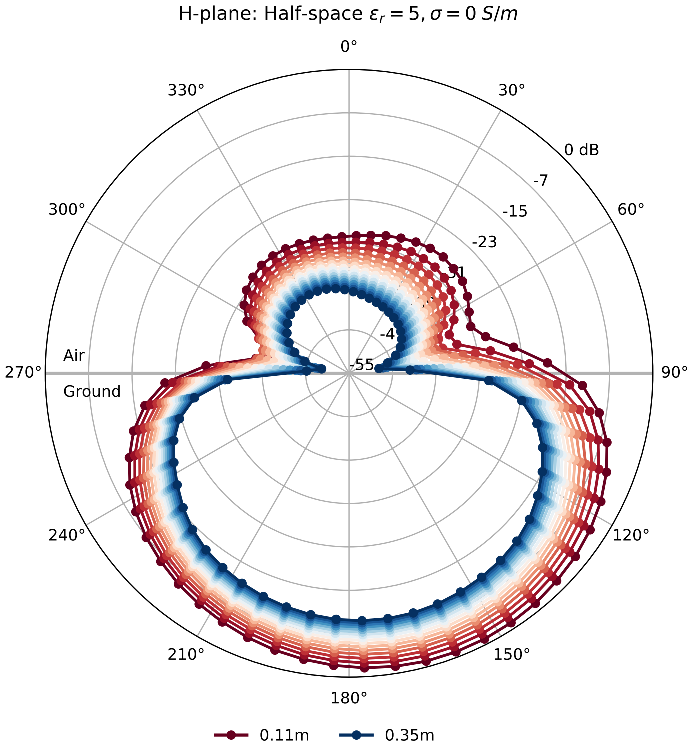


Figure 4.31: Radiation patterns of the 2GHz modelled antenna for observation distances between 0.11-0.35m from the center of the antenna for free-space: a) E-plane b) H-plane.

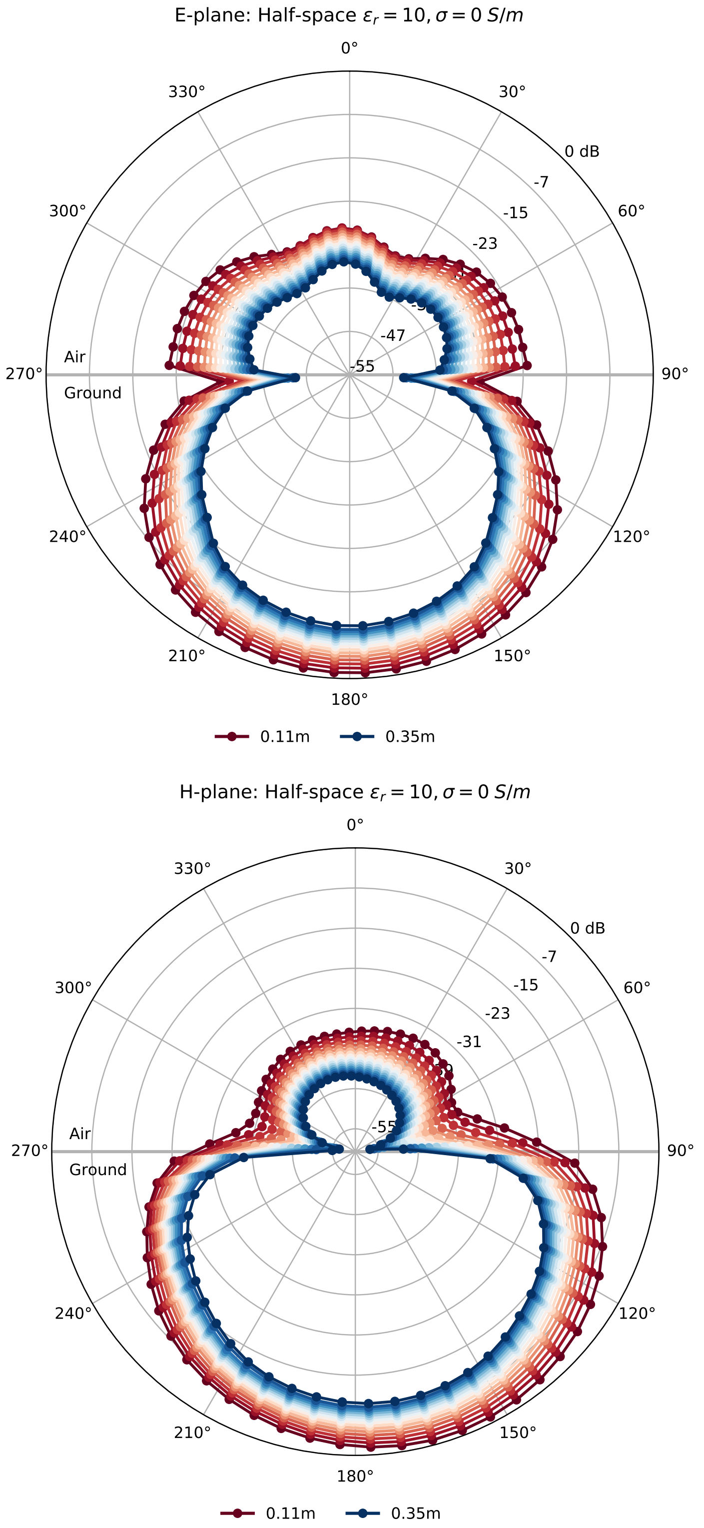


a)



b)

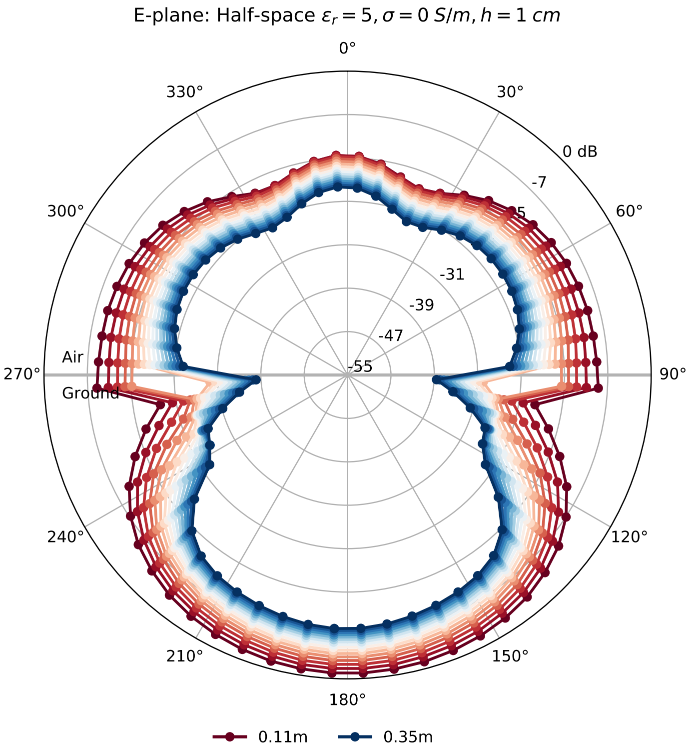
Figure 4.32: Radiation patterns of the 2GHz modelled antenna for observation distances between 0.11-0.35m from the center of the antenna for a lossless half-space with $\epsilon_r = 5$: a) E-plane b) H-plane.



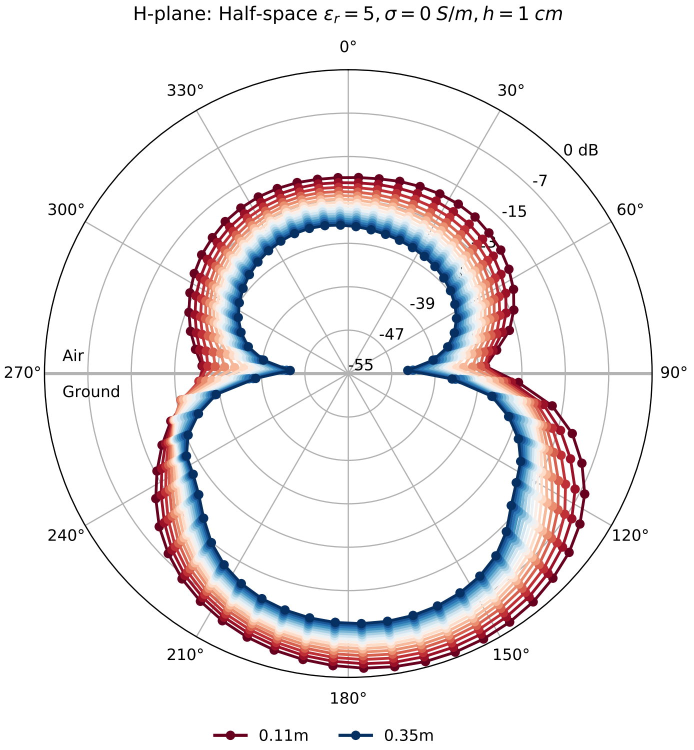
a)

b)

Figure 4.33: Radiation patterns of the 2GHz modelled antenna for observation distances between 0.11-0.35m from the center of the antenna for a lossless half-space with $\epsilon_r = 10$: a) E-plane b) H-plane.



a)



b)

Figure 4.34: Radiation patterns of the 2GHz modelled antenna for observation distances between 0.11-0.35m from the center of the antenna for a lossless half-space with $\epsilon_r = 5$ and antenna height $h = 1cm$ above ground: a) E-plane b) H-plane.

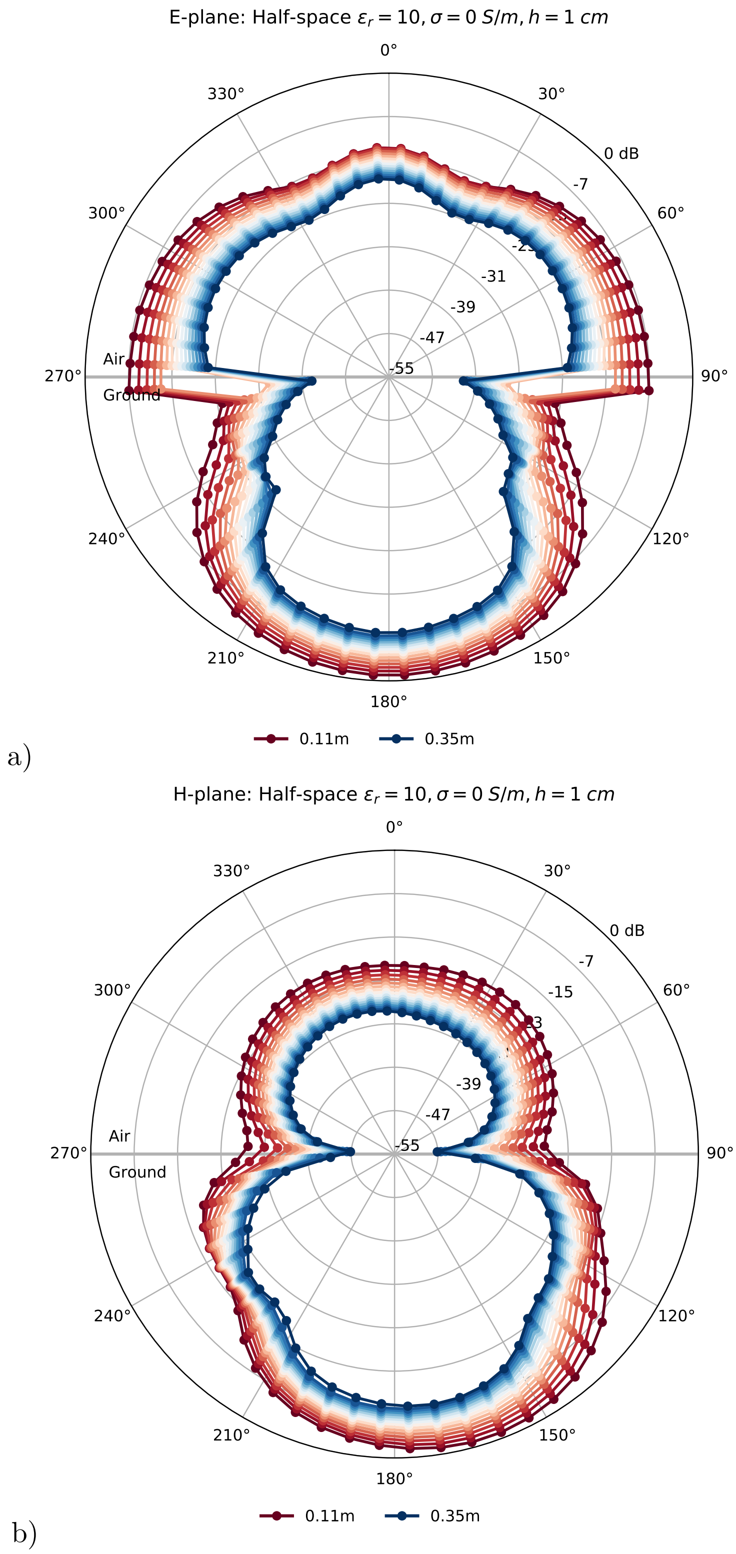


Figure 4.35: Radiation patterns of the 2GHz modelled antenna for observation distances between 0.11-0.35m from the center of the antenna for a lossless half-space with $\epsilon_r = 10$ and antenna height $h = 1cm$ above ground: a) E-plane b) H-plane.

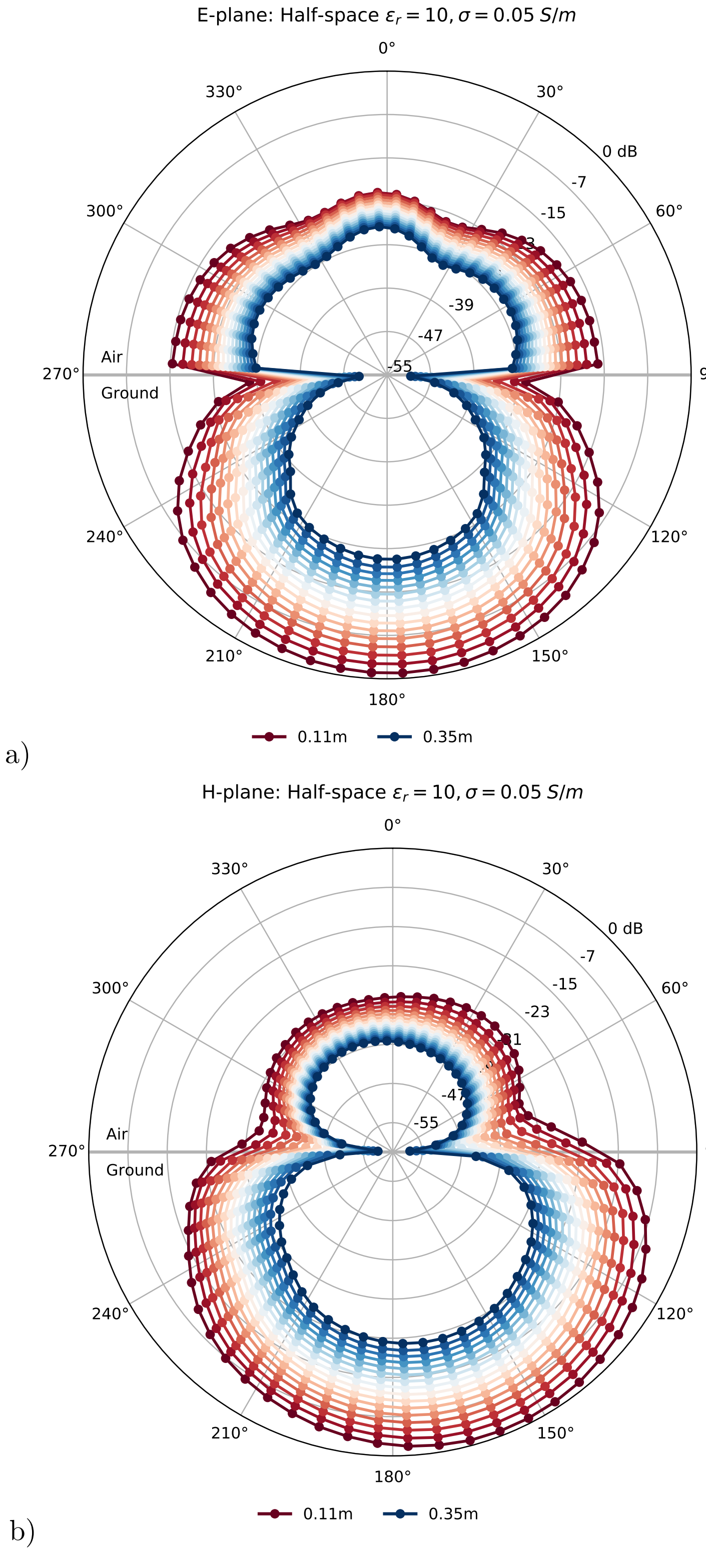


Figure 4.36: Radiation patterns of the 2GHz modelled antenna for observation distances between 0.11-0.35m from the center of the antenna for a lossy half-space with $\epsilon_r = 10$ and $\sigma = 0.05S/m$: a) E-plane b) H-plane.

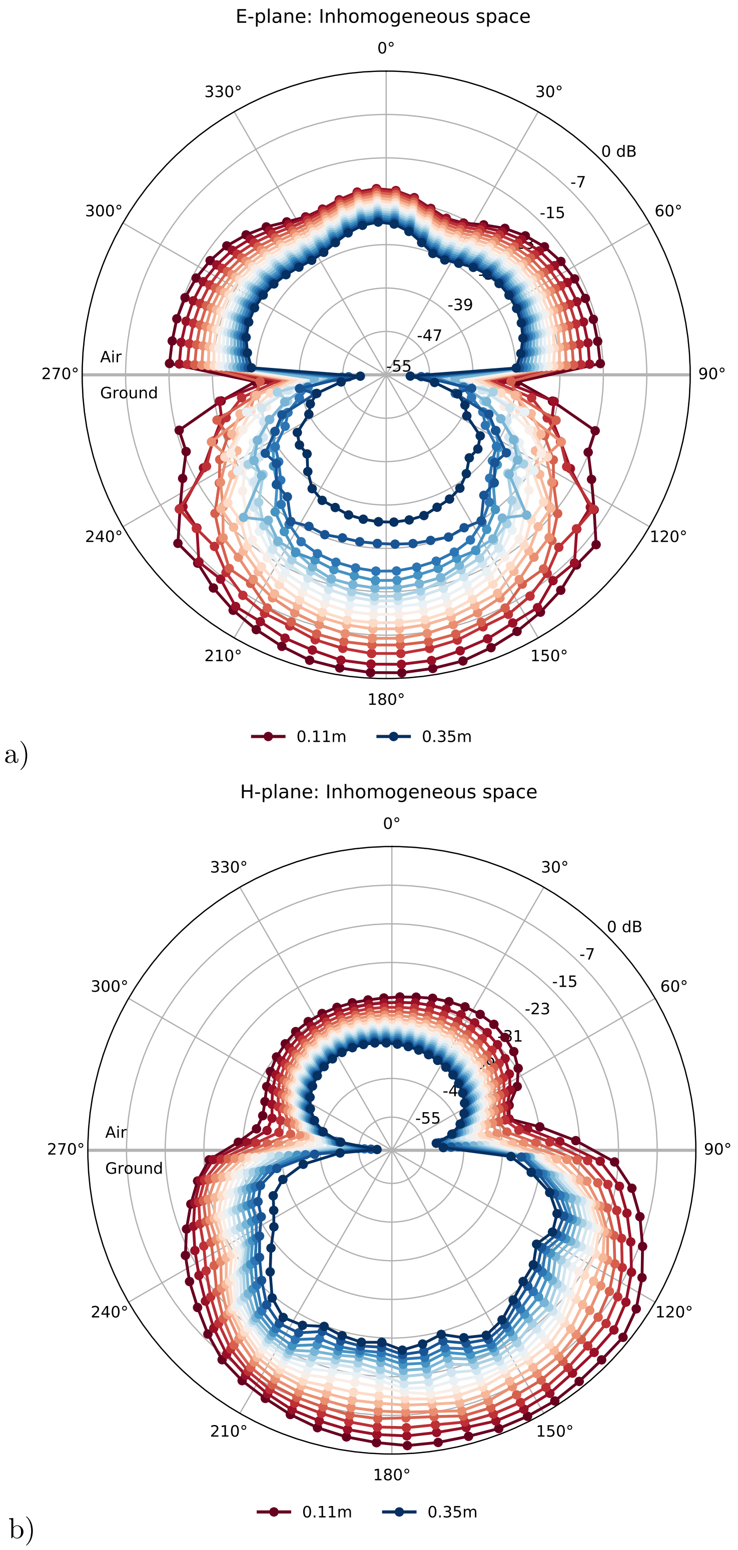
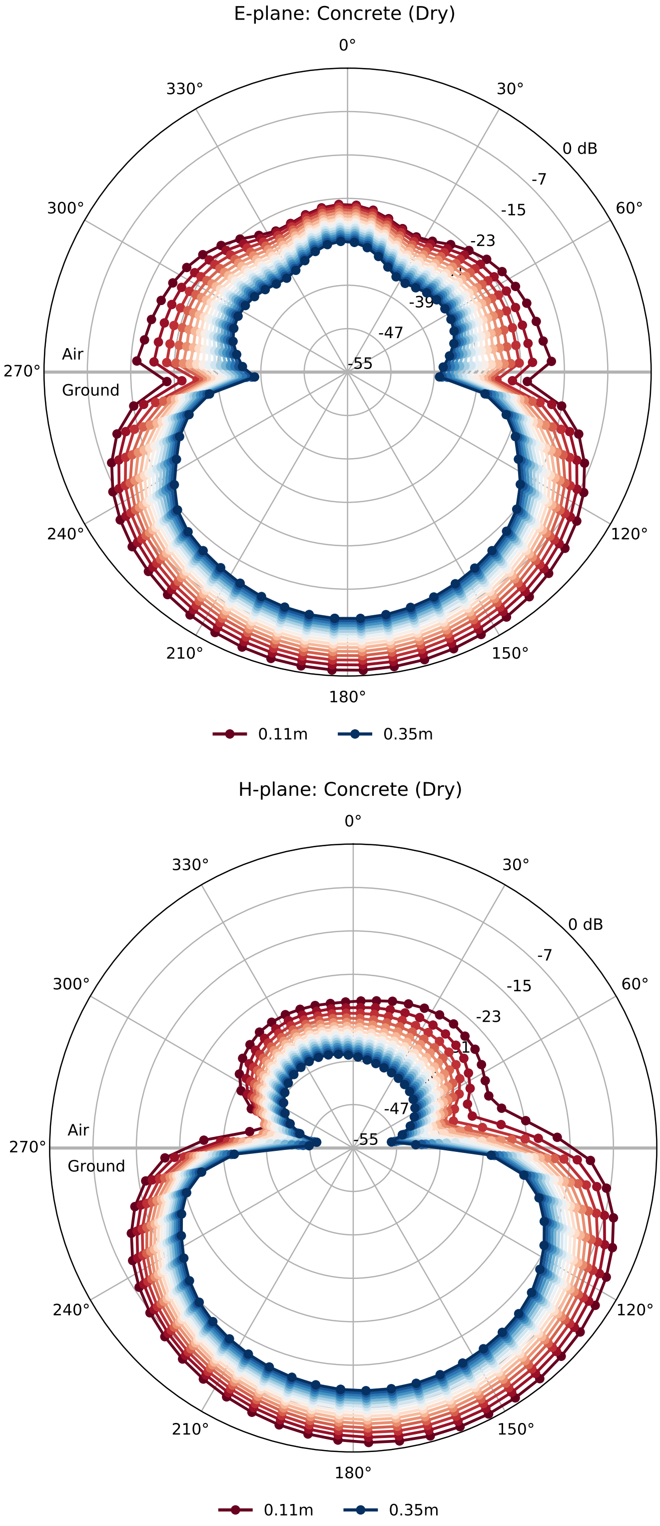


Figure 4.37: Radiation patterns of the 2GHz modelled antenna for observation distances between 0.11-0.35m from the center of the antenna for a lossy inhomogeneous space: a) E-plane b) H-plane.



a)

b)

Figure 4.38: Radiation patterns of the 2GHz modelled antenna for observation distances between 0.11-0.35m from the center of the antenna over a dry concrete slab simulated using a single pole Debye model: a) E-plane b) H-plane.

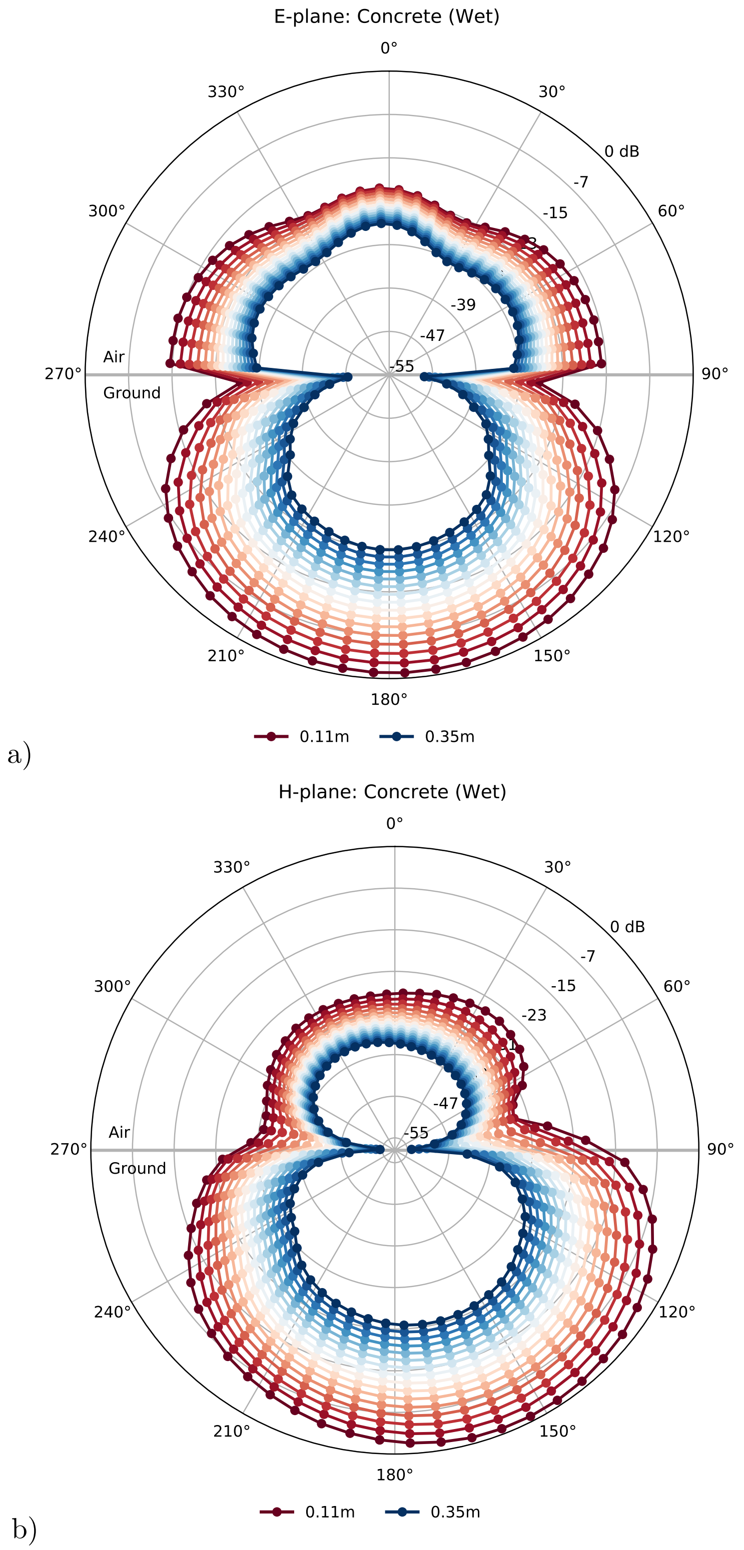
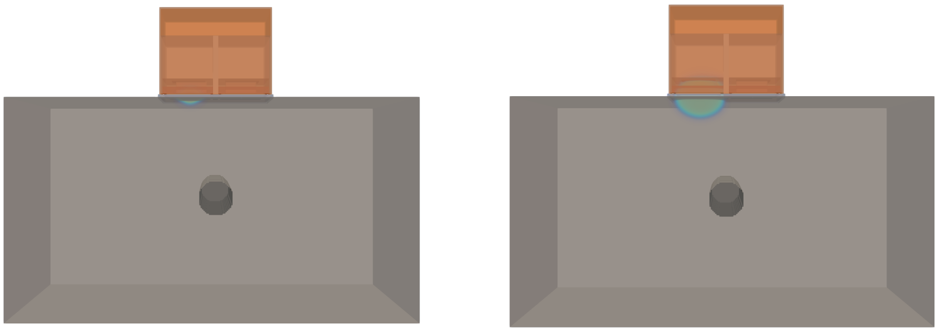
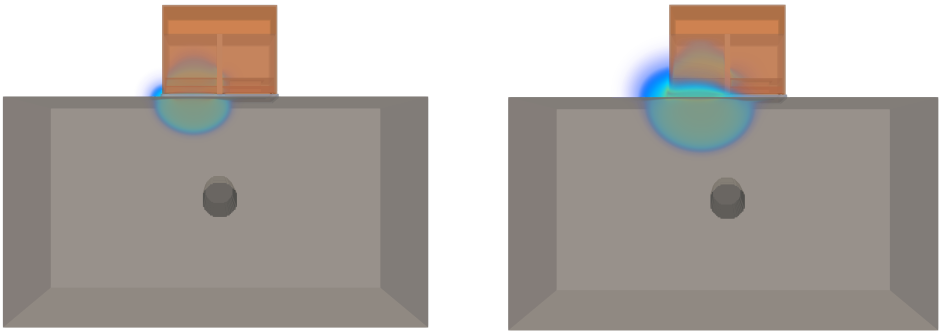


Figure 4.39: Radiation patterns of the 2GHz modelled antenna for observation distances between 0.11-0.35m from the center of the antenna over a wet concrete slab simulated using a single pole Debye model: a) E-plane b) H-plane.



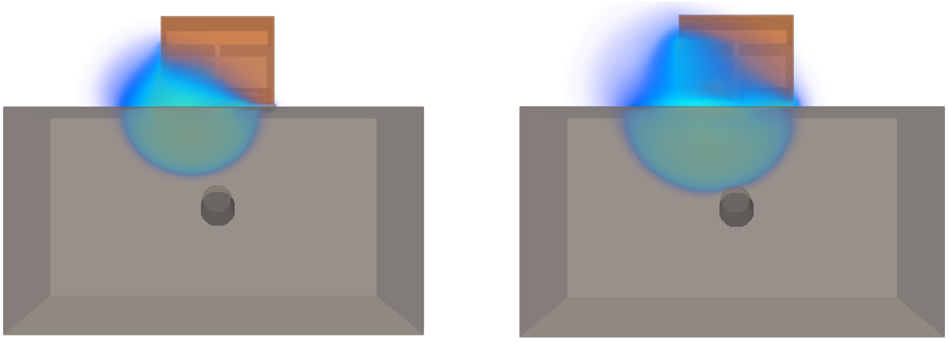
(a) t = 0.2 ns

(b) t = 0.3 ns



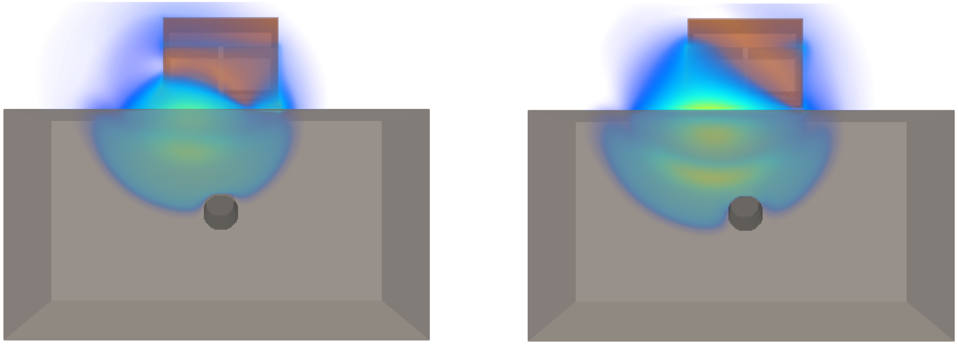
(c) t = 0.4 ns

(d) t = 0.5 ns



(e) t = 0.6 ns

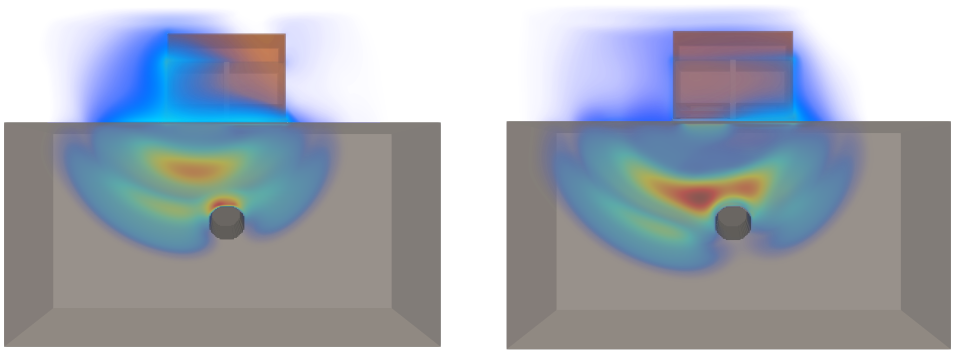
(f) t = 0.7 ns



(g) t = 0.8 ns

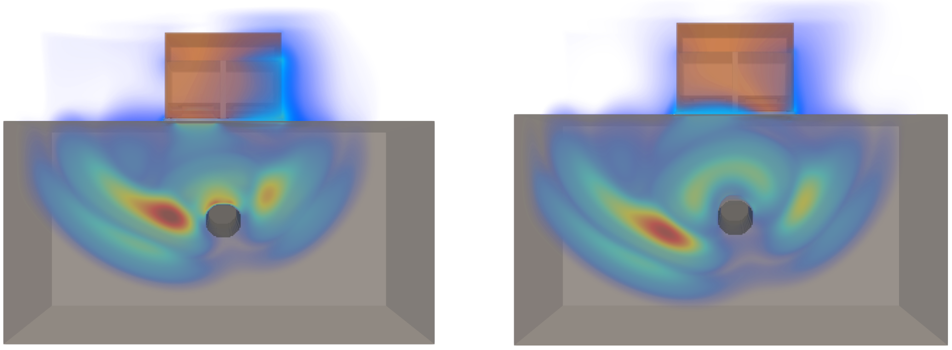
(h) t = 0.9 ns

Figure 4.40: E-field snapshots over a pec target in lossy half-space for t=0.2-0.9 ns.



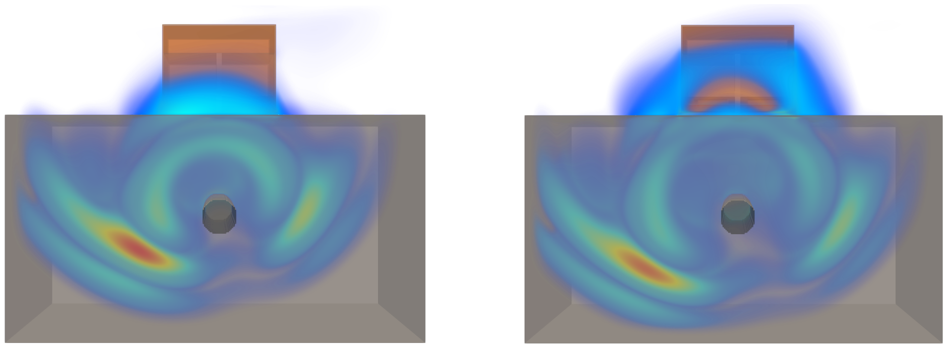
(a) t = 1.0 ns

(b) t = 1.1 ns



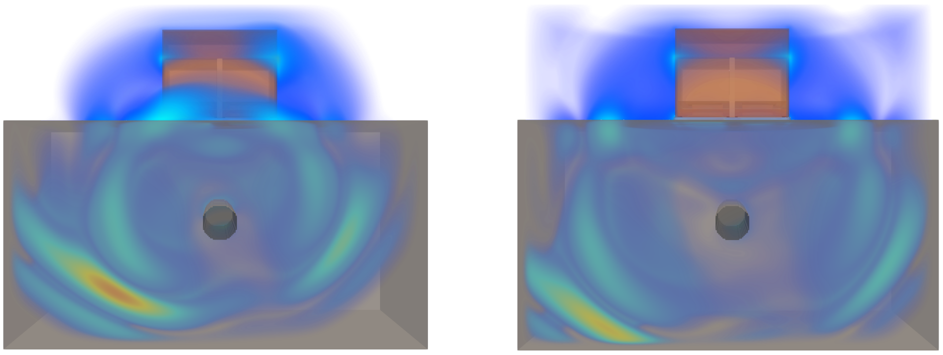
(c) t = 1.2 ns

(d) t = 1.3 ns



(e) t = 1.4 ns

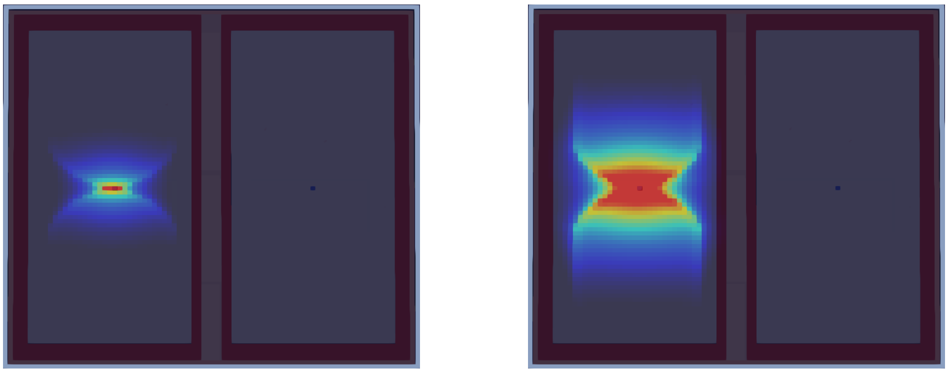
(f) t = 1.5 ns



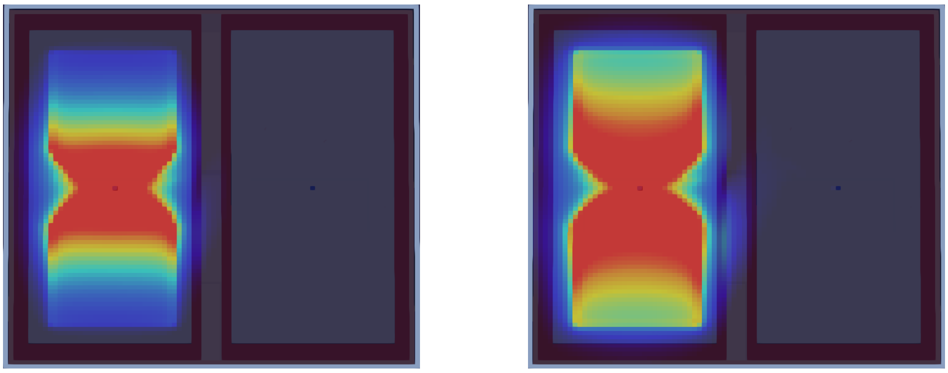
(g) t = 1.6 ns

(h) t = 1.9 ns

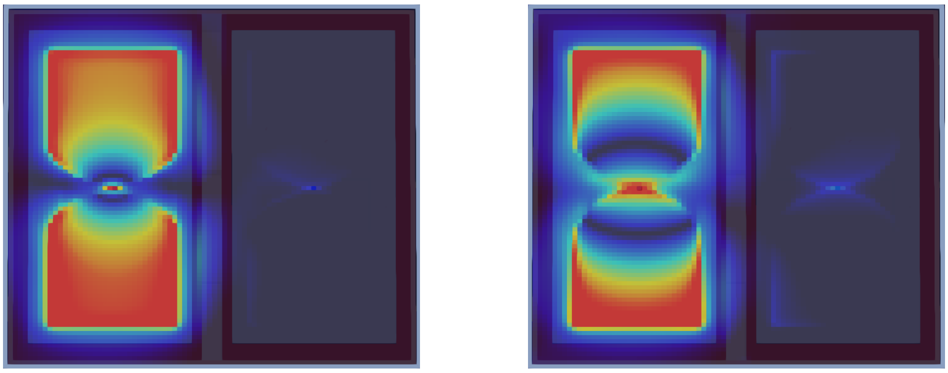
Figure 4.41: E-field snapshots over a pec target in lossy half-space for t=1.0-1.9 ns.



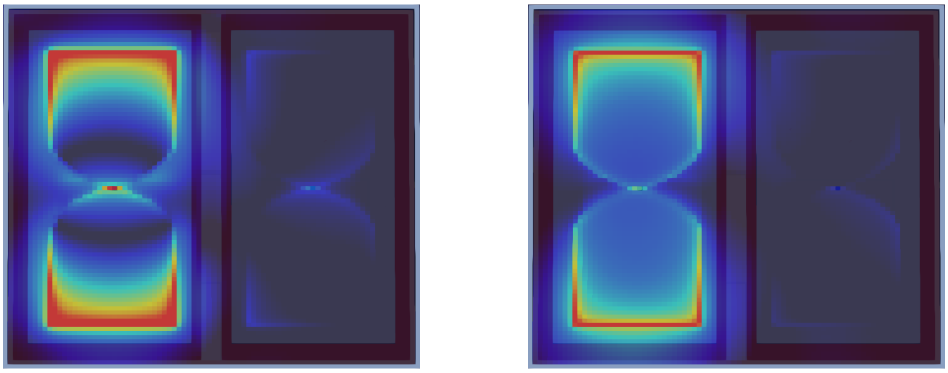
(a) t = 0.26 ns (b) t = 0.33 ns



(c) t = 0.4 ns (d) t = 0.47 ns



(e) t = 0.54 ns (f) t = 0.61 ns



(g) t = 0.68 ns (h) t = 0.75 ns

Figure 4.42: Current distribution snapshots for free-space for t=0.26-0.75 ns.

towards the open ends. Is is also shown, that current is present in the shield plate that separates the transmitter and the receiver.

4.10 GRID INDEPENDENCE ANALYSIS

One of the most important parameters in an FDTD simulation is the choice of the discretization step. The FDTD grid must be fine enough to resolve with a high degree of accuracy the smallest geometrical feature of the model.

In order to choose an acceptable step size for the modelled transducer, a grid independence analysis (Roache, 1998) is carried out. Grid independence analysis compares solutions for a number of different grid sizes and tries to estimate the discretization step for which the solution will be almost invariant even when using successively smaller cell sizes. The cell sizes which were used are 0.25, 0.5, 1 and 2 mm with the responses acquired in free space. First, Richardson extrapolation (Richardson, 1911) is employed to demonstrate the convergence of the solution. This method computes a higher-order estimate of the value that would have been obtained if a zero grid spacing was used from the existing lower-order numerical solutions. It can be applied to distinct points or to solution functionals. The following integrated quantity from the lower-order solutions is needed in order to acquire the value at zero grid spacing

$$U = \int_0^T u^2 dt \tag{4.15}$$

where u represents the A-scan, T is the time window and dt is the time step used based on the CFL condition. Assuming a second-order numerical solution and using the solutions obtained for the two finer grids, an approximate value at zero grid spacing can be obtained as:

$$U_{dx=0} = U_1 + \frac{U_1 - U_2}{r^p - 1} \tag{4.16}$$

where U_1 and U_2 are the integrated quantities for the solutions using a step size of 0.25 and 0.5 mm, respectively, r is the ratio of the cell sizes and p is the order of convergence of the method. For a second-order scheme, the theoretical order of convergence is p = 2, but the observed order is usually lower due to numerical approximations, boundary conditions etc. and can be calculated using three grids (Roache, 1998) as:

$$p = \ln \frac{U_3 - U_2}{U_2 - U_1} / \ln r \tag{4.17}$$

which was shown to be p = 1.73 for this case. Figure 4.43 displays the convergence of the solution from coarser to finer grids to the value at zero spacing.

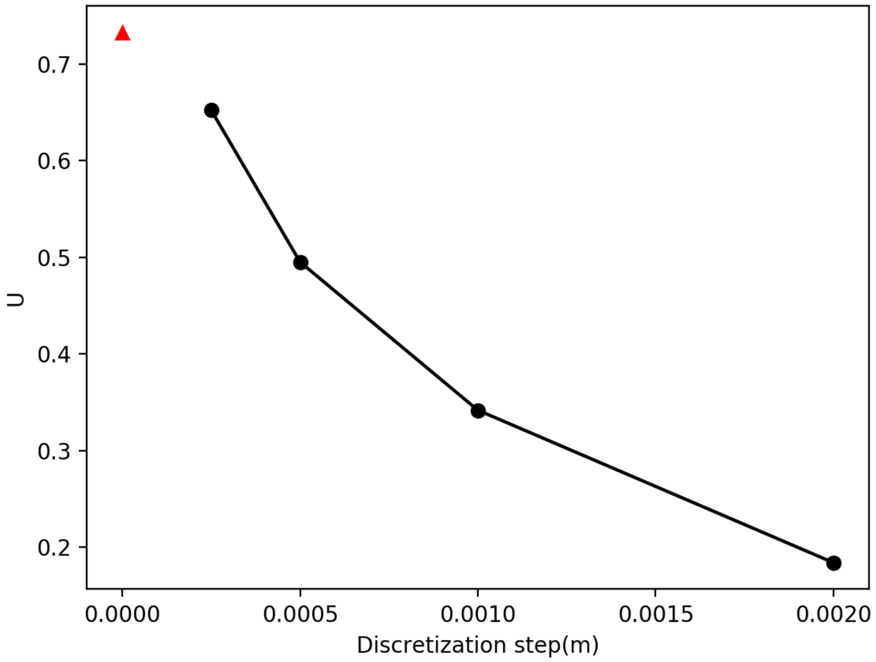


Figure 4.43: Richardson Extrapolation: The triangle indicates the value at zero spacing, while the dots are the integrated quantities for different cell sizes.

With increasingly finer grid, the time step has to be reduced and the temporal and spatial discretization errors should asymptotically get close to zero. Grid convergence index (GCI) (Roache, 1994, 1998) is a measure that tries to quantify if the solutions are within the asymptotic range of convergence. The GCI value indicates how far a solution is from the asymptotic numerical value and is computed with the use of 2 grids as:

$$\mathbf{GCI_{12}} = \frac{F_s|d_{12}|}{r^p - 1} \tag{4.18}$$

where F_s is a factor of safety chosen as 1.25 when comparing three grids or more (Roache, 1998) and d is the relative difference of the integrated quantities U of 2 grids. If the ratio between two GCIs is close to the value of 1, the solution is within the asymptotic range and assumed to be grid independent for the three cell sizes. Even in case a solution is close to an asymptotic value, it still might have errors when compared to the true solution.

Using four different cell sizes, three GCIs were obtained. By comparing the GCIs, a ratio of 1.09 is obtained for the three finer grids indicating that a solution with a cell size of Δl $\leq$ 1 mm is mesh independent. Consequently, a spatial discretization of Δx = Δy = Δz = 1 mm is chosen for all the simulations.

4.11 COMPARISON OF ANTENNAS

Usually many transducer units that represent the same GPR system are constructed. However, these units might produce slightly different responses when tested in the same scenarios and using the same system settings. In general, this is the case for all transducers. One reason is that it is extremely difficult to acquire or manufacture the required materials, that are needed to construct a specific transducer, with the exact same dielectric properties. Especially when it comes to EM absorber foams, the manufacturers provide a range of values that the true properties of a certain batch lie in, since the exact properties are difficult to determine. Units constructed using absorber foams from the same batch are likely to produce almost identical responses, while units that include absorbers from different batches, might have small differences. Thus, for each transducer unit, slightly different material properties might have been used, which affect the antenna radiation and lead to variations in the responses. In addition to that, due to degradation in the antenna components, which happens in all systems, the antenna performance is degrading over time.

To demonstrate this in practice, data from the same scenarios and with the same system settings were collected in the same NDT laboratory using two different units of the GSSI 2000 MHz “palm” antenna. The two transducer units are visible together in Figure 4.44, while in Figure 4.45 the normalised free space and the metal plate responses of both units are compared, where No-1 represents the real transducer that was used to build the model described above and No-2 represents the second unit. It is obvious that the traces from the two units are not identical, resulting from differences in the dielectric properties of their elements.

To obtain an understanding of which components differ the most, a model of the second unit is also optimised, expecting that differences in the optimisation outputs will provide some insight. The modelled geometry remains the same as for the first unit described above. The same antenna properties are optimised following the process of the GA scheme. For both units, the same initial random state is used to initialise the optimisation, which would have resulted in the same parameter values if the two units were identical.

Figure 4.46 illustrates the real responses compared with the responses resulted from optimisation for the second transducer unit. Similarly with the previous unit, the responses are again in a very good agreement, with the model predicting accurately both the amplitudes and the timings. Table 4.3 shows the optimised values of the parameters for the the second transducer unit along with the ones for the first. The optimisation of both units resulted to similar values for most parameters, with only small differences observed relative to the ranges of values each variable can take. Given this, most of the parameters will affect the radiation

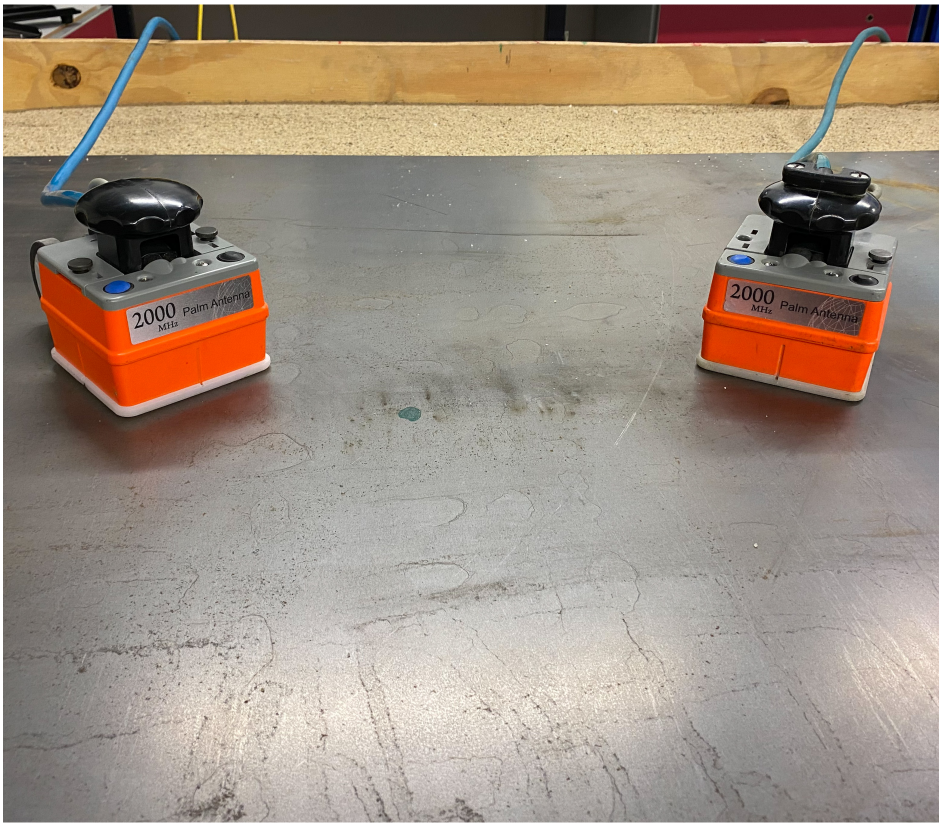
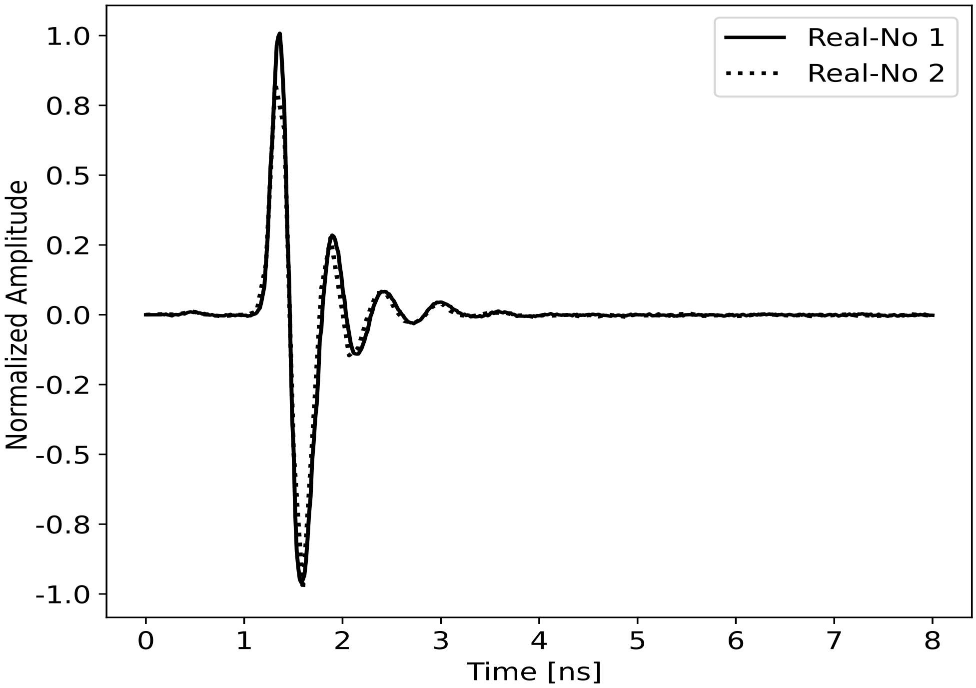


Figure 4.44: The two units of the GSSI 2000 MHz “palm” transducer.

of both units in almost the same way. However, the optimisation resulted in significant differences in the case of the absorbers, which had the largest contrasts. The permittivity estimates of the two absorber foams for the first unit are close to 1, while for the second unit are ∼3. The conductivity estimates have greater contrasts (based on the ranges of values the properties obtain) with the first absorber obtaining $\sigma = 1.08$ (S/m) and $\sigma = 0.45$ (S/m) for the first and the second unit, respectively, while the conductivity of the second absorber is estimated with $\sigma = 1.26$ (S/m) and $\sigma = 1.42$ (S/m) for the two units, respectively. Note that neither of the optimisation values for each model correspond to the true dielectric properties of the absorbers or the other parameters, as mentioned earlier. The optimisation resulting in larger contrasts for the absorbers compared to the rest of the properties, simply supports the premise that the absorbers are the key components that cause most of the differences in the responses from each unit due to the variation in their material properties.



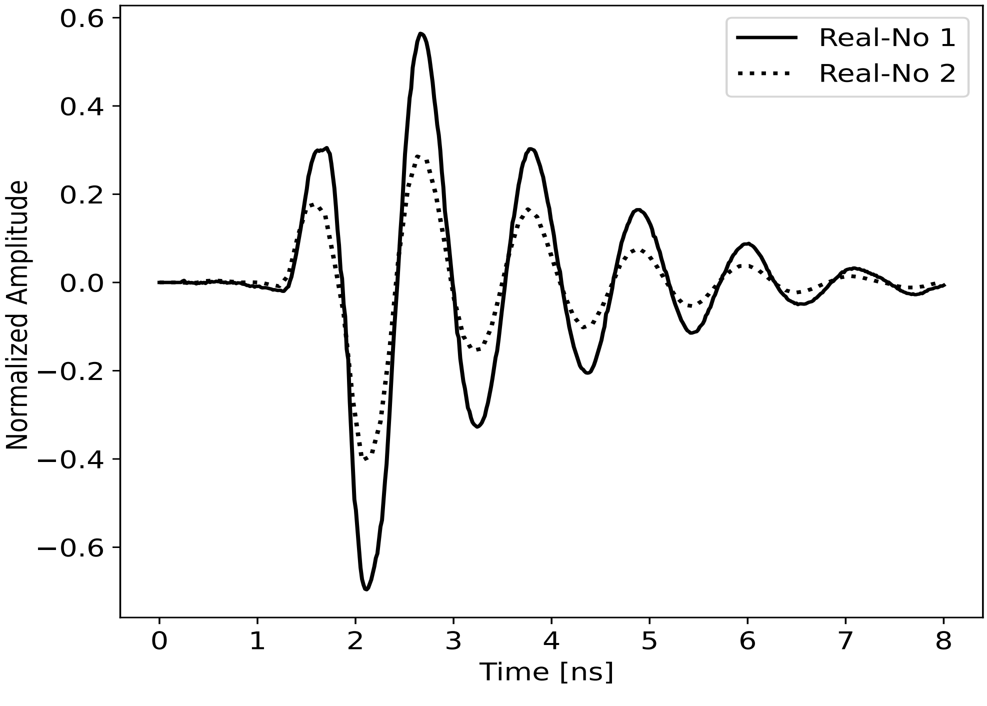


Figure 4.45: Comparison between the real responses of the two transducer units for the free space (top) and the metal plate (bottom). No-1 represents the real transducer unit that was used to build the model above, while No-2 represents the second unit.

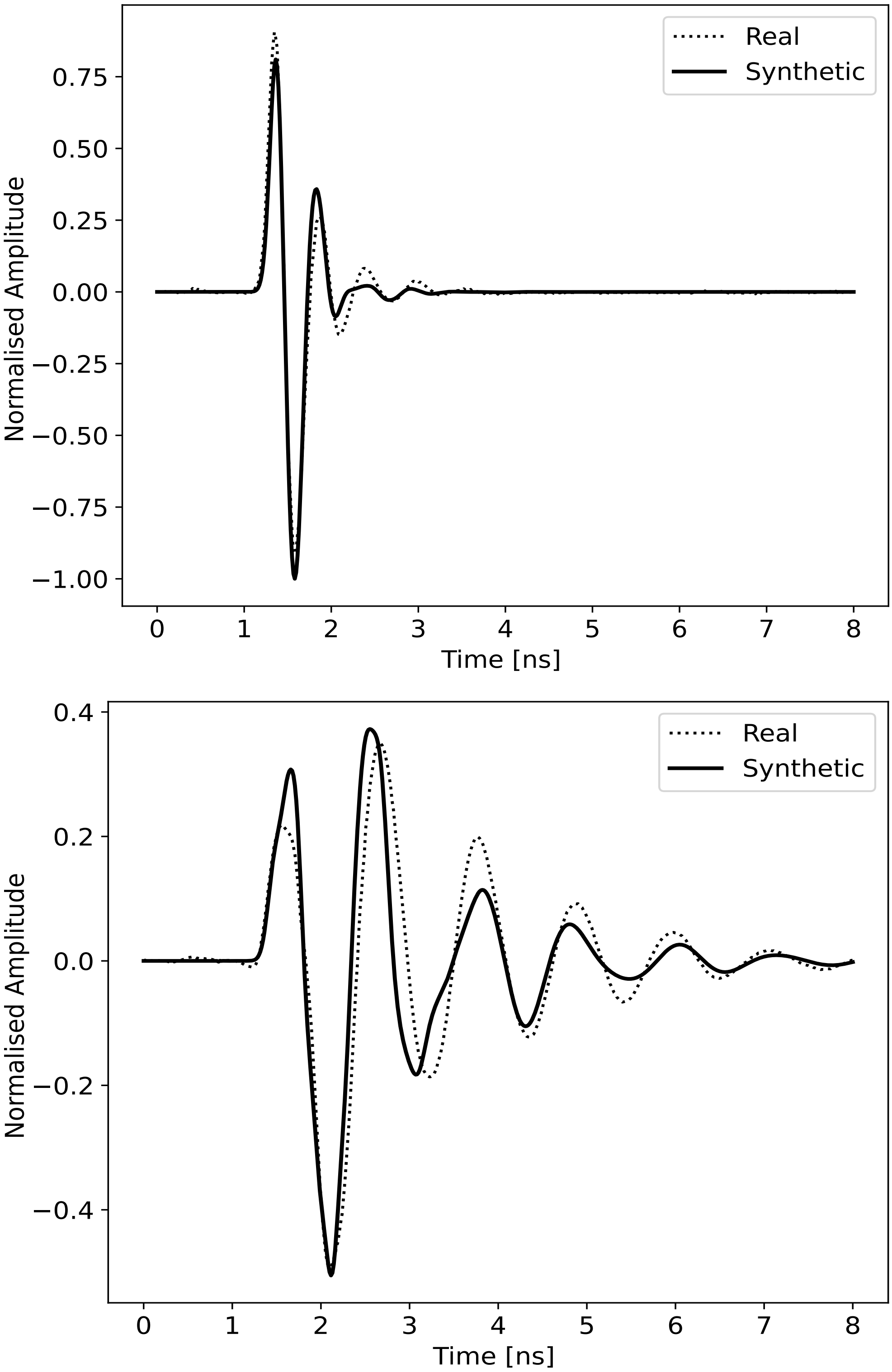


Figure 4.46: Comparison of the real responses and the responses obtained from optimisation for the second unit of the GSSI 2 GHz "palm" antenna for a) Free-space, b) PEC.

Optimised Parameters	**Unit No-1**	**Unit No-2**
Absorber 1 ϵ_{a1}	1.10	2.96
Absorber 2 ϵ_{a2}	1.07	2.77
PCB ϵ_{pcb}	1.52	1.40
Skid ϵ_s	2.67	2.17
Plastic Case ϵ_p	6.1	5.88
Shield ϵ_{sh}	1	1
Absorber 1 σ_{a1} (S/m)	1.08	0.45
Absorber 2 σ_{a2} (S/m)	1.26	1.42
PCB σ_{pcb} (S/m)	0.023	0.020
Skid σ_s (S/m)	0.005	0.007
Plastic Case σ_p (S/m)	0.0029	0.004
Shield σ_{sh} (S/m)	4.5×10^7	5.2×10^8
$R_s(\Omega)$	560	535
$\sigma_R(\Omega^{-1})$	0.005	0.012
f (GHz)	2.12	2.09

Table 4.3: Optimised antenna parameters.

5

MACHINE LEARNING AND NEURAL NETWORKS

In this chapter, the theory behind machine learning and specifically artificial neural networks is described. The way a deep learning model is trained, the activation functions and the backpropagation algorithm are analytically presented. In addition, hyperparameters, their importance and hyperparameters tuning are discussed.

5.1 MACHINE LEARNING INTRODUCTION

Today, artificial intelligence (AI) is a rapidly progressing field trying to create intelligent machines that are able to perform certain tasks which normally require human intelligence. AI has a vast number of applications in everyday life and is an active area of research. Machine Learning (ML) is a subset of AI, which enables machines to learn from data without being programmed explicitly and make accurate predictions. While the term was coined by Arthur Samuel in 1959, a formal definition of what ML is, is given by (Mitchell Tom, 1999) as

> A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.

Essentially, a ML algorithm learns to perform a certain task by observing examples, which is the experience. Given the examples, called a training set or training data, a mathematical relationship is build in order to make future predictions. Evaluation of the algorithm and the accuracy of its predictions is carried out by a quantitative performance measure, which is usually a measure of error or a measure of accuracy.

Each example is a set of measurable properties or characteristics, called features, that represent an object or event. An example is typically presented in the form of an n-dimensional feature vector $\mathbf{x} \in \mathbb{R}^n$, where each element x_i of the vector is an individual feature and n is the number of features. For instance, in computer vision the features of an image can be the intensity of the pixels in the image. Choosing features is a very important and challenging process as these have to include sufficient information for the inherent characteristics of the object or event they represent. The features influence directly the training of the ML model and can significantly improve its performance if chosen properly. To train an ML model, an algorithm that improves the weights is required in a way that reduces the error or maximises the accuracy.

A wide variety of tasks can be performed by an ML algorithm. Some of the most common tasks are the following:

1. Classification: The program has to specify to which class amongst k classes an input belongs to. Examples of a classification task are to identify whether an email is spam or not spam or the object recognition in images.
2. Regression: In this task, which is the most common along with classification, the ML model has to predict a numerical value given an input. An example of a regression task is to predict future prices. To solve this, the ML algorithm has built a relationship based on learning from previous historical data.
3. Machine translation: Given a sequence of symbols in a certain language, the program has to convert it to another language. This is commonly applied to natural language processing (NLP), such as translating from English to Spanish.
4. Anomaly detection: In this task, the algorithm learns to detect and flag unusual events, such as in credit card fraud where ML flags unusual transactions.
5. Denoising: Given a corrupted signal, the program learns to filter out the noise and produce a clean signal, such as image or audio denoising.

Depending on the task, different performance measures must be used. For classification tasks, the accuracy of the model can be measured, which is the proportion of the data that are classified correctly, whereas in a regression task, the performance measure is an error measure such as mse, where a small error value is indicative of a good performance. Minimising the error or maximising the accuracy on the training set is the goal of the learning process but usually, we are interested in how well the ML model performs on data that it has not seen before and its ability to generalise. Therefore, its performance is evaluated on a test set, which includes unseen data that were not part of the training. A good model results in both a small test and training error. A large training and test error means that the algorithm is not able to model the training data nor to generalise to new cases. This is the case of underfitting. The second case of poor performance is overfitting, where the algorithm results in a very small training error but the test error is large. In this situation, the model has learned the training data too well, memorising them in a way, and also the noise that is contained in the data making it hard to generalise.

ML algorithms can be broadly classified in two categories, the supervised and the unsupervised learning depending on the training data they are provided. In supervised learning, the algorithm is provided with both input **x** and output **y** data and tries to map a relationship between the two. In this case, each input example is associated with a label or target. For instance, in case of spam filtering, an email is the input example and its corresponding label is spam or not spam. In unsupervised learning, the algorithm is provided with input data only and tries to identify hidden patterns from these data and their inherent structure. These classes of algorithms are used for clustering to group the data based on their similarities and differences, for data dimensionality reduction or density estimation to determine how the data is distributed in space. In addition to these categories, there is also the semi-supervised learning, which is a combination between the previous two and the reinforcement learning, where the model is trained by rewarding or punishing certain behaviors.

Some of the traditional ML algorithms are linear regression, logistic regression, random forest, k-nearest neighbors (KNN), K-means, naive Bayes, while deep learning is the most popular class of ML methods, which is widely used today. In recent years, ML has been applied to a number of GPR applications. In the work presented by Gamba and Lossani (2000), a feed-forward neural network was used to automatically detect hyperbolic signatures from pipes in B-scan images. Training the ML model was performed using real GPR data, which were pre-processed before learning. The results of the algorithm were compared with manually analysed by a human B-scans, showing good accuracy. Similarly, Singh and Nene (2013) employed neural networks to detect hyperbolas in B-scans, which are subsequently used to estimate the location and depth of a buried object using a curve fitting algorithm. Generative adversarial networks (GANs) along with convolutional neural networks (CNNs) were proposed by Zhang et al. (2021) for the same purpose. The difference with the previous approaches is that the model is training using synthetic data with GANs used to generate more training data from the existing ones. A network based on CNNs is then used to train a model for automatic hyperbola detection.

A supervised deep learning classification scheme was applied by Giannakis et al. (2015) for landmine detection using GPR. A diverse 2D FDTD training set was used that included different soils, landmines with varying depth, false alarm targets among others. The suggested scheme was a two-step process, where first a classification between landmine and false target is performed and then the landmine type is recognised. CNNs were also implemented by Lameri et al. (2017) for landmine detection from GPR images. The training was performed using synthetic B-scans using minimal pre-processing. The suggested algorithm was tested on real GPR data, demonstrating a very good accuracy. In the work presented by Kafedziski et al. (2018) faster region CNN (Faster R-CNN) is used for detecting and classifying landmines. The algorithm is trained to classify two types of objects, anti-tank mines and objects with hyperbolic signatures using both synthetic and real data. Elsaadouny et al. (2020) employed transfer learning for target classification, where the learned features from the popular LeNet-5 network were transferred to the GPR training set used. It was shown that transfer learning improved the performance and resulted in a higher accuracy.

5.2 DEEP LEARNING

Deep Learning is a class of ML methods, which is based on artificial neural networks (ANNs), or more simply neural networks (NNs). ANNs are inspired by the biological neural networks and the way the information is processed by the human mind to identify patterns. They are trying to learn from experience and mimic the problem-solving capabilities of the human brain. ANNs are used when the relationship between the input and the output is a complex non-linear relationship, which cannot be modelled using traditional ML techniques and due to their ability to handle large datasets. Advancements in computer infrastructure, both in software and hardware, have made it possible to develop enormous networks that are trained using a vast number of data and to solve increasingly complicated applications over time.

An ANN consists of hundreds or even millions of interconnected nodes or neurons which are arranged in layers. Each network has an input layer, one or more hidden layers and an output layer. Figure 5.1 demonstrates a NN of 2 hidden layers with 4 nodes each, an input layer with 3 nodes and an output with 1. Nodes in the input layer represent the features of the training examples, while the output layer represents the predicted values. Each node has an associated weight. Finding the optimal weights $\mathbf{W}$ is the goal of the deep learning algorithm. Training the ML model is an iterative process that starts with randomly initialised weights and updates these weights at each iteration based on a criterion, the objective function that was chosen as a performance measure.

Consider the NN architecture shown in Figure 5.1 used to solve a regression problem. The training set consists of m training examples, where the ith training example is represented by a $\mathbf{x}^{(i)} \in \mathbb{R}^n$ feature vector with n features and its corresponding label $y^{(i)}$. The total training set can be represented as pairs by

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m-1)}, y^{(m-1)}), (x^{(m)}, y^{(m)}) \tag{5.1}$$

or in matrix notation as $\mathbf{X} \in \mathbb{R}^{n \times m}$ and $\mathbf{Y} \in \mathbb{R}^{1 \times m}$. Each input feature vector is multiplied by a set of weights $\mathbf{w}$, the terms are summed and added to a bias term as

$$z_j^{[1]} = \mathbf{w}_j^{[0]T} \mathbf{x}^{(i)} + b \tag{5.2}$$

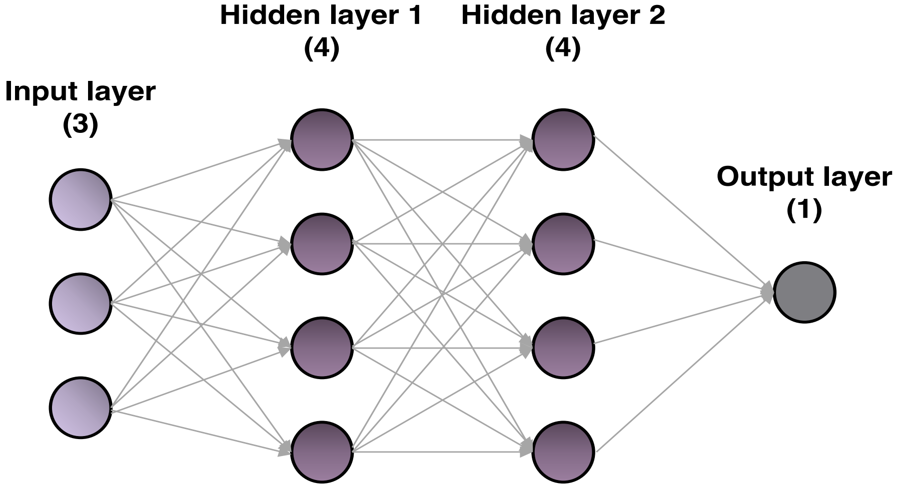


Figure 5.1: Example of a neural network with 3 input nodes, two hidden layers with 4 nodes each and an output layer with 1 node.

where the superscript of z is the layer, which is 1 for the first layer and the subscript j is the node of this layer. The bias term serves like an intercept in a linear equation and helps the algorithm produce a better fit but its use is not always necessary and is usually not included in the input layer but only in the hidden layers. Adding bias terms results in the network shown in Figure 5.2. Then, the weighted sum $z_j^{[1]}$ is passed through an activation function as

$$a_j^{[1]} = \sigma(z_j^{[1]}) \tag{5.3}$$

The activation function is the component that introduces the non-linearity and allows for complex relationships to be built. Activation functions will be analytically described at a latter chapter. The value of $a_j^{[1]}$ is the output of the jth node of the first layer. These equations will be calculated for every node in the first layer

$$z_1^{[1]} = \mathbf{w}_1^{[0]T}\mathbf{x}^{(i)} + b_1^{[0]} \quad , \quad a_1^{[1]} = \sigma(z_1^{[1]}) \tag{5.4}$$

$$z_2^{[1]} = \mathbf{w}_2^{[0]T}\mathbf{x}^{(i)} + b_2^{[0]} \quad , \quad a_2^{[1]} = \sigma(z_2^{[1]}) \tag{5.5}$$

$$z_3^{[1]} = \mathbf{w}_3^{[0]T}\mathbf{x}^{(i)} + b_3^{[0]} \quad , \quad a_3^{[1]} = \sigma(z_3^{[1]}) \tag{5.6}$$

$$z_4^{[1]} = \mathbf{w}_4^{[0]T}\mathbf{x}^{(i)} + b_4^{[0]} \quad , \quad a_4^{[1]} = \sigma(z_4^{[1]}) \tag{5.7}$$

The resultant values of $\mathbf{a}^{[1]}$ will be the input to the next layer and used to calculate its weighted sums by

$$z_1^{[2]} = \mathbf{w}_1^{[1]T}\mathbf{a}^{[1]} + b_1^{[1]} \quad , \quad a_1^{[2]} = \sigma(z_1^{[2]}) \tag{5.8}$$

$$z_2^{[2]} = \mathbf{w}_2^{[1]T}\mathbf{a}^{[1]} + b_2^{[1]} \quad , \quad a_2^{[2]} = \sigma(z_2^{[2]}) \tag{5.9}$$

$$z_3^{[2]} = \mathbf{w}_3^{[1]T}\mathbf{a}^{[1]} + b_3^{[1]} \quad , \quad a_3^{[2]} = \sigma(z_3^{[2]}) \tag{5.10}$$

$$z_4^{[2]} = \mathbf{w}_4^{[1]T}\mathbf{a}^{[1]} + b_4^{[1]} \quad , \quad a_4^{[2]} = \sigma(z_4^{[2]}) \tag{5.11}$$

In its general form, for the j-th node in layer l, z_j^l is defined as

$$z_j^l = \sum_{k=1}^{K} w_{jk}^{l-1} a_k^{l-1} + b_j^l \tag{5.12}$$

where K is the number of nodes in $l-1$ layer or in matrix notation

$$\mathbf{z}^l = \mathbf{W}^{l-1}\mathbf{a}^{l-1} + \mathbf{b}^{l-1} \tag{5.13}$$

$$\mathbf{a}^l = \sigma(\mathbf{z}^l) \tag{5.14}$$

The final output $\hat{y}$ is calculated by

$$\hat{y}^{(i)} = a_1^{[3]} = g(\mathbf{w}_1^{[2]T}\mathbf{a}^{[2]}) \tag{5.15}$$

Note that the last activation function for the output layer is different from the previous ones and in general each layer can use a different activation function. Since it is a regression problem, the output layer uses a linear activation function. It is obvious that $a^{[0]} = \mathbf{x}$. The steps followed above, where an output from the NN is computed by propagating the input signal through the hidden layers until the output layer is called forward propagation or forward pass. This process is repeated for all the examples in the training data. After calculating the predicted values, they are compared with their corresponding labels, which are known since this is a supervised learning algorithm. There are many different loss functions that can be used, with the mse being a very common option for regression, which is given by

$$MSE_{train} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2 \tag{5.16}$$

The model then updates the weights through iteration and aims to find an optimal set of weights that minimises the mse. To minimise the error, an optimisation scheme along with an algorithm for computing the gradient of the loss, called backpropagation is used, which will be described at later sections. The same process is followed for larger networks with a different number or layers and

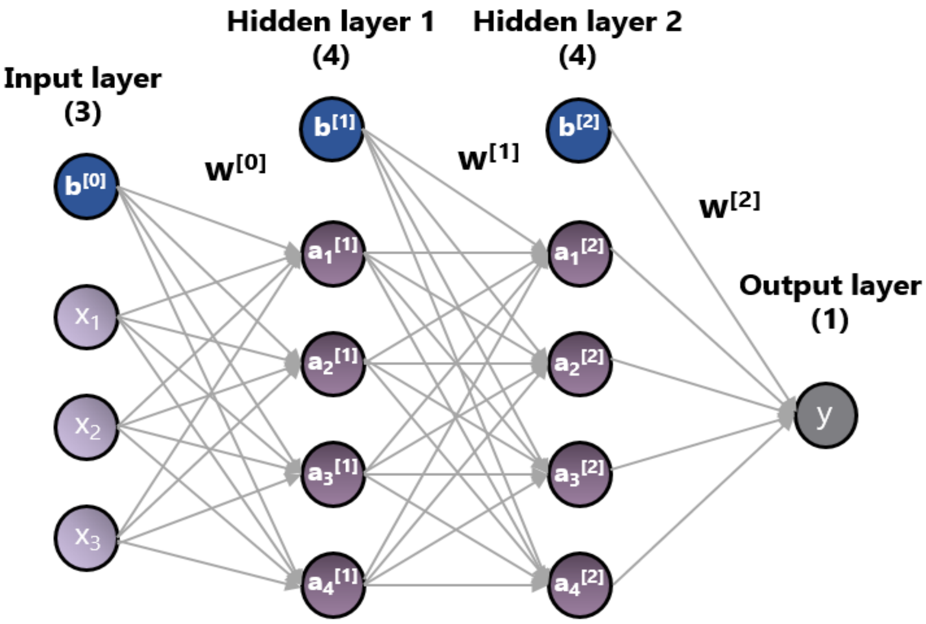


Figure 5.2: Forward propagation in a neural network with 3 input nodes, two hidden layers with 4 nodes each and an output layer with 1 node.

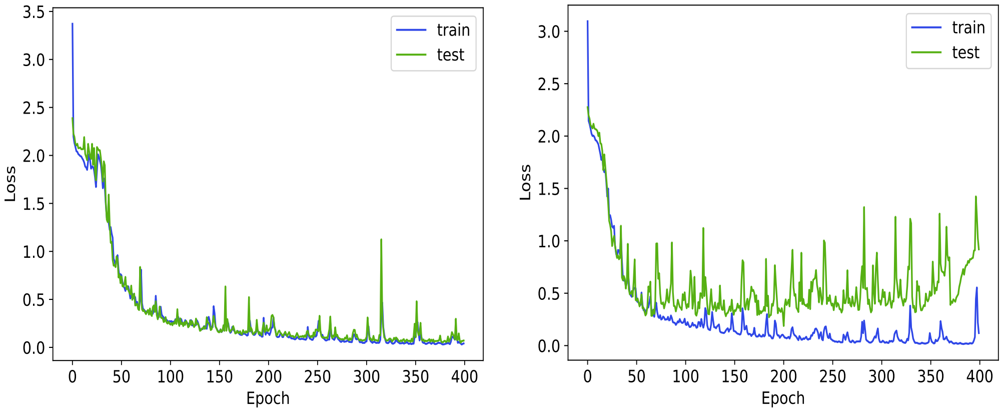
nodes. Finally, after training the model is evaluated on the test set and refined in case of poor performance.

The network described above is the most basic deep learning model, called a feed forward neural network. It is called feed forward because there are no connections feeding outputs of the model back to itself and information flows only from the input to the hidden layers and finally to the output. NNs with feedback connections are called recurrent neural networks (RNNs) and are mostly used for time series and natural language processing (NLP) applications. The above NN is also a fully-connected NN, meaning that all nodes from one layer are connected with all the nodes from the next layer, which is not always the case.

5.3 HYPERPARAMETERS

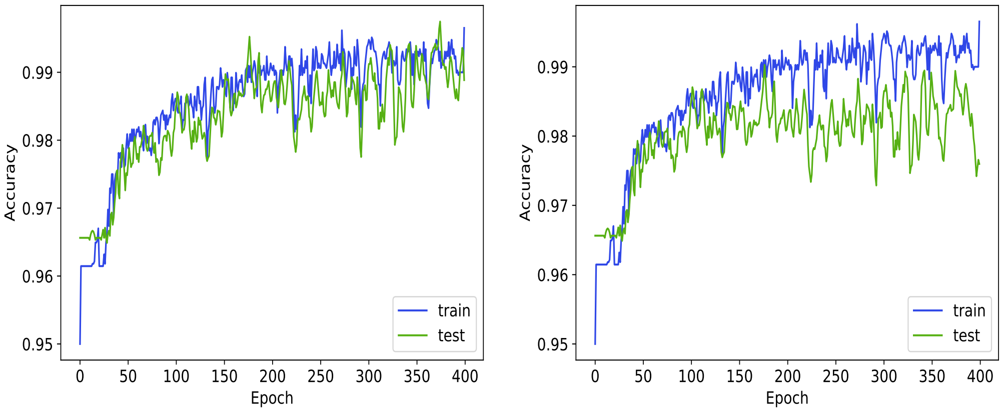
NNs require a set of settings, called hyperparameters, which must be determined before the training process begins. Hyperparameters are basically all the parameters that are not learned by the training process but control its behavior. These are parameters such as the number of hidden layers and nodes in the network, the learning rate, the choice of loss and activation functions, the optimization algorithm and others, which have to be predefined.

The choice of the hidden layers and units in a network is critical. A small number of units might not be able to capture the relationship between the input



(a) Loss: Good fit

(b) Loss: Overfitting



(c) Accuracy: Good fit

(d) Accuracy: Overfitting

Figure 5.3: Plots of loss (top) and accuracy (bottom) on the training and test sets over the training epochs showing the cases of a good fit (left) and overfitting (right).

and the output data and thus, lead to underfitting. In contrast, having a vast number of units can result in overfitting the training set and making the network unable to generalise to new cases. A sufficient number of units accompanied by a regularisation technique is usually the best choice for increasing the efficiency of the network.

The learning rate controls how quickly a network updates its parameters. It is the step size that the weights are updated during training at each iteration while moving towards the minimum of the loss function. A small learning rate will result in a slow training process but the model will converge smoothly and probably close to the minimum as shown in Figure 5.4a. In contrast, using a quite large learning rate, where big update steps are taken, it is highly likely for the algorithm to overshoot the minimum and may never converge. This is illustrated in Figure 5.4b where the value of the loss bounces back and forth in a convex function using gradient descent.

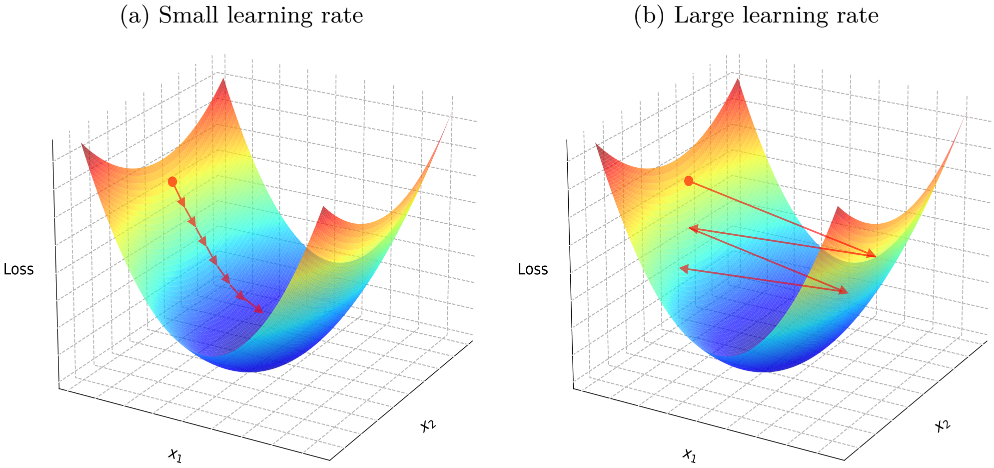


Figure 5.4: Convergence of the solution towards the minimum using a) a small learning rate and b) a large learning rate.

An epoch is defined as one full pass through the full training dataset, with one full pass accounting as a forward and a backward pass. Usually, the training data are partitioned in a number of batches. Iteration is the number of batches required to complete one epoch. If the training has $m = 3000$ training examples and it is divided into batches of 500, there will be a total of 6 iterations to complete 1 epoch.

Tuning or hyperparameter optimisation is the process of finding the optimal hyperparameters for a ML model. Every problem requires a different set of hyperparameters, which is determined via trial-and-error. Automated ways of testing different values have been developed with the most famous being grid search. Grid search trains multiple NN models for different combinations of hyperparameters and finally chooses a model based on the performance metric. Using grid search can be extremely computationally expensive, but once performed the optimal NN model is found.

5.3.1 *Activation functions*

An activation function defines how the weighted sum of input nodes from one layer is transformed into the output of a node from the next layer. Activation functions are a critical part of a neural network, which directly affect the training model and must be chosen carefully. Usually activation functions are non-linear and are used to map non-linear relationships between input and output data, although they can be linear as well. All layers can have different activation functions but hidden layers usually use the same activation function. The output layer is usually assigned a different activation function from the hidden layers, which is dependent upon the type of problem and prediction required, e.g. regression requires a linear activation function. Since neural networks are trained using the backpropagation algorithm, which requires calculation of derivatives, the activation function chosen must be differentiable. The most common activation functions are the following:

1. Sigmoid function: The sigmoid activation function, also called logistic function is the function used for the logistic regression classification algorithm

   $$\sigma(z) = \frac{1}{1+e^{-z}} \tag{5.17}$$

   The function takes a real value as input and outputs a value in the range [0, 1]. Its familiar S-shape is illustrated in Figure 5.5. In the past it was used for binary classification problems, but it has been replaced by other functions due to vanishing gradient problems and not being zero-centered. The vanishing gradient problem refers to the situation where the gradient decreases dramatically as it propagates backwards through the network and slows down the learning process.

2. Softmax function: For multi-class classification problems where a multinomial probability distribution is required, the softmax function can be used. The softmax function can be thought of as a generalised form of the sigmoid. This function takes as input a vector $\mathbf{z}$ of real values with K elements, normalises it into a probability distribution and outputs a vector of the same length that consists of K probabilities

   $$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum\limits_{j=1}^{K} e^{z_j}} \tag{5.18}$$

   where i=1,2,...,K and $\mathbf{z} \in \mathbb{R}^K$.

3. Tanh function: The hyperbolic tangent activation function, or simply tanh, is similar to the sigmoid function, but it is zero-centered. Thus, it takes a real value as input and outputs a value in the range of [-1, 1]. Tanh is calculated as

   $$tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{5.19}$$

   The S-shape of the tanh activation function is shown in Figure 5.6.

4. ReLU function: The rectified linear activation function, or ReLU is the most common function that is used in practice as the activation of the hidden layers. ReLU is very simple and easy to implement. It is more effective than

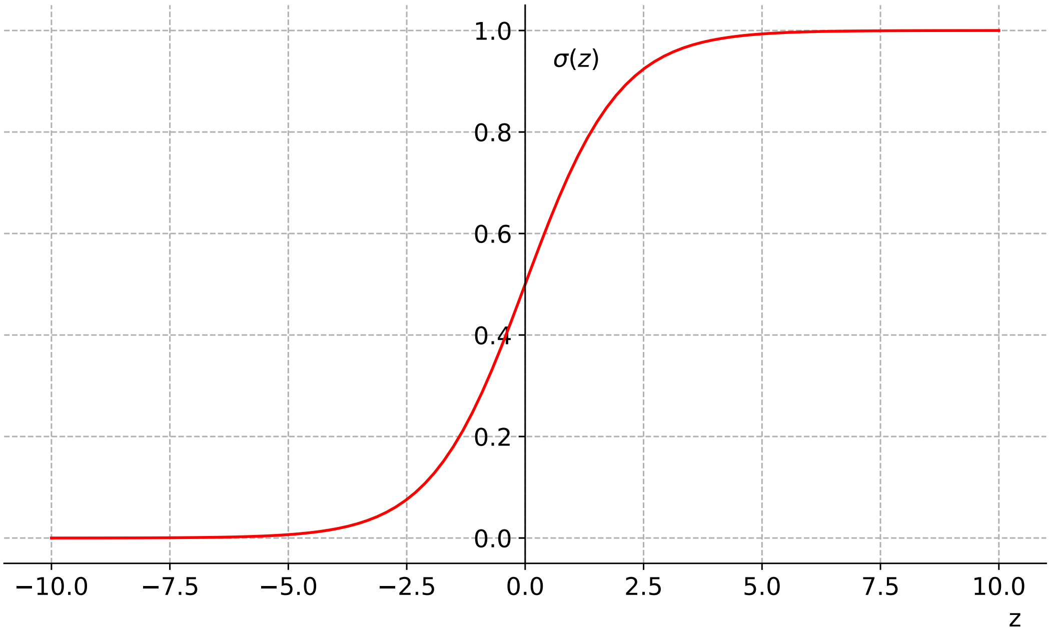


Figure 5.5: The sigmoid activation function.

the other functions due to its ability of overcoming the vanishing gradient problem and is defined as

$$ReLU(z) = z^+ = max(0, x) \tag{5.20}$$

Given a negative input value, ReLU will output 0, otherwise the value itself is returned, as shown in Figure 5.7. There are many modified versions of ReLU, such as Leaky ReLU, Parametric ReLU and others, used to address certain problems.

5. Linear function: The linear activation function, also called identity or no activation, is the case where no activation function is used. This means that it does not change the weighted sum of the inputs in any way and simply returns the value itself. A linear activation is commonly used for the output layer in regression problems.

5.3.2 *Backpropagation*

A network requires an efficient optimisation technique that repeatedly updates the weights and biases so as to minimise the loss function, that is the differences between predicted and actual outputs. The optimisation techniques usually require the computation of the gradient of the loss in order to update the parameters. The mainstream technique used is the backpropagation algorithm or simply backprop,

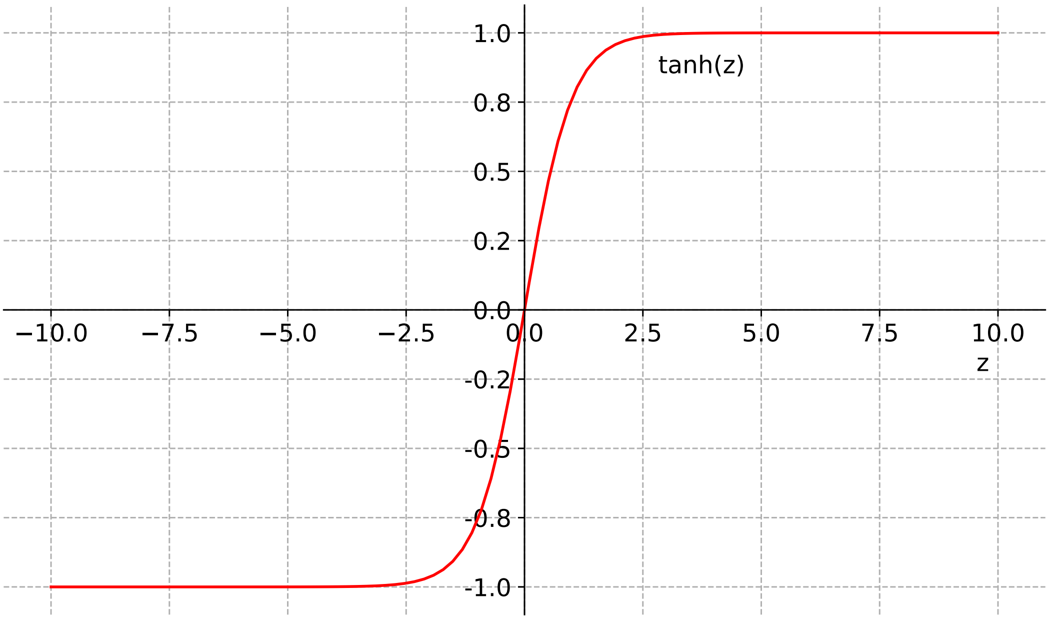


Figure 5.6: The hyperbolic tangent (tanh) activation function.

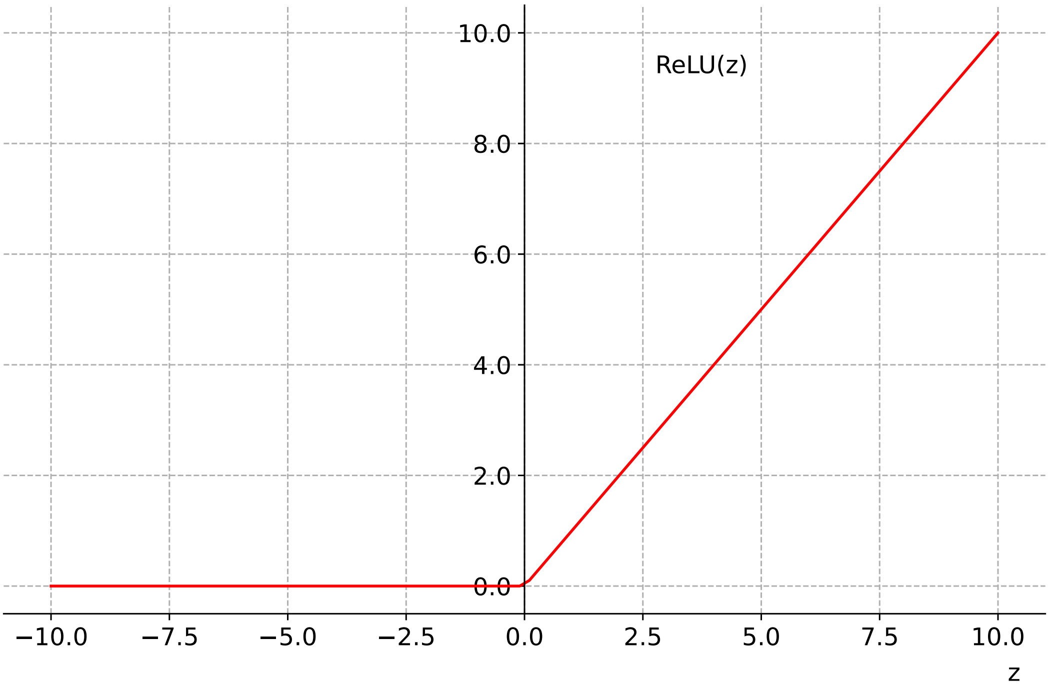


Figure 5.7: The rectified linear activation function (ReLU) activation function.

which is a method of computing gradients in directed graphs of computations and propagate the error backwards through the network. After calculating the cost function for the outputs, the algorithm travels back from the output layer to the hidden layers to adjust their weights such that the error is reduced.

Backpropagation calculates the gradient of the loss function with respect to the weights of the network in order to determine the level of adjustment the parameters need. To compute the gradients, backpropagation uses a method called the chain rule. If C_k is the cost or loss function, the gradient of a single weight w^l_{jk} using chain rule is given as

$$\frac{\partial C_k}{\partial w^l_{jk}} = \frac{\partial C_k}{\partial z^l_j}\frac{\partial z^l_j}{\partial w^l_{jk}} \tag{5.21}$$

where $w_j k^l$ is the weight between the k-th node in layer $l-1$ and the j-th node in layer l. Using again the chain rule, the gradient of the cost function with respect to the weight is given by

$$\frac{\partial C_k}{\partial w^l_{jk}} = \frac{\partial C_k}{\partial a^l_k}\frac{\partial a^l_k}{\partial z^l_j}\frac{\partial z^l_j}{\partial w^l_{jk}} \tag{5.22}$$

Assuming a cost function of $C = (a^l - y)^2$

$$\frac{\partial C_k}{\partial a^l_k} = 2(a^l - y) \tag{5.23}$$

$$\frac{\partial a^l_k}{\partial z^l_j} = \sigma'(z^l) \tag{5.24}$$

$$\frac{\partial z^l_j}{\partial w^l_{jk}} = a^{l-1}_k \tag{5.25}$$

Similar equations are obtained for the bias

$$\frac{\partial C_k}{\partial b^l_j} = \frac{\partial C_k}{\partial z^l_j}\frac{\partial z^l_j}{\partial b^l_j} = \frac{\partial C_k}{\partial a^l_k}\frac{\partial a^l_k}{\partial z^l_j}\frac{\partial z^l_j}{\partial b^l_j} \tag{5.26}$$

$$\frac{\partial C_k}{\partial a^l_k} = 2(a^l - y) \tag{5.27}$$

$$\frac{\partial a^l_k}{\partial z^l_j} = \sigma'(z^l) \tag{5.28}$$

$$\frac{\partial z^l_j}{\partial b^l_j} = 1 \tag{5.29}$$

The common fraction in both the weight and bias equations is called the local gradient and is defined as

$$\delta_j^l = \frac{\partial C_k}{\partial z_j^l} \tag{5.30}$$

This quantity represents the error of the j-th node in the l-th layer. Using again the chain rule δ_j^l is given by

$$\delta_j^l = \frac{\partial C_k}{\partial a_k^l}\frac{\partial a_k^l}{\partial z_j^l} \tag{5.31}$$

Since, a_k^l depends only on ∂z_j^l, only one term remains

$$\delta_j^l = \frac{\partial C_k}{\partial a_k^l}\frac{\partial a_k^l}{\partial z_j^l} \tag{5.32}$$

Likewise, calculating the gradient with respect to a weight and the bias of a node in the previous layer would be

$$\frac{\partial C_k}{\partial w_{jk}^{l-1}} = \frac{\partial C_k}{\partial a_k^l}\frac{\partial a_k^l}{\partial z_j^l}\frac{\partial z_j^l}{\partial a_k^{l-1}}\frac{\partial a_k^{l-1}}{\partial z_j^{l-1}}\frac{\partial z_j^{l-1}}{\partial w_{jk}^{l-1}} \tag{5.33}$$

$$\frac{\partial C_k}{\partial b_j^{l-1}} = \frac{\partial C_k}{\partial a_k^l}\frac{\partial a_k^l}{\partial z_j^l}\frac{\partial z_j^l}{\partial a_k^{l-1}}\frac{\partial a_k^{l-1}}{\partial z_j^{l-1}}\frac{\partial z_j^{l-1}}{\partial b_j^{l-1}} \tag{5.34}$$

Since the total cost function involves averaging the costs functions over many training examples, its derivative requires also averaging across these training examples

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{1}{m}\sum_{k=0}^{n-1}\frac{\partial C_k}{\partial w_{jk}^l} \tag{5.35}$$

In addition, this process is repeated for all the nodes in order to calculate the gradients with respect to all the weights and biases in the network. Finally, having calculated the gradients, the weights and biases can be updated using gradient descent with the following equations

$$\mathbf{w} := \mathbf{w} - \alpha\frac{\partial \mathbf{C}}{\partial \mathbf{w}} \tag{5.36}$$

$$\mathbf{b} := \mathbf{b} - \alpha\frac{\partial \mathbf{C}}{\partial \mathbf{b}} \tag{5.37}$$

where α is the learning rate. The process of a forward pass and backward pass is repeated until convergence is achieved. For backpropagation to work, a certain form of a cost function is required, that is that the cost function can be written as the average of the cost functions for individual training examples and also that it is a function of the outputs of the network.

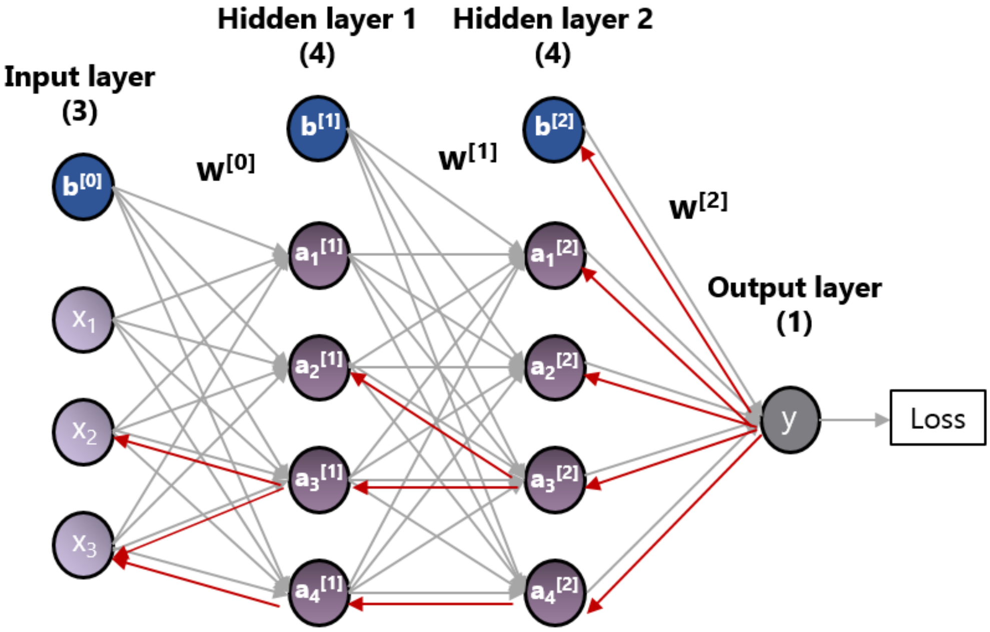


Figure 5.8: Backpropagation in a neural network with 3 input nodes, two hidden layers with 4 nodes each and an output layer with 1 node.

5.4 OPTIMISATION

To train a network, an optimisation scheme is required to update the parameters by reducing the losses with iterative gradient-based optimisation algorithms being the most common choices. The parameters are usually randomly initialised and adjusted through an iterative optimisation algorithm. The most common optimisers in ML are gradient descent, stochastic gradient descent (SGD), mini-batch gradient descent, SGD with momentum, AdaGrad, RMSprop and Adam optimiser.

The equations for gradient descent, which is a well-known and easy to implement optimisation algorithm were presented in the previous section for backpropagation. For gradient descent to work the cost function has to be differentiable and convex. Traditional gradient descent performs a pass on all of the training examples before making the first update to the parameters, resulting to large execution times and large memory requirements.

SGD is a variant of gradient descent, in which the parameters of the model are updated after computing the loss on a single training example. For the weights, this is given as

$$\mathbf{w} := \mathbf{w} - \alpha \nabla \mathbf{C}(w; x^{(i)}, y^{(i)}) \tag{5.38}$$

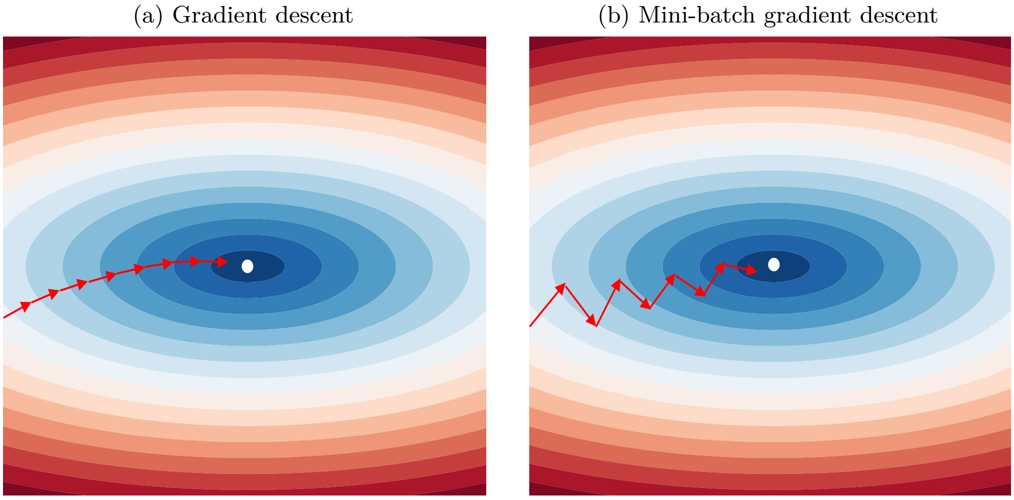


Figure 5.9: Convergence towards the minimum (white dot): a) Gradient descent, b) Mini-batch gradient descent.

Therefore, for a dataset with 500 training examples, SGD will update the model parameters 500 times in one pass of the total dataset. Optimising using SGD leads to fluctuations and greater variance in the updated model parameters and may overshoot the global minimum. However, this results in less memory requirements and converge is achieved much faster than traditional gradient descent.

Mini-batch gradient descent uses a batch of a number of training examples, called a mini-batch, the size of which is less than the total training set. So, the entire training set is divided in batches and the parameters are updated after every pass on a batch. This method lies in between gradient descent and SGD and takes a step on adjusting the parameters on a mini-batch at a time

$$\mathbf{w} := \mathbf{w} - \alpha \nabla \mathbf{C}(w; B(i)) \tag{5.39}$$

where $B(i)$ is the i-th mini-batch of training examples. This technique results in less variance and medium needs of memory. Mini-batches allow for the use of vectorised implementations, resulting in faster computations. For small datasets, it is recommended to use the traditional gradient descent, while for large datasets the mini-batch algorithm by splitting to batches with size based on the size of the entire training set and the computer resources.

A drawback of the mini-batch algorithm is that the updates of the weights are still noisy. SGD with momentum handles this problem by denoising the gradients. The method of momentum is used to accelerate training, especially in case of noisy gradients. It is based on adding history of the gradients from previous updates to the current update equations by using an exponentially decaying moving average of the previous gradients. This method requires an additional hyperparameter $\gamma \in [0, 1)$, known as momentum, that controls the contribution of the previous gradients to the update equations and is usually set to 0.9. The greater γ is, the more the previous gradients control the direction of the current update. Adjusting the weights using SGD with momentum is given by

$$\mathbf{v}_t = \gamma \mathbf{v}_{t-1} - \alpha \nabla \mathbf{C}(\mathbf{w}) \tag{5.40}$$

$$\mathbf{w} := \mathbf{w} + \mathbf{v}_t \tag{5.41}$$

where the variable $\mathbf{v}$ plays the role of the velocity, meaning the direction and speed that the parameters move through the optimisation space. The velocity is computed as an exponentially decaying average of the negative gradient. Time t here means the current update of the weights.

While the previous algorithms have fixed learning rates, algorithms with adaptive learning rates have been designed. The magnitude of each gradient can vastly vary for different weights, making it difficult to find a single global learning rate and thus, adapting the learning rate of each weight separately is needed. The AdaGrad algorithm by Duchi et al. (2011) modifies the learning rate at each time step of the model parameters by scaling them inversely proportional to the square root of the sum of squares of their past gradients. For a given weight, this is given by

$$w_{t,i} := w_{t-1,i} - \frac{\alpha}{\sqrt{G_{t-1,ii} + \epsilon}} \nabla C(w_{t-1,i}) \tag{5.42}$$

$$G_{t-1,ii} = \sum_{\tau=1}^{t} g_\tau g_\tau^T \tag{5.43}$$

where α is the initial learning rate, $\mathbf{G}_t$ is the sum of squares of the past gradients g_τ until time t and ϵ is a small quantity (usually 1e-8) used to avoid division by zero. Parameters that have large gradient of the loss have a rapid decrease in their learning rate, whereas parameters with small gradients have a small drop in their learning rate. A drawback of the method is that the accumulation of the squared gradients decreases the learning rate and slows down the training process.

The RMSprop algorithm, an unpublished optimisation algorithm by Geoffrey Hinton, is a modification of AdaGrad where the sum of squared gradients is replaced by an exponentially weighted moving average $E[g^2]$ of past gradients

$$w_{t,i} := w_{t-1,i} - \frac{\alpha}{\sqrt{E[g^2]_{t-1} + \epsilon}} \nabla C(w_{t-1,i}) \tag{5.44}$$

$$E[g^2]_{t-1,i} = \beta E[g^2]_{t-2,i} + (1-\beta)(\nabla C(w_{t-1,i}))^2 \tag{5.45}$$

Hinton suggests a value of 0.9 as the optimal for β. While AdaGrad decreases the learning rate based on the total history of the squared gradients, the RMSprop algorithm removes the history of the very first gradient states using the exponentially decaying average and thus converges faster.

Adaptive moment estimation, known as Adam optimiser, is a combination of RMSprop and SGD with momentum, originally presented by Kingma and Ba (2014). It is a widely used and robust optimisation scheme that computes individual adaptive learning rates for each parameter of the network, which are calculated using estimates of the first (the mean) moment and second (the uncentered variance) moments of the gradients. To estimate the moments, Adam uses exponentially moving averages of a mini-batch gradient. For a given weight, the update equation is

$$m_{t,i} = \beta_1 m_{t-1} + (1 - \beta_1) g_{t,i} \tag{5.46}$$

$$v_{t,i} = \beta_2 v_{t-1} + (1 - \beta_2) g_{t,i}^2 \tag{5.47}$$

$$\hat{m_{t,i}} = \frac{m_{t,i}}{1 - \beta_1^t} \tag{5.48}$$

$$\hat{v_{t,i}} = \frac{v_{t,i}}{1 - \beta_2^t} \tag{5.49}$$

$$w_{t,i} := w_{t-1,i} - \alpha \frac{\hat{m_{t,i}}}{\sqrt{\hat{v_{t,i}}} + \epsilon} \tag{5.50}$$

where $m_{t,i}$ and $v_{t,i}$ are the first and second moment, respectively. The hyperparameters $\beta_1, \beta_2 \in [0, 1)$ control the exponential decay of the moving averages with recommended values of 0.9 and 0.999, respectively. A suggested default value of the initial learning rate is given as 0.001.

To this date, there is not a single best algorithm but the class of algorithms with adaptive learning rates have shown to behave more efficiently and robustly.

6

NEURAL NETWORKS FOR BACKGROUND REMOVAL AND PERMITTIVITY ESTIMATION

In this chapter, a novel deep learning-based method for background removal is proposed, where the background response and its dielectric properties are predicted. At first, a review of background removal methods for GPR data is provided. Afterwards, the data preprocessing steps and the development of the NN model are provided. The scheme is validated with both modelled and real data and is used as part of a reverse time migration scheme.

6.1 BACKGROUND REMOVAL

As mentioned earlier, background removal is one of the main processing steps and apart from simple mean subtraction, a large number of different signal processing techniques that perform background removal can be found in the literature. A review of different signal processing techniques used for background clutter reduction is provided by Jung-Ho Kim et al. (2007). The basic principles of each method are given and their performances are compared using field GPR data.

Hamran et al. (1995) propose a simple time-gating approach, that reduces the early part of the received signal to zero. However, this approach has the drawback that it might also remove target responses, especially when shallow buried objects are present and careful selection of the time window removed must be made. A filtering process in the frequency-wavenumber (f-k) domain over a window of the GPR data has also been implemented for direct air wave removal (Young and Sun, 1999). The process was tested on data from a mid-continent fluvial environment of sand and discontinuous clay layers. Another background removal technique, which uses frequency domain basis functions to represent the clutter and target signal, is presented by van der Merwe and Gupta (2000). The unknown parameters in the basis functions are approximated using an iterative algorithm. The scheme was evaluated using both modelled and measured GPR data.

Nuzzo and Quarta (2004) used linear radon transform and discrete wavelet transform for coherent noise attenuation. Comparison was made between the results of the transforms with the classical f-k and running average subtraction methods. It was shown that Radon-based methods are superior compared to the rest of the methods but computationally are more expensive. In the work presented by Jung-Ho Kim et al. (2005) filtering in the wavenumber domain and predictive deconvolution was employed after testing several approaches to remove the ringing. The filtering was used as part of processing of GPR data acquired from an archaeological exploration survey. Solimene et al. (2014) used an entropy-based time-gating approach for background noise removal. This method assumes that the clutter has similar characteristics across the profile. The time-gating window is chosen based on an entropy criterion. The performance of the method is compared with the mean subtraction and the subspace projection methods. A technique based on low-rank and sparse priors was presented by Tivive et al. (2019) for mitigating the background clutter. A low-rank representation of the background clutter is extracted from the radar responses and a sparse component for the target signal. The proposed scheme takes into account also the noise present in the data.

Principal component analysis (PCA), independent component analysis (ICA) and singular value decomposition (SVD) have also been used for background and other clutter reduction. These techniques decompose a signal into a number of components, where the components which represent the useful signal are separated by the ones that correspond to the clutter. The idea behind using these techniques for background removal is that the first component with the maximum variability represents the background information. By discarding this component, the background response in removed.

SVD was used by Cagnoli and Ulrych (2001) to eliminate the flat horizontal reflections from a GPR line. The high-pass eigenimages of the B-scan data were obtained, which include only the reflections of interest, while the highly correlated parts that represent the flat responses have been removed. Liu et al. (2017) employed SVD for denoising and direct wave removal. A quantitative criterion for choosing the singular values for denoising a GPR profile was given, however it requires knowledge of the signal to noise ratio (SNR). The results were compared with the wavelet threshold denoising method, bandpass filtering and mean removal for both modelled and field data.

In the work presented by (Tebchrany et al., 2014), both PCA and ICA have been implemented to remove the direct air and direct ground wave. The performance of the techniques was evaluated on both synthetic and real GPR images of a single target buried in a two-layered medium. It was concluded that PCA was more efficient than ICA in the presence of shallow targets, while the opposite was true for targets buried deeper. Song et al. (2017) suggests a robust PCA (RPCA) scheme for clutter suppression to landmine detection problems in order to assist in landmine recognition in heavy clutter environments. The RPCA decomposition was used prior to migration of numerical and real GPR data, showing that it significantly reduced the clutter.

Verma et al. (2009) applied an ICA-based technique and compared the results with SVD, PCA and factor analysis methods. These techniques were applied on experimental data and was shown that ICA was more efficient and resulted to a higher SNR value compared to the rest of the methods. Morphological component analysis (MCA), which decomposes an image into a number of morphological components has also been utilised for clutter suppression by separating the morphological components that correspond to the targets (Temlioglu and Erer, 2016). MCA assumes that a GPR image is a linear mixture of several morphological components, where each of them can be sparsely represented by a specific dictionary and which are morphologically distinct, e.g. sines and spikes. However, a drawback of the MCA method is that it is dependent on the selection of the dictionaries.

A first attempt on clutter suppression using deep learning for GPR data is proposed by Ni et al. (2020) and is based on autoencoders (AE), which are a form of unsupervised ANNs, which are most commonly used for data compression. An autoencoder has two main parts: the encoder, which compress the input into a lower-dimensional representation (latent-space representation) and the decoder, which reconstructs the output from this compressed representation. The suggested scheme utilizes robust AE (RAE), a type of autoencoder that can be regarded as an extension of the RPCA technique, to separate the clutter component from the target component in a low-rank and sparse matrix representation problem.

The main drawback of the methods discussed above is that in addition to the background, information about other flat-lying responses, that is included in this first component, is lost. These targets might be of interest and removing them will significantly affect the interpretation of the data. In this chapter, an ML scheme is proposed, that predicts the background response from GPR data, performs background removal and subsequently estimates both the permittivity and conductivity of the background. The scheme removes only the background response without affecting other flat-lying targets. In deep learning, SVD and PCA are also used for data compression and will be discussed analytically in the following sections since they were used as part of the ML scheme. All the ML schemes presented throughout this thesis were developed using the machine learning platform TensorFlow (Abadi et al., 2015).

6.2 TRAINING SET

To train a NN, a vast amount of training data that captures diverse scenarios is required. Since, collecting real GPR data from diverse environments and scenarios is time-consuming, and the ground truth is usually unknown and not easy to obtain, synthetic data were chosen instead, generated using gprMax

(Warren et al., 2016). Two different sets of training data were generated for the two different ML problems, the background prediction and the prediction of properties. For the background prediction case, the dataset consists of 6500 A-scans, where each A-scan falls into one of the three general cases:

1. 1 layer + reinforcing bar (rebar)
2. 1 layer + half-space
3. 1 layer + rebar + half-space

with 2500, 2000 and 2000 A-scans produced for each case respectively. The size of the training set was decided based on the variability of the model parameters and the number of different scenarios that are sufficient to capture the pattern between the input and output data. Using too many data is not the best practice since this can result in overfitting the dataset and thus, a balance between capturing the relationship and avoiding overfitting should be sought. The total responses, which include both the background and the signal from the targets are the input data to the network, while the output data include only the background response. In the case of the output data, the background was modelled as a homogeneous half-space in the simulations, spanning the entire domain with no targets included. Therefore, for $m = 6500$ traces with n discrete points, the training data $\mathbf{X} \in \mathbb{R}^{n \times m}$ and their corresponding labels $\mathbf{Y} \in \mathbb{R}^{n \times m}$, consist of traces $\mathbf{X}_i \in \mathbb{R}^n$ and their corresponding background clutter $\mathbf{Y}_i \in \mathbb{R}^n$, i.e. $\mathbf{X}_i = \mathbf{Y}_i + \mathbf{S}_i \ \forall i \in [0 \ m]$, where $\mathbf{S}_i$ is the resulting scattering field from the targets for the ith trace. For estimating the dielectric properties (ϵ_r, σ_r) of the background, traces with a background response only (without any targets) are used as input data $\mathbf{Y} \in \mathbb{R}^{n \times m}$ with their corresponding labels $\mathbf{P} \in \mathbb{R}^{2 \times m}$, where $\mathbf{P}_i = [\epsilon_i, \sigma i]^T$ is a column vector that contains the permittivity and conductivity of the the ith trace.

The dielectric properties of the background, the half-space depth and dielectric properties and the depth of the center and diameter of rebars are all randomly generated. These values can vary randomly between certain specified ranges. More specifically, for both the background and half-space, the permittivity can take values $\epsilon_r = 2 - 22$, the conductivity, $\sigma = 0.00001 - 0.5$ S/m, while the depth and diameter of the rebars are varying between 4 - 20 cm and 2 - 7 cm, respectively. For the last two cases that include the background layer and a half-space, the thickness of background layer varies between 2 - 17 cm, whereas in the first case the background is modelled as a half-space. All materials are considered non-magnetic and have been defined with a relative permeability $\mu_r = 1$ and a zero magnetic loss $\sigma_\mu = 0$, while the rebars are modelled as perfect electric conductors (PECs). In all cases, the rebars are placed directly below the midpoint of the antenna. Centering the rebar was chosen for simplicity, but it is later proven that it was sufficient for the algorithm to capture the pattern and perform well even in cases where rebars are not placed directly below the GPR transducer. For all the simulations, a domain size of $300 \times 300 \times 400$ mm with a step size of 1 mm for all three directions was used, while the time step was set to $\Delta t = 1.92$ ps as calculated by the Courant-Friedrichs-Lewy (CFL) condition (Taflove and Hagness, 2000). The simulations were performed for a time window of 8 ns, resulting in a total of 4156 time steps. Generating this dataset required a running time of 1.5 days on 4 NVIDIA TITAN RTX 24GB GPUs, utilizing the GPU execution capabilities of gprMax (Warren et al., 2019).

6.3 PRE-PROCESSING

Before training a machine learning algorithm, the training set requires pre-processing in order to transform the data to a more suitable form for ML. This usually involves data quality assessment, normalization of the data or feature scaling and dimensionality reduction. In this case, four processing steps were implemented that are described below.

6.3.1 *Normalization of the data*

Since it is desired for the ML schemes to be applicable to real data as well, the modelled data should be comparable with the real ones. For this reason, the responses in the training set were normalised to the maximum absolute amplitude value of the simulated free space response, which was 8.08. When making predictions using real data as input, these should be normalised to the maximum absolute amplitude value of the free space response acquired with the real GPR system and using the same system settings.

6.3.2 *Resampling*

As an initial step to reduce the dimensionality of the data, each A-scan response is downsampled using the Fourier method from 4156 to 512 sample points, which is the number of samples that the real data were collected with. There is no information lost during this process, since for the specific antenna system used of 2GHz center frequency, it is expected for the data to contain useful information in frequencies significantly less than 32 GHz, which is half the sampling frequency corresponding to 512 samples in a time window of 8 ns, as calculated from the Nyquist theorem.

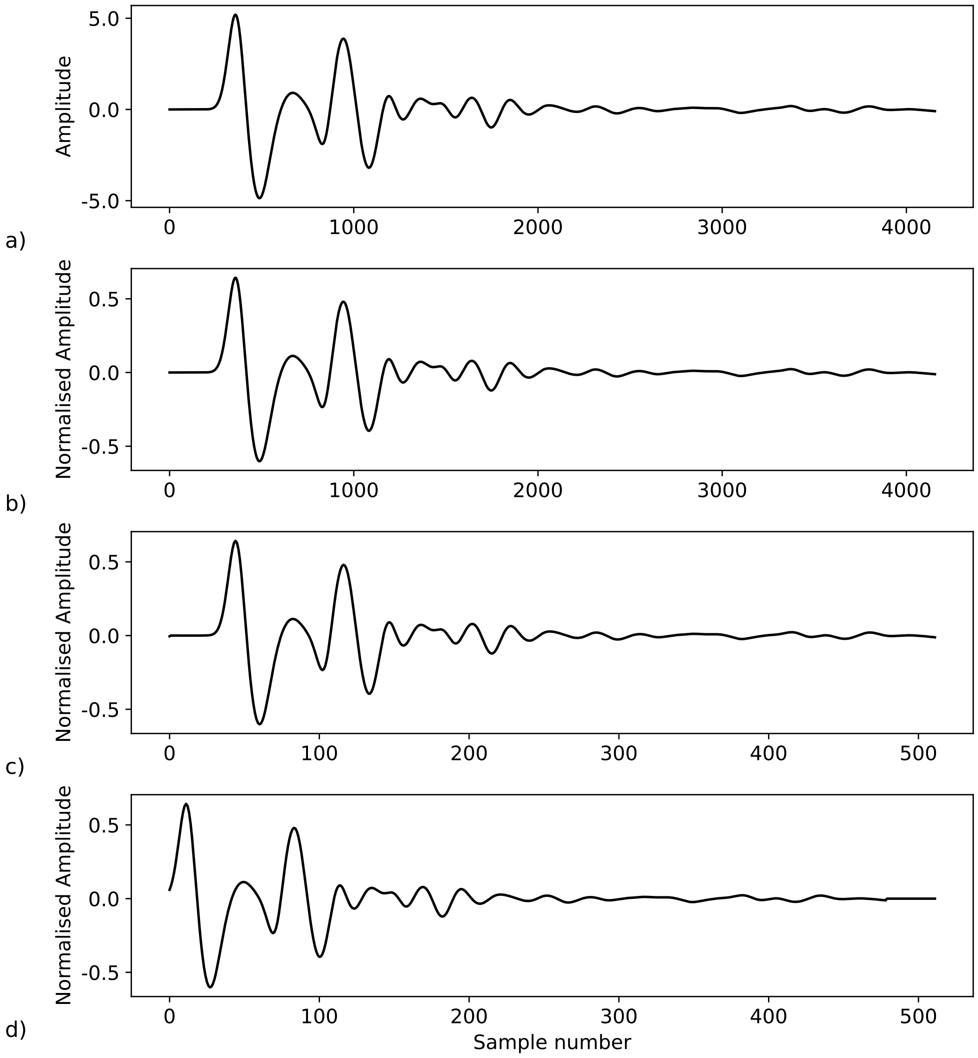


Figure 6.1: Pre-processing of a single A-scan: a) Raw A-scan, b) Normalised A-scan, c) Response after resampling and d) Shifted response.

6.3.3 *Shifting the data*

In order for the deep learning algorithms to be applicable to real GPR data, synthetic and real data must have the same starting point (first sample), meaning that all data should be shifted so that they have the same origin. Whilst the time zero value can be clearly determined in the simulations, the true time zero of the real GPR system cannot be determined due to the inherent characteristics of the transducer electronics and thus, the time zero position cannot be chosen as the starting point. By shifting the A-scans to the maximum positive or negative amplitude value, could result in losing important background information, especially in cases where targets are located at a shallow depth. Due to the above reasons, a small positive amplitude value of 0.05 was chosen as an optimal origin, meaning that both input and output normalised A-scans are shifted so that their first sample point corresponds to the first point with 0.05 amplitude. This process is demonstrated in Figure 6.1, where in a) the raw trace is presented, whereas in b)-d) the trace after the processes of normalization, resampling and shifting is presented. Note that this shifting is not regarded as an optimal time zero correction but as a way of making the modelled data comparable with the real data, along with normalisation.

6.3.4 *Principal Component Analysis for Dimensionality Reduction*

Dimensionality reduction is the transformation of a data set from a high-dimensional space to a lower-dimensional space that is performed in order to decrease the number of variables in a problem. In addition, both the complexity of the problem and the required computational cost will be reduced. A large number of input or output variables will make it harder for a machine learning algorithm to establish an accurate relationship between the two, while leading to larger running times. Therefore, in order to increase efficiency, the dimensions of the data should be decreased prior to training.

The lower-dimensional representation should contain the meaningful information of the original data set, while discarding any unnecessary information. Ideally, the low-dimensional space should be as close as possible to the intrinsic dimension of the data, which is the minimum number of variables required to obtain an accurate approximation of the original signal. Depending on the problem, dimensionality reduction can be applied to both input and output data.

Consider one A-scan from the training set. Its original length is 4156 samples, while after downsampling its length reduces to 512 sample points. This reduced length is still substantial, considering the fact that there is a total of 6500 A-scans for the input and another 6500 A-scans for the output of the network. Therefore, both the input and output data should be transformed to a lower-dimensional space.

Most of the dimensionality reduction techniques fall into one of two categories. The first is the feature elimination, where the feature space is reduced by discarding variables that are considered to not have significant contribution in predicted the output. However, if the wrong variables are discarded then important information will be lost. The second category is feature extraction. Methods belonging to this class create a new set of independent variables, where each of these new variables is a combination of the original variables in the problem. After creating the new set of variables, the ones with the least significance are discarded. Because all new variables are combinations of the original ones, even

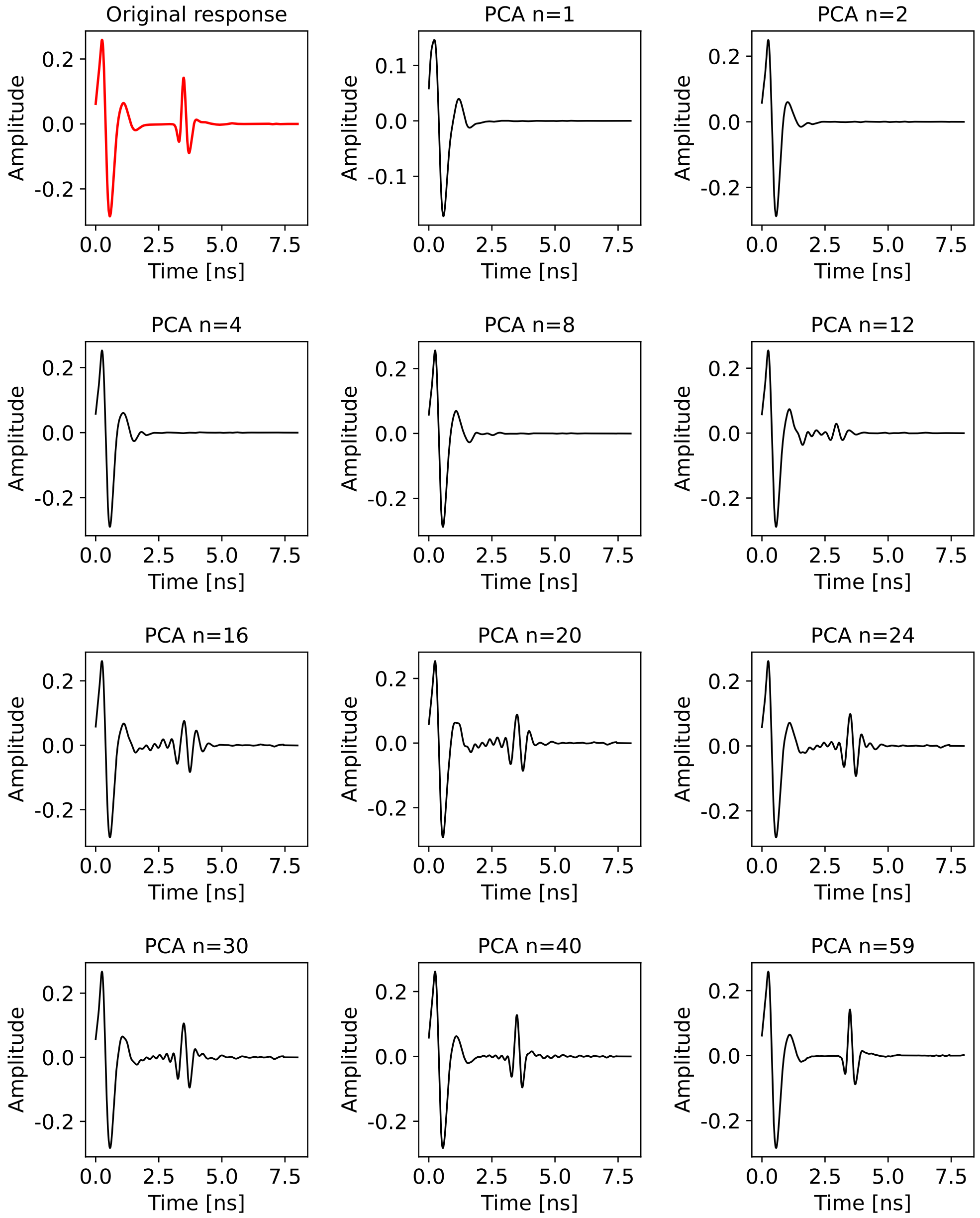


Figure 6.2: PCA representation of an A-scan for different number of components.

when dropping the least important parameters, the most valuable information of the old variables is retained.

One of the oldest and main methods for performing dimensionality reduction is the principal component analysis (PCA), which is a feature extraction technique. PCA implements a linear mapping of the original data to a lower-dimensional space. The idea behind this technique is to reduce the dimensionality, while preserving as best as possible the variability in the data.

In order to perform PCA, the computation of the covariance matrix of the data is required. The covariance matrix is used to deduct if there is any relationship between the variables in the feature space. In many cases, some of the variables

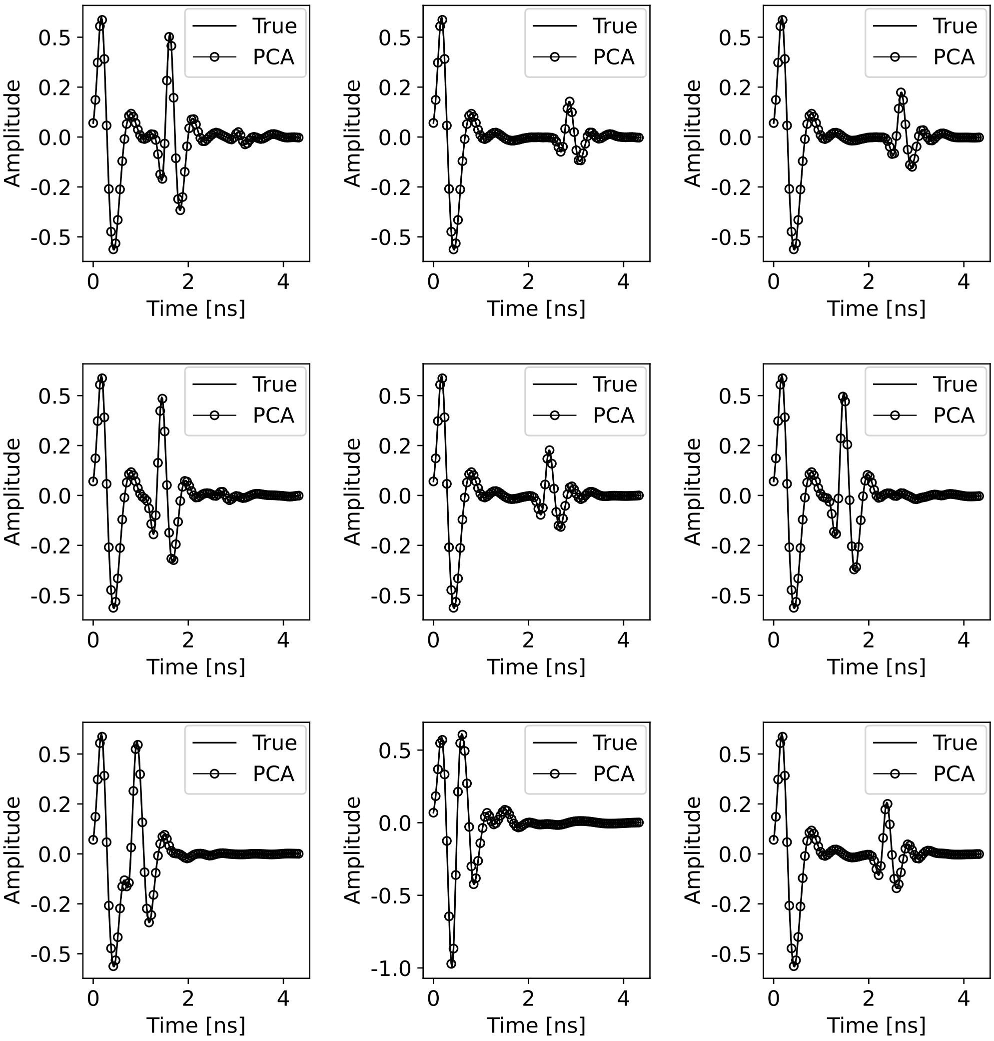


Figure 6.3: Reconstructed PCA A-scans versus true A-scans randomly selected from the training set.

are highly correlated, meaning that they contain redundant information. These correlations can be identified using the covariance matrix. This matrix is an $n \times n$ square symmetric matrix (where n is the dimension of the data) that contains as entries the covariances between each possible pair of the initial variables. The covariance between two variables x and y is defined as:

$$\sigma(x, y) = \frac{1}{m-1} \sum_{i=1}^{m} (x_i - \bar{x})(y_i - \bar{y}) \tag{6.1}$$

where m is the number of samples and $\bar{x}$, $\bar{y}$ are the mean values of x and y, respectively. Given this equation, the covariance matrix for three variables can be represented as:

$$\begin{bmatrix} \sigma(x_1,x_1) & \sigma(x_1,x_2) & \sigma(x_1,x_3) \\ \sigma(x_2,x_1) & \sigma(x_2,x_2) & \sigma(x_2,x_3) \\ \sigma(x_3,x_1) & \sigma(x_3,x_2) & \sigma(x_3,x_3) \end{bmatrix} \tag{6.2}$$

A positive covariance means that two variables increase together (positively correlated), while a negative means that one variable increases when the other decreases (negatively correlated). Since the covariance of a variable with itself is just the variance of that variable, the elements on the main diagonal are actually the variance of each variable.

The next step is to compute the eigenvectors and eigenvalues of the covariance matrix. In linear algebra, when performing a linear transformation of a vector space V, which is performed by multiplying the vector space with a transformation matrix A, there are certain vectors that will not change their direction. They can be stretched or shrinked by a scalar value but their direction remains the same. These vectors are called eigenvectors and the amount of scaling is called an eigevalue. For these vectors, multiplying them with the matrix A is equivalent to multiplying them with just a scalar value. This can be written as:

$$Ax = \lambda x \tag{6.3}$$

where x are the eigevectors and λ is a vector containing the eigenvalues that correspond to each eigenvector.

The eigenvectors of the covariance matrix are actually the directions with the highest variance containing most of the information and these are called principal components. The corresponding eigenvalues are coefficients that represent the amount of variance contained in each principal component. These principal components are the new uncorrelated variables that are created by linearly combining the initial variables. If there are n number of variables, then n number or principal components are calculated.

The combinations of the original variables in the principal components are performed in such a way that most of the information is contained in the first few components. PCA arranges the components in a way that the first component has the maximum possible information (the line that maximizes variance of the data), the second component the maximum remaining information, with the condition that is uncorrelated with the first, and so on by ranking the eigenvectors in the descending order (highest to lowest) of the eigenvalues. This means that the first few components explain most of the variability in the data and the rest which have minor contribution can be neglected without losing much information. Therefore by discarding the non-important components, the data are projected from a higher-dimensional space to a lower-dimensional space. It is worth noting that although the principal components will increase the efficiency and convergence of an ML scheme, they have no real physical meaning and are not easily interpretable since they are linear combinations of the original variables.

Consequently, to further decrease the dimensionality of both the input and output dataset, PCA is implemented. Thus, one PCA model is constructed to transform the input A-scans and a second PCA model to transform the output A-scans. A linear combination of all the principal components via weighting factors can reconstruct exactly the original dataset. In this case, each A-scan in the dataset can be represented as:

$$a_i = Cw_i \tag{6.4}$$

where a_i is the ith A-scan, C is the matrix that contains the principal components and w_i is the vector with the weights for the ith A-scan. Since each A-scan consists of 512 sample points after downsampling, the total number of principal components is 512. Each of the principal components explain a certain amount of variance in the data and are sorted by descending variance. The number of principal components kept were 59 and 6, for the input and output data, respectively, which explain a total of 0.999 variance in the data, while the rest of the components were discarded. The weights are also 59 and 6, for each case, respectively, and are distinct for each A-scan in the dataset. The PCA compressed forms of the data are the final variables that are used as the input/output nodes of the neural network. The total data set can be then written in a matrix form as:

$$\begin{aligned} \mathbf{T} &= [t_1, t_2, \ldots, t_n] \in \Re^{6500\times 59} \\ \mathbf{B} &= [b_1, b_2, \ldots, b_n] \in \Re^{6500\times 6} \end{aligned} \tag{6.5}$$

where t_i, b_i are the vectors with the input and output PCA compressed versions, respectively, for the ith training example. To apply PCA on the data, the Python module scikit-learn was used (Pedregosa et al., 2011). Figure 6.2 illustrates an A-scan from the training set, shown in the first subfigure, and its reconstructed signal using different number of PCA components. It is obvious that using the first few components, only the background response is reconstructed. With increasingly more components used, the approximated signal resembles more the original signal with the response of the target starting to appear. Using $n = 59$, the reconstructed signal obtained is identical to the original response. To verify that 59 components are sufficient, a number of A-scans from the training set are presented in Figure 6.3 along with their reconstructed PCA approximations, where it is obvious that these are almost indistinguishable. For illustrative purposes, the A-scans have been truncated to 4.5 ns in the Figure, although the full 8 ns have been actually used for PCA. Note, that although the total data set is transformed to a compressed PCA form, only the training set was used to calculate the PCA model and find the principal components.

The rest of the data, that form the validation and test sets, were not part of fitting the PCA model but were transformed afterwards using the PCA model the resulted from the training set.

6.4 DEEP LEARNING SCHEME

After generating and pre-processing the A-scans, the data set is split into training, test and validation data. First, the data are shuffled and then randomly split in the three sets. In particular, 80% of the data is allocated to the training set, while the test and validation sets are allocated 10% each.

The proposed deep learning scheme uses two connected networks, meaning that the output of the first network is passed as an input to the second. The first neural network predicts the compressed background response $\mathbf{B}_i$ given the compressed total response $\mathbf{T}_i$ as an input. Subsequently, the predicted compressed background is feeded as an input to the second neural network in order to predict the background relative permittivity and conductivity $\mathbf{P}_i$. The predicted compressed background $\mathbf{B}_i$ in then transformed back to time domain to acquire the background response as an A-scan Y_i, which is subsequently subtracted from the raw time-domain data $\mathbf{X}_i$ to perform the background removal. Performing background removal using the suggested NN scheme predicts and removes only the signal that corresponds to the direct wave and does not affect flat or other targets, in contrast to the methods discussed earlier.

Having a permittivity estimate allows for the calculation of the EM propagation velocity in the background using Equation 2.2 and therefore to transform the time to depth scale and approximate the depth to targets. Usually, a velocity estimate is obtained using the hyperbola fitting technique, where a hyperbola equation is fitted to the measured hyperbola. This technique relies on the existence of a hyperbolic signature in the data and certain simplifications. Al-Nuaimy et al. (2000) modelled the hyperbolic signatures using the equation

$$\left(\frac{t}{t_0}\right)^2 - \left(\frac{2(x-x_0)}{ut_0}\right)^2 = 1 \tag{6.6}$$

which relates the two-way travel time t with position x in the horizontal direction and the velocity u. The (x_0, t_0) coordinates represent the apex of the hyperbola. This model assumes a target of 0 m radius and tries to find the optimal velocity by solving the hyperbola equation and acquiring a hyperbola that resembles as best as possible the hyperbola corresponding to the target. Usually, a visual fitting is performed, were hyperbolas for different velocities are calculated and plotted overlaying the measured hyperbolas until the shapes of the two hyperbolas match. The assumption of zero radius is clearly incorrect as all cylindrical objects have

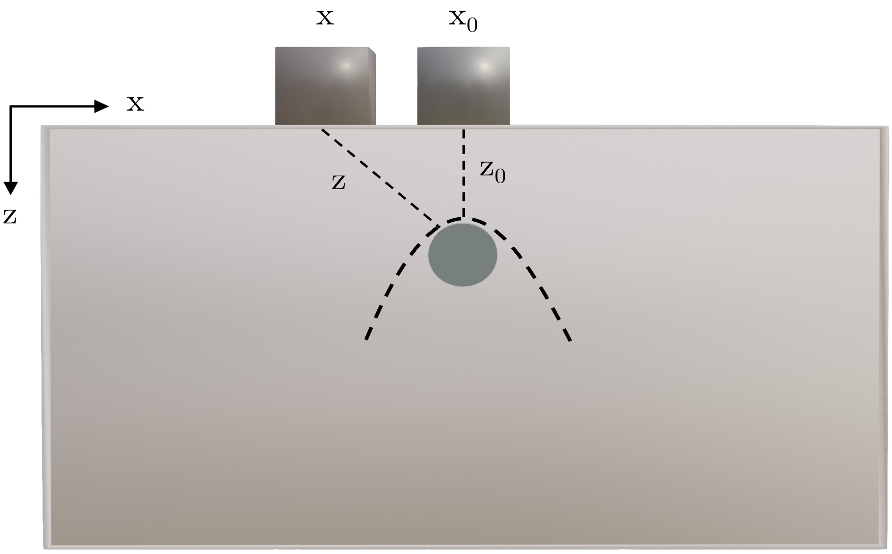


Figure 6.4: Geometry for the hyperbola acquisition over a cylindrical target

a finite radius and thus, works well only for a quite small radius where the target resembles a point reflector. In contrast, for increasing radius, this method leads to an inaccurate velocity estimate since the spreading of the hyperbola is attributed to a higher value of the velocity by the algorithm, although it is actually formed by a smaller velocity and the finite radius of the target.

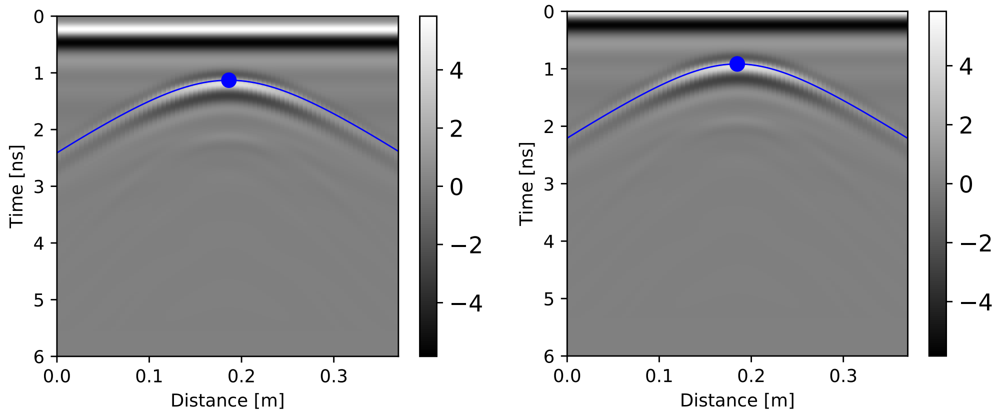
An equation that takes into account the radius R of an object was presented by Shihab et al. (2004) as

$$(z+R)^2 = (z_0+R)^2 + (x-x_0)^2 \tag{6.7}$$

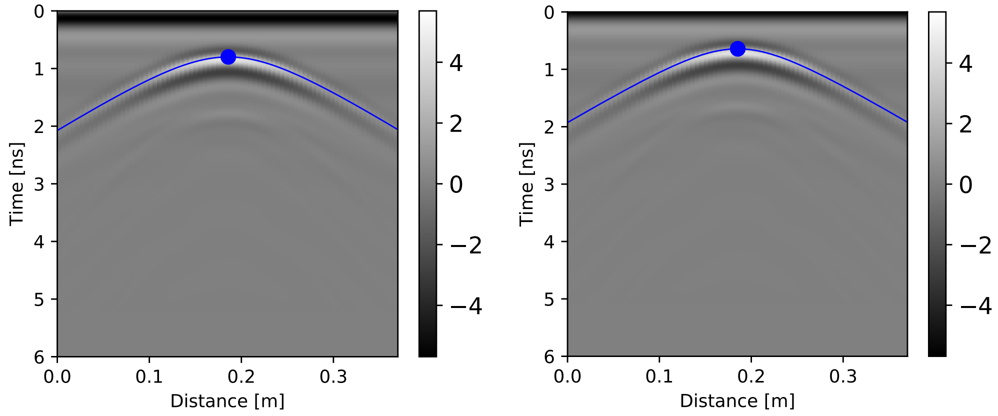
where z is the apparent depth of the cylinder from a receiver position x, z_0 is the depth of the top of a cylinder and x_0 corresponds to the horizontal position of the cylinder as displayed in Figure 6.4. Substituting $z = \frac{ut}{2}$ and $z_0 = \frac{ut_0}{2}$ results in

$$\left(\frac{t+\frac{2R}{u}}{t_0+\frac{2R}{u}}\right)^2 - \left(\frac{x-x_0}{\frac{u}{2}t_0+R}\right)^2 = 1 \tag{6.8}$$

For a zero radius, this equation reduces to equation 6.7. It is obvious that this equation has two unknown quantities, R and u. To acquire estimates of these parameters, an optimisation fitting method is implemented which is commonly a least-squares fitting method. Nonetheless, this fitting problem suffers from non-uniqueness of the solution, where different combinations of radii and velocities might lead to the same exact hyperbola and therefore is difficult to characterise accurately the target.



(a) Estimated velocity v=0.175 m/ns (b) Estimated velocity v=0.184 m/ns



(c) Estimated velocity v=0.194 m/ns (d) Estimated velocity v=0.203 m/ns

Figure 6.5: Estimated velocities from hyperbola fitting for different time zero corrections.

Furthermore, these techniques are highly dependent on the time zero correction and the direction of scanning relative to the position of the target. Figure 6.5 illustrates an example of hyperbola fitting for different time zero corrections, namely for selecting the first break, the maximum positive peak, the maximum negative peak and the midpoint between the positive and negative peak. The estimated velocity for each case is given in the Figure, showing that different velocities are obtained for different time zero corrections. Crossing a target at an angle different than 90° will result to a wider hyperbola and will overestimate the velocity. Therefore for optimal results the target should be aligned in the same direction as the E-field of the antenna. But even when crossing a target at a 90° angle, for different time zero positions, a different velocity value will be obtained in addition to the error from the 0 radius assumption, if the first method is used. In contrast, the deep learning method of finding the velocity is independent of the direction of scanning relative to a target, does not make any assumptions and does not require the existence of a hyperbola in the data. The

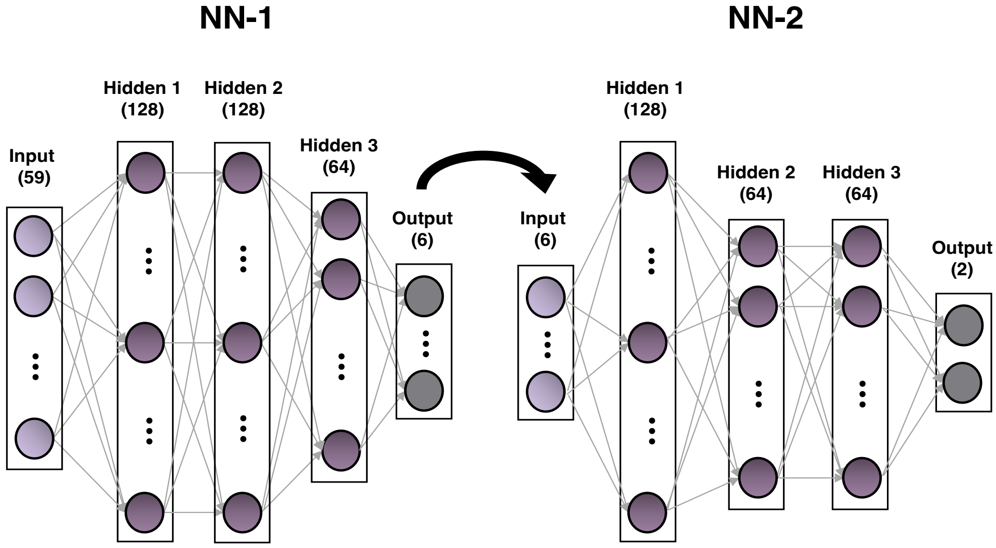


Figure 6.6: Proposed architecture of the two coupled neural networks.

velocity is estimated using entirely background information and is more accurate than conventional hyperbola fitting.

Hyperparameter tuning was used to define the architecture of the two networks, the learning rate and other settings of the ML scheme in order to acquire an optimal ML model. Grid search was utilised to find the optimal hyperparameters for the specified problem by trying different combinations of certain parameters. For both networks, the number of layers, number of nodes, the learning rate and the activation functions were tuned using a grid search. The ranges used for each hyperparameter in grid search are shown in Table 6.1. The architectures and the final hyperparameters are discussed below.

6.4.1 *Neural Network 1 Architecture*

For the first network (NN-1), a feed-forward fully-connected (Dense) network was selected for the training. The input and output layers of the model consist of 59 and 6 nodes, respectively, as specified previously. For this network, it was deducted after grid search that three hidden layers, that are composed of 128, 128 and 64 nodes, respectively, were sufficient to capture the relationship between the input and output data. All layers use ReLU as an activation function, expect from the output layer, which uses a linear activation function, since it is a regression problem. The Adam optimiser is used to decrease the mse with a learning rate of

	Lower limit	**Upper limit**
Number of layers	2	8
Number of nodes	16	256
Learning rate	0.0001	0.1
	Options	
Optimiser	SGD RMSprop Adagrad Adadelta Adam	
Activation function	ReLU Tanh Sigmoid	

Table 6.1: Ranges and options of hyperparameters used in grid search for the background prediction ML models.

0.001 and a decay of 1e-7 at each iteration. As a loss function the mse between the predictions and the true labels is chosen, given by

$$MSE_{train} = \frac{1}{m}\sum_{i=1}^{m}(\hat{\mathbf{y}}_i^{train} - \mathbf{y}_i^{train})^2 \tag{6.9}$$

where $\hat{\mathbf{y}}_i^{train}$ is the predicted PCA values for the ith training example and $\mathbf{y}_i^{train}$ its the true label. The network was trained for 200 epochs.

6.4.2 *Neural network 2 architecture*

The second network (NN-2) is also a dense network that takes as input the output predictions from the previous network, resulting in an input layer with 6 nodes. It consists of three hidden layers with 128, 64 and 64 nodes, respectively and an output layer with 2 nodes. The rest of the hyperparameters are common in the two networks. The mse between the predicted and the true ϵ_r and σ was used as a loss function. The architecture of both networks is illustrated in Figure 6.6. For both models checkpoints were utilised in order to select as final model the one that achieved the best performance on the validation set through the epochs.

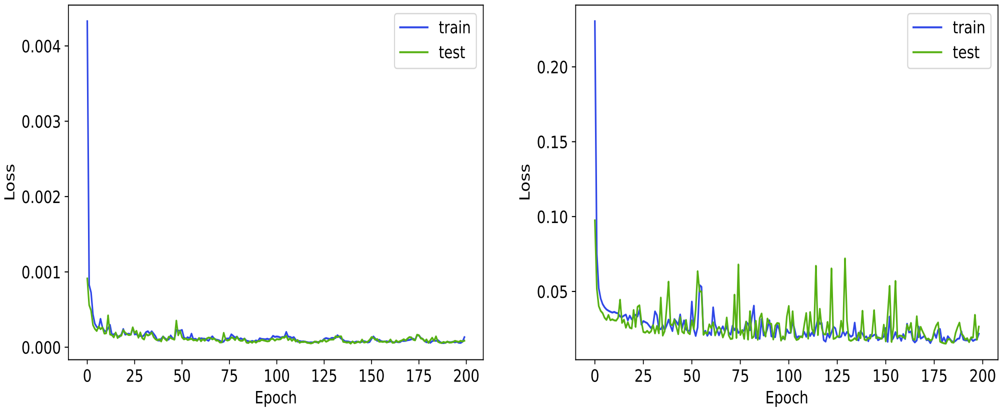
	Predicted value	**True value**
A-scan 1 permittivity ϵ_r	5.81	5.77
A-scan 2 permittivity ϵ_r	6.09	6.18
A-scan 1 conductivity σ (S/m)	0.00039	0.00034
A-scan 2 conductivity σ (S/m)	0.0205	0.0200

Table 6.2: Synthetic A-scan predicted dielectric properties.

6.5 RESULTS

Training the first NN model resulted in an mse value of 2.6e-4 and 7.0e-4 for the training and test set, respectively. Transforming the training data from PCA back to time domain and calculating the errors, resulted in an mse of 6.7e-6 and 8.7e-6 for the training and test set, respectively. Permittivity was estimated with an accuracy of ± 0.2 for the training and ± 0.3 for the test set, whilst conductivity predictions had an accuracy of ± 0.001 S/m for the former and ± 0.0015 S/m for the latter. The loss values plotted per epoch for each of the two networks are displayed in Figure 6.7. In the graphs, both the train and the test losses are plotted, showing the convergence and the ability of the networks to generalise since, not only low training but also low test set errors are achieved. Oscillations in the loss are visible in the data, which are expected especially in the case of the test set, before achieving convergence to a minimum loss. The first network converged faster as illustrated in Figure 6.7a, showing that a very small loss is obtained even with a small number of epochs, in contrast with the second network which was oscillating for many epochs before reducing the loss, as shown in Figure 6.7b. To demonstrate the convergence of the first network that predicts the background, the true background response is plotted versus the ML background predictions per different epochs for two randomly selected A-scans from the test set in Figures 6.8 and 6.9, respectively. Each plot corresponds to a certain epoch, which is reported in the plot title along with its corresponding mse for the training set. Note that the mse reported here was calculated after transforming the predictions from PCA back to time domain. From both figures, it is obvious that in the first iterations with the highest error, the predicted responses do not resemble the ground truth. As the training progresses, the predictions of the network are improved until reaching a very low error in the last iterations, where the true and predicted responses are almost identical, as shown in the Figures.

However, a low loss does not necessarily mean that the network predictions will be accurate and the performance of the network should always be evaluated



(a) Background prediction

(b) Dielectric properties prediction

Figure 6.7: Train and test set loss values per epoch for a) Background prediction and b) Permittivity and conductivity prediction.

by investigating the data. In this section, the performance of the algorithm is tested by applying the scheme to both numerical and real data. Subsequently, the estimated permittivity profile is used in a reverse-time migration (RTM) scheme. This allows to perform migration using the actual permittivity profile, which increases the reliability and accuracy of the reconstructed images.

6.5.1 *Synthetic A-scan data*

To evaluate the performance of the ML model, the algorithm was applied to the test set, which was not part of the training process. Two randomly chosen A-scans are presented in Figure 6.10. In both cases, the model consists of a homogeneous background with a buried metal cylinder. For the first case, the background medium is simulated with an $\epsilon_r = 5.77$ and $\sigma = 0.00034 S/m$ and a cylinder with radius 2 cm is buried at 17 cm. The second case has a background with $\epsilon_r = 6.18$ and $\sigma = 0.02 S/m$ and a cylinder with a radius of 1.3 cm and depth of 4 cm.

The raw data are presented in the top row of the Figure, the predicted background compared with the true background in the middle row, whereas the bottom row illustrates the responses after the predicted background has been subtracted. It is obvious that in both cases, the network was able to successfully predict the background, which is indistinguishable from the ground truth. Even in the presence of a shallow buried object, the ML model was able to identify the background signal accurately. Subsequently, the predicted background is subtracted from the raw total traces to perform background removal. Finally, given the predicted background response, its dielectric properties are estimated and are presented in Table 6.2 along with the true ones. From the Table, it is evident

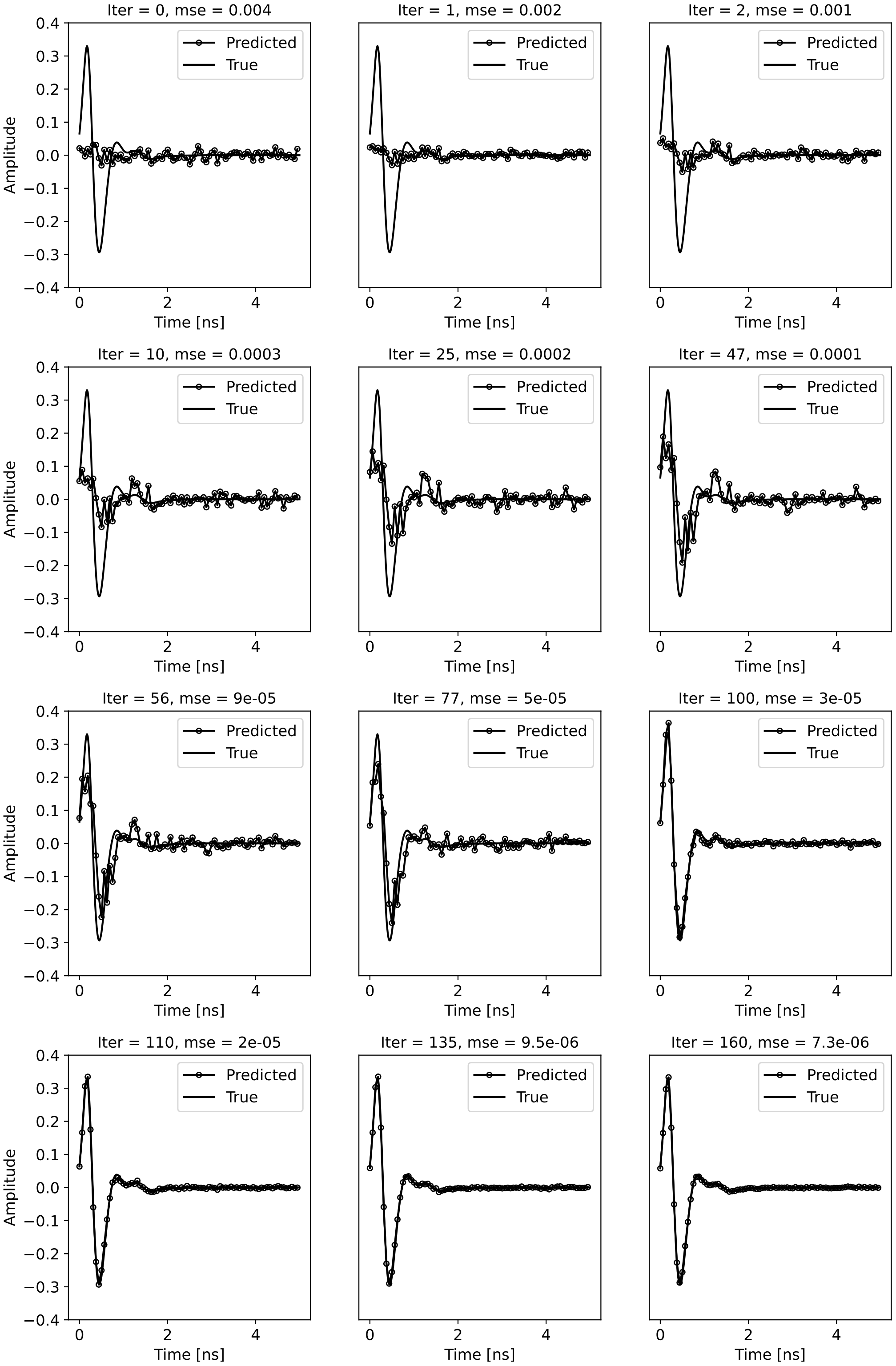


Figure 6.8: Background response predictions versus true per different epochs from high to low error for the first random input A-scan chosen from the test set.

that for both A-scans, the algorithm predicted permittivity and conductivity values very close to the true ones.

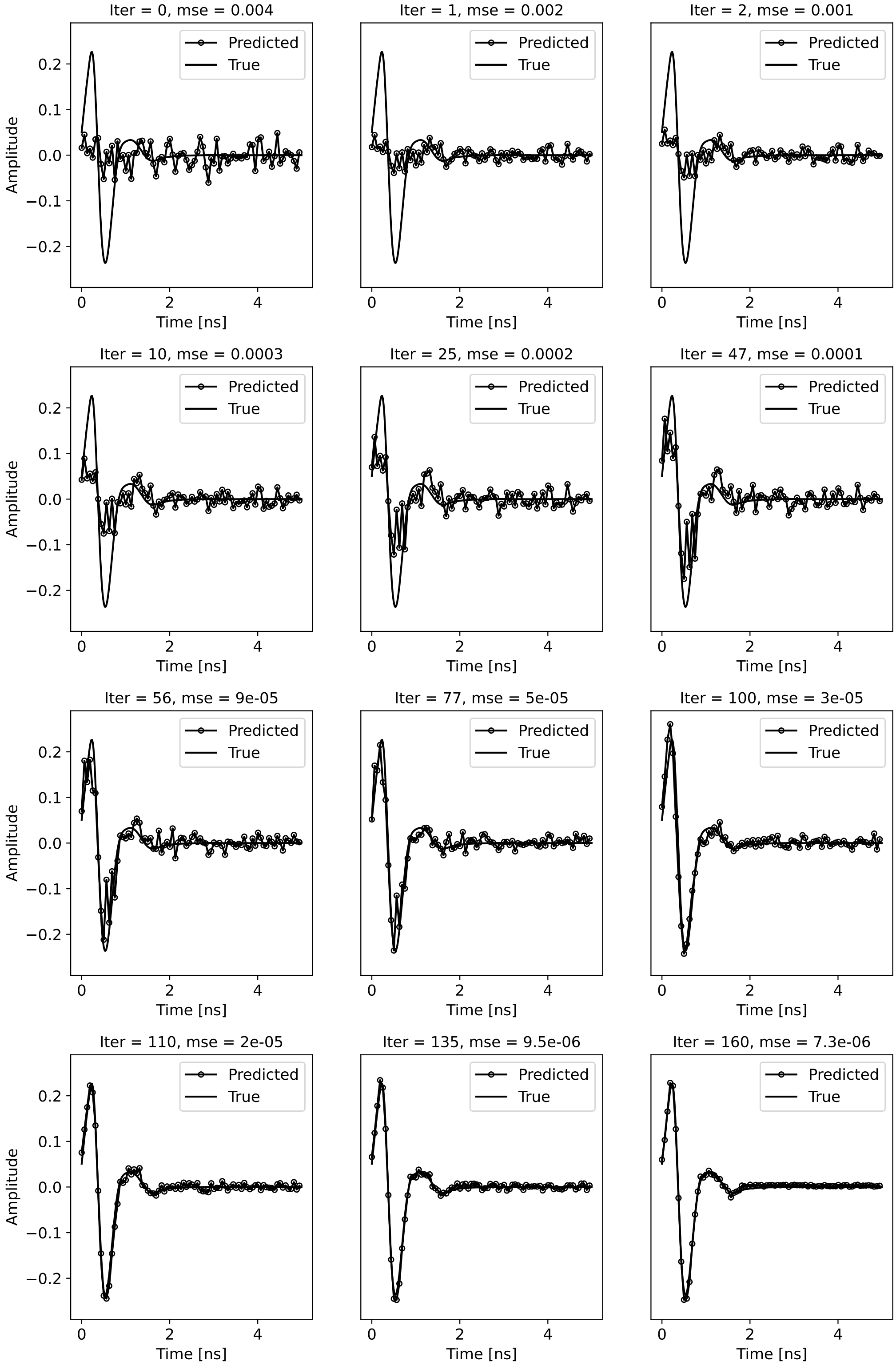
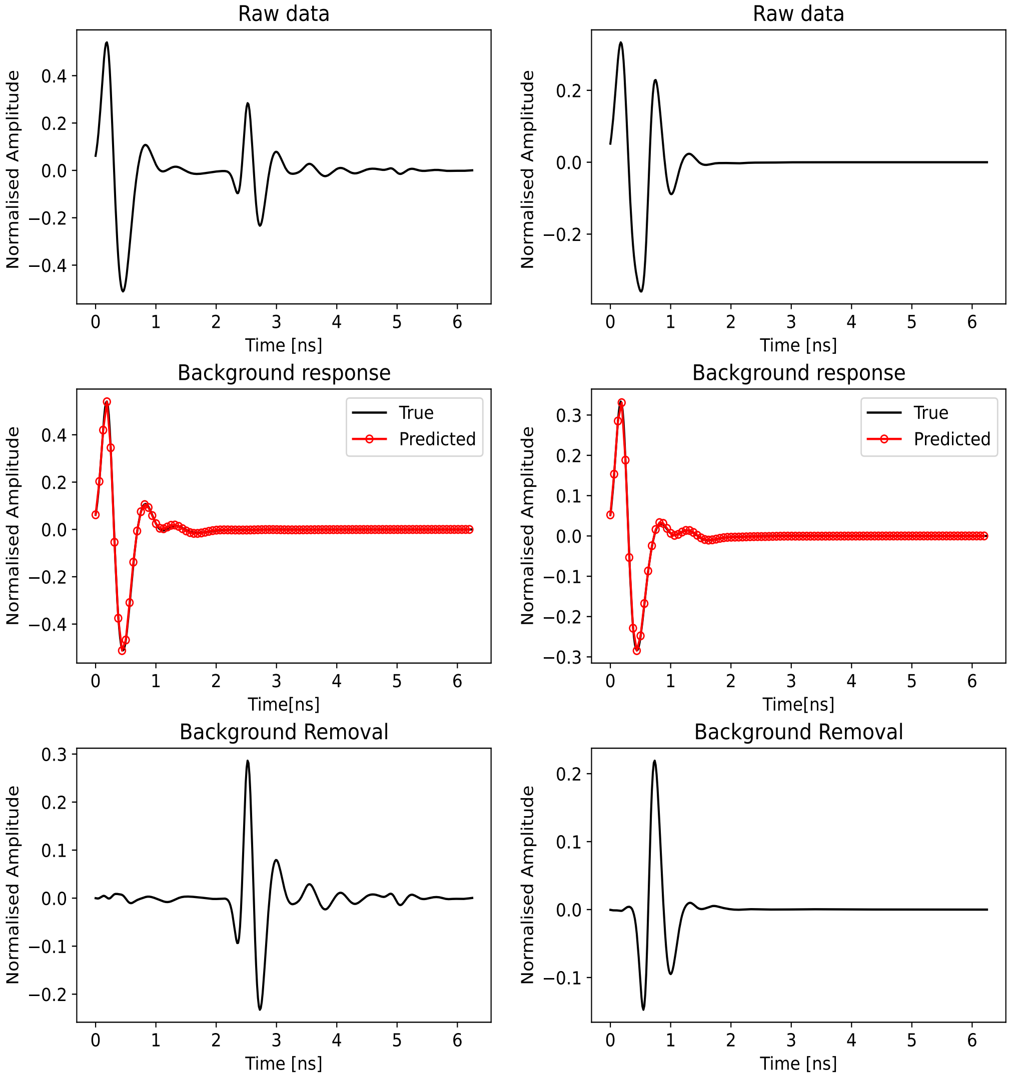


Figure 6.9: Background response predictions versus true per different epochs from high to low error for the second random input A-scan chosen from the test set.

The suggested deep learning scheme was applied on 50 different A-scans randomly selected from the test set and the predicted dielectric properties were



a) b)

Figure 6.10: The NN background removal applied on two synthetic A-scans. Top: Raw data, Middle: Predicted versus true background, Bottom: Data after background removal.

compared with the ground truth. Figure 6.11 displays the comparison of the permittivities, whereas Figure 6.12 for the conductivities. The plotted lines in the two graphs are used to indicate how far the predictions are from the true properties, where the point will fall on the line if the true value is predicted exactly. In all cases, the predicted values are very close to the true ones, proving the accuracy of the ML scheme.

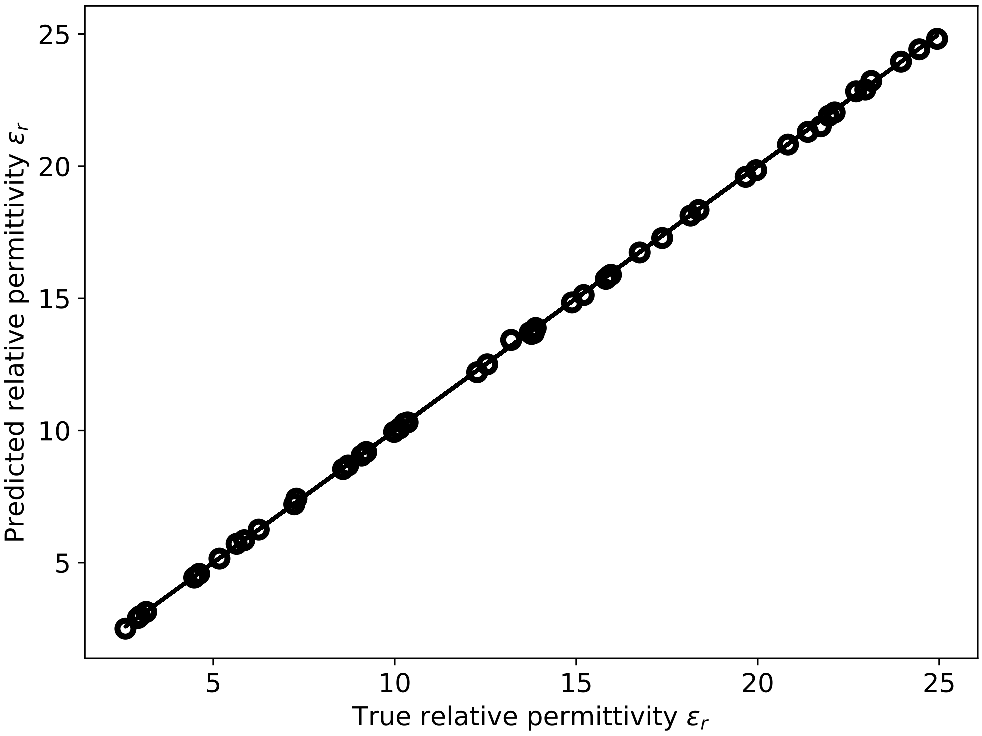


Figure 6.11: Comparison between true and predicted permittivity values for 50 different randomly selected cases.

6.5.2 *Synthetic B-scan data*

Although the deep learning scheme is trained using A-scan data, it can be applied on B-scan data as well. In this case, a background signal is predicted by NN-1 for each A-scan included in a B-scan and subsequently subtracted from the corresponding A-scan. Thus, a background response is subtracted trace by trace, in contrast with other methods where the same single background response is removed from all A-scans. A permittivity and a conductivity estimate is acquired by NN-2 for each A-scan and all the estimates are averaged out to obtain a single mean permittivity and mean conductivity that characterise the entire B-scan. However, in case of a considerably inhomogeneous background medium, a permittivity and conductivity distribution, which characterise the area locally around an A-scan, are more useful than a mean value for the total area of interest.

A number of synthetic B-Scans were also generated to evaluate the deep learning scheme and compare its performance with the mean subtraction method. These data were not part of the training process and are originated from different and more complex scenarios than the ones used in the training. Four different B-scans are discussed here, which are visible in Figure 6.14, whereas the geometry of the models used to generate the B-scans is shown in Figure 6.13. The Bscans on the left of Figure 6.14 show the raw data, in the middle the data after NN

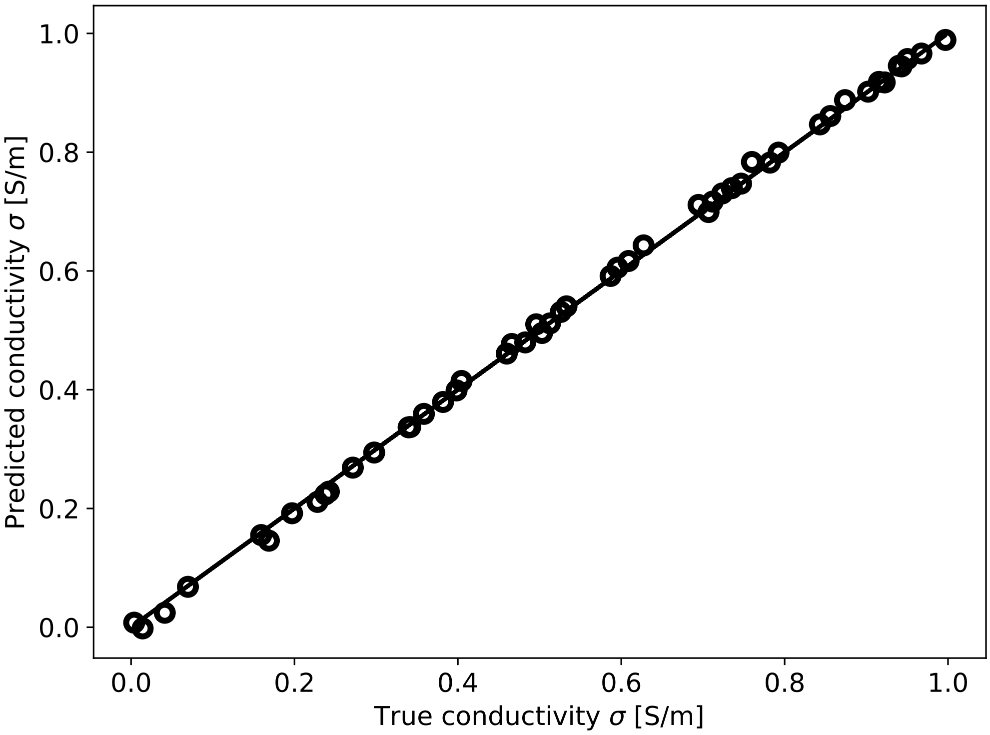
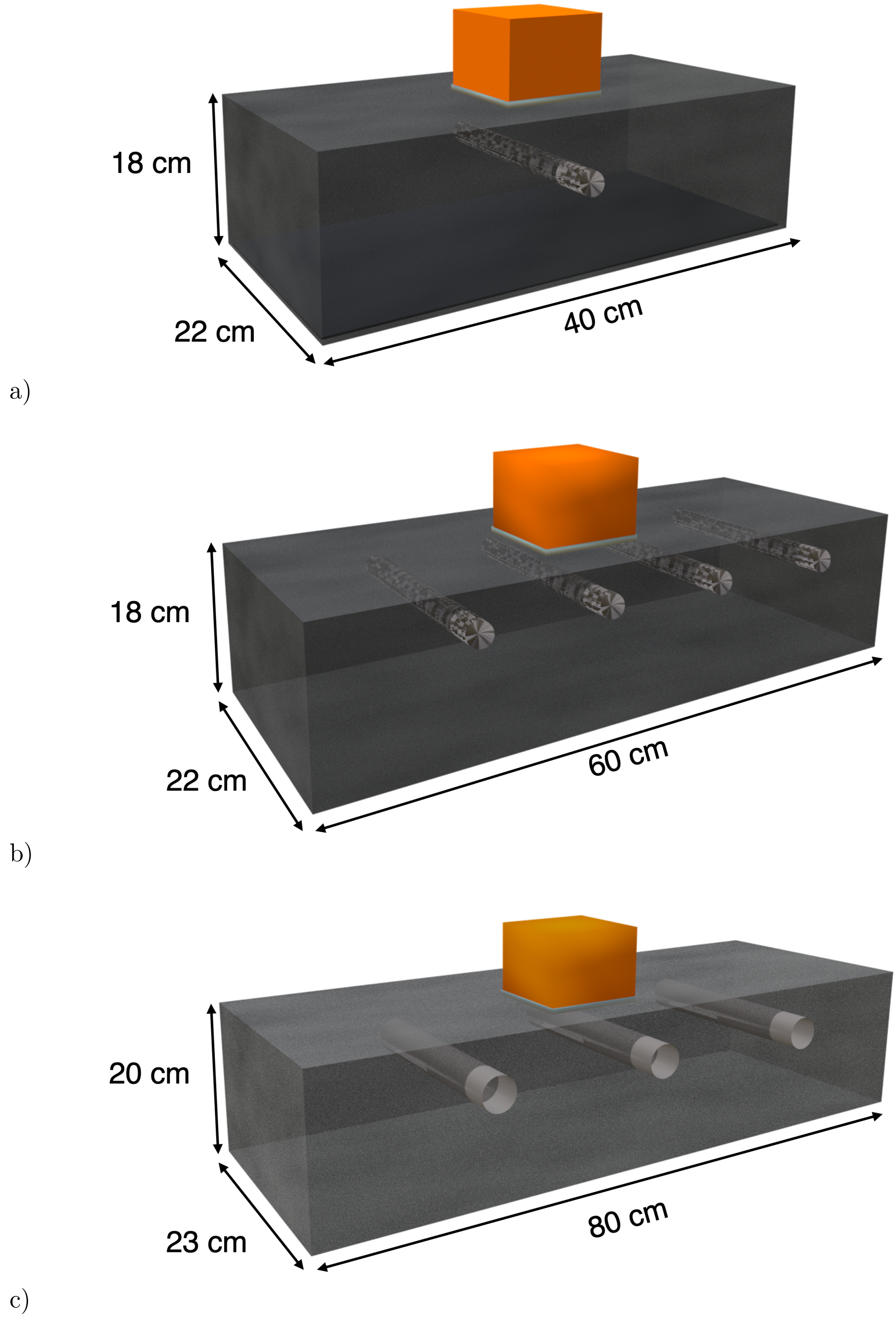
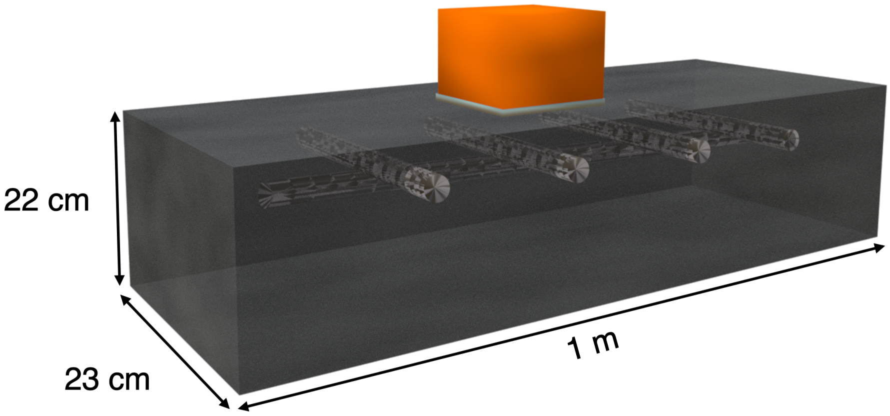


Figure 6.12: Comparison between true and predicted conductivity values for 50 different randomly selected cases.

background removal and on the right the Bscans after mean subtraction. In the first scenario, a rebar of radius 1.5 cm is buried at a depth of 8 cm along with a metal plate buried at 18 cm depth and spanning the entire Bscan. The background medium is simulated with an $\epsilon_r = 3$ and $\sigma = 0.001 S/m$, whereas the targets as PECs. From Figure 6.14a it is obvious that the NN model predicted accurately the background response, which was subsequently subtracted from the original data. Furthermore, the NN scheme did not affect the flat response from the plate, in contrast to the mean background removal, which partially removed the it in addition to creating artifacts in the data. The second scenario consists of four rebars closely spaced together and buried at a shallow depth of 4 cm. The background is modelled with an $\epsilon_r = 6$ and $\sigma = 0.01 S/m$ and the rebars as PECs with radius 1.5 cm. It is shown that even in the presence of shallow buried targets and closely spaced together, the network is able to recognise the background successfully. The mean background removal created again an artifact around 0.5 ns time.

For the third case, a background medium with $\epsilon_r = 9$ and $\sigma = 0.015 S/m$ is simulated along with four empty plastic pipes of 2 cm radius and 4 mm wall thickness, which are considered as PVC with $\epsilon_r = 4$ and $\sigma = 0 S/m$ and are placed at a depth of 6 cm. This scenario demonstrates that the scheme, although trained with metallic rebars only, works even in cases where the data include


18 cm
22 cm
40 cm
a)
18 cm
22 cm
60 cm
b)
20 cm
23 cm
80 cm
c)



d)

Figure 6.13: Modelled geometry of the B-scans generated for evaluating NN background removal.

responses from targets with different properties. The background signal was again removed accurately using the ML scheme, while the average subtraction created an artifact around 1.5 ns. The fourth and last case represents a more complex scenario, where five targets are placed in different depths. In particular, the direction of scanning is perpendicular to the long axis of 4 rebars of radius 1.2 cm, which are buried at 4, 5, 5, 6 cm, respectively. The fifth rebar of radius 1.2 cm is parallel to the direction of scanning and buried below the rest of the targets. This rebar goes from a depth of 10 cm for the left end to 11.3 cm for the right end of the space and therefore is a dipping target as illustrated in Figure 6.14d. This is a typical scenario of a reinforcing mesh found in concrete slabs. The background was modelled using $\epsilon_r = 6.5$ and $\sigma = 0.01 S/m$. Again, even in this complex scenario, the ML model distinguished the background response, without affecting the dipping linear response from the rebar. The average subtraction behaved similarly to the previous cases.

Table 6.3 presents the mean predicted dielectric properties, which again are in very good agreement with the real ones. The estimated permittivity across the A-Scans was ± 0.2 from the mean value, while the conductivity was ± 0.00086 S/m. Using the predicted permittivity, the velocities were calculated and used to convert from time to depth and thus, a depth scale is included in the NN results, as shown in Figure 6.14. So far, only homogeneous and non-dispersive media cases were considered, since only homogeneous and non-dispersive materials were used as part of the training. However, to evaluate the capabilities of the algorithm, the scheme was also applied to dispersive and inhomogeneous half-spaces. Figure 6.16 shows two B-scans obtained over the same inhomogeneous half-space first with no targets, and then with two rebar targets of 1.6 cm radius buried at 7 cm depth. The soil was simulated with a sand fraction of $S = 0.5$,

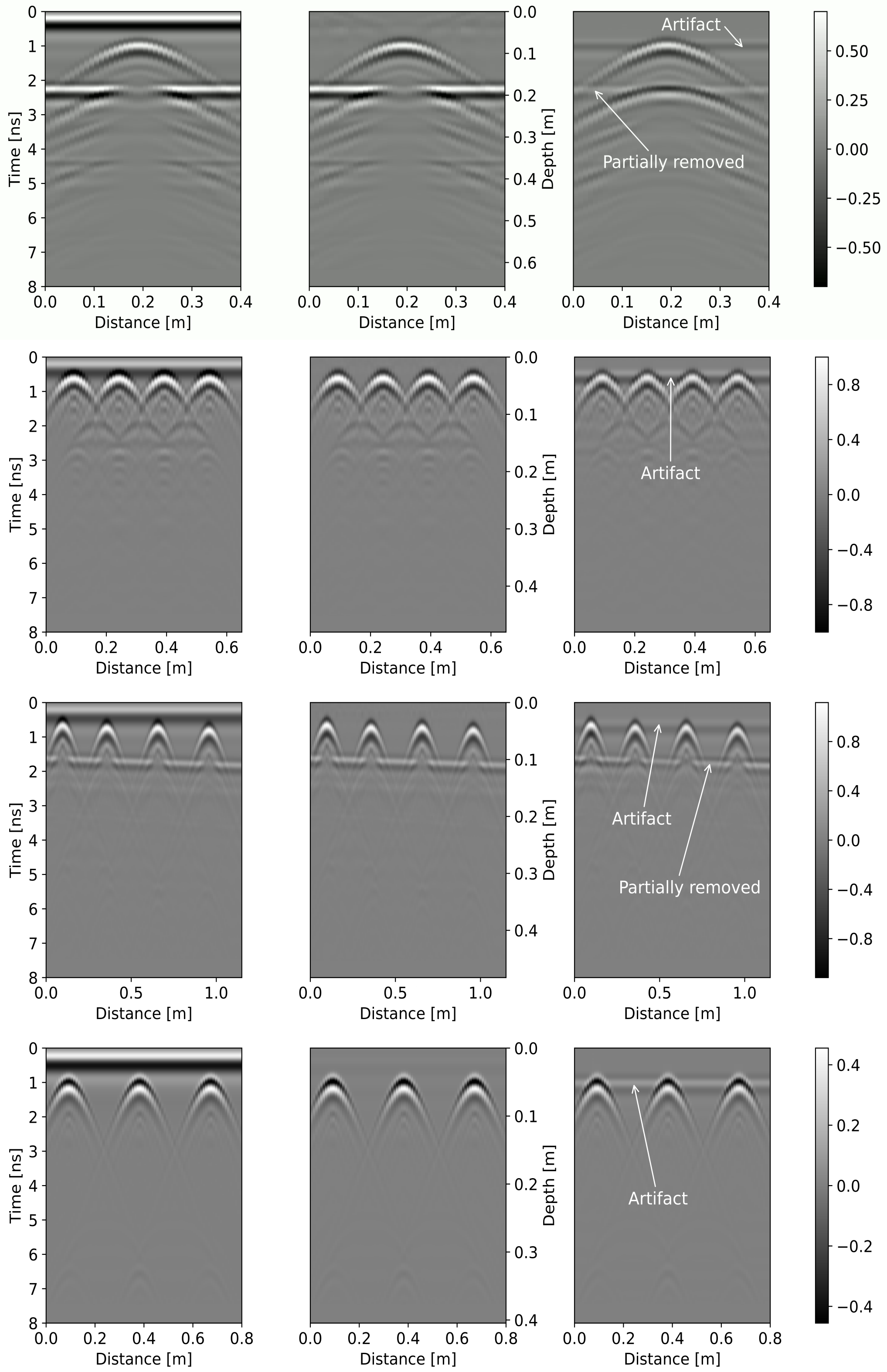


Figure 6.14: Background removal on synthetic data. Left: Raw data. Middle: Data after NN background removal. Right: Data after mean subtraction.

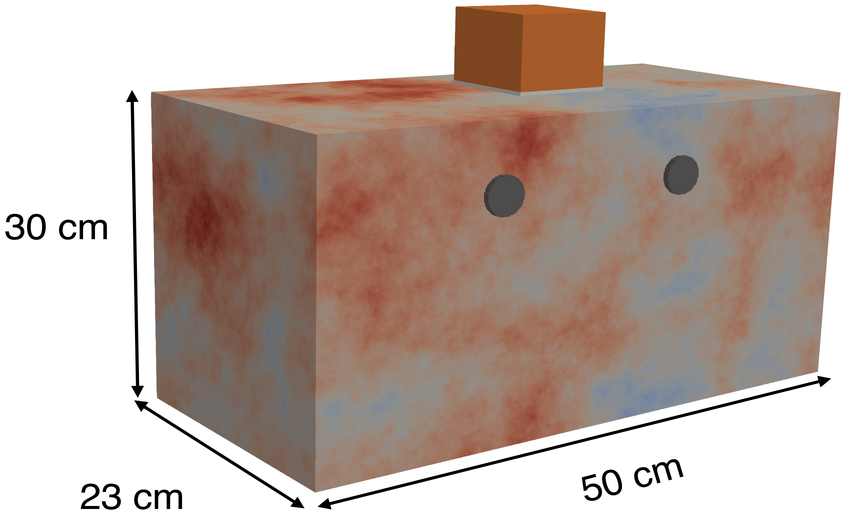


Figure 6.15: Modelled geometry of the inhomogeneous background medium.

a clay fraction of $C = 0.5$, bulk density of 2 g/cm^3, sand particle density of 2.66 g/cm^3, and a volumetric water fraction ranging between 0.001 - 0.2 with 40 materials with different dispersive properties used to capture this range. The geometry of the model is given in Figure 6.15. Since the ML scheme uses local information contained in an A-scan to predict the background, in contrast to other methods which use the entire B-scan to predict a single background, it was able to accurately predict the background response even in an inhomogeneous environment with and without targets included. An inhomogeneous environment has variations at its extent, which can be better captured using local information.

Finally, noise was introduced to the synthetic B-scans to test the performance of the scheme with noisy data. The first and the second synthetic scenario with added noise are used here to demonstrate the performance of the network. Gaussian white noise was added to the data, where samples are drawn randomly from a normal distribution with zero mean and the noise has a constant power spectral density

$$\mathbf{S}(i)^* = \mathbf{S}(i) + \mathbf{n}(i) \tag{6.10}$$

where $\mathbf{S(i)}$ represents the i-th A-scan of a B-scan and $\mathbf{n(i)}$ is the noise added to the A-scan. Figure 6.17 displays the noisy B-scan data along with the B-scans after NN background removal. In addition, randomly selected A-scans from the two B-scans are plotted in Figure 6.18 along with their predicted background. Even in the case of noisy data, the algorithm was able to recognise the background responses. However, having excessive noise in the data leads to incorrect shifting of the traces and thus, in this case the performance of the algorithm is degraded.

The diversity of cases presented, demonstrate that the deep learning scheme performs very well in a variety of complex scenarios that were not part of the

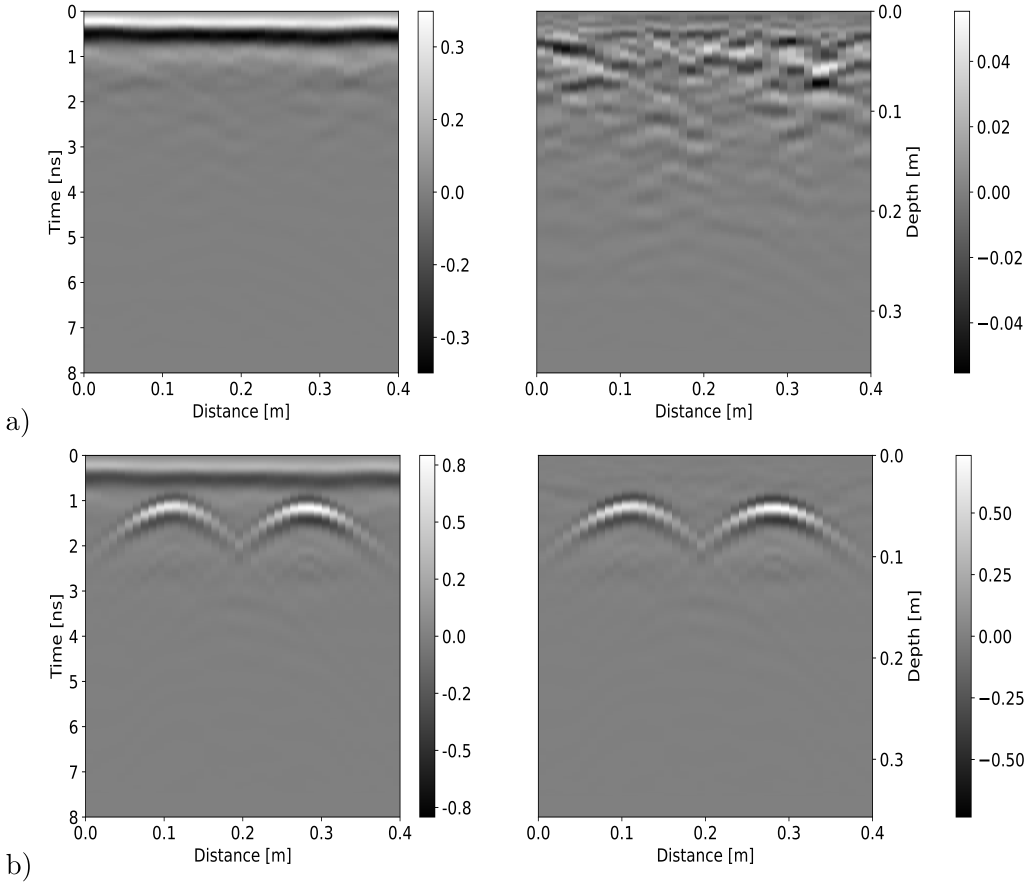


Figure 6.16: NN Background removal on synthetic data over inhomogeneous environment. Left: Raw data. Right: Data after NN background removal.

training. NN-1 was able to accurately predict the background response regardless of the background properties and the depth, size, orientation, type or number of targets. Following the efficiency of NN-1, NN-2 provided accurate estimates of the background dielectric properties in all cases.

6.5.3 *Real data*

The proposed scheme is now applied on real data to evaluate its accuracy with real case studies. Data from 7 different cases were collected using the GSSI 2 GHz “palm” antenna system. Cases a) to e) were collected on the ground floor of the William Rankine building at the School of Engineering, The University of Edinburgh, which is essentially a concrete slab, whereas cases f) and g) are from two experimental reinforced concrete slabs located in the NDT laboratory in the same building. The B-scans acquired are shown in Figure 6.20, while pictures of the survey areas are presented in Figure 6.19. No gain was applied at the raw or processed data of Figure 6.20. It is obvious that these B-scans are far more complex than the scenarios used in the training set and the synthetic B-scans

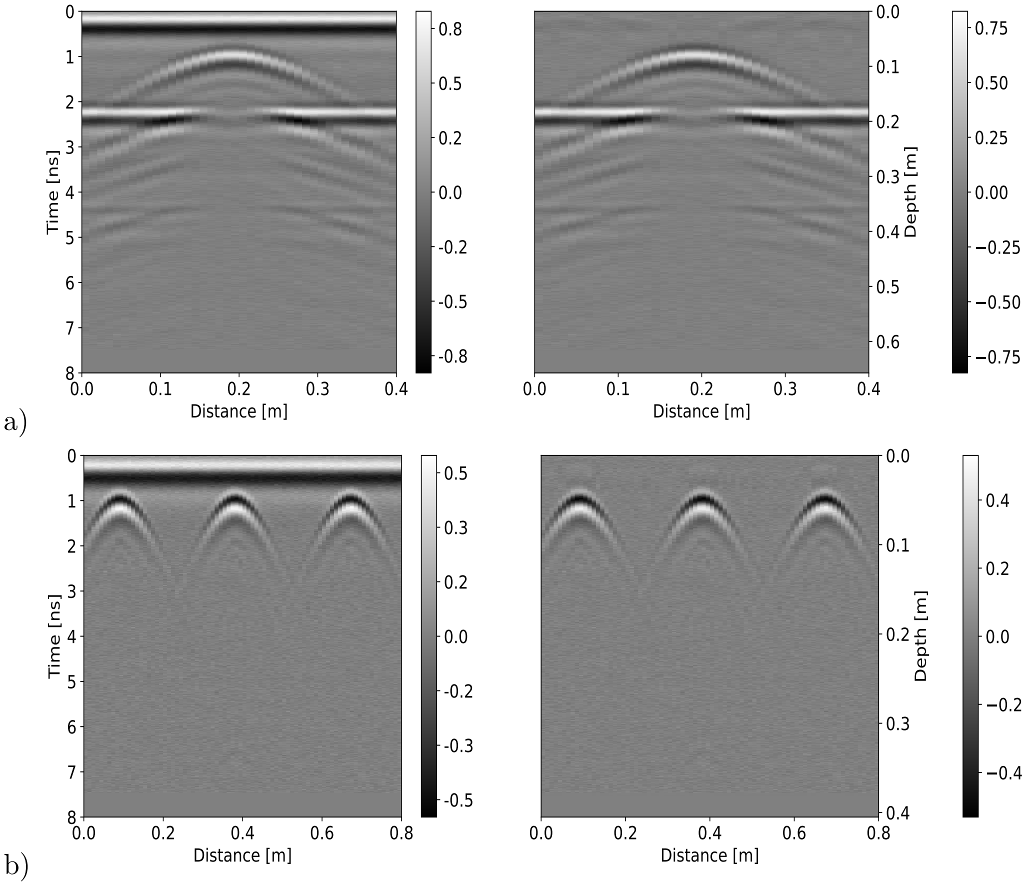


Figure 6.17: NN Background removal on noisy synthetic data. Left: Raw data. Right: Data after NN background removal.

used for evaluation. Multiple targets at different depths can be seen, linear and dipping features and targets closely spaced together. In addition to that, as with all real measurements, noise is contained in the data.

A concrete slab is a structural element of most modern buildings, that consists of cast concrete of certain thickness with flat, horizontal surfaces. A steel-reinforced concrete slab is most often used to construct floors and ceilings, where a reinforcing rebar mesh and post-tension cables are embedded in concrete to compensate for concrete's low tensile strength. Apart from the reinforcement, metallic or non-metallic conduits can also be found embedded into concrete. Therefore, in cases a) to e), it is expected to find a number of targets as included in a typical reinforced concrete floor slab. The hyperbolic signatures were obtained by scanning perpendicular to the targets, whereas the flat responses by scanning parallel, along the main of the targets. Although these five cases represent complex scenarios that were not part of the training, the scheme predicted the background signal successfully, as illustrated in Figure 6.20. Performing average background subtraction did not remove the background completely, with remaining parts

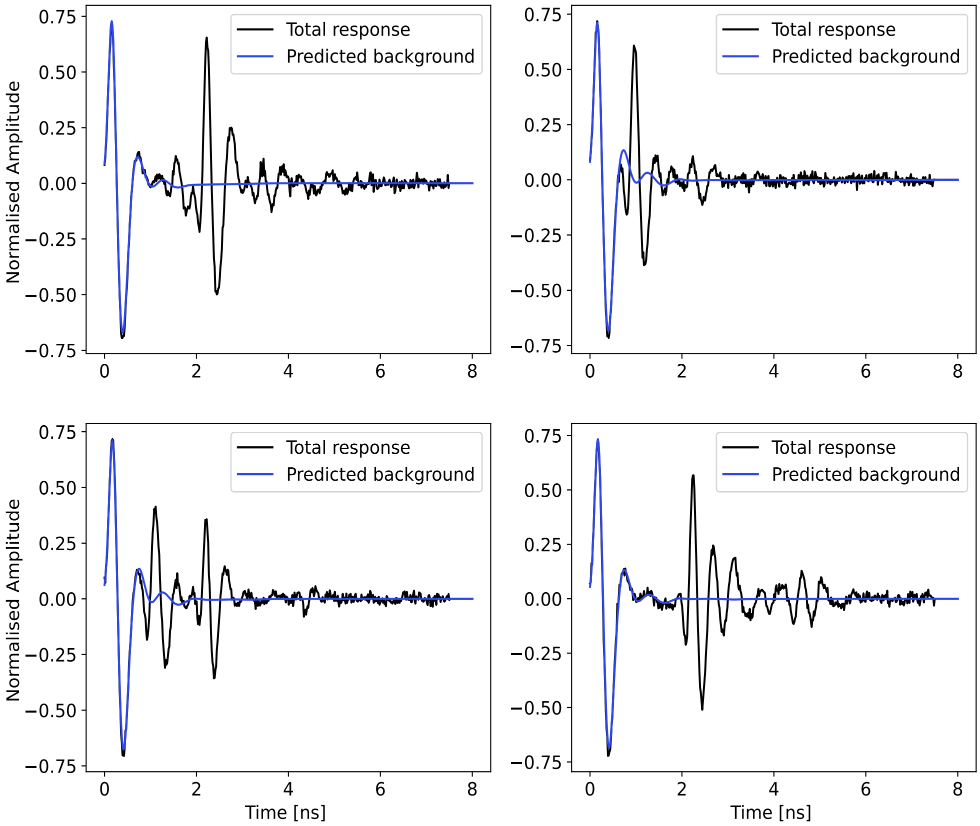


Figure 6.18: NN Background removal on noisy synthetic data: Selected A-scans plotted with their predicted background.

still visible in the data, in addition to partially removing the flat responses from targets.

Scenarios d) and e) consist of hyperbolic signatures from the steel rebars existing in the concrete slab, while in d) a flat response from the bottom of the slab is also present. The background signal was effectively removed using ML in both cases, whilst the average subtraction introduced flat-laying artifacts in the data. In addition, the bottom of the slab is clearly visible in the NN results even without gain, but eliminated using the mean subtraction. It is clear that in all cases, the deep learning scheme outperformed the mean removal method and produced GPR images with higher signal to clutter ratio. Note, that the NN scheme is trained to predict only the background response and no other clutter that might be present in the data.

After the background signals are estimated, the dielectric properties are predicted. The real investigated cases are considered relative homogeneous and their predicted mean dielectric properties are presented in Table 6.4 for all cases. For the first three cases, which were measured on the same ground floor, the resultant permittivity values are between 4.6 - 5.7, which are considered typical permittivity values for dry concrete Bourdi et al. (2012). For the last two cases,

	Predicted value	**True value**
B-scan 1 permittivity ϵ_r	3.18	3.00
B-scan 2 permittivity ϵ_r	6.12	6.0
B-scan 3 permittivity ϵ_r	8.8	9
B-scan 4 permittivity ϵ_r	6.4	6.5
B-scan 1 conductivity σ (S/m)	0.0016	0.001
B-scan 2 conductivity σ (S/m)	0.013	0.01
B-scan 3 conductivity σ (S/m)	0.0019	0.015
B-scan 4 conductivity σ (S/m)	0.009	0.01

Table 6.3: Synthetic B-scan predicted dielectric properties.

Case	**Predicted mean ϵ_r**	**Predicted mean σ (S/m)**
a)	5.71	0.012
b)	5.05	0.040
c)	5.32	0.036
d)	4.87	0.043
e)	4.65	0.070
f)	9.83	0.059
g)	9.03	0.051

Table 6.4: Real B-scan predicted dielectric properties.

the resulting permittivities were larger with values that vary from $\epsilon = 9 - 9.83$, indicating that these concrete slabs have greater water content. Using the predicted permittivities, the velocity of the materials and therefore, the depth of the targets are calculated, as shown in the second column of Figure 6.20 by the existing depth scale in the NN results.

Although for the ground floor scenarios, the true depths are unknown and not easy to obtain, for the concrete slab in the NDT lab, the true depth of targets is known and used for evaluation of the scheme. The comparison showed that the NN-resultant depths of the rebars match the true depths, validating the effectiveness of the deep learning scheme on real data. The network predicted accurately both the depth of the rebars and the thickness of the slab, which was 22 cm and was estimated successfully as shown in scenario d). Since the true conductivity values of the background media were unknown, independent tests with synthetic data showed that the scheme overestimates the conductivity values

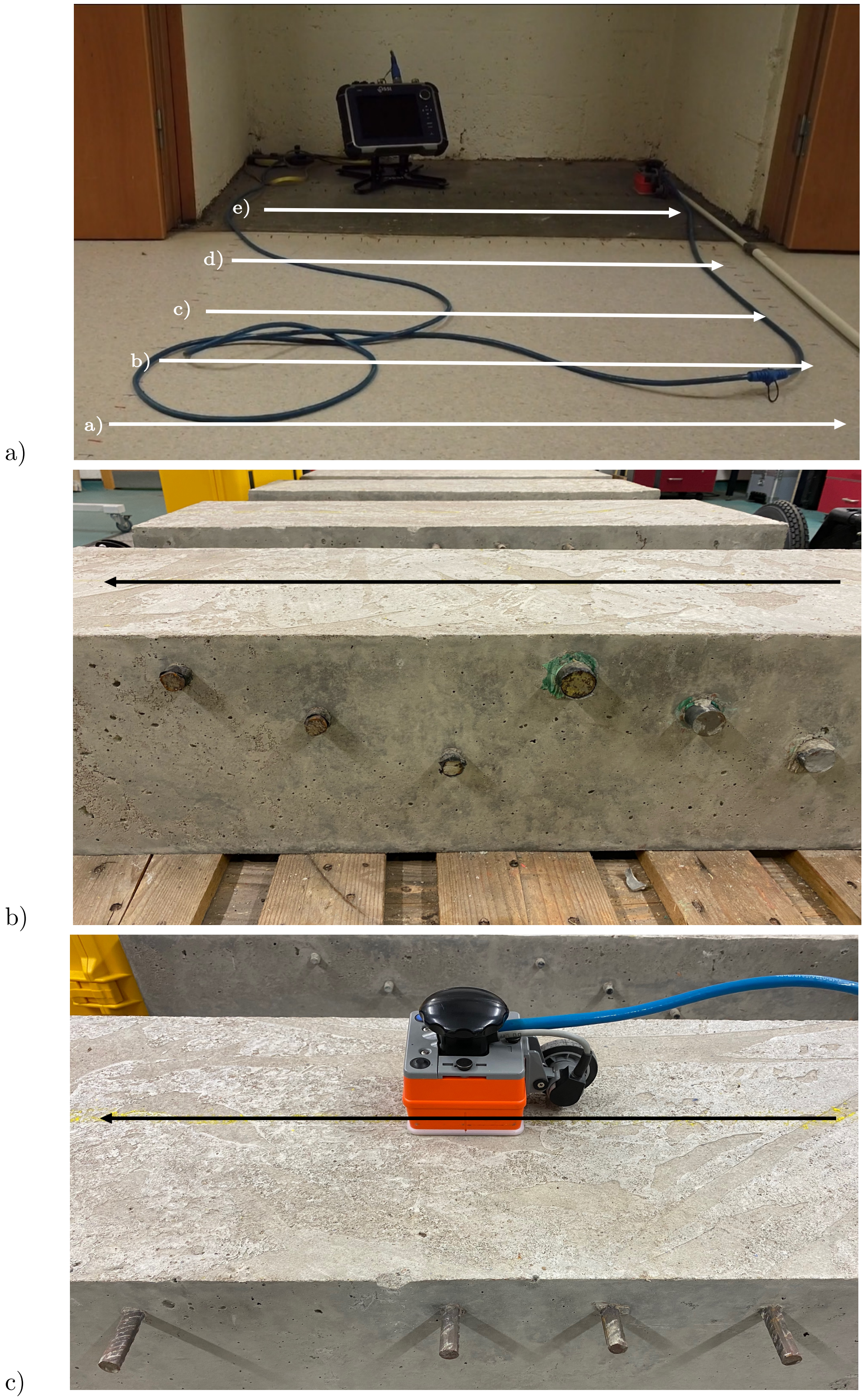


Figure 6.19: Locations of where the real B-scans from Figure 6.20 were acquired for a) B-scans a-e, b) B-scan f and c) B-scan g.

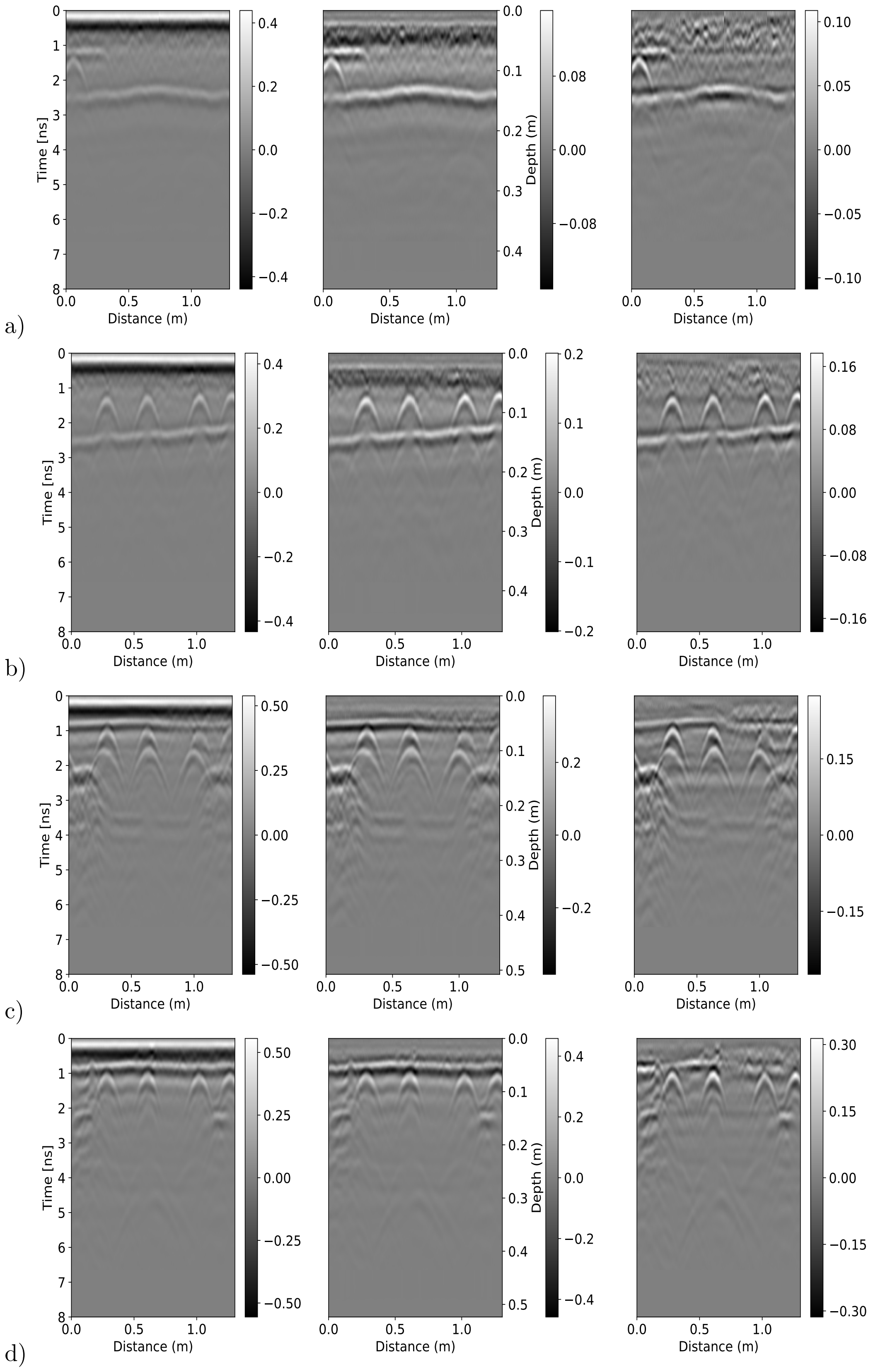


Figure 6.20: Real B-scan data. Left: Raw data. Middle: Data after NN background removal. Right: Data after mean subtraction.

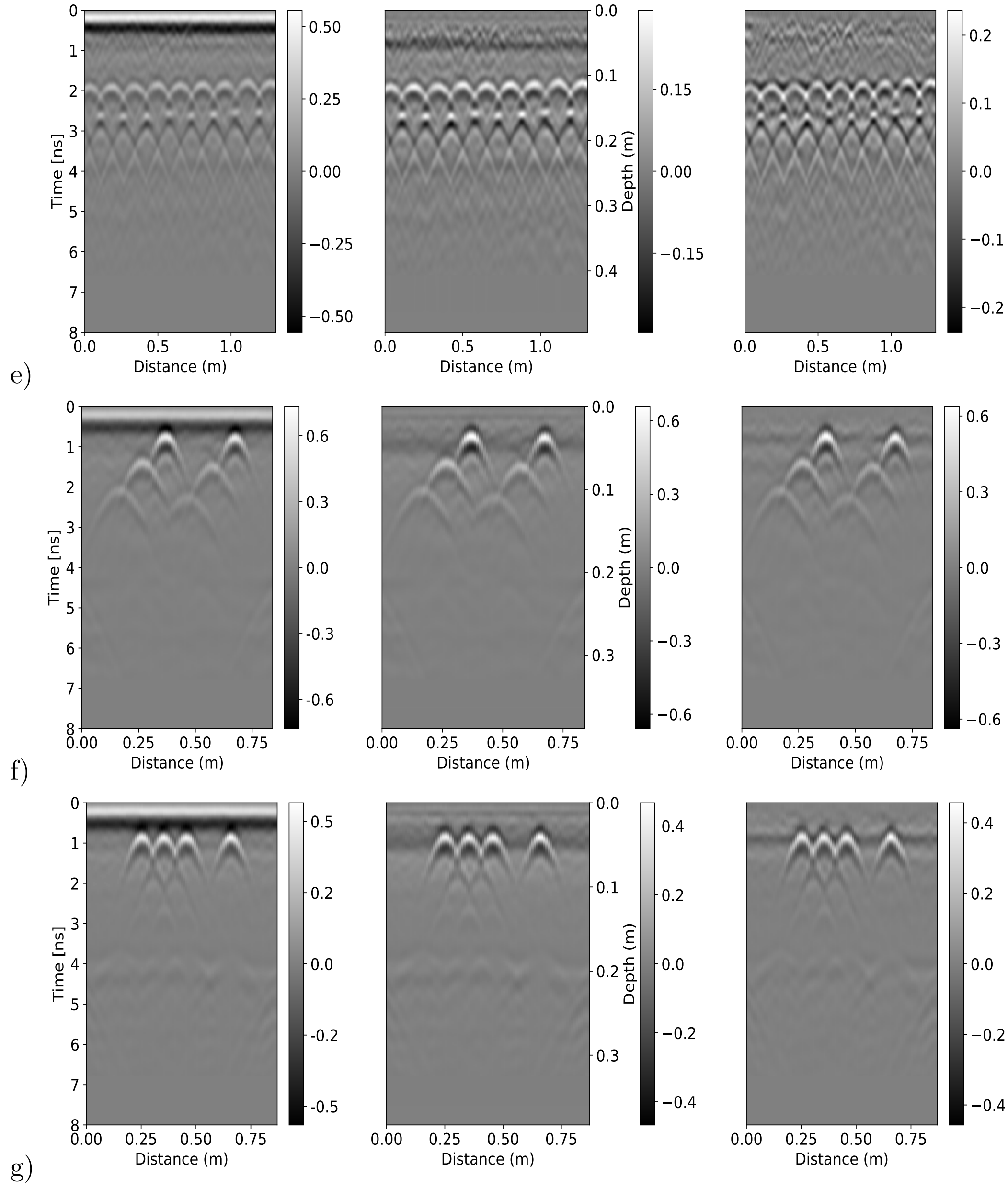


Figure 6.20: (Cont.) Real B-scan data. Left: Raw data. Middle: Data after NN background removal. Right: Data after mean subtraction.

when applied to dispersive media, since they were not included in the training process. For all cases, the estimated dielectric properties across the A-scans were very close to the mean value, with a maximum observable difference of 0.8 for the permittivity and 0.01 S/m for the conductivity.

6.6 REVERSE TIME MIGRATION APPLICATION

As described briefly in Chapter 2, migration is an advanced imaging technique, which is frequently applied on GPR data to move scattering events to their true spatial position and focus hyperbolic signatures (Yilmaz, 2001). Migration is based on the assumption that point scatterers are acting as impressed sources, which are initiated at time $t = 0$ (Yilmaz, 2001). The goal of migration methods is to make the B-scans appear as similar as possible to the true cross-section of the subsurface being investigated. To acquire a section close to the true cross-section, which is in depth, not only the hyperbolas have to be collapsed, but also an accurate estimate of the EM wave velocity is required. However, obtaining an accurate velocity estimate is not easy and therefore migrated images are in many cases presented in time, leading to time migration. When there is knowledge of the velocity, depth migration can be performed. There are many different migration algorithms which can be broadly classified into three categories: 1) Algorithms based on FD, 2) frequency-wavenumber (f-k) solutions and 3) algorithms based on the integral solution to the scalar wave equation. Migration was originally developed for seismic data but has been shifted to GPR surveys as well.

One of the first migration methods is the diffraction-summation technique, which is based on a summation of the amplitudes along a diffraction hyperbola (Yilmaz, 2001). A method similar to the diffraction-summation technique with additional corrections is the well-known Kirchhoff diffraction migration introduced by Schneider (1978), which is based on an integral formulation for the solution of the scalar wave equation. Hogan (1988) first applied it to GPR data and discussed all the parameters that affect the migration algorithm such as geological inhomogeneities and medium velocity. Kirchhoff migration does not perform well when significant lateral variations in velocity are present. Another migration method is the f-k migration or Stolt migration given by Stolt (1978), which is based on the scalar wave equation and Fourier transform to backpropagate the signal back to the scatterer. The phase-shift method by Gazdag (1978), also belonging to the category of f-k solution is based on a phase shift in the f-k domain. The frequency components are summed every depth step to acquire the GPR image at t=0. Migration has been applied successfully to different GPR surveys. Stolt migration was employed by González-Huici et al. (2014) to reconstruct the GPR image for landmine detection. The performance was compared with other reconstruction techniques, showing some limitations of the method.

A common migration method, originally developed for seismic data (Baysal et al., 1983), is the reverse time migration (RTM) (Leuschen and Plumb, 2001), which was developed based on the exploding reflectors model (Loewenthal et al., 1976). One of the first applications of RTM on GPR data was presented by Fisher et al. (1992), where RTM was applied to constant-offset data. A modified RTM scheme was described by Schofield et al. (2014) to increase the probability of detection of AT landmines. For the modified RTM, a stacking technique was implemented which stacks multiple GPR images with differing diffraction signatures and reduces the effect of varying ground conditions. Lei et al. (2018) implemented RTM for cross-borehole numerical data. The wavefield were initially decomposed into left and right-going wavefields using hilbert transform before applying RTM, which seemed to reduce the low-frequency migration artifacts.

Based on this model, it is assumed that the zero offset data are almost the same as if they had been generated using a number of sources at the positions where the target reflectors are located, and a series of receivers placed on the surface at each antenna location along the profile. The sources are triggered simultaneously and back-propagate upward towards the location of the antenna following the same path as the incident field. For this purpose, the received signals are reversed with respect to time and used as excitation sources, thus the name reverse time migration. If the EM waves propagate back in time, at $t = 0$ they will collapse at their cluttering sources (Leuschen and Plumb, 2001). The wavefield at $t = 0$ is then the migrated wavefield, where all energy has converged back to the reflectors positions.

Assuming a B-scan $B(\mathbf{q}, \mathbf{t})$, for the time interval $\mathbf{t}\{[0, t_{max}] \in \mathbb{R}^m \ t_{max} > 0\}$ and $\mathbf{q}_j = ||\langle x_j,\ y_j,\ z_j \rangle||$, where $\{x_j, y_j, z_j\} \in \mathbb{R}$ are the coordinates of the jth measurement. $B(\mathbf{q}, \mathbf{t})$ is first reversed with respect to time $B(\mathbf{q}, t_{max} - \mathbf{t})$, and afterwards the reversed traces are used as impressed current sources $J_u(\mathbf{q}, \mathbf{t}) = B(\mathbf{q}, t_{max} - \mathbf{t})$ where $u \in \{x, y, z\}$ is the polarisation of the receiver (Giannakis et al., 2020). The polarisation of impressed sources is set perpendicular to the line of measurements, and following the work by Giannakis et al. (2020), soft line sources were used. For RTM, $B(\mathbf{q}, t_{max} - \mathbf{t})$ needs to be interpolated in time to be synchronous to the time step of FDTD (Giannakis et al., 2020). For all the cases presented below, a discretisation step of $\Delta x = \Delta y = 0.001m$ was used, while Δt was calculated by the CFL condition. The FDTD domain was truncated using a PML with a 20-cell thickness. To compensate for the two-way travel time, the velocity of the medium should be set to half of its actual velocity i.e. the permittivity should be set to four times the actual permittivity value.

A second-order in both space and time FDTD-TM mode is implemented to propagate the waves back in time. The TM_z mode, which includes the components E_z, H_x and H_y, was described in Chapter 3. Using a 2D scheme reduces the computational requirements without changing the shapes of the investigated targets. Although a 2D FDTD scheme is used for the migration, all the simulated data to which migration was applied, were acquired using a fully-3D FDTD scheme, as mentioned earlier.

For successful implementation, RTM requires an accurate estimate of the medium permittivity. For a relatively homogeneous background, even an average permittivity for its total extent, can yield an accurate migrated image. In contrast, using a mean permittivity to describe a non-homogeneous material will result in under and overmigrated sections in the image, that originate from the use of low or high EM wave velocities. Instead, performing RTM with a permittivity distribution that captures the variations in permittivity along the B-scan, will produce a migrated image with sufficient accuracy. RTM has the ability to handle velocity lateral variations in contrast to other commonly used migration algorithms in GPR processing that assume a constant velocity throughout the material. Using an accurate velocity structure, the fields are backpropagated until the diffractions collapse back to their origin at $t = 0$.

Here an RTM approach using the ML-based background subtraction/prediction that was developed is proposed, where a permittivity distribution $\epsilon(\mathbf{q})$ is estimated using the deep learning scheme described previously and is assumed to vary only along the direction of the survey $\mathbf{q}$. The predicted background is used to perform background removal on the responses, which is an essential step prior to migration. A flow chart of the process followed is illustrated in Figure 6.21 for both RTM and general use of the proposed ML scheme. To demonstrate the performance of the scheme it is applied on both synthetic and real data.

6.6.1 *RTM-synthetic data*

The proposed processing scheme is now evaluated on two synthetic case studies with varying permittivity along the line of measurements. The first case includes 4 rebars of 8 mm radius buried in a depth of 7 cm. The background layer is modelled with a conductivity $\sigma = 0.001 S/m$ and a linearly increasing permittivity along the direction of scanning, starting from $\epsilon_r = 4$ at the beginning of the profile and advancing to $\epsilon_r = 8.5$ at its end. The half-space included in the model starts at a depth of 12 cm and was modelled with $\epsilon_r = 5$ and $\sigma = 0.001 S/m$. The modelled geometry is illustrated in Figure 6.22a, whereas the corresponding B-scan in Figure 6.23b. The differences in shading for the first layer are used to indicate that the properties of the material are changing along its largest

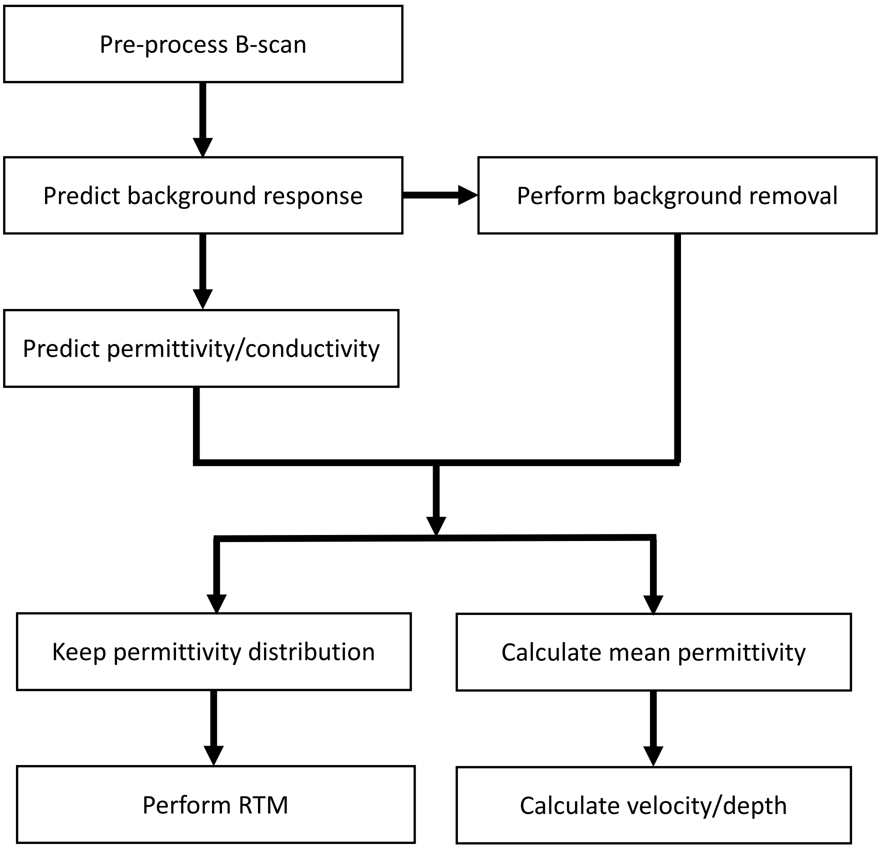


Figure 6.21: Flow chart of the steps followed for the proposed scheme.

dimension. Figure 6.23 shows the true permittivity distribution plotted against the NN predicted permittivities, which are in a very good agreement.

At first SVD was used to filter the background out of the data and migration was performed using a mean permittivity value for the entire B-scan. From Figure 6.23c, it is obvious that with a mean permittivity, the hyperbolas did not collapse and a noisy migrated image was produced. In addition, SVD was not able to remove the background successfully with sufficient parts of it remaining in the data. Due to the mean permittivity used for RTM, the response from the half-space still appears as a response from a dipping target, whereas the true interface is flat as illustrated in Figure 6.22a. In contrast, performing RTM using the ML predicted permittivities, produced a clear migrated image where the hyperbolic events have collapsed and the reflectors have been relocated to their true positions, including the response from the half-space (see Figure 6.23d). Furthermore, the deep learning scheme predicted accurately and removed the background signal from the original data. Finally, a hilbert transform was applied on the migrated signals produced by RTM with ML to acquire the traces envelope with all positive components as shown in Figure 6.23e. Note that the migrated images for both synthetic and real data have been truncated to a depth of 0.3

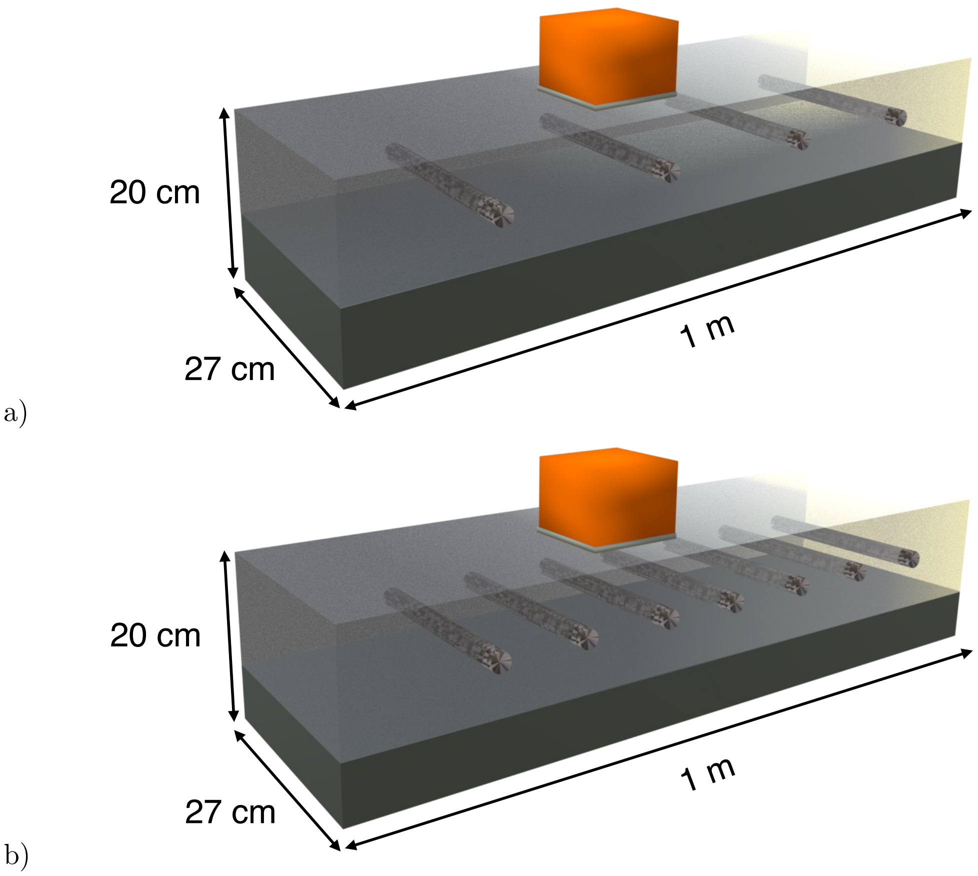


Figure 6.22: Geometry of the synthetic models used for the RTM scheme: a) First synthetic scenario, b) Second synthetic scenario.

m since no significant information exists beyond this depth and in order for the targets to be seen clearer.

The second scenario consists of 7 rebars of 1 cm radius, which are placed at 8 cm depth. In this case, the variation in permittivity of the background is greater than the first case study, starting from $\epsilon_r = 3$ and linearly increasing up to $\epsilon_r = 12$. Similarly to the first example, the conductivity was set to $\sigma = 0.001 S/m$. The half-space starting at 13 cm depth was simulated with $\epsilon_r = 8$ and $\sigma = 0.008 S/m$. The modelled geometry of the second case is shown in Figure 6.22b and the B-scan data in Figure 6.24b. The permittivity variations for the second case study are more pronounced and due to the increasing permittivity and therefore decreasing velocity, the rebar targets appear to be at different depths, as shown in Figure 6.24b, although they are actually at the same depth. The same holds for the half-space response. The predicted permittivity distribution is very close to the true distribution as displayed in Figure 6.24a.

Using a mean permittivity for the total area in RTM, resulted in over and undermigrated sections in the image, where the hyperbolas have been focused only in the sections that had a permittivity value close to the mean value and the interface with the half-space still appears to be dipping. Again, SVD did not remove large part of the background energy as shown in Figure 6.24c. The proposed ML scheme with the permittivity distribution for RTM was able to successfully collapse the events to their true positions even in this case of large permittivity variations across the profile. The final migrated image using the proposed ML scheme is presented in Figure 6.24d and its corresponding Hilbert transform in Figure 6.24e. The results demonstrate that the suggested scheme outperformed in both cases the commonly used RTM scheme with a mean permittivity and SVD filtering and can be applied successfully to datasets from environments with permittivity variations.

6.6.2 *RTM- Real data*

In this section, RTM is performed on two cases of the real data that were collected from the ground floor of the William Rankine building, as described previously. Since the floor is a concrete slab, some small variations in the dielectric properties throughout its extent are expected due to differences in the moisture content. From the first B-scan in Figure 6.25b, it is obvious that there are 4 targets below a flat target. The predicted permittivities along the measurement line are ranging between 4.6-6, as displayed in Figure 6.25a. As expected, the predicted permittivity distribution for the real data is noisy and therefore, a moving-average filter was applied prior migration producing the smooth distribution shown in the Figure. In this survey area, significant variations in permittivity are not encountered and therefore, even a mean permittivity in RTM resulted in an acceptable migrated image as displayed in Figure 6.25c. However, SVD removed partially the flat response resulted in a lower resolution image. The deep learning model removed only the background energy and focused successfully the hyperbolas. The result is a clear migrated image, which is shown in Figure 6.25d, where both the hyperbolic responses have collapsed and also the flat response is clearly visible in the data. Similarly to the synthetic data, Hilbert transform was applied after migration, which is displayed in Figure 6.25e.

In the second case study, the B-scan shows 4 hyperbolic responses around the same depth and a dipping linear reflector at a depth below the hyperbolas, as illustrated in Figure 6.26b. The estimated permittivity values, presented in Figure 6.26a, are varying between 6.5 and 7.2 and similarly to the first case study, the noisy predicted permittivity distribution was smoothed. Similarly with the first case, both schemes were applied on the data, with the SVD removing again partially the dipping response in contrast with the ML scheme which did not affect it, as shown in Figures 6.26c and 6.26d. Since the variation in permittivity

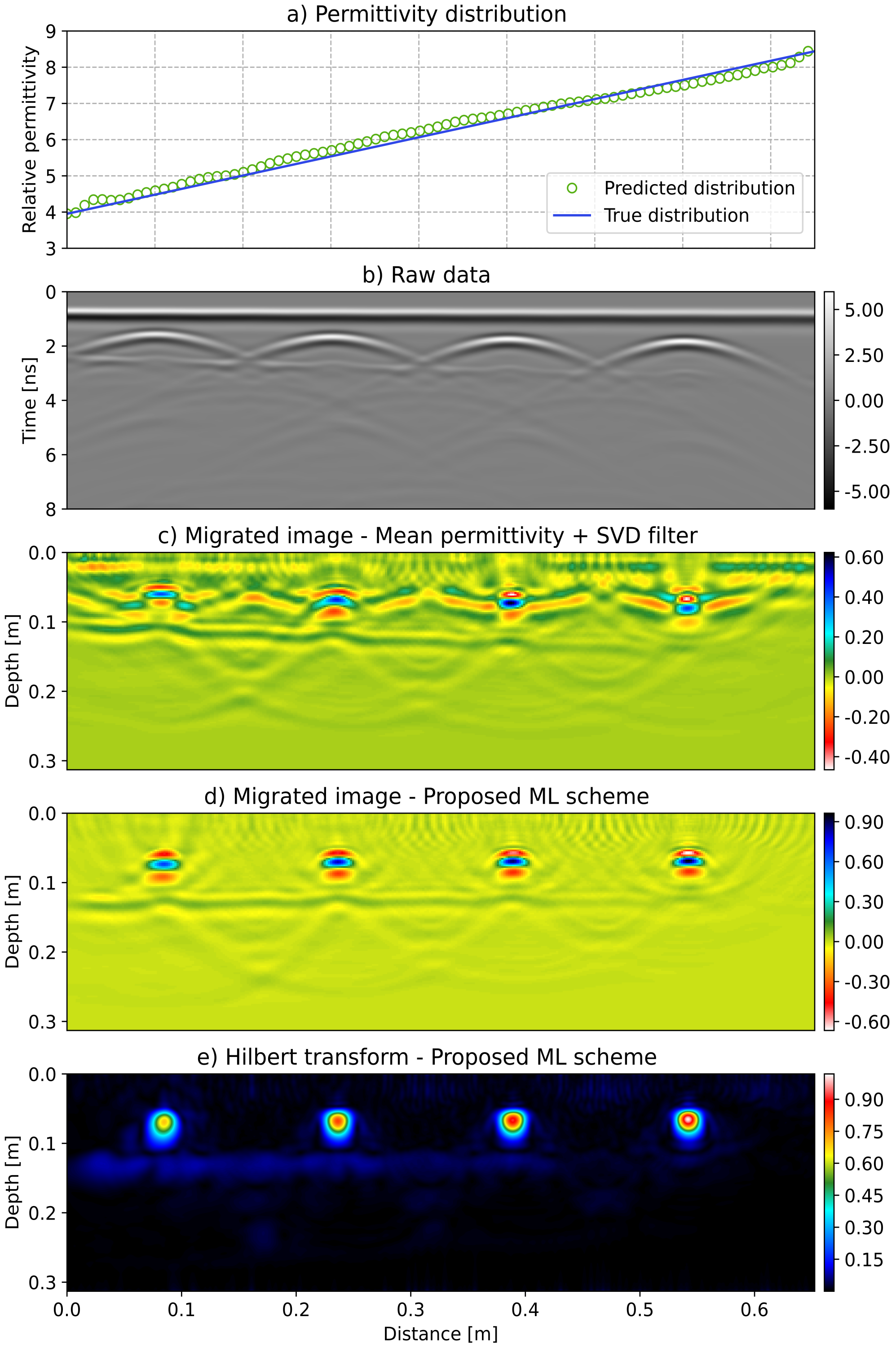


Figure 6.23: RTM of synthetic scenario 1: a) Predicted versus true permittivity distribution, b) Raw data, c) RTM using a mean permittivity and an SVD filter, d) RTM using the proposed ML scheme and e) Hilbert transform after RTM using the proposed ML scheme.

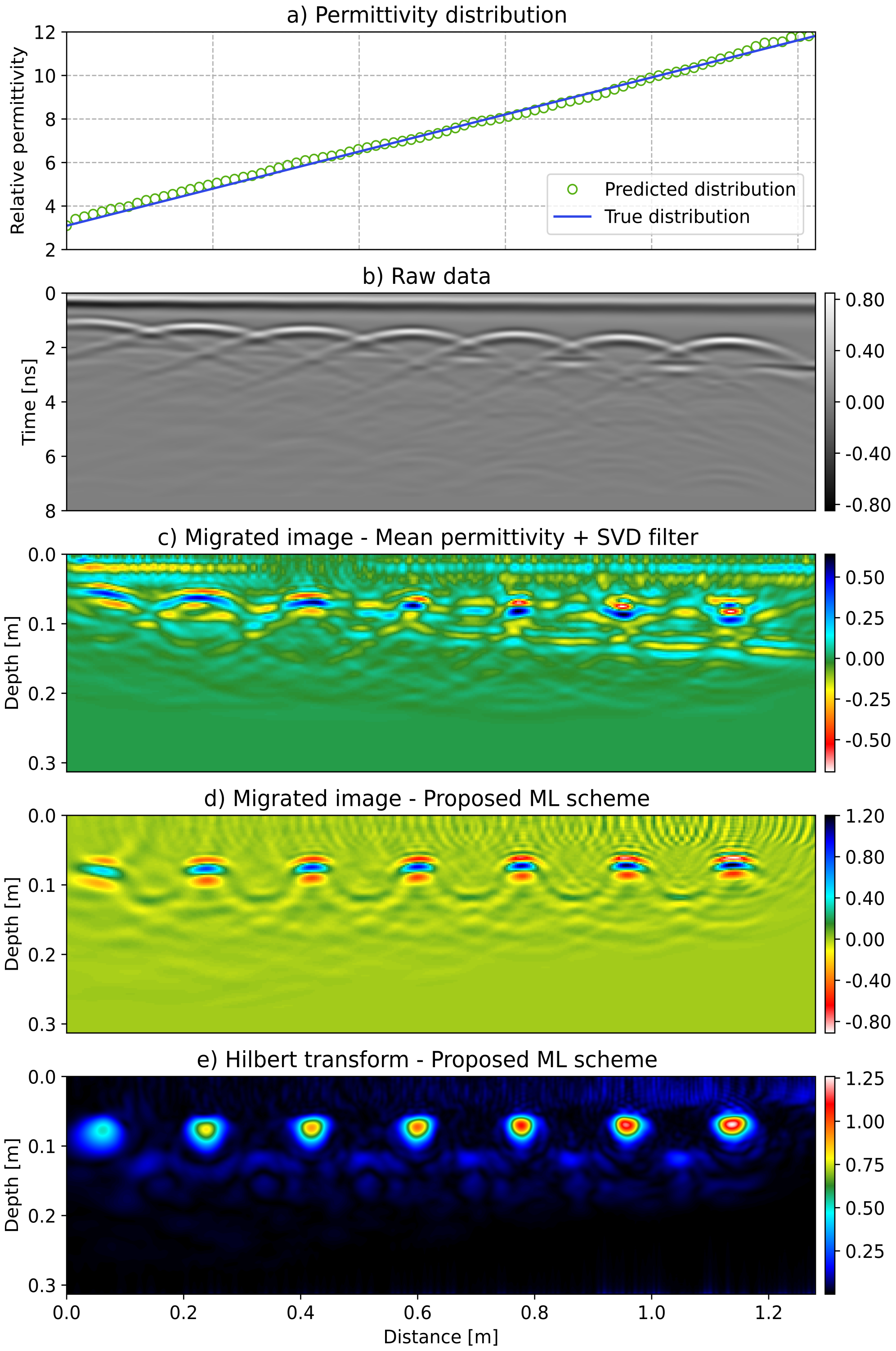


Figure 6.24: RTM of synthetic scenario 2: a) Predicted versus true permittivity distribution, b) Raw data, c) RTM using a mean permittivity and an SVD filter, d) RTM using the proposed ML scheme and e) Hilbert transform after RTM using the proposed ML scheme.

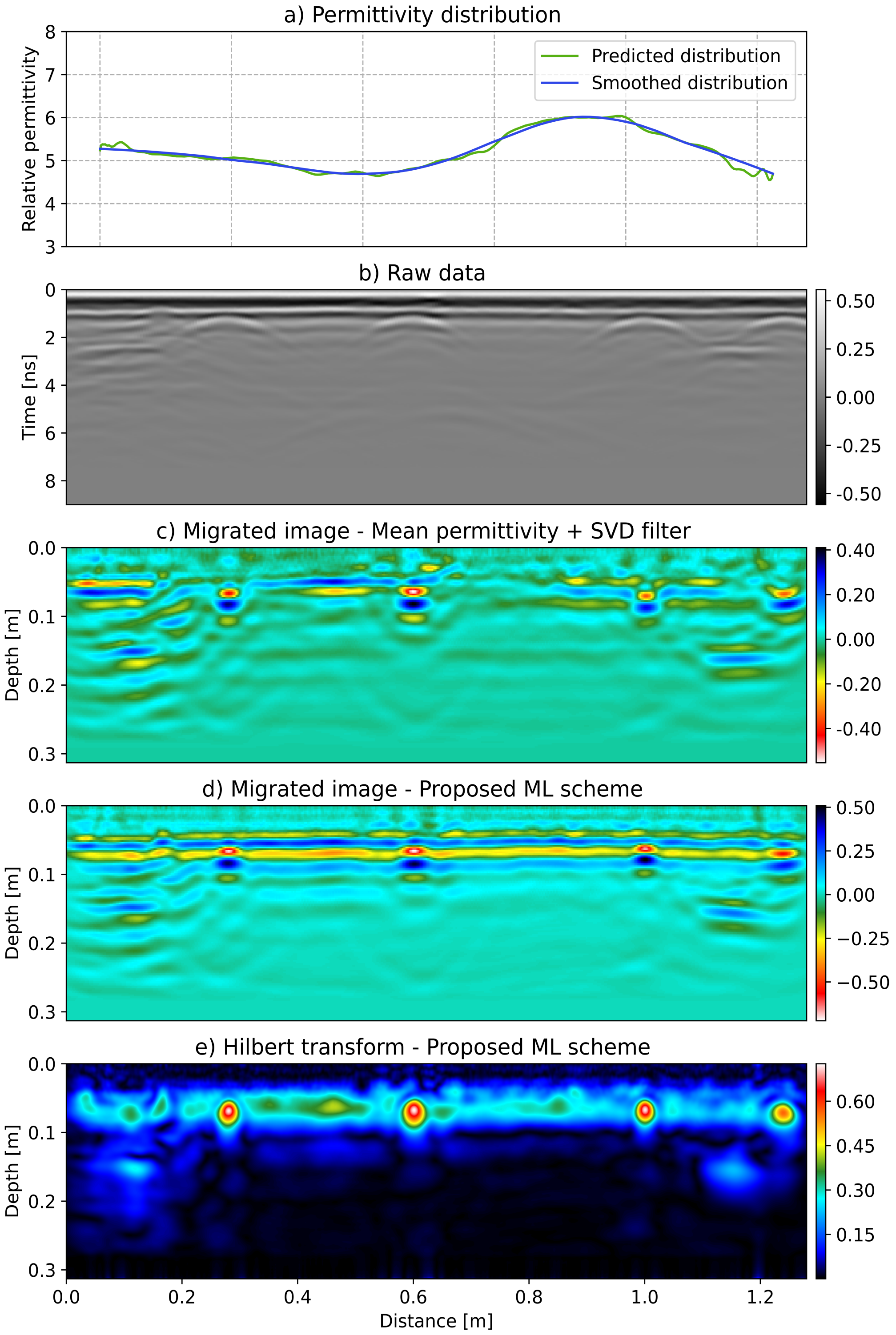


Figure 6.25: RTM of real case study 1: a) Predicted permittivity distribution, b) Raw data, c) RTM using a mean permittivity and an SVD filter, d) RTM using the proposed ML scheme and e) Hilbert transform after RTM using the proposed ML scheme.

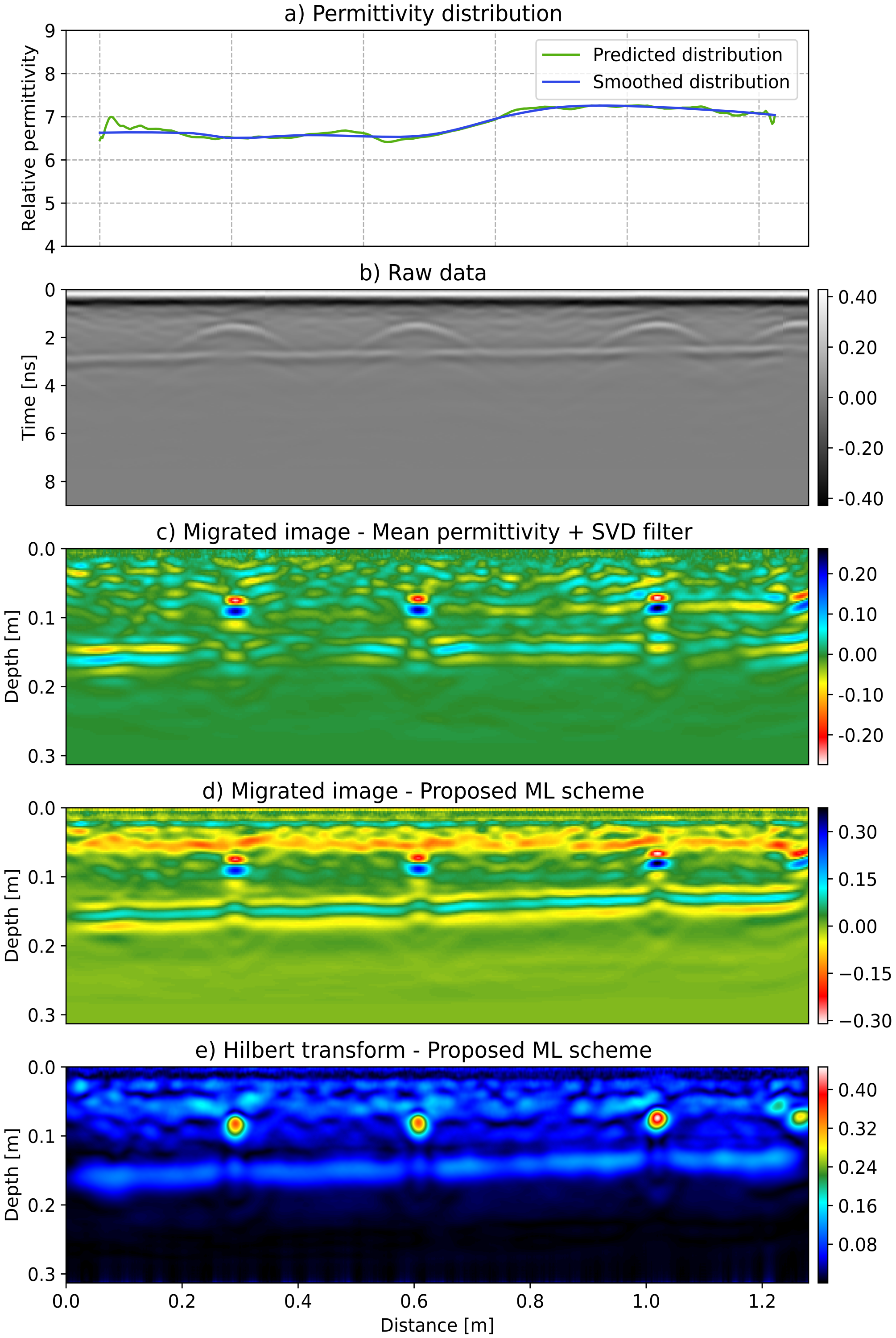


Figure 6.26: RTM of real case study 2: a) Predicted permittivity distribution, b) Raw data, c) RTM using a mean permittivity and an SVD filter, d) RTM using the proposed ML scheme and e) Hilbert transform after RTM using the proposed ML scheme.

was not significant again, RTM performed similarly on both cases. The Hilbert transform of the final migrated image is shown in 6.26e.

The above results on both synthetic and real data demonstrate the accuracy of the proposed ML scheme to perform RTM and its superiority compared to commonly used RTM with SVD and a mean permittivity. The suggested scheme removes the background response successfully, does not remove other flat targets and can produce high resolution images even in complex environments with large permittivity variations, therefore enhancing the interpretation of GPR data.

7

ML-BASED FORWARD SOLVER

In this chapter, a deep learning scheme is presented, that is used to predict B-scan responses from reinforced concrete slabs in almost real time given certain input parameters and act as a forward solver. The deep learning forward solver is used as part of a full waveform inversion algorithm to acquire estimates of the depth and diameter of rebars, as well as to characterise the concrete. The proposed scheme is evaluated using both synthetic and real data and finally compared with a conventional FDTD forward solver.

7.1 INTRODUCTION

In Chapter 3, a detailed description of the FDTD forward solver was presented. As mentioned earlier, numerical modelling is extensively used to understand the EM wave behavior and enhance the interpretation of real data. In addition to that, a forward solver is an essential component of full-waveform inversion (FWI), which is an inverse problem used to estimate key parameters of the earth model that generated the GPR data. To solve an inverse problem, solving the forward problem multiple times is required. FDTD is a computationally expensive algorithm and therefore leads to large computational times when used as part of an FWI scheme. This makes FWI not applicable in practice in the absence of high performance computing (HPC). Therefore, finding a fast forward solver is essential in order to be able to use FWI in practice.

An attractive approach would be to utilise ML to create a forward solver that can provide real time predictions, given certain earth model parameters as an input. To train the ML model, a large and well-labeled training set with A-scans of diverse scenarios would be needed. After training, the ML model would be able to produce 3D forward solutions instantly but will be applicable only to specific scenarios that were part of the training. Therefore, in this section a deep learning forward solver is presented that generates responses for concrete slab scenarios. The ML solver is used as a component of FWI to acquire estimates of the depth and diameter of rebars that exist in concrete slabs and characterize the concrete itself. Although the ML is tuned for concrete slab scenarios, it can be trained to include other cases as well. Before discussing the development of the ML-based forward solver, a description of the FWI method is presented.

7.2 FULL-WAVEFORM INVERSION

The inverse problem is the opposite case of the forward problem. In the inverse case, the GPR responses are known, whereas the earth model that generated these responses is unknown. This is the most common case given that the field GPR data collected are known and is sought to infer the dielectric properties of the subsurface from the measured data. Inversion problems are usually formulated as iterative optimisation problems and can be implemented using either local search algorithms such as gradient-based methods or global optimisers such as GA and PSO, mentioned earlier. Regardless of the method used, a set of parameters is sought that minimises as possible the misfit between the observed and synthetic responses. Inverse problems are ill-posed problems and thus, the problem of non-uniqueness arises, meaning that different sets of parameters can produce the same or similar responses. Therefore, constraints based on a priori information are needed in order to acquire a realistic model and also for the inferred parameters to be closer to some preferred model.

FWI, as the name suggests is an inversion method that utilises entire waveform information in contrast to other methods, such as ray-based methods that use only a portion of the responses, i.e., the first arrival and the maximum amplitude of the first cycle. It was originally developed for seismic data (Tarantola, 1984a,b) and is usually implemented using a gradient-based method along with an FDTD solver, whereas a global optimiser is used less often due to the extensive computational costs.

A typical gradient-based FWI algorithm starts with an initial guess of the parameters which is consecutively corrected through the iterations based on the data misfit. At each iteration the forward problem is solved one or usually more times to acquire the GPR synthetic responses, corresponding to the current model parameters, which are subsequently compared with the real responses to calculate the misfit and update these parameters. The process is advanced until an acceptable misfit value is reached. This type of methods is more suitable to large scale problems with many parameters due to less computational times compared to global optimisers. However, these methods are highly dependent on the choice of the initial model and suffer from the local minimum problem, from which they cannot escape to reach a better or global minimum.

Global optimisers do not suffer from the local minima problem and are not dependent on an initial model. At each iteration of FWI, multiple possible solutions are generated, for which the forward problem must be solved. If the number of model parameters sought is increased, the number of possible solutions should be increased accordingly. Therefore global optimisers are very computationally expensive algorithms and lead to increased execution times as the number of parameters increases.

Usually FWI is performed without including a model of the system used to collect the GPR data but using theoretical sources in the simulations. Therefore, it is required to calculate an effective wavelet in order to estimate the unknown source wavelet. For off-ground surveys this can be easily obtained using measurements over a steel plate. However, for surface GPR data, where the antenna is coupled with the ground and the wavelet is changing with the medium properties, it is required to calculate an effective wavelet for each measurement position. In addition, the effective wavelet needs to be optimised simultaneously with the medium dielectric properties since these are coupled together and the properties affect the resultant wavelet (Busch et al., 2014).

A detailed overview of FWI for GPR is provided by van der Kruk et al. (2018) along with applications. A number of FWI applications for crosshole GPR data can be found in the literature (Gueting et al., 2015; Klotzsche et al., 2010, 2013). Ernst et al. (2007) implemented FWI with a conjugate gradient technique and a 2D FDTD forward solver to invert first for the permittivity while keeping conductivity fixed and then invert for conductivity while keeping the permittivity fixed in a stepwise manner in order to avoid sensitivity issues between these two properties. This way high-resolution permittivity and conductivity distributions between boreholes were produced. The scheme was applied to a number of synthetic data, both simple and complex, to evaluate its performance and address its limitations. In all cases, it was shown that FWI performed better than conventional ray methods.

An FWI algorithm that deals with the sensitivity problem and simultaneously updates and solves for the permittivity and conductivity distributions was developed by Meles et al. (2010) using again a 2D FDTD scheme due to limited available computational resources at that time. The proposed FWI algorithm was applied to a number of synthetic 2D models with a realistic permittivity and conductivity distributions and was used to invert both crosshole and surface-to-borehole GPR data. The tests showed that the scheme improved the FWI results compared with previous studies which were based on a cascaded updating of the inversion parameters. This vectorial FWI described above was applied by Klotzsche et al. (2010) to invert crosshole GPR data from a gravel aquifer in Switzerland. To acquire the measurements, MALÅ 250 MHz antennas were used with limited number of transmitter positions and large number of receiver positions. To apply the 2D FWI approach on real 3D data, a 3D to 2D transformation was performed. The results were compared with a ray-based approach, showing a higher resolution inverted images. The same FWI scheme was used by Klotzsche et al. (2013) to invert six crosshole GPR cross-sections of four boreholes arranged in a squared configuration, along with a novel amplitude analysis. Low-velocity waveguides were detected and interpret as high-porosity zones. The inversion results were compared with porosity values inferred from Neutron–Neutron data and permeability logs of the same boreholes. The vectorial FWI was also applied by Gueting et al. (2015) on crosshole GPR data to characterise facies heterogeneity in an aquifer. First, FWI was used to obtain the subsurface permittivity and conductivity and afterwards the results were compared with other investigation tools.

A frequency domain 3D FWI scheme is employed by Busch et al. (2012) on surface GPR data. FWI was implemented using a combined global and local optimisation approach. The scheme was used on synthetic and measured CMP data to acquire permittivity and conductivity estimates over a horizontal layered subsurface. It was shown that an effective wavelet needs to be measured at each position and also that the wavelet must be simultaneously updated with the properties of the medium since it is dependent on the medium. One application of FWI on surface GPR data is presented by Liu et al. (2018), where FWI along with a fully 3D FDTD solver and a theoretical source was used to estimate the radius of cylindrical targets, the medium dielectric properties and the effective source wavelet. The scheme was tested with both synthetic and real GPR data.

A crosshole/borehole-to-surface FWI algorithm that is constrained by a compressed representation of the fractal distribution of the water existing in the subsurface is presented by Giannakis et al. (2022) for reconstructing the water distribution between boreholes. Imposing the fractal constraints, the FWI execution time was decreased significantly compared to a traditional FWI scheme.

The first development of a fast ML forward solver for GPR data of concrete slabs was introduced in Giannakis et al. (2019b, 2018). The scheme utilises a dense NN along with principal component analysis (PCA) to predict single A-scan responses for which the target is buried exactly below the antenna center, given certain model parameters as input. The model was trained to predict concrete slab scenarios for rebars with various diameters and different depths in a homogeneous concrete slab with varying dielectric properties. This forward solver was used as part of FWI to obtain estimates of the water content in concrete, depth and diameter of rebars. The scheme included a model of a 1500 MHz transducer from GSSI (Warren and Giannopoulos, 2011; Giannakis et al., 2019a) in the synthetic training data and therefore, can be used for real GPR data. FWI using the ML model as a forward solver was validated with both modelled and real data demonstrating the accuracy and validity of the scheme.

The scheme presented here follows the work by Giannakis et al. (2019b) and is extended to make predictions for different rebar lateral positions relative to the antenna center, meaning that it does not require for the transducer to be directly above the target in contrast to Giannakis et al. (2019b), which can only predict A-scans on top of rebars. Therefore, entire B-scans for reinforced concrete slab scenarios can be now generated fast in almost real time using the proposed ML scheme, reducing significantly the overall FWI computational costs. The ML-based forward solver for reinforced concrete slab scenarios is analytically described in the following sections.

7.3 TRAINING SET

To form the training set, numerous data from diverse concrete slab scenarios are needed. The training was performed again using FDTD synthetic data only, which include the numerical model of the GSSI 2 GHz antenna. The simulations consist of different cases of concrete as background medium with a single rebar buried in the concrete at different depths and positions.

The dispersive behavior of concrete can be described using an extended Debye model, given by Equation 3.67 presented in Chapter 3. Values of the extended Debye model parameters for concrete with different water contents by volume are provided in Table 7.1 and were obtained in Bourdi et al. (2012) by fitting experimental data. As shown in the Table, the Debye parameters for concrete can be represented using a single moisture content value, reducing the number of parameters from 4 to 1. This helps avoid the instabilities that arise in FWI due to the sensitivity differences between $\epsilon_s, \epsilon_\infty, \sigma$ and t_0 since these 4 parameters span vastly different ranges, and also accelerates the training process. To acquire a large set of diverse scenarios and capture the variations in water content and their effect, cubic interpolation was used to interpolate between the values shown in Table 7.1 and express different water content values with respect to the dielectric properties of concrete.

The training set consists of 4000 different scenarios drawn from a uniform distribution of the following parameters

- $WC \in [0.2 - 10]\%$
- $R \in [1, 6]$ cm
- $D \in [3, 25]$ cm
- $X \in [-11, 11]$ cm

where, WC is the water content, R is the diameter of the rebar, D is the burial depth and X is the horizontal distance relative to the center of the antenna. The main axis of the antenna (E-field) was parallel to the main axis of the rebars in all of the simulations. The materials used in the simulations are considered to be non-magnetic, while the rebars are simulated as PECs. The domain size

Water Fraction (%)	ϵ_s	ϵ_∞	t_0 (ns)	σ (S/m)
12	12.84	7.42	0.611	20.6×10^{-3}
9.3	11.19	7.20	0.73	23×10^{-3}
6.2	9.14	5.93	0.80	6.7×10^{-3}
5.5	8.63	6.02	1.00	5.15×10^{-3}
2.8	6.75	5.50	2.28	2.03×10^{-3}
0.2	4.8	4.50	0.82	6.06×10^{-4}

Table 7.1: Extended Debye properties of concrete Bourdi et al. (2012).

is set to 300 × 300 × 400 mm with 1 mm step size and a time window of 8 ns, resulting in 4156 time steps from CFL condition. Although the water content values were used to represent concrete in the ML scheme, the corresponding Debye properties were used to simulate concrete and generate the training data in FDTD. Generating this dataset required a running time of 1 day executed on 4 NVIDIA TITAN RTX 24GB GPUs and using gprMax.

The 4 parameters, WC, R, D and X are the inputs to the network, and their corresponding outputs are the simulated A-scans. Therefore, the training set consists of the input data $\mathbf{X} \in \mathbb{R}^{4\times m}$ with their corresponding labels $\mathbf{Y} \in \mathbb{R}^{n\times m}$, where $m = 4000$ is the number of traces and n is the number of samples per trace and $\mathbf{Y}_i = [WC_i, R_i, D_i, X_i]$ is a vector that contains the input parameters for the ith trace. Before training, both input and output data are subjected to pre-processing as described below.

7.4 DATA PRE-PROCESSING

Because the input parameters lie in considerable different ranges, parameters with larger values, such as the permittivity and the water content, will intrinsically influence the ML model more than the rest of the parameters, due to their larger values. To handle this, the 4 input parameters are normalised to the maximum value of their specified ranges. The same pre-processing steps of normalizing, resampling and shifting the data that were applied for the backround removal case, were applied on the output A-scans for the ML-based forward solver, as well. However, the dimensionality reduction in this case was implemented using singular value decomposition (SVD) as described below, which proved to perform better in this case compared to PCA decomposition. To demonstrate this, 4 different A-scans have been randomly selected from the training set and compressed using both SVD and PCA with the same number of components. Figure 7.1 compares

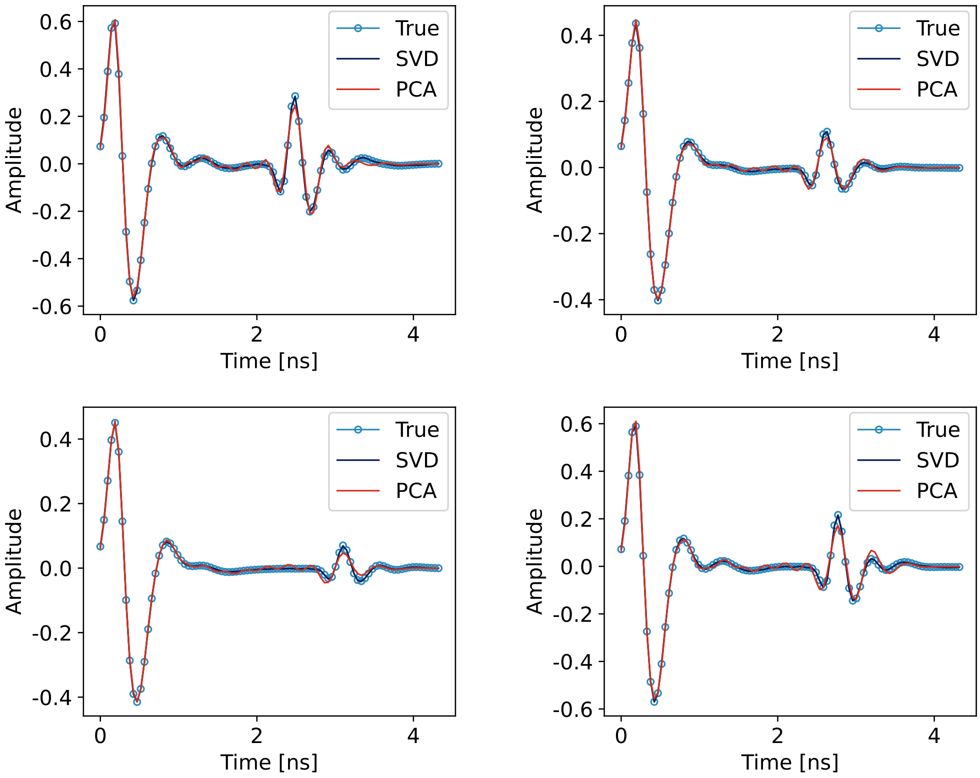


Figure 7.1: Comparison of SVD and PCA representations of different randomly selected A-scans from the training set. For both PCA SVD and PCA the same number of components was used.

the true A-scans with their compressed representations, where it is obvious that the SVD representation approximates the true A-scans more accurately than PCA.

7.4.1 *Singular Value Decomposition for Dimensionality Reduction*

Similarly to PCA, SVD has been used by the GPR community for direct wave removal and clutter reduction (Cagnoli and Ulrych, 2001; Liu et al., 2017), while in ML schemes is commonly used as a data compression method (Brunton and Kutz, 2019). PCA is a method similar to SVD, and in fact, the principal components can also be obtained via SVD. SVD factorises a $m \times n$ matrix $\mathbf{A}$ into the product of three matrices

$$A = U\Lambda V^T \tag{7.1}$$

where $\mathbf{U}$ is define to be an $m \times m$ matrix, Λ an $m \times n$ matrix and $\mathbf{V}$ an $n \times n$ matrix. The matrices $\mathbf{U}$ and $\mathbf{V}$ are both orthogonal matrices containing the

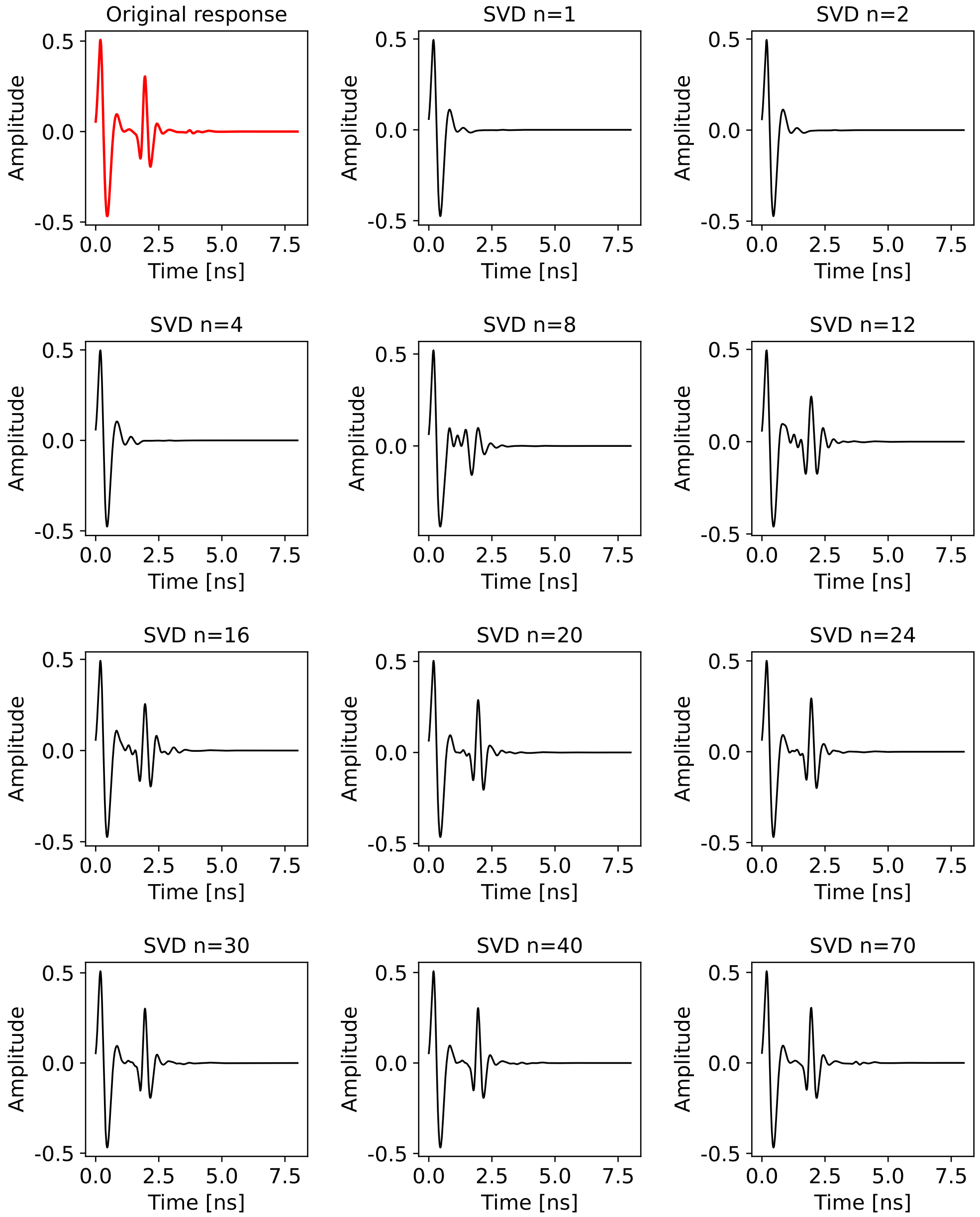


Figure 7.2: SVD representation of an A-scan for different number of components.

left-singular vectors and right-singular vectors, respectively, while Λ is a diagonal matrix containing the singular values. Alternatively, it can be written as

$$A = \sum_{i=1}^{n} \lambda_i u_i v_i \tag{7.2}$$

The columns of U are hierarchically arranged so that the first column explains most of the variance in the data, the second column, the second best variance and so on, similarly to PCA described previously. The singular values in Λ are arranged in a descending order and $\lambda_i \geq 0$ for all i. The importance of each column

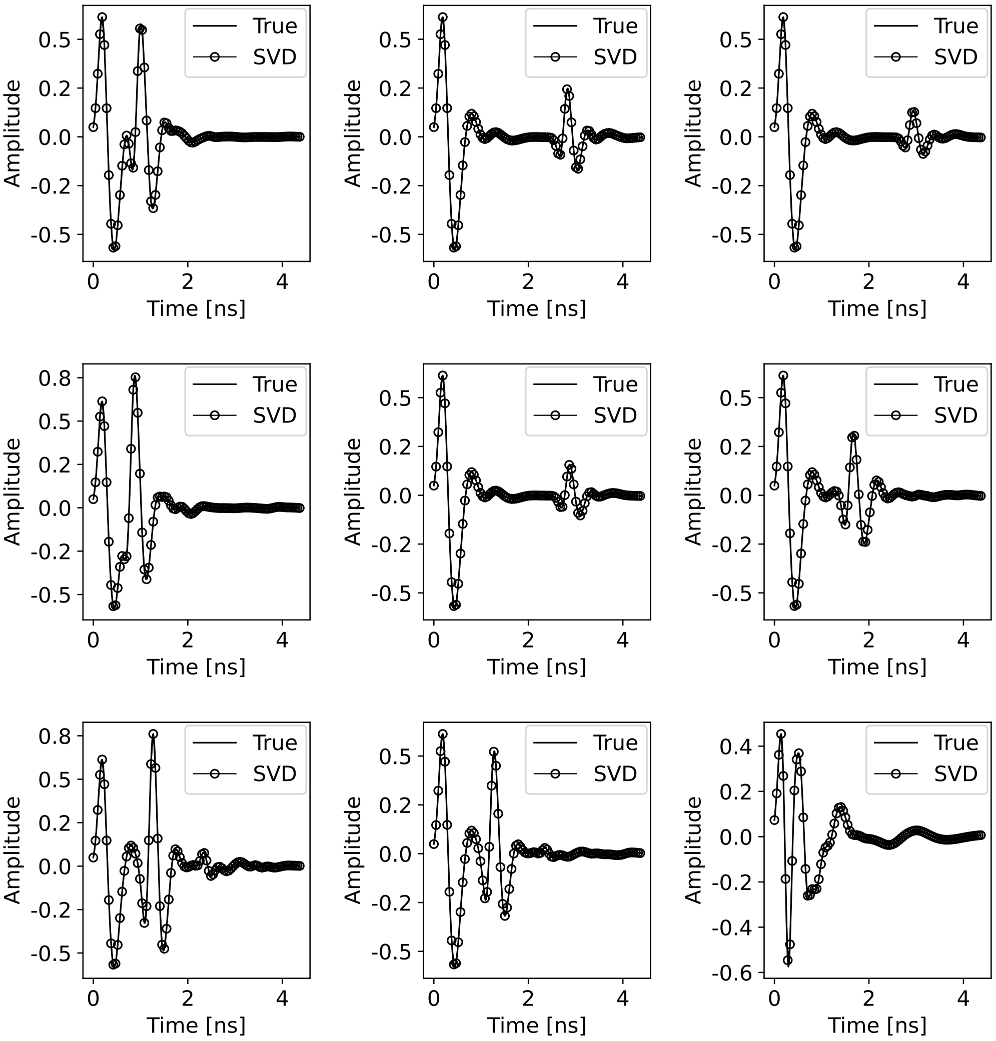


Figure 7.3: Reconstructed SVD A-scans versus true A-scans randomly selected from the training set.

of U or V is given by their corresponding λ_i value. By discarding the components of U and V that correspond to very small λ_i values, we can approximate matrix A using fewer components with negligible loss of information. 70 SVD components found to be sufficient to capture the variability in the A-scan data. Thus, the compressed representation $\mathbf{A} \in \mathbb{R}^{70\times m}$ of the A-scans are finally used as the output labels of the network. Similarly with the PCA, the SVD decomposition was implemented using the scikit-learn module (Pedregosa et al., 2011). Figure 7.2 shows a reconstruction of a training set A-scan using different number of SVD components. The original response is shown on the first subfigure, plotted with a red line. Similarly to PCA, by using more SVD components, the reconstructed signal resembles more the original response, whereas using the chosen number of 70 components, the reconstructed signal is indistinguishable from the true case, as shown in the last subfigure. A number of SVD reconstructed A-scans

	Lower limit	**Upper limit**
Number of layers	2	8
Number of nodes	16	512
Learning rate	0.0001	0.1
	Options	
Optimiser	SGD RMSprop Adagrad Adadelta Adam	
Activation function	ReLU Tanh Sigmoid	

Table 7.2: Ranges and options of hyperparameters used in grid search for the ML forward model.

from the training set plotted versus the true responses are displayed in Figure 7.3, demonstrating that 70 SVD components are sufficient to acquire accurate approximations. PCA of the data was also investigated, however SVD resulted in a better performance of the network and chosen as the optimal dimensionality reduction technique in this case.

7.5 DEEP LEARNING SCHEME

Similarly to the ML background prediction scheme, after creating and preprocessing the A-scans, the data set is shuffled and split into training, test and validation sets with portions of 80%, 10% and 10% each.

Grid search was performed to determine the optimal structure of the network, where the number of layers, number of nodes, the learning rate and the activation functions for each layer were optimised. The final architecture of the network is shown in Figure 7.4, whereas the ranges and possible options for each hyperparameter used in grid search are given in Table 7.2. A feed-forward fully-connected network consisting of 5 hidden layers with 300 nodes at each layer was found to be the optimal architecture for this problem. The input layer contains 4 nodes for the 4 parameters, whereas the output layer consists of 70 nodes which represent the SVD compressed form of an A-scan. ReLU is used

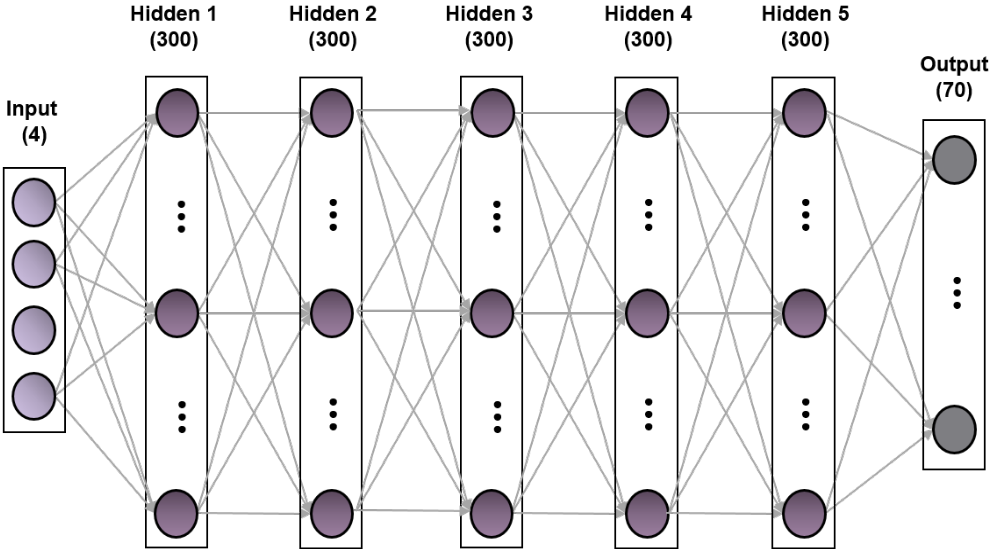


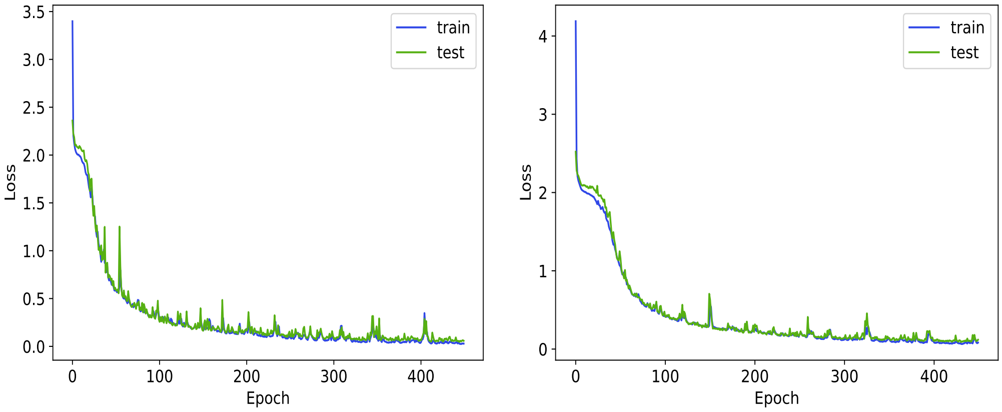
Figure 7.4: Neural network architecture for forward solver.

as the activation function of all layers, expect of the output layer, which uses a linear activation function since it is again a regression problem. The loss function that is minimised with Adam optimiser is the mse between the predicted data and the labels. The learning rate of the model is $a = 0.001$ with a decay of 1e-7 at each iteration.

In this case ensemble averaging is utilised, where 40 distinct ML models are trained. Each ML model is trained separately from the rest, an output is predicted from each model and all the outputs are averaged to provide the final result. Using multiple models, instead of a single, resulted in a better accuracy, prevented overfitting and helped the network to generalise. Each ML model was trained for 500 epochs. The final ML model given an input vector $\mathbf{X}_i$ with the 4 parameters (R, D, X, WC) predicts its SVD compressed response $\mathbf{A}_i$ as an output, which is subsequently decompressed and transformed back to time domain to obtain the response as an A-scan $\mathbf{Y}_i$. In contrast to Giannakis et al. (2019b), the current ML scheme incorporates the X distance from the antenna center, giving the ability to generate entire B-Scans over an investigated rebar.

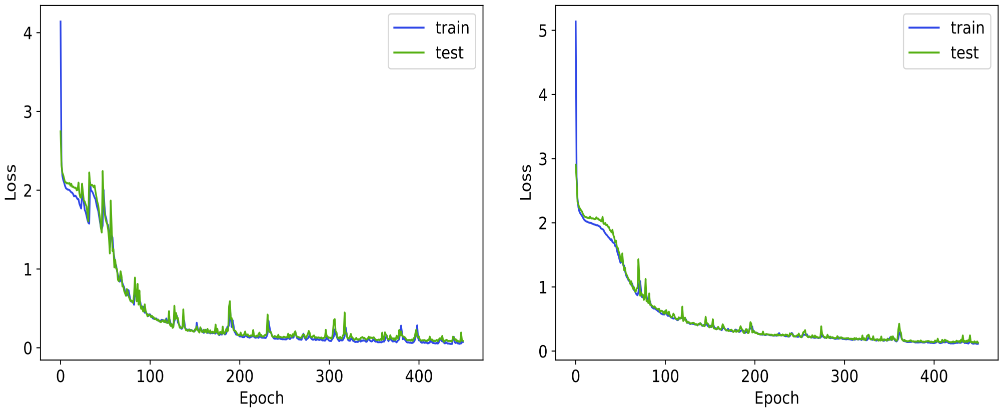
7.6 RESULTS

The final ML model resulted in an mse error of 4.64e-5 for the training set and 6.1e-4 for the test set. The low testing errors demonstrate that the model has not overfitted the training set and can generalize to new unseen cases. Figure 7.5 shows the training and test set loss values plotted per epoch for 4 different ML models used in the ensemble averaging. It is shown that all the models converge similarly towards a low loss for both the training and test set and thus,



(a) ML model 1

(b) ML model 10



(c) ML model 20

(d) ML model 35

Figure 7.5: Train and test set loss values per epoch for 4 of the models used in the ensemble averaging.

demonstrate their ability to generalise. The spikes observed in the loss curves are an effect of using mini-batch gradient descent in Adam optimizer, where the weights are updated by “looking” at one batch of the training data at a time, with some batches leading to weights that minimise the error, whereas other batches might include training examples that cannot be captured accurately and divergence the weights of the ML model resulting to an increase in the error. To evaluate the accuracy of the forward solver, the scheme is applied to various synthetic and real unseen scenarios that were not included in the training process and as part of a FWI framework used to obtain estimates of key properties. FWI is also implemented using an FDTD forward solver and the performance is compared with the results of FWI using the ML forward solver in order to demonstrate its similarity with an FDTD solver, as well as its limitations.

7.6.1 *Forward modelling for synthetic data*

Part of the synthetic data was allocated for evaluating the accuracy of the A-scan predictions and the ability of the ML model to generalise. Fifteen different cases were randomly chosen from the test set, which are shown in Figure 7.6, and compared with the ground truth. The 4 input model parameters used for each A-scan are presented in Table 7.3, ordered by row. From the table is obvious that the selected A-scans capture a wide variety of concrete slab scenarios from dry to wet concrete conditions and different rebar properties. In all scenarios, the ML predictions are almost indistinguishable from the FDTD traces. To support this, the cross-correlation between the true and predicted A-scans was computed, which resulted in a highest correlation value between 98.7% - 99.8% for all cases. It is apparent that the deep learning forward solver predicts successfully both the waveform shape and the arrival times of the responses for different water content, depth and diameter of rebars, even in cases where the targets are not buried directly below the transducer, demonstrating its ability of generating entire B-scans. Given sufficient training data the scheme was able to capture the pattern and the complexity of the EM responses and predict the traces with almost the same accuracy as the FDTD method for the specified scenario it was trained for.

7.6.2 *FWI-synthetic data*

The deep learning model is finally coupled with an FWI scheme to infer estimates of the concrete water content, the depth and diameter of rebars. Initially, the ML-FWI scheme is evaluated using synthetic data to determine its limitations and capabilities regarding accuracy, resolution and speed. For this reason, a number of B-scans for diverse scenarios were generated using FDTD, with two case studies shown here. The two case studies represent a reinforced concrete slab with high and low water content, respectively and are used to validate the scheme's performance in both conditions. The 3D geometry of the two models is displayed in Figures 7.7 and 7.8, respectively, while their corresponding B-scans are illustrated in Figure 7.9. The first model includes two rebars with diameter $R = 1$ cm buried at the same depth $D = 5.8$ cm, while concrete was simulated with a water content of 9%. The size of the slab was set to be $75 \times 30 \times 20$ cm. For the inversion, a total of 10 traces with 1 cm spacing were used for each rebar, which were chosen relative to the apex of each hyperbola, as demonstrated in Figure 7.9b. For the second model, the concrete slab was simulated with a $WC = 3.5\%$ and a single rebar buried at $D = 7.8$ cm and $R = 3$ cm. The total B-scan and the selected 10 traces used for FWI can be seen in subfigures c)

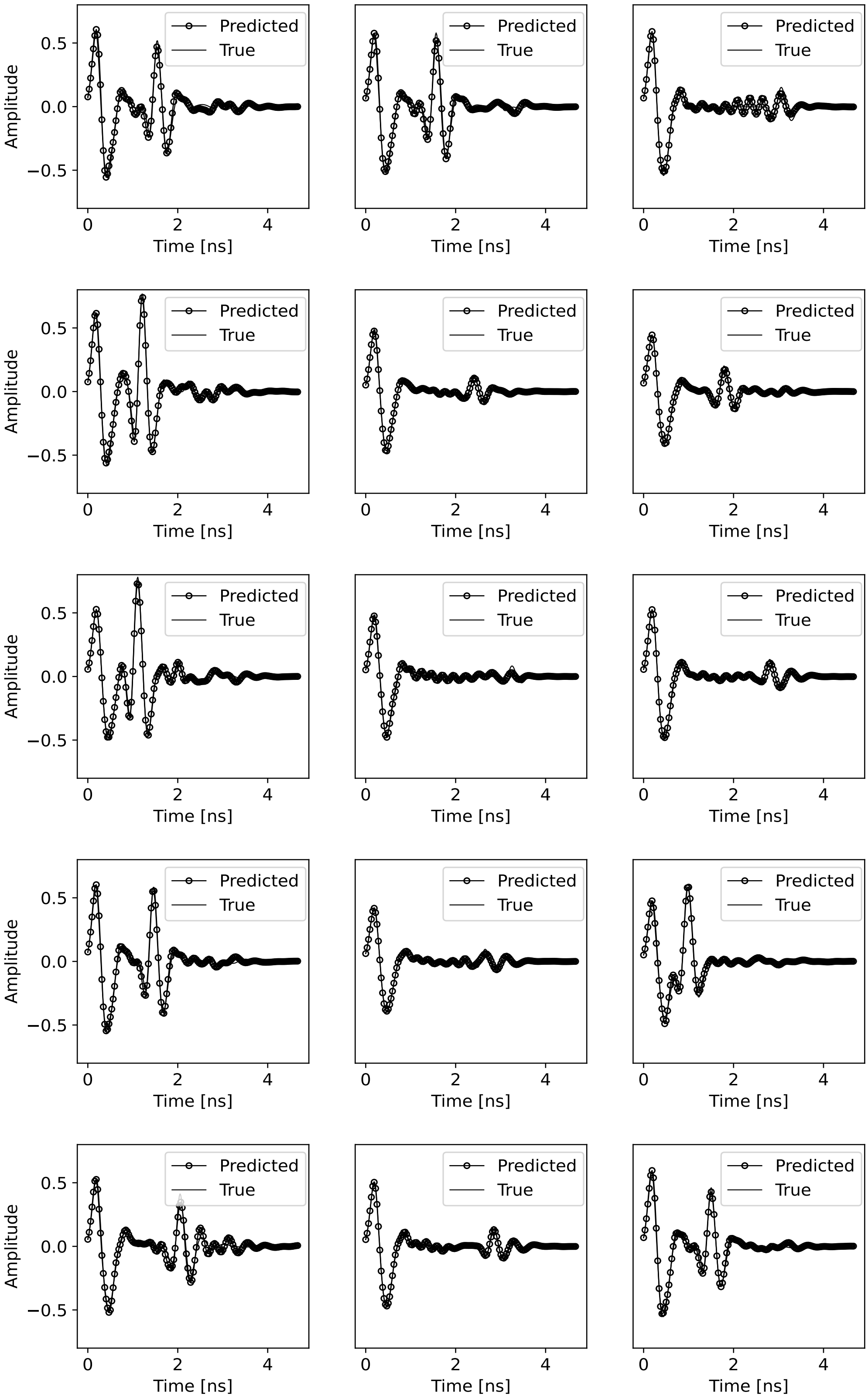


Figure 7.6: Predicted responses randomly chosen from the test set compared with the true responses.

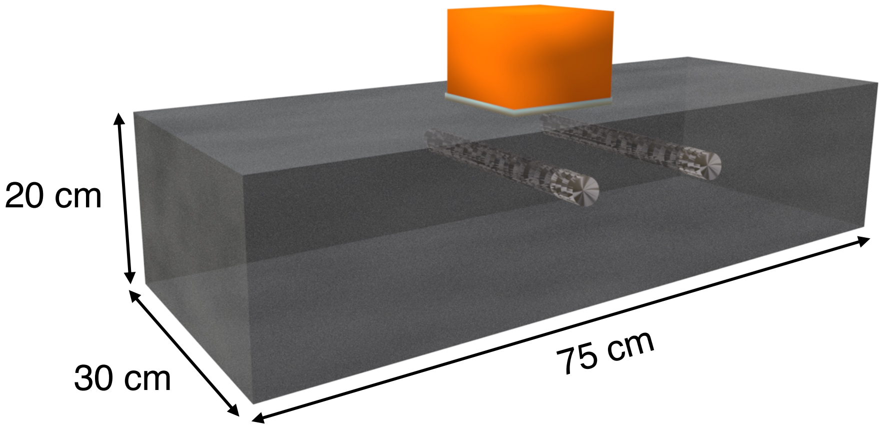


Figure 7.7: Modelled geometry of the first synthetic B-scan used for FWI.

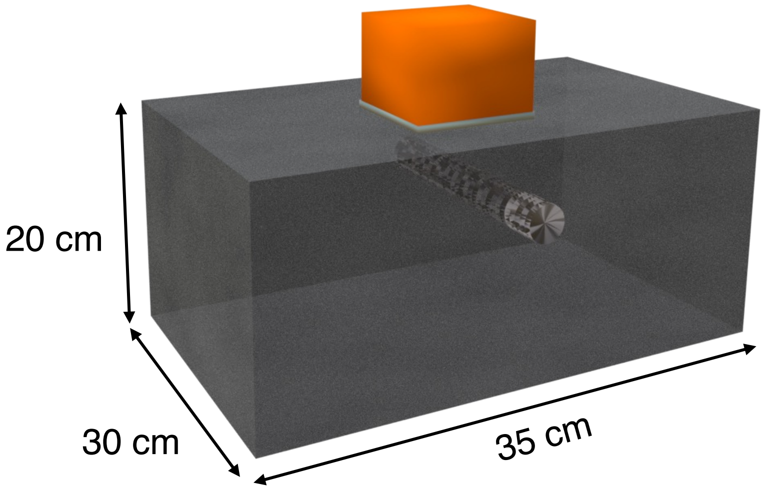


Figure 7.8: Modelled geometry of the second synthetic B-scan used for FWI.

and d) of Figure 7.9. The size of the modelled slab was set to 35 × 30 × 20 cm. Although 10 traces were used in the above cases, inversion can be performed successfully with even less number of traces depending on the application.

To perform the inversion, PSO was implemented, which was analytically described in Chapter 4. A global optimiser was chosen instead of a traditional gradient-based optimiser, in order to avoid local minima and the dependency on the initial model chosen. Although global optimisers are computationally intense algorithms, using ML as a forward model, that provides A-scans in real time, significantly reduces the execution time and makes FWI with PSO feasible even in commercial computers. For the PSO scheme, a number of particles, i.e. solutions, of 60 was chosen with a total of 70 iterations performed. The number of particles and iterations can be adjusted depending on the number of parameters sought and the problem being investigated. In general, with more parameters

A-scan	WC(%)	D (cm)	R (cm)	X (cm)
1	1.13	0.191	0.017	0.010
2	2.99	0.189	0.018	0.030
3	1.96	0.269	0.017	0.090
4	0.42	0.177	0.023	0.0
5	6.85	0.223	0.009	0.0
6	4.87	0.142	0.028	-0.010
7	5.11	0.154	0.016	0.015
8	6.95	0.285	0.022	0.0
9	5.03	0.247	0.006	0.022
10	0.97	0.189	0.019	0.045
11	10.49	0.236	0.019	0.01
12	7.31	0.134	0.006	0.011
13	4.64	0.225	0.028	0.008
14	5.93	0.270	0.027	-0.014
15	1.45	0.180	0.009	-0.005

Table 7.3: Model parameters used to generate each A-scan in Figure 7.6.

being sought a greater number of particles is required. The loss function selected to be minimised is the mean absolute error (mae) between the ground truth traces and the A-scans which result from the inversion

$$C(p) = \sum_k \sum_t |(E^{obs}_{k,t} - E^{synth}_{k,t}(\mathbf{p})| \tag{7.3}$$

where E^{obs} are the observed traces and $E^{synth}(\mathbf{p})$ are the modelled traces which are a function of the input parameters **p** and these are summed over time and over all traces used in the inversion with k representing the k-th trace.

The process followed in the FWI scheme using the ML-forwars solver is illustrated using a flow chart in Figure 7.10. First, the selected traces from the B-scan are loaded to the algorithm. Then the initial particles, in this case models with 4 parameters, are randomly chosen by PSO based on the set of constraints determined by the ranges of values each parameter can take. In case a priori information exists, it can be used to constrain and decrease further the solution space. The parameters of each model are normalised and given as an input to the ML solver in order to predict the responses that correspond to each model

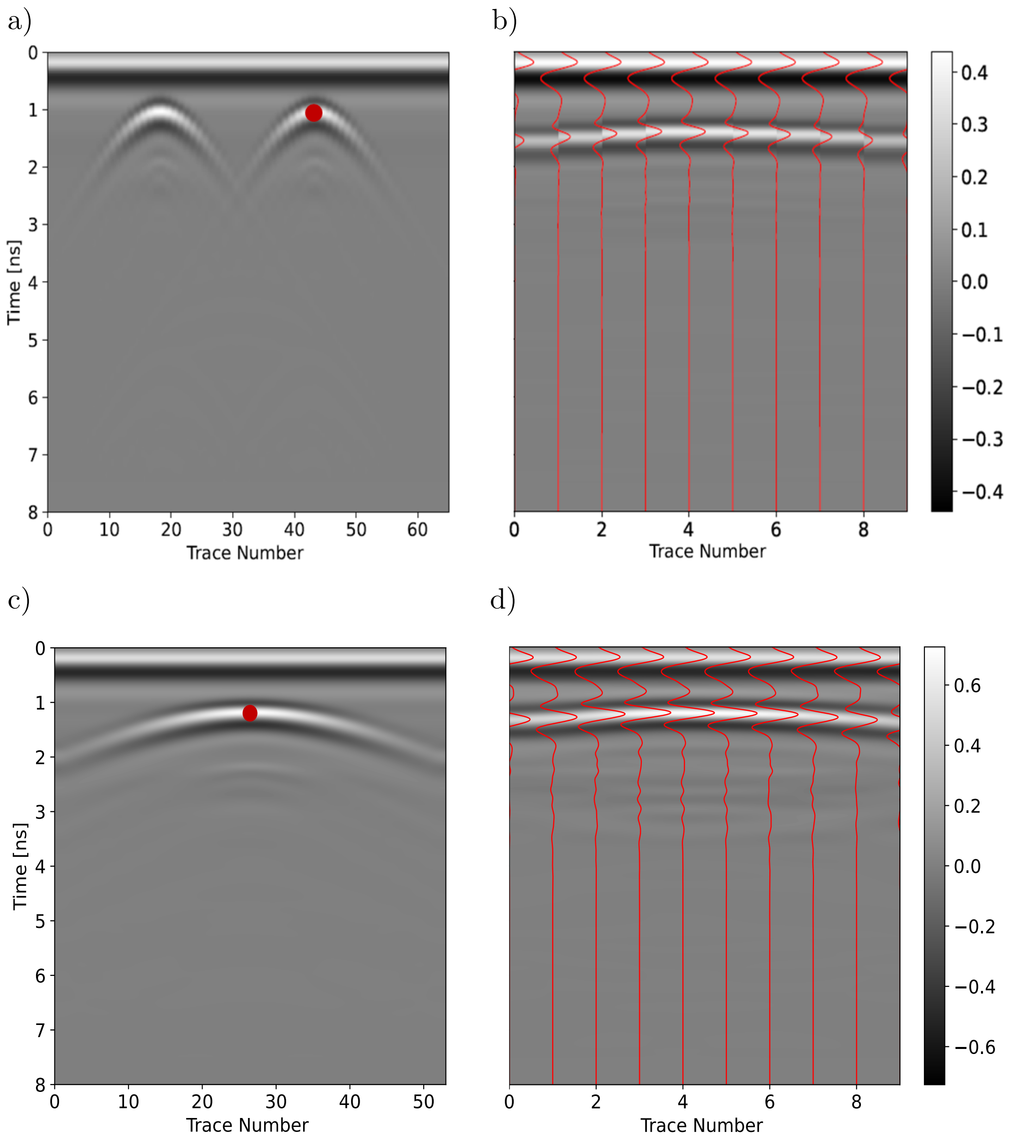


Figure 7.9: a) The resulting B-scan for the synthetic scenario shown in Fig. 7.7, b) The 10 traces selected from each hyperbola to be used in FWI, c) The resulting B-scan for the synthetic scenario shown in Figure 7.8 and d) The 10 traces selected from the chosen hyperbola to be used in FWI.

and solve the forward problem. The network predicts the A-scan compressed responses for different antenna positions, which essentially forms a compressed B-scan for each model. The compressed B-scans are decompressed to obtain the time domain responses and mae is calculated between each predicted B-scan and the true B-scan data. Subsequently, the parameters of each model are updated, the forward problem is solved for the updated parameters and the same process continues until convergence is achieved and the final model with a low error is obtained.

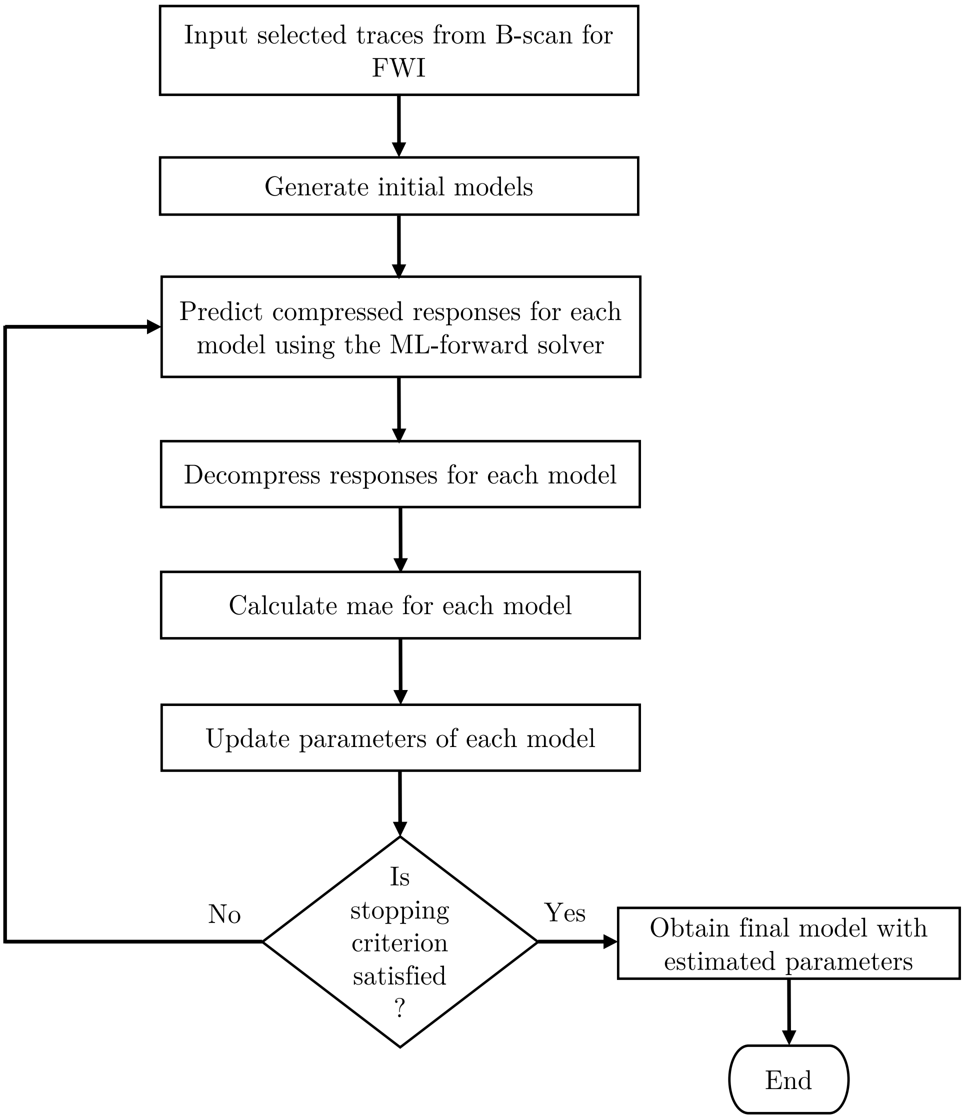


Figure 7.10: Flow chart illustrating the FWI scheme using the ML-based forward solver.

For the first case study, FWI was completed with a loss of 8.4e-6. The recovered parameters from the inversion are given in Table 7.4 along with the ground truth

B-scan 1	**WC(%)**	**D (cm)**	**R (cm)**
Ground truth	9	5.8	1
FWI	8.95	5.5	1.1
B-scan-2			
Ground truth	3.5	7.8	3
FWI	3.44	7.4	2.95

Table 7.4: FWI results for the synthetic B-scans.

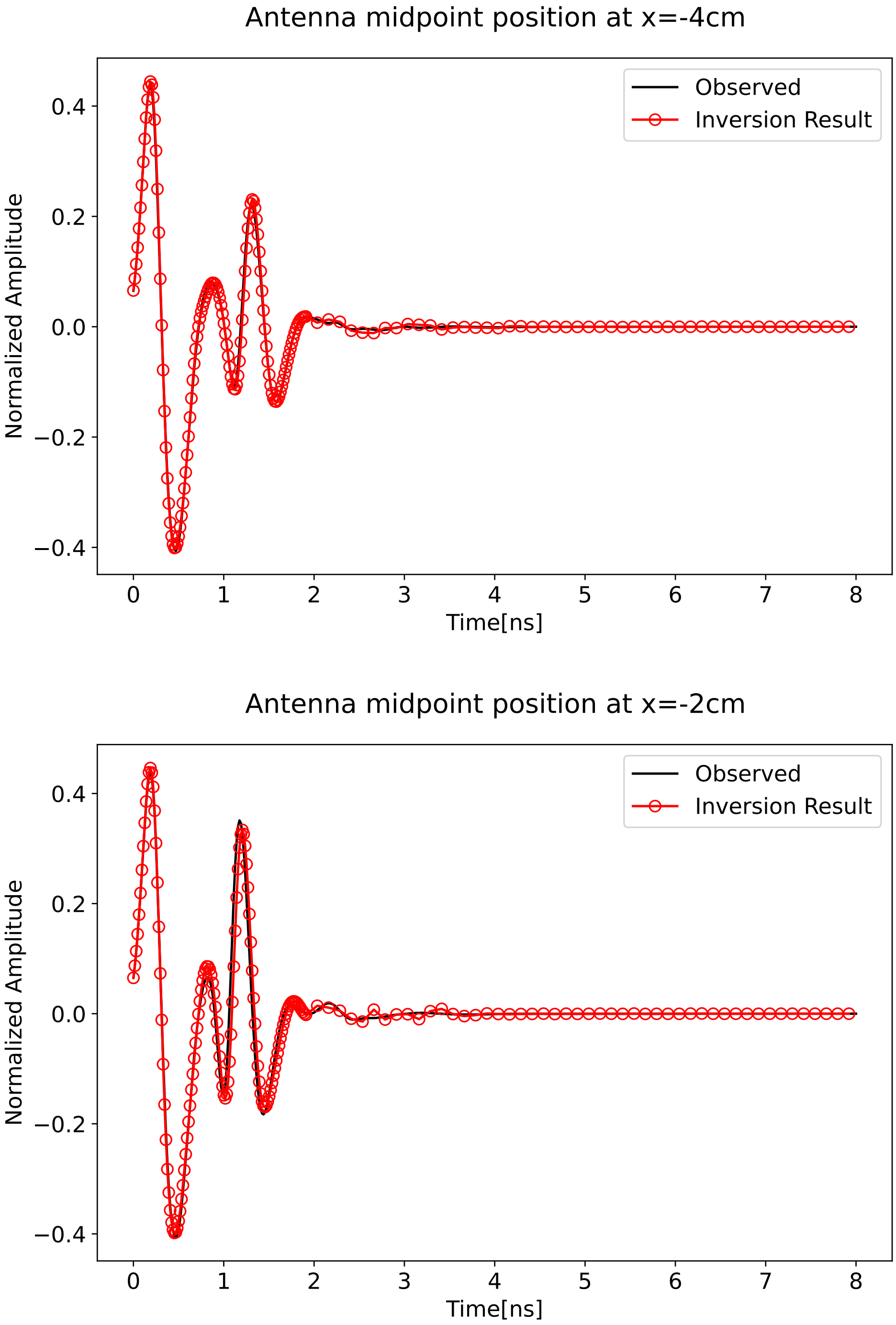


Figure 7.11: Comparison between the synthetic traces from the concrete slab scenario 1 with the resultant A-scans using the parameters predicted by the FWI-ML algorithm for antenna positions x=-4 cm and x=-2 cm relative to the target's position.

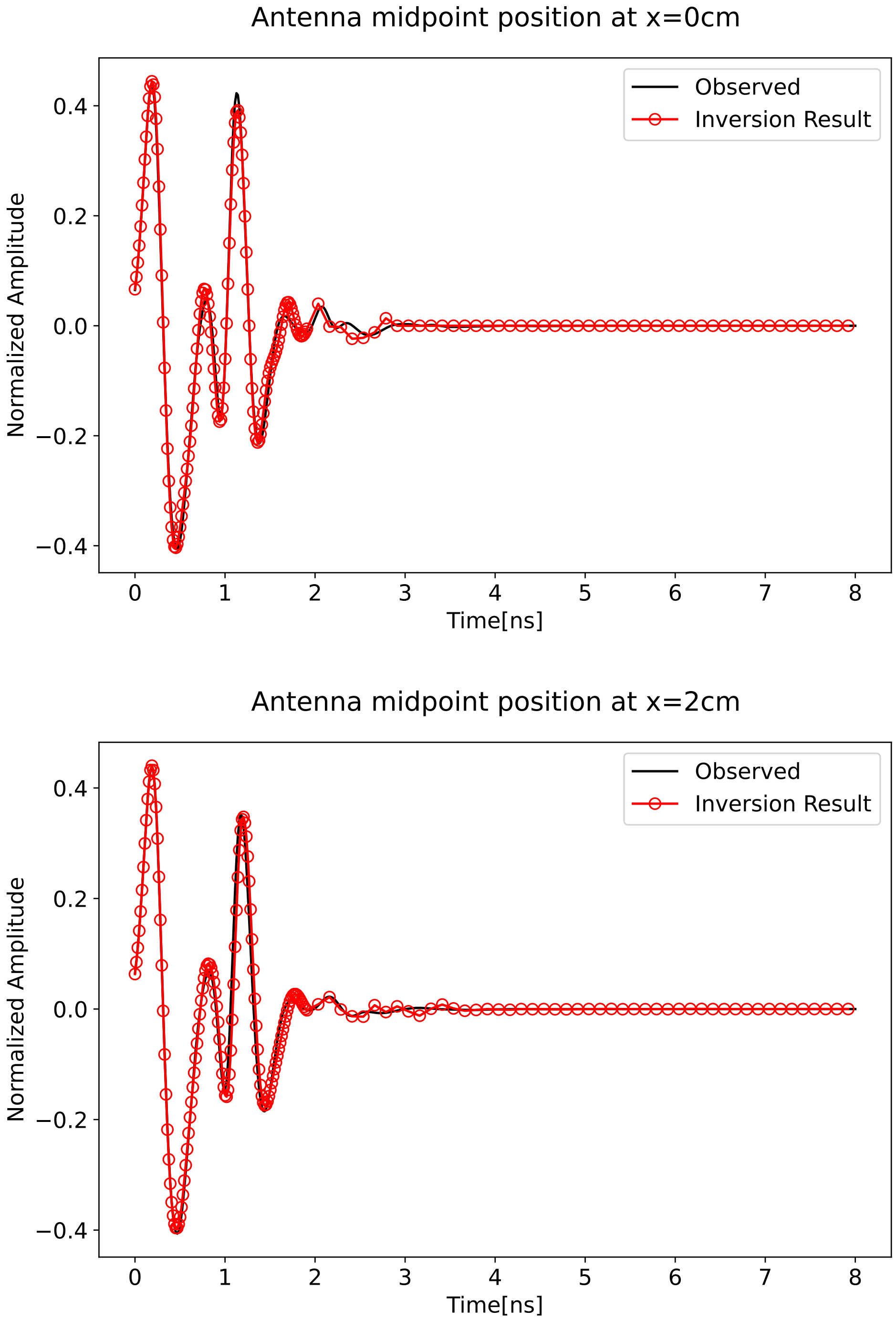


Figure 7.12: Comparison between the synthetic traces from the concrete slab scenario 1 with the resultant A-scans using the parameters predicted by the FWI-ML algorithm for antenna positions x=0 cm and x=2 cm relative to the target's position.

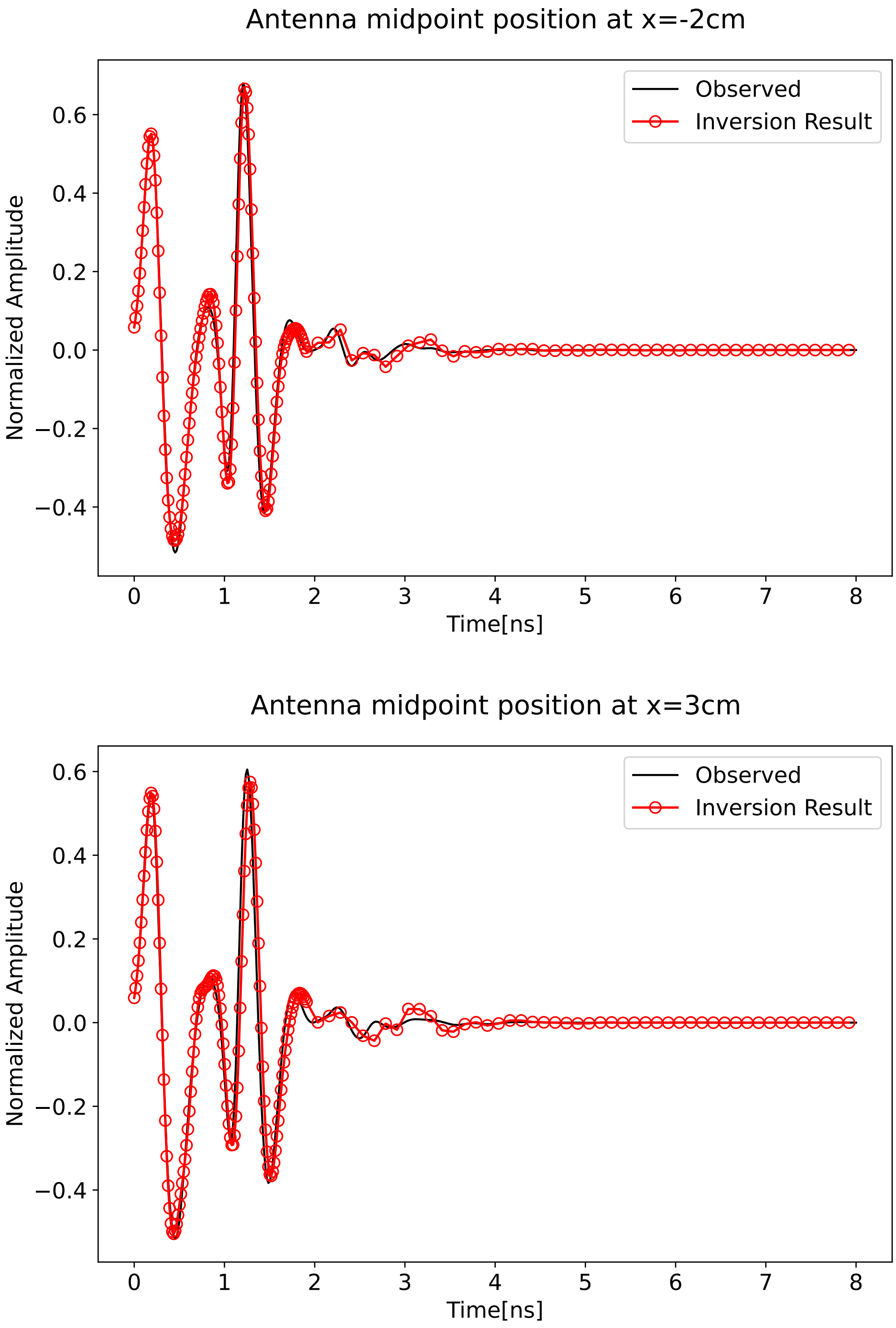


Figure 7.13: Comparison between the synthetic traces from the concrete slab scenario 2 with the resultant A-scans using the parameters predicted by the FWI-ML algorithm for antenna positions x=-2 cm and x=3 cm relative to the target's position.

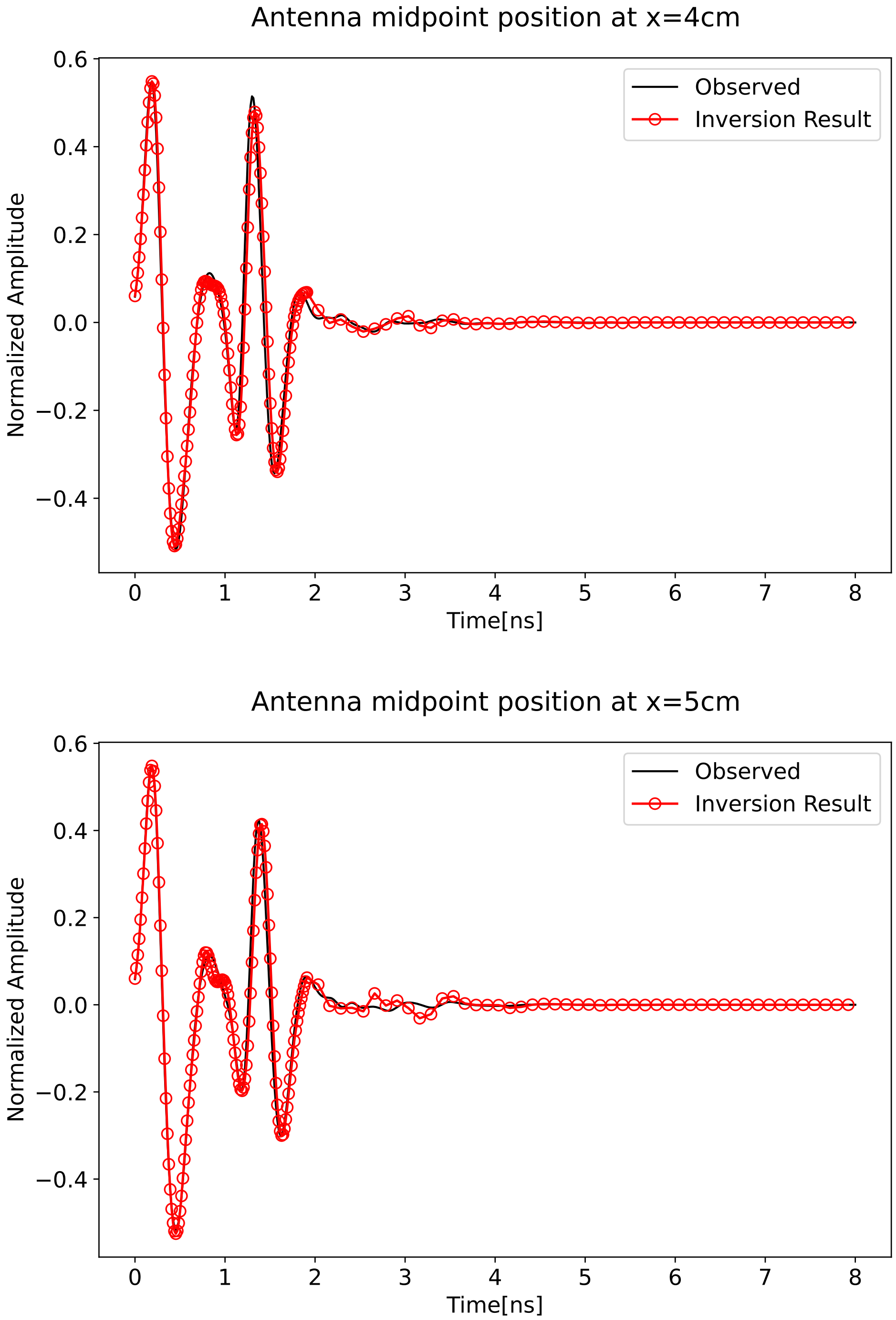


Figure 7.14: Comparison between the synthetic traces from the concrete slab scenario 2 with the resultant A-scans using the parameters predicted by the FWI-ML algorithm for antenna positions x=4 cm and x=5 cm relative to the target's position.

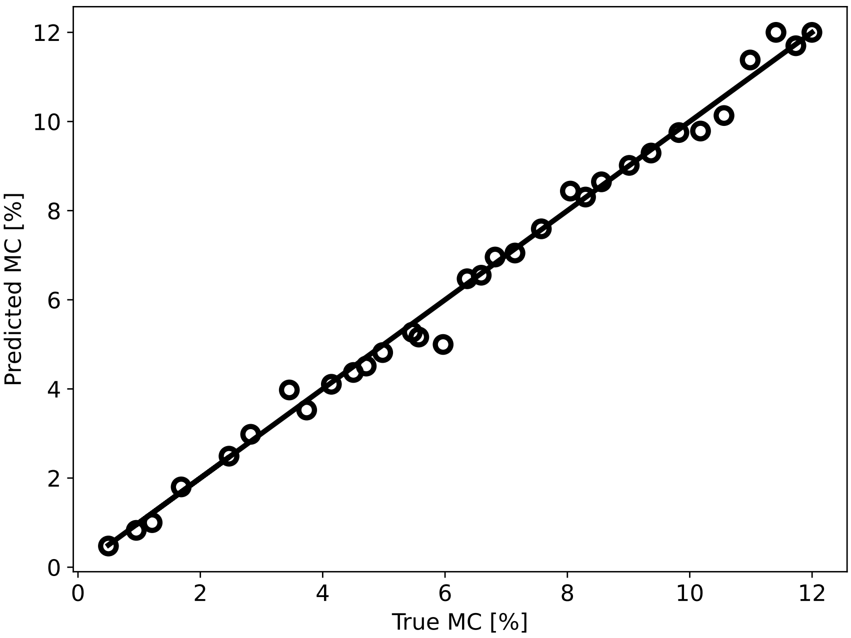


Figure 7.15: True moisture content versus predicted from FWI for 30 different cases.

values. It is obvious that the obtained values are very close to the true parameters and to further validate the scheme, the A-scans that correspond to the recovered model parameters are investigated. Figures 7.11 and 7.12 displays the observed A-scans compared with the fitted responses for different antenna positions and also for the case where the target is located directly below the antenna center. In all cases, the traces are almost identical, validating the accuracy of the solution and that the predicted parameters reproduce successfully the original traces. Similarly for the second scenario, the FWI estimated parameters, presented in Table 7.4 and their corresponding fitted responses are in a very good agreement with the ground truth. The synthetic traces compared with the observed ones are displayed in Figures 7.13 and 7.14 for different antenna positions relative to the target. FWI converged with a loss of 1.1e-5 for the second scenario.

The same ML-FWI algorithm was applied on 30 different randomly generated simulated B-scans and the predicted FWI parameters were compared with the true parameters as shown in Figures 7.15, 7.16 and 7.17 for the moisture content, the depth and the diameter, respectively. It is obvious that the three parameters can be accurately predicted within a reasonable margin of error. The water content predictions are the most accurate, with the scheme predicting values very close to the true ones at all cases. The depth and diameter follow with an error of ±1 cm and ±0.5 cm, respectively, which are acceptable values of error. Note, that although 10 traces are used in the inversion, ML-FWI can provide predictions with similar accuracy with even less number of traces.

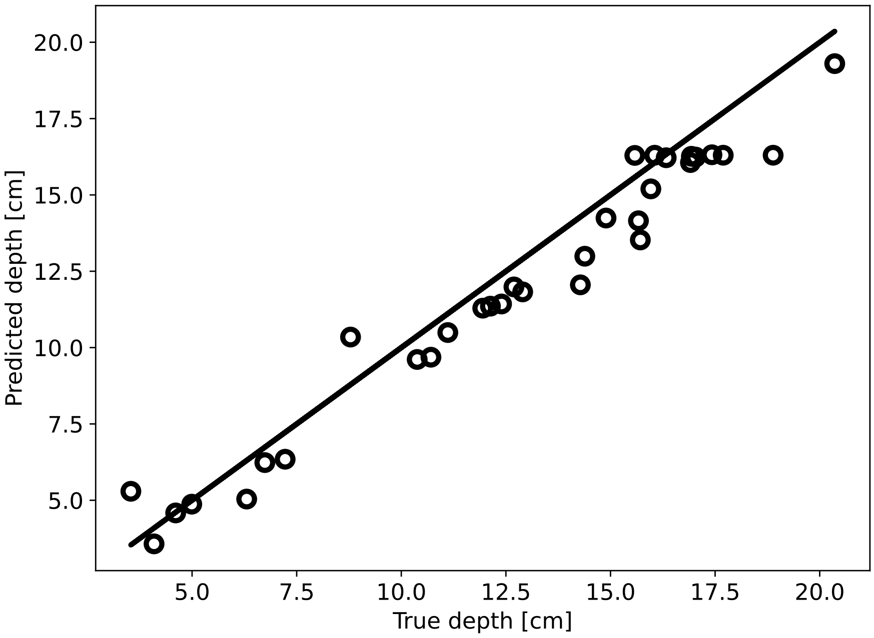


Figure 7.16: True target depth versus predicted from FWI for 30 different cases.

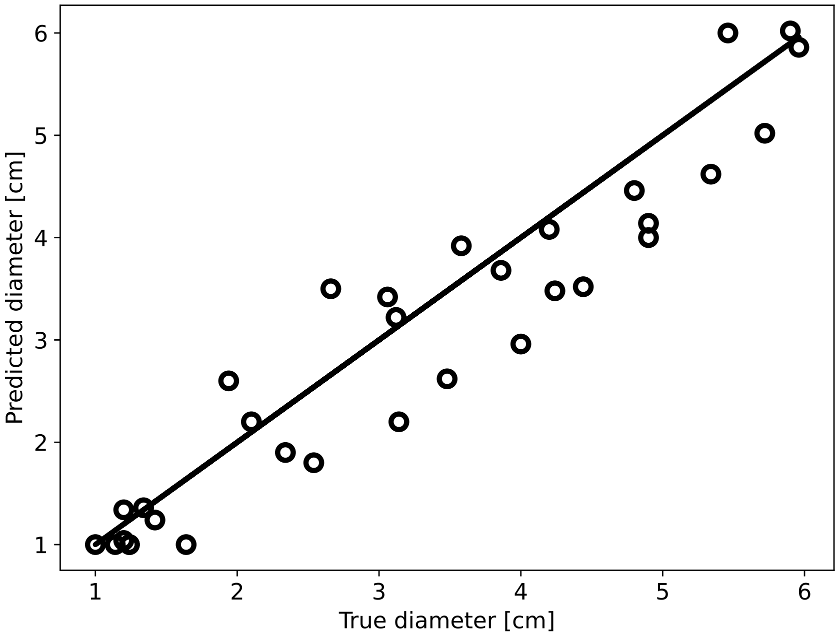


Figure 7.17: True target diameter versus predicted from FWI for 30 different cases.

B-scan 1	**WC (%)**	**D (cm)**	**R (cm)**
Ground truth	-	3	1.2
ML-FWI	11.2	3.6	1.26
FDTD-FWI	10.9	3.8	1.22
B-scan 2			
Ground truth	-	6.2	2.5
ML-FWI	11.0	6.7	2.6
Giannakis et al. (2019b)	11.5	-	-

Table 7.5: FWI results for the real B-scans.

7.6.3 *FWI-real data*

Since a replica of a real GPR system was used in the simulations, the ML-FWI scheme can be applied on real data measured with the GSSI 2000 MHz antenna. Real data from reinforced concrete slabs, were collected in the NDT laboratory at the University of Edinburgh in the School of Engineering using the GSSI transducer along with the SIR-4000 system. These scenarios were used to evaluate the performance of the algorithm on real data, with two case studies presented here. The setup of the two investigated scenarios are illustrated in Figures 7.19 7.20, for the first and the second, respectively, and their corresponding B-scans in Figure 7.18. For the first case, the transducer was moved on top of a single rebar and 10 traces, shown in Figure 7.18b, were chosen for the inversion. In the second case, it is obvious that the slab included multiple rebars and thus, multiple responses were acquired. However, the algorithm is trained for slabs with a single rebar and responses from other targets need to be removed. The rebar, from which 10 traces were selected for FWI, is annotated with a red dot in Figure 7.18c. The selected traces were truncated in order to eliminate later responses from the deeper rebar. The same FWI scheme, that was applied on the synthetic data, was also applied on the real case study.

For the first case, the resulting D and R of the rebar are 3.6 cm and 1.26 cm, respectively, which are very close to the true measured values of $D = 3$ cm and $R = 1.2$ cm, as shown in Table 7.5. The fitted traces, which were obtained using the FWI predicted parameters, compared with the real responses are displayed in Figures 7.21 and 7.22 for different antenna positions, showing a good agreement in all cases. FWI for the second scenario converged with $D = 6.7$ cm and $R = 2.6$ cm, which again are accurate estimates of the true values $D = 6.2$ cm and $R = 2.5$ cm. The fitted traces for the second case are plotted in Figures 7.23 and 7.24,

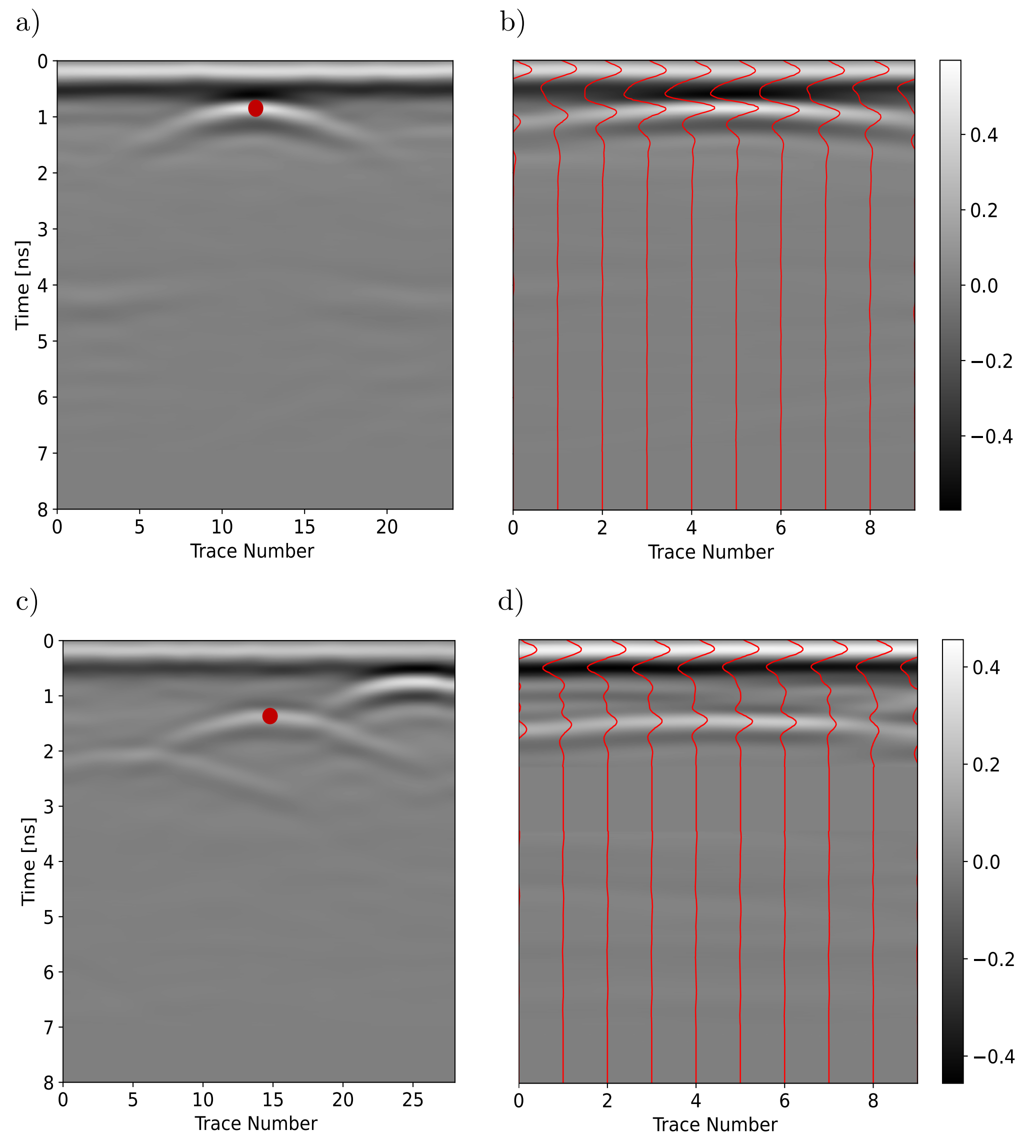


Figure 7.18: a) The resulting B-scan for the real scenario shown in Figure 7.19, b) The 10 traces selected from the hyperbola to be used in FWI, c) The resulting B-scan for the real scenario shown in Figure 7.20 and d) The 10 traces selected from the chosen hyperbola to be used in FWI.

showing again a very good fit between the real and the FWI traces. Although the true water content of the concrete was unknown and could not be easily determined, the resultant value $WC = 11.2\%$ and $WC = 11.0\%$ for the first and second slab, respectively, is in a good agreement with the value of $WC = 11.5\%$, which was estimated in Giannakis et al. (2019b) for the same concrete batch and using a different GSSI transducer of 1.5 GHz. Therefore, providing a way of validating the moisture content estimate. The small mismatches between the responses can be attributed to small errors in the ML model predictions or to the limitations of the FWI algorithm itself in addition to the mismatch error

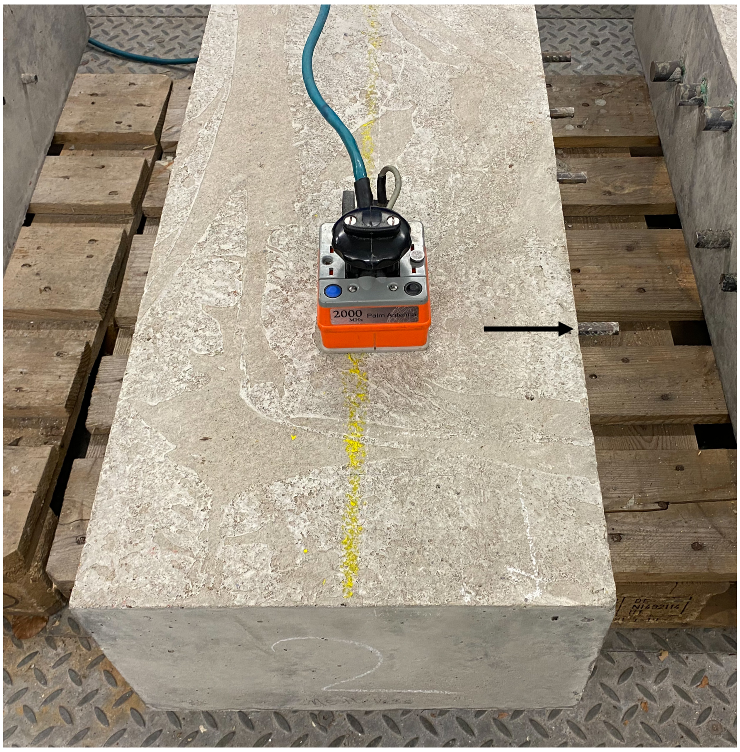


Figure 7.19: Image of the slab for the first case study: 10 traces were selected were selected for FWI from the B-scan taken over the rebar shown with a black arrow.

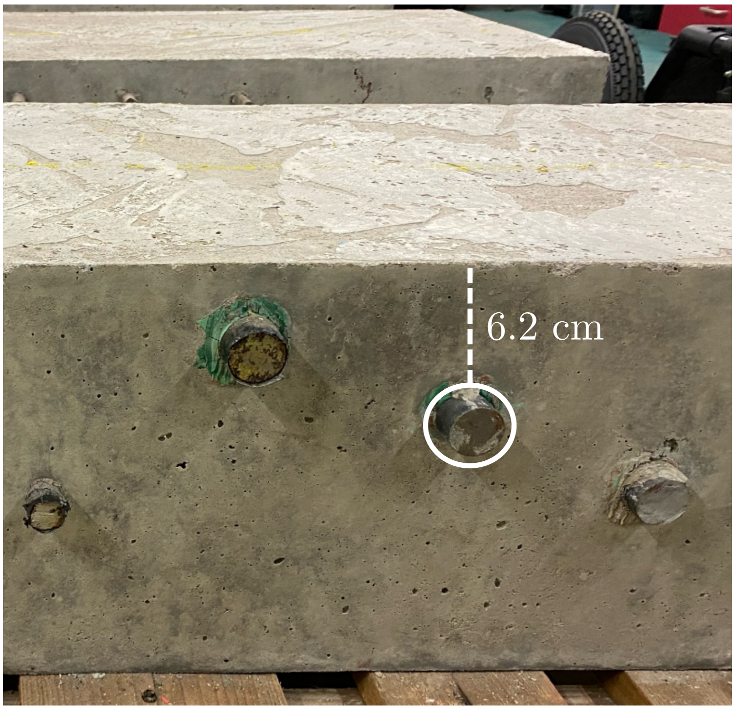


Figure 7.20: Image of the slab for the second case study: 10 traces were selected for FWI from the B-scan taken over the circled rebar, shown in white.

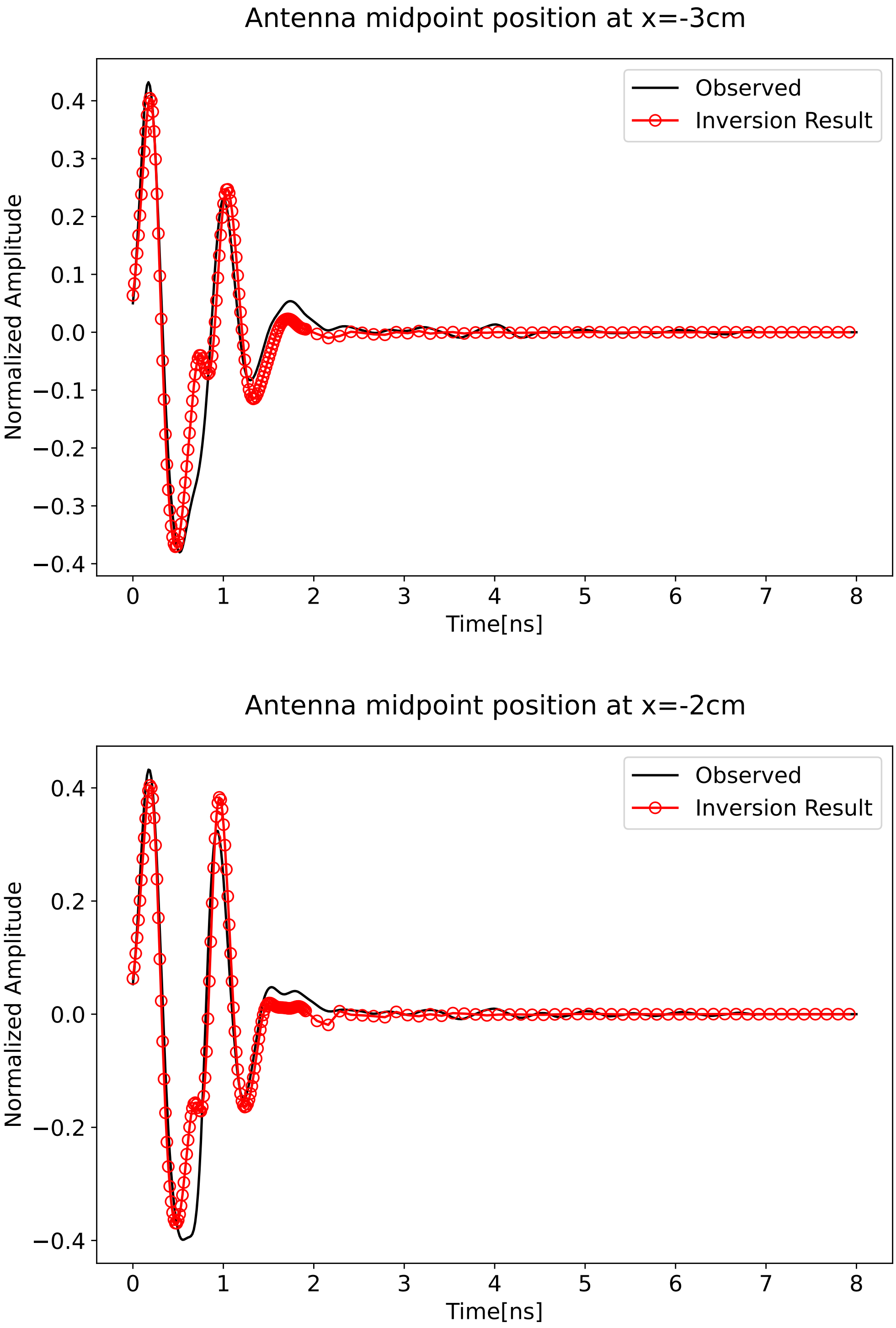


Figure 7.21: Comparison between the real traces from the concrete slab 1 with the resultant A-scans using the parameters predicted by the FWI-ML algorithm for antenna positions x=-1 cm and x=0 cm relative to the target's position.

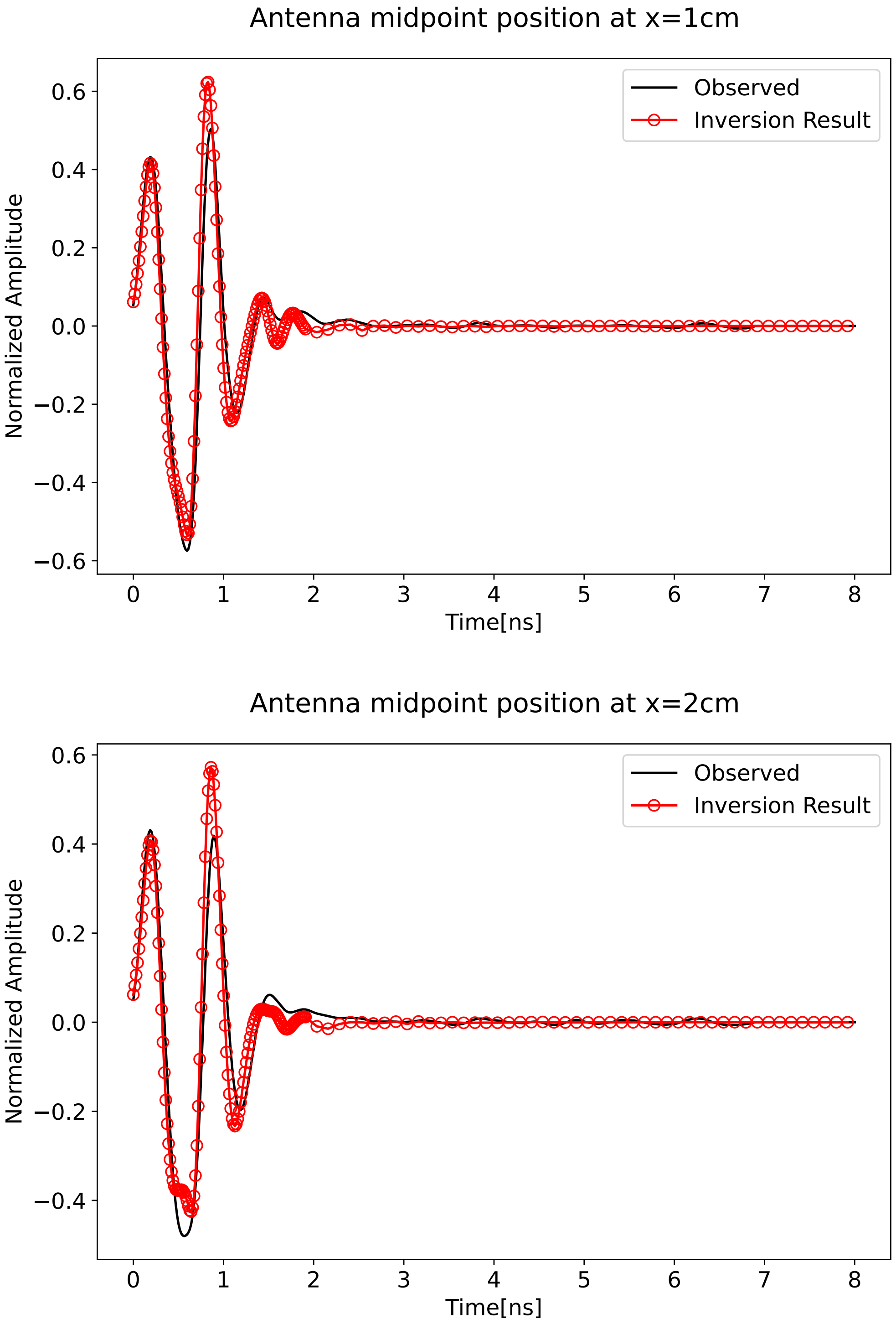


Figure 7.22: Comparison between the real traces from the concrete slab 1 with the resultant A-scans using the parameters predicted by the FWI-ML algorithm for antenna positions x=2 cm and x=3 cm relative to the target's position.

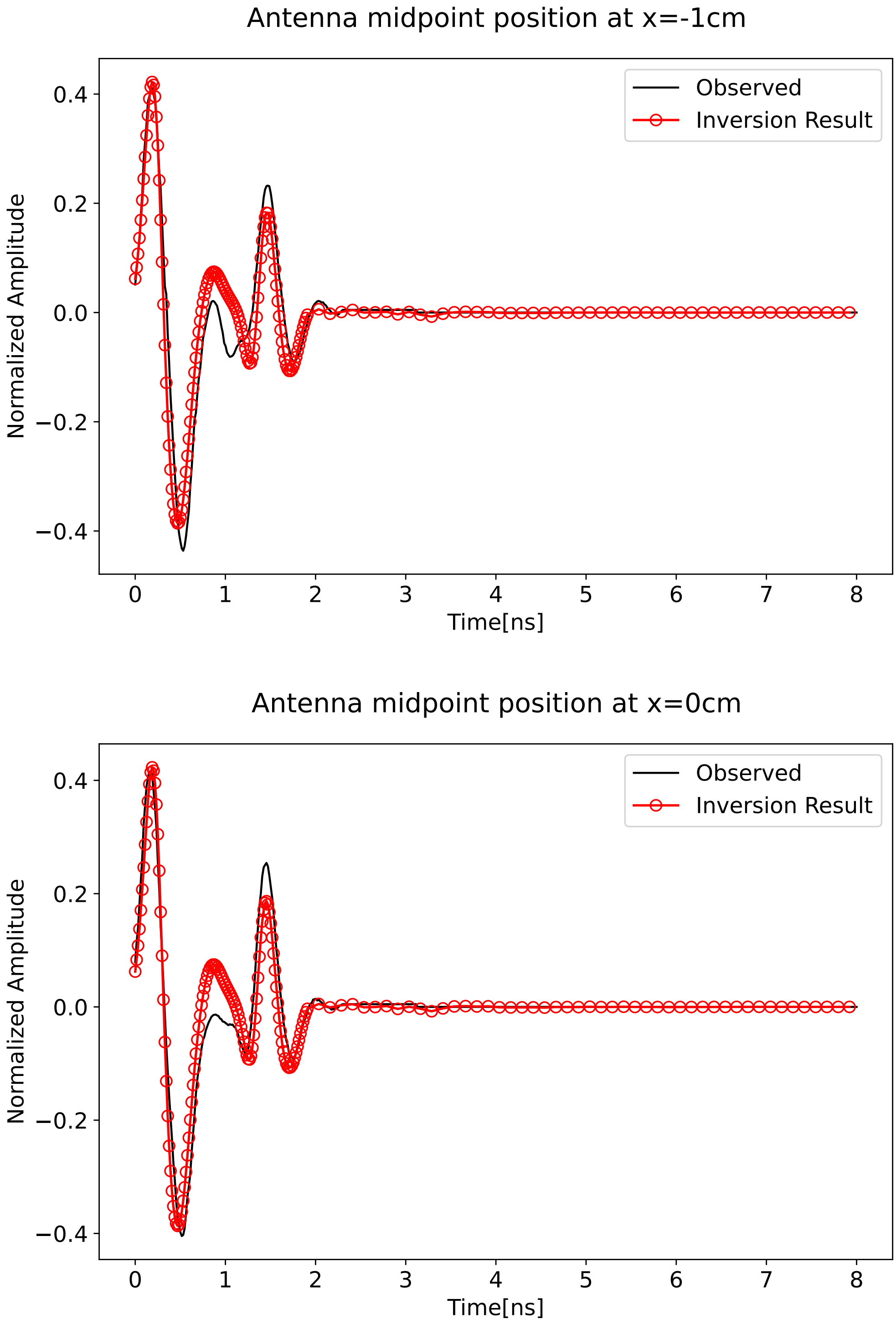


Figure 7.23: Comparison between the real traces from the concrete slab 2 with the resultant A-scans using the parameters predicted by the FWI-ML algorithm for antenna positions x=-3 cm and x=-2 cm relative to the target's position.

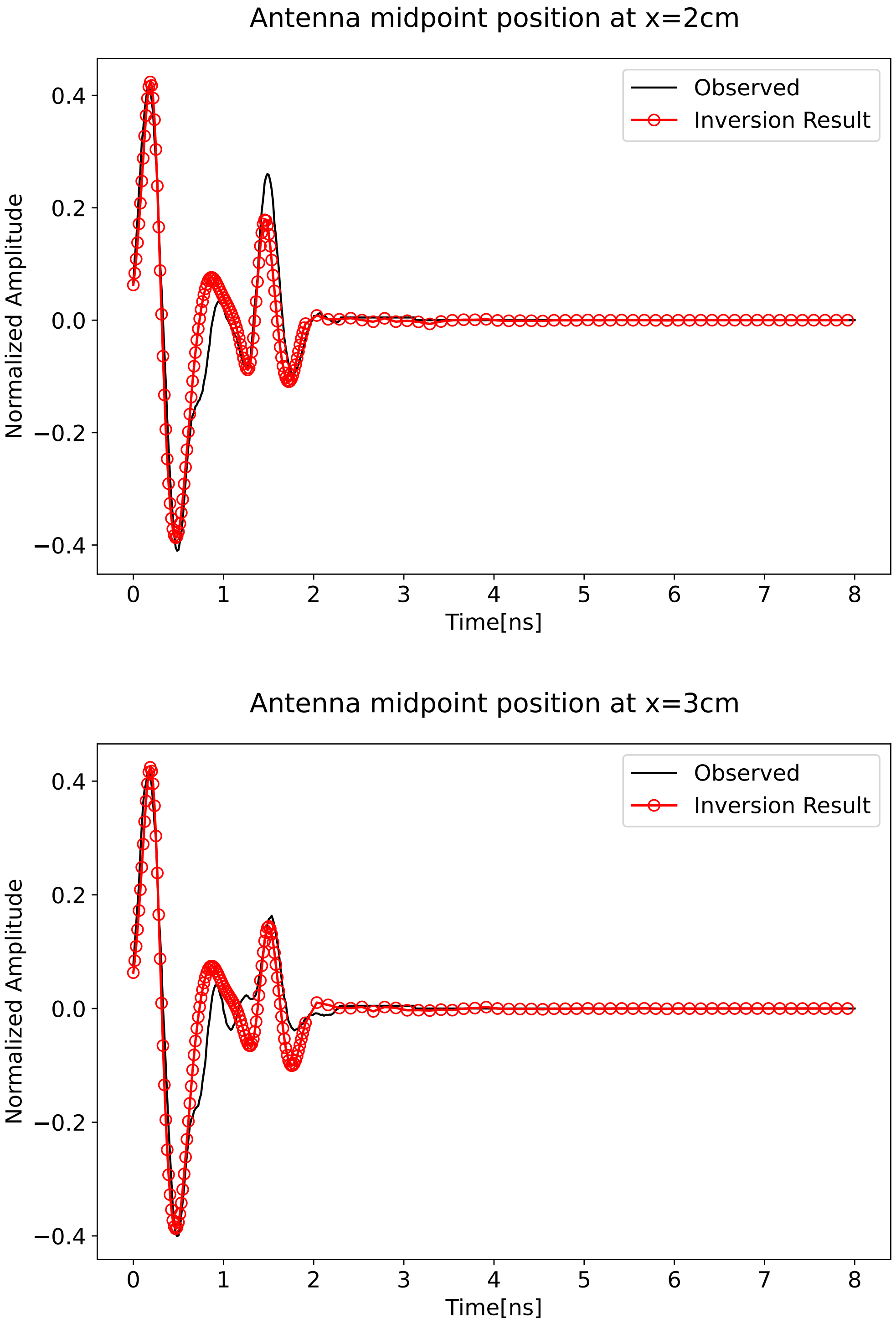


Figure 7.24: Comparison between the real traces from the concrete slab 2 with the resultant A-scans using the parameters predicted by the FWI-ML algorithm for antenna positions x=1 cm and x=2 cm relative to the target's position.

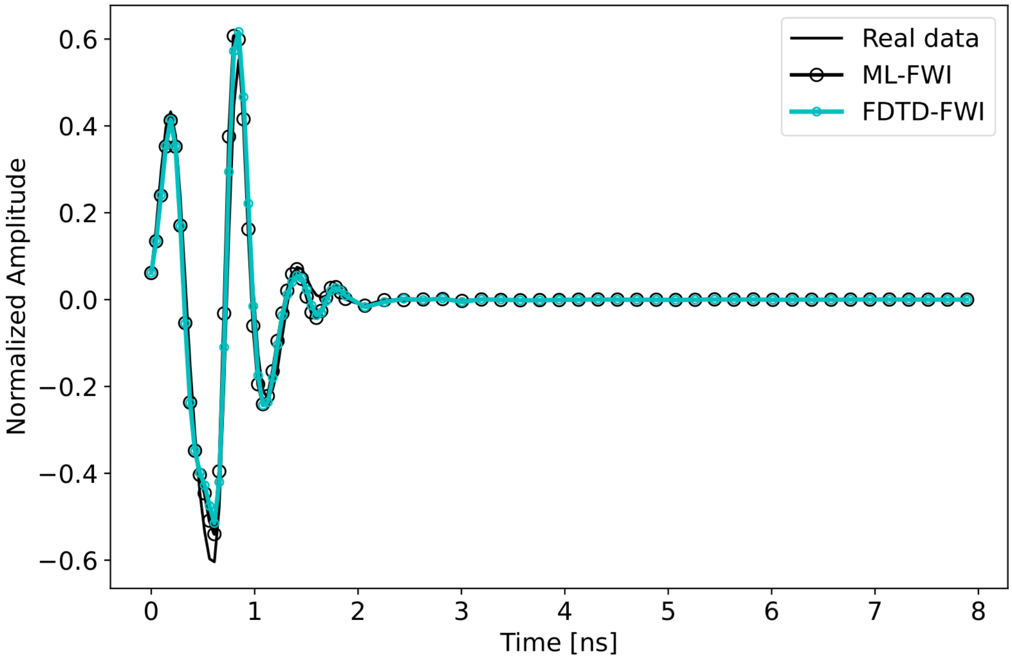


Figure 7.25: Comparison between the fitted traces using ML-FWI and FDTD-FWI.

between the true antenna and the model. Nevertheless, the results prove that the scheme can provide a highly accurate solution.

To get a sense of how similar the ML-based forward solver behaves to FDTD as part of FWI, the same FWI scheme was applied to the first real case study using gprMax as its FDTD solver. The resultant FWI values for WC, D and R of the rebar were 10.9%, 3.8 cm and 1.22 cm, respectively, which are in a very good agreement with both the ground truth and the values obtained using the ML solver, as displayed in Table 7.5. To further evaluate the accuracy and similarity of the schemes, the FWI traces were investigated. Figure 7.25 shows a comparison between the real A-scan and the fitted ML-FWI and FDTD-FWI traces for rebar position at x=1 cm. The results are almost identical, indicating that the ML model can behave similarly to an FDTD solver for the cases it has been trained for, Therefore, conventional solvers such as FDTD can be replaced for certain scenarios by ML models without compromising the overall accuracy of FWI.

7.6.4 *Execution times*

Generating the entire training set consisting of 4000 traces required a run time of ∼2 days on an NVIDIA TITAN RTX 24 GB GPU, given that a single FDTD A-scan takes ∼40-45 seconds by utilizing the GPU capabilities of gprMax (Warren et al., 2019). The ML forward solver can predict an A-scan instantly in real-time, regardless of CPU or GPU execution.

For the FWI of the real data, a model of the real scenario is required, since FWI requires for the forward problem to be solved multiple times. Based on the domain size and time window used for the model of the real scenario described above, each A-scan required ∼20-25 seconds. Given this, the FWI algorithm with FDTD as forward solver and using 10 traces, 40 iterations and 40 particles for the PSO required a runtime of ∼8.5 hours on 4 NVIDIA TITAN RTX 24 GB GPUs, which would have been increased to ∼35 hours if a single GPU was used. In contrast, the FWI-ML scheme using the same inversion options as the FWI-FDTD, required a runtime of 1.5 hours on an Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz without utilizing parallelization which would have decreased the execution time a little further. Therefore, this proves that the ML-forward solver can lead to orders of magnitude less FWI execution times compared to an FDTD solver.

8

CONCLUSIONS AND RECOMMENDATIONS

This chapter presents a summary of each of the previous chapters and provides the main findings and outputs of the thesis. In addition, recommendations for future research on the subject are given.

8.1 CONCLUSIONS

The main objective of the thesis was to develop a realistic numerical equivalent of a widely-used commercial high-frequency GPR antenna and use this to develop realistic models for training ML processing and interpretation schemes. The reason for developing such ML schemes was to automate some of the processing and interpretation steps of GPR data, which often pose difficult problems to solve. The current processing is based on traditional algorithms which have certain drawbacks and the interpretation of the data is subjective and depends on the knowledge and experience of the human use, whereas an ML approach can provide objective and more accurate solutions. In order for the ML algorithms to be effective, the training data should be representative of real GPR data and thus, a replica of a real GPR system was essential.

8.1.1 *Development and optimisation of the transducer model*

An accurate model of the GSSI 2 GHz “palm” commercial antenna was developed using global optimisers to obtain estimates of unknown parameters. To replicate the behavior of a real antenna, both the main components and their dielectric properties needed to be accurately implemented in the simulations. The process followed to built the model can be summarised as follows:

- The components and the geometrical characteristics of the transducer were obtained by visually inspecting the interior of the transducer enclosure. The essential components for developing the model were the transmitter and receiver bowties, two EM foam absorbers, the EMI gaskets, the shielding, the plastic case and the skid plate.
- The true pulse was unknown and a voltage source with a Gaussian-shaped pulse was used with a center frequency chosen through optimisation.

- Global optimisation was employed to acquire estimates of unknown parameters of the transducer. GA and PSO were both implemented with the criterion of minimising the mismatch between synthetic and real data from two scenarios. The scenarios selected were the free space direct coupling and the response over a metal plate.
- Both schemes converged with high accuracy, with the GA model resulting in the smallest error and chosen as the optimal model of the GSSI transducer.
- The antenna model was further validated using real GPR data from different scenarios, showing a very good agreement predicting accurately both the arrivals and the amplitudes of the responses.
- A grid convergence analysis was implemented to determine the optimal FDTD step size for the model.
- The E-plane and H-plane radiation patterns of the antenna model were developed in free space and over lossless and lossy media to investigate the radiation characteristics of the antenna.
- A model of a second unit of the same transducer was also built, showing that units corresponding to the same system are not in general identical, which is usually attributed to the difficulty of creating absorber foams with the exact same dielectric properties for all units of the same system.

All the components included in the model proved to be essential in order to acquire an accurate numerical equivalent of the real system. Good knowledge of the geometry is essential, otherwise if the geometry of the model does not correspond in a high degree to the real geometry, an optimised model that can fit the real data well and reproduce the real system's responses might not exist. A Gaussian-shaped pulse, although not physically realisable, was proved to be a good approximation to the real pulse. To minimise the mismatch between the real and the synthetic data in optimisation, two scenarios were found to be sufficient as long as these scenarios are quite different from each other. Using two similar scenarios would lead to a model that is not able to generalise, whereas more than two scenarios resulted to a negligible improvement in the model. For the specific problem, GA performed better than the PSO method for a number of different initial conditions, although this might not be the case for different antenna models. The final GA model was able to predict accurately the timings, shapes and amplitudes of both real responses used in optimisation with an RMS error of 5.57% and of other validation scenarios which were not part of the optimisation, concluding that model is a reliable representation of the real GPR system.

The optimisation results showed that knowledge of the true antenna parameters was not essential to built an accurate numerical equivalent of a real antenna. The acquired optimised parameter values might not even be close to the true antenna parameters, but their combined effect, which was constrained to be realistic, was able to reproduce the real responses. This is the case for other antenna systems, as also shown in Warren (2009). Thus, it was concluded that with only a small amount of information regarding the antenna and the implementation of global optimisers, a numerical equivalent of a real transducer can be built efficiently.

The grid convergence analysis showed that the optimal spatial step for modelling the specific antenna system is 1 mm and also demonstrated the significance of choosing the right spatial step in order to resolve all the antenna elements, while considering the computational costs. A very coarse grid might not be able to resolve accurately the smallest antenna elements. For other antenna systems, the choice of the optimal spatial step might be different and is always dependent on the size of the elements required to be modelled.

Studying the radiation characteristics of the antenna model showed that, as expected, the main lobes became more directive with increasing permittivity of the half-space, whereas the minor back lobes were decreasing with increasingly more energy going into the subsurface. Introducing a finite conductivity resulted in an amplitude attenuation in the subsurface corresponding to the conduction losses.

8.1.2 *Background prediction using ML*

After developing the numerical equivalent of the GPR transducer, different ML applications were investigated. The first application was the prediction of the background response and its dielectric properties. The suggested scheme can be summarised as follows:

- A large training set from diverse scenarios was developed using the antenna model in the simulations. The training set included scenarios of a half-space with a rebar, a layer and a half-space and a half-space including both. The dielectric properties, the depth and radius of rebars and the thickness of the layers were all randomly generated. All half-spaces and layers used in the training set were modelled as non-dispersive homogeneous materials.

- Pre-processing the A-scan training data was performed to transform them to a suitable form for ML.

- Deep learning, which is the most popular class of ML techniques was implemented. In particular, two coupled feed-forward fully-connected NNs were used for the scheme. The first network predicts the background response given a total A-scan response. The estimated background response is subtracted from the original A-scan response to perform background removal.

- The predicted background from the first network is fed as an input to the second network, which subsequently predicts the background dielectric properties. A permittivity estimate enables the calculation of the EM wave propagation velocity estimate in the background medium and therefore to approximate the depth of targets.
- The scheme was validated with both modelled unseen test data and real data from different concrete slab scenarios, demonstrating its effectiveness and performance.
- A comparison between the deep learning outputs and the results from traditional techniques for background removal and velocity estimation was made, showing the superiority of the scheme.
- An RTM scheme was proposed that used the NN scheme to perform background removal and the NN predicted permittivity distribution instead of using a mean permittivity for an entire GPR profile and traditional background removal techniques.

Training a NN with pairs of input total GPR responses and their corresponding output background response proved to be an efficient scheme leading to an ML model that has captured the pattern and was able to predict background responses given a total A-scan as input. This novel approach was shown to remove only the background, since this was the only task the model was trained for, without removing other flat-lying responses, in contrast with the current background removal techniques. Training the first network with only homogeneous non-dispersive background media was proven to be sufficient and the network performed with a great accuracy even when tested to inhomogeneous and dispersive media. This can be attributed to the A-scans used for the training, which represent local information, instead of B-scan responses that common techniques use. Even in cases where targets that were not included in the training data were present in the responses, such as non-metallic targets or multiple targets placed closely spaced together, the network was still capable of predicting the background signal.

Targets placed very close to the surface, where the background signal is not clear, did not seem to affect the performance of the NN model, which again recognised the background responses. This is very important since most of the targets sought with this particular GPR system are expected to be shallow placed. The high accuracy of the model is highly attributed to the pre-processing scheme of the training data. Inappropriate pre-processing of the data can lead to an inefficient ML model and to incorrectly assume that the ML suggested scheme with the particular training set is not feasible and thus, the choice of the processing techniques is significant. The dimensionality reduction helped significantly the training of the model, as it decreased the complexity and the computational costs by using fewer variables in the problem with no significant loss of information. Shifting of the responses and normalising the data was proven to be essential in order for the network to perform well on real data.

It was deducted that the novel idea of using only background information to predict its dielectric properties is effective with the second network accurately predicting the background permittivity and conductivity given the predicted output background response from the first network. The permittivity value allows for the calculation of the velocity and therefore, the depth to targets of interest. This ML scheme does not require the presence of a hyperbola in the data to acquire a permittivity estimate, in contrast to the commonly used hyperbola fitting, with the latter being also dependent on the time zero correction chosen.

When the scheme is applied to dispersive media, it is still predicting an accurate bulk permittivity estimate, whereas the conductivity value is harder to predict. Given that dispersive materials were not part of the training, the sensitivity of the two parameters and the complexity in acquiring a conductivity estimate, this result was expected.

Both networks converged with very small test set errors between the true and predicted outputs, which indicates that the networks have not overfitted the training set. This was further confirmed with a number of real and synthetic A-scan an B-scan examples, where in all cases the proposed scheme was able to recognise the background signal and provide accurate estimates of its properties. The performance of the ML scheme was compared with the mean removal method with the results showing that the proposed scheme outperforms the mean subtraction method. The mean subtraction method removed both the background and other flat lying targets in addition to creating artifacts in the data in contrast to the ML scheme, which removed only the background signal and did not create any artifacts in the data.

The proposed ML algorithm can provide background prediction with a mean permittivity and conductivity estimate for an entire B-scan or a distribution around the area of interest in case of a considerably inhomogeneous background. An RTM scheme that utilises the ML background removal and the predicted permittivity distribution, instead of a mean value, was developed and evaluated with both synthetic and real cases. RTM was performed on the same data using also SVD filtering for background removal with a mean permittivity value and the results were compared with the proposed RTM-ML algorithm. In both synthetic and real scenarios, the ML model predicted more accurately the background response from the SVD filter, which along with previous examples shows the superiority of the ML scheme from both the mean subtraction and SVD methods. A predicted permittivity distribution assisted RTM to successfully collapse the events to their true positions in contrast to using a mean permittivity which lead to over and undermigrated sections in the data. For relative homogeneous environments, RTM using a mean permittivity for the entire profile performs well, however in cases where permittivity variations exist, a single permittivity is not representative of the total area and a permittivity distribution results in a more accurate migrated image. This limitation of RTM with a mean permittivity value becomes more evident when large permittivity variations are present throughout a profile, where, as demonstrated, RTM fails to collapse the hyperbolas in the GPR image.

8.1.3 *ML-based forward solver*

The second ML application investigated was a deep learning-based forward solver used to predict entire B-scan data for reinforced concrete slab scenarios. The deep learning forward solver was used as part of an FWI algorithm to predict key properties. The steps follow for developing the ML model are as follows:

- A vast amount of simulated data representing different concrete slab scenarios was created and used to train the ML model. Concrete was simulated as a dispersive material using a Debye model with different Debye properties for each scenario. The rebars were modelled with random radii, depths and horizontal positions relative to the center of the antenna.
- In order to tackle sensitivity issues that arise between the Debye properties, these were represented by a single parameter, the water content of the concrete. This also reduces the number of input variables needed for the network.
- A feed-forward fully-connected NN was developed, which given four input model parameters predicts an A-scan response in real time and acts as a forward solver. The input parameters used are the water content, the depth, radius and position of the rebar. The network is tuned only for concrete slab scenarios but can be adapted to other cases, as well.
- The ML forward solver was used as part of an FWI scheme, which was implemented using PSO to invert a number of traces and estimate the radius and depth of the rebar and characterise concrete by its water content.

- The ML-FWI scheme was tested using both simulated and real data to evaluate its performance.
- Finally, the ML-FWI scheme was compared with the same FWI algorithm but using FDTD as its forward solver.

The Debye properties for concrete and in general are vastly differing in sensitivity, with the deep learning forward solver on initial attempts not being able to predict the GPR responses well, even after normalising the input Debye properties and for a number of different hyperparameters. In addition, the FWI algorithm was not able to predict accurately all Debye properties, as expected from its limitation to handle sensitivity issues. Using the water content as one of the inputs to the network, instead of the Debye properties, seemed to tackle this problem for the specific case and both the ML model and the ML-FWI scheme predicted successfully the GPR responses and the inversion parameters, respectively.

The same pre-processing as with the background prediction case was applied, which was essential for the convergence of the network, with the difference that SVD was used instead of PCA to reduce the dimensionality of the training GPR responses. For the ML-based forward solver, SVD was able to capture the pattern of the A-scan responses better than PCA for the same number of components. However, in this case, 70 SVD components were required to reproduce accurately the responses, in contrast to the smaller number of 59 PCA components required for the total responses in background prediction. This can be attributed to the training set models for the forward solver consisting of dispersive materials, which are more complex than the homogeneous media used for the ML background prediction.

Initially, a single ML model was trained to predict the GPR responses. However, utilising ensemble averaging with 40 different ML models used to produce the final model, resulted in better performance and more accurate predictions on unseen scenarios due to the better generalisation of the model resulting from ensemble averaging. The predictions made by the deep learning model were compared with the FDTD output A-scans, for a number of different cases, showing a great match between the two. The novel proposed ML scheme can predict at near real-time A-scans for a number of antenna positions relative to a target's position, and thus, generate entire B-scans fast with almost the same accuracy as the FDTD method.

The evaluation of the ML-FWI scheme on both synthetic and real data showed that in all cases the estimated parameters were very close to the ground truth and the resultant traces corresponding to the FWI parameters were in a good agreement both in amplitude and in phase with the original traces used for the inversion. Assessing the performance of the scheme on multiple synthetic examples showed that ML-FWI estimates the depth and diameter of rebars with an error of ±1 cm and ±0.5 cm, respectively, whereas the water content estimate was always close to the true value. FWI was performed using a global optimiser, which the fast ML forward solver made easy to implement in practice.

The comparison of ML-FWI with FDTD-FWI showed that the two schemes produced similar results, demonstrating that ML can provide A-scans with similar accuracy to FDTD but in orders less computational time. This makes FWI, which is a computationally demanding algorithm, applicable to conventional computers.

The above ML applications, apart from proving the success of ML schemes on processing and interpreting GPR data, they also demonstrate the significance of including a GPR transducer model in the simulations, which makes feasible the use of ML algorithms to process and interpret real GPR data. Without a model that can replicate the real responses in the simulations, the training of a NN would be carried out using responses that do not resemble the real ones and thus, would not perform well on real cases and would lead to misinterpretations of the data. Therefore, a numerical equivalent of a GPR system is an essential component of the training data simulations.

8.2 RECOMMENDATIONS

The research of the present thesis is part of an effort made in order to assist the processing and interpretation of GPR data using numerical modelling and machine learning. Recommendations for further research on the topic are provided in this section. The following recommendations are divided into two parts, for improving the existing antenna model and for improving the existing and developing other ML schemes.

8.2.1 *Improving the antenna model*

In the antenna model, a Gaussian-shaped pulse was used since the true pulse that the GPR manufacturers use is unknown, which is not realisable in practice. An improvement to the model would be to optimise the source pulse in addition to the antenna parameters, in order to obtain a more realistic pulse and possibly acquire a better match between synthetic and real responses. Furthermore, since this is simply a model, small details of the antenna interior have not been included. Investigating if the mismatch between the model and the real antenna is only due to the pulse differences or if more details regarding the antenna components are required is recommended for further research.

Measurements of the real data used in the optimisation can be performed in a facility with minimum environmental error, such as an anechoic chamber in order to characterise better the mismatch of the model with the real antenna. The antenna model requires a small step size of 1mm, but this might not be the case for the rest of the FDTD domain. Implementing a sub-gridding technique that would allow the use of different step sizes in different parts of the domain would greatly reduce the computational requirements.

Having an accurate digital representation that reproduces the radiation of a real antenna provides the ability to be used to different and not only machine learning applications. The behaviour of the antenna and how it changes with different types of materials can be investigated.

8.2.2 *Further developments for the deep learning schemes*

Although both of the ML schemes showed a high accuracy, some improvements can be made. For the background removal case, although the network performed well to unseen complex scenarios, including more complex scenarios in the training data, such as dispersive and inhomogeneous background media might increase the accuracy of its predictions further. In addition, different types of targets, non-metallic and of different shape could be used for training.

The ML-forward solver can also be tuned to other applications or trained to include more cases. FDTD can solve different scenarios but with a large computational cost, while ML can instantly predict responses but only for the scenarios it has been trained for and therefore it has to be tuned to different cases depending on the application. Moreover, different antenna polarisations can be investigated, apart from the E-field of the antenna being parallel to the long axis of the targets.

The ML solver currently predicts scenarios of concrete slabs with a single rebar. Improvements to the network and the training data can be made in order to include multiple targets, which can also be closely spaced together either horizontally or vertically to each other or even at an angle and thus, predict more realistic scenarios. Different network architectures and types of networks that might perform better with more complex scenarios can be investigated.

Different antenna elevations can also be used in the simulations for both schemes. Currently, in all the training data, the antenna is coupled to the ground. However, for some applications, such as in landmine detection, the antennas have to be lifted off ground and therefore, ML algorithms working for different antenna elevations would be beneficial. Additionally, including both simulated and real data for the training process can be considered, with the prerequisite that the real data are well-characterised. If this is not the case and the dielectric properties of the materials and targets included in the real scenarios are not accurately characterised, the performance of the ML models will be degraded.

Moreover, the performance of the algorithms can be examined by training the networks using data from different antenna models.

A

FDTD UPDATE EQUATIONS

$$\begin{aligned} E_x\Big|_{i+1/2,j,k}^{q+1/2} = \frac{1-\frac{\sigma\Delta t}{2\epsilon}}{1+\frac{\sigma\Delta t}{2\epsilon}} E_x\Big|_{i+1/2,j,k}^{q-1/2} + \frac{\frac{\Delta t}{\epsilon}}{1+\frac{\sigma\Delta t}{2\epsilon}} \Bigg(& \frac{H_z\Big|_{i+1/2,j+1/2,k}^{q} - H_z\Big|_{i+1/2,j-1/2,k}^{q}}{\Delta y} \\ & - \frac{H_y\Big|_{i+1/2,j,k+1/2}^{q} - H_y\Big|_{i+1/2,j,k-1/2}^{q}}{\Delta z} \\ & - J_{Sx}\Big|_{i+1/2,j,k}^{q} \Bigg) \end{aligned} \tag{A.1}$$

$$\begin{aligned} E_y\Big|_{i,j+1/2,k}^{q+1/2} = \frac{1-\frac{\sigma\Delta t}{2\epsilon}}{1+\frac{\sigma\Delta t}{2\epsilon}} E_y\Big|_{i,j+1/2,k}^{q-1/2} + \frac{\frac{\Delta t}{\epsilon}}{1+\frac{\sigma\Delta t}{2\epsilon}} \Bigg(& \frac{H_x\Big|_{i,j+1/2,k+1/2}^{q} - H_x\Big|_{i,j+1/2,k-1/2}^{q}}{\Delta y} \\ & - \frac{H_z\Big|_{i+1/2,j+1/2,k}^{q} - H_z\Big|_{i-1/2,j+1/2,k}^{q}}{\Delta z} \\ & - J_{Sy}\Big|_{i,j+1/2,k}^{q} \Bigg) \end{aligned} \tag{A.2}$$

$$\begin{aligned} E_z\Big|_{i,j,k+1/2}^{q+1/2} = \frac{1-\frac{\sigma\Delta t}{2\epsilon}}{1+\frac{\sigma\Delta t}{2\epsilon}} E_z\Big|_{i,j,k+1/2}^{q-1/2} + \frac{\frac{\Delta t}{\epsilon}}{1+\frac{\sigma\Delta t}{2\epsilon}} \Bigg(& \frac{H_y\Big|_{i+1/2,j,k+1/2}^{q} - H_y\Big|_{i-1/2,j-1/2,k}^{q}}{\Delta y} \\ & - \frac{H_x\Big|_{i,j+1/2,k+1/2}^{q} - H_x\Big|_{i,j-1/2,k+1/2}^{q}}{\Delta z} \\ & - J_{Sz}\Big|_{i,j,k+1/2}^{q} \Bigg) \end{aligned} \tag{A.3}$$

$$\begin{aligned} H_x\Big|_{i,j+1/2,k+1/2}^{q} = \frac{1-\frac{\sigma_m\Delta t}{2\mu}}{1+\frac{\sigma_m\Delta t}{2\mu}} H_x\Big|_{i,j+1/2,k+1/2}^{q-1} + \frac{\frac{\Delta t}{\mu}}{1+\frac{\sigma_m\Delta t}{2\mu}} \Bigg(& \frac{E_y\Big|_{i,j+1/2,k+1}^{q+1/2} - E_y\Big|_{i,j+1/2,k}^{q+1/2}}{\Delta z} \\ & - \frac{E_z\Big|_{i,j,k+1/2}^{q+1/2} - E_z\Big|_{i,j+1,k+1/2}^{q+1/2}}{\Delta y} \\ & - M_{Sx}\Big|_{i,j+1/2,k+1/2}^{q+1/2} \Bigg) \end{aligned} \tag{A.4}$$

$$\begin{aligned} H_y\Big|_{i+1/2,j,k+1/2}^{q} = \frac{1-\frac{\sigma_m\Delta t}{2\mu}}{1+\frac{\sigma_m\Delta t}{2\mu}} H_y\Big|_{i+1/2,j,k+1/2}^{q-1} + \frac{\frac{\Delta t}{\mu}}{1+\frac{\sigma_m\Delta t}{2\mu}} \Bigg(& \frac{E_z\Big|_{i+1,j,k+1/2}^{q+1/2} - E_z\Big|_{i,j,k+1/2}^{q+1/2}}{\Delta x} \\ & - \frac{E_x\Big|_{i+1/2,j,k}^{q+1/2} - E_x\Big|_{i+1/2,j,k}^{q+1/2}}{\Delta z} \\ & - M_{Sy}\Big|_{i+1/2,j,k+1/2}^{q+1/2} \Bigg) \end{aligned} \tag{A.5}$$

$$\begin{aligned} H_z\Big|_{i+1/2,j+1/2,k}^{q} = \frac{1-\frac{\sigma_m\Delta t}{2\mu}}{1+\frac{\sigma_m\Delta t}{2\mu}} H_z\Big|_{i+1/2,j+1/2,k}^{q-1} + \frac{\frac{\Delta t}{\mu}}{1+\frac{\sigma_m\Delta t}{2\mu}} \Bigg(& \frac{E_x\Big|_{i+1/2,j+1,k}^{q+1/2} - E_x\Big|_{i+1/2,j,k}^{q+1/2}}{\Delta y} \\ & - \frac{E_y\Big|_{i+1,j+1/2,k}^{q+1/2} - E_y\Big|_{i,j+1/2,k}^{q+1/2}}{\Delta x} \\ & - M_{Sz}\Big|_{i+1/2,j+1/2,k}^{q+1/2} \Bigg) \end{aligned} \tag{A.6}$$

B

INPUT GPRMAX FILES FOR BACKGROUND PREDICTION TRAINING SET

---

1. Background half-space + rebar

```
#domain: 0.300 0.300 0.400
#dx_dy_dz: 0.001 0.001 0.001
#time_window: 8e-9

#python:

import numpy as np
from user_libs.antennas.GSSI2000 import GSSI_2

# Rebar properties

d = 0.127 + 0.16*np.random.rand()            # Depth
x = 0.150                                     # X position
y = 0                                         # Y start
l = 0.300                                     # Y end - length
r = 0.01+0.025*np.random.rand()               # Radius

# Background properties

e1 = 2 + 20*np.random.rand()                  # Permittivity of
    background
s1 = 0.00001 + 0.5*np.random.rand()           # Conductivity of
    background

# Define materials

print("#material: {} {} 1 0 background".format(e1, s1))

# Set geometry

print("#box: 0 0 0.087 0.3 0.3 0.4 background")
print("#cylinder: {} {} {} {} {} {} {} pec".format(x, y, d, x, l, d
    , r))
```

```
# Load antenna model

GSSI_2(0.107, 0.110, 0.020)

#end_python:

#geometry_view: 0 0 0 0.300 0.300 0.400 0.001 0.001 0.001
    model_geometry n
```

2. Background layer + half-space

```
#domain: 0.300 0.300 0.400
#dx_dy_dz: 0.001 0.001 0.001
#time_window: 8e-9

#python:

import numpy as np
from user_libs.antennas.GSSI2000 import GSSI_2

# Background properties

e1 = 2 + 20*np.random.rand()                 # Permittivity of
    background
s1 = 0.00001 + 0.5*np.random.rand()          # Conductivity of
    background
bg_bottom = 0.107 +0.150*np.random.rand()    # Thickness of
    background layer

# Half-space properties

e2 = 2 + 20*np.random.rand()                 # Permittivity of
    half-space
s2 = 0.00001 + 0.5*np.random.rand()          # Conductivity of
    half-space

# Define materials

print("#material: {} {} 1 0 background".format(e1, s1))
print("#material: {} {} 1 0 half_space".format(e2, s2))

# Set geometry

print("#box: 0 0 0.087 0.3 0.3 {} background".format(bg_bottom))
```

```
print("#box: 0 0 {} 0.3 0.3 0.4 half_space".format(bg_bottom))

# Load antenna model

GSSI_2(0.107, 0.110, 0.020)

#end_python:

#geometry_view: 0 0 0 0.300 0.300 0.400 0.001 0.001 0.001
    model_geometry n
```

3. Background layer + half-space + rebar

```
#domain: 0.300 0.300 0.400
#dx_dy_dz: 0.001 0.001 0.001
#time_window: 8e-9

#python:

import numpy as np
from user_libs.antennas.GSSI2000 import GSSI_2

# Rebar properties

d = 0.127 + 0.16*np.random.rand()             # Depth
x = 0.150                                     # X position
y = 0                                         # Y start
l = 0.300                                     # Y end - length
r = 0.01+0.025*np.random.rand()               # Radius

# Background properties

e1 = 2 + 20*np.random.rand()                  # Permittivity of
    background
s1 = 0.00001 + 0.5*np.random.rand()           # Conductivity of
    background
bg_bottom = 0.107 +0.150*np.random.rand()     # Thickness of
    background layer

# Half-space properties

e2 = 2 + 20*np.random.rand()                  # Permittivity of
    half-space
```

```
s2 = 0.00001 + 0.5*np.random.rand()           # Conductivity of
    half-space

# Define materials

print("#material: {} {} 1 0 background".format(e1, s1))
print("#material: {} {} 1 0 half_space".format(e2, s2))

# Set geometry

print("#box: 0 0 0.087 0.3 0.3 {} background".format(bg_bottom))
print("#box: 0 0 {} 0.3 0.3 0.4 half_space".format(bg_bottom))

print("#cylinder: {} {} {} {} {} {} {} pec".format(x, y, d, x, l, d
    , r))

# Load antenna model

GSSI_2(0.107, 0.110, 0.020)

#end_python:

#geometry_view: 0 0 0 0.300 0.300 0.400 0.001 0.001 0.001
    model_geometry n
```

C

SCRIPTS FOR GENERATING THE ML FORWARD SOLVER TRAINING SET

1. Interpolation script

```
import numpy as np
from scipy import interpolate

# Debye properties of concrete

debye_properties = np.array([[4.8, 4.50, 0.82, 6.06e-4],
                             [6.75, 5.50, 2.28, 2.03e-3],
                             [8.63, 6.02, 1.00, 5.15e-3],
                             [9.14, 5.93, 0.80, 6.7e-3],
                             [11.19, 7.20, 0.73, 23e-3],
                             [12.84, 7.42, 0.611, 20.6e-3]])

# Number the rows

x = np.array(range(debye_properties.shape[0]))

# Set number of A-scans

no_Ascans = 4000

# Set interpolation points

no_data = np.linspace(0, x.max(), no_Ascans)

# Interpolation

f = interpolate.interp1d(x, debye_properties, axis=0, kind='cubic')

# Generate interpolation data

interp_data = f(no_data)

# Save properties to file
```

```
np.savetxt('debyeProp.txt', interp_data)
```

2. Input gprMax file

```
#domain: 0.300 0.300 0.400
#dx_dy_dz: 0.001 0.001 0.001
#time_window: 8e-9

#python:

import numpy as np
from user_libs.antennas.GSSI2000 import GSSI_2

# Rebar properties

d = 0.117 + 0.22*np.random.rand()             # Depth
r = 0.005 + 0.025*np.random.rand()            # Radius
x = 0.050 + 0.220*np.random.rand()            # X position
y = 0                                         # Y start
l = 0.300                                     # Y end - length

# Load file with the interpolated concrete debye properties

prop = np.genfromtxt('debyeProp.txt', delimiter=' ')

# Get debye properties for current model

ind = current_model_run      # Current A-scan running
es = prop[ind-1,0]           # Zero-frequency relative permittivity
einf = prop[ind-1,1]         # Relative permittivity at infinite
    frequency
t = prop[ind-1,2]*1e-9       # Relaxation time
s = prop[ind-1,3]            # Conductivity
de = es - einf

# Define materials

print("#material: {} {} 1 0 medium".format(einf,s))
print("#add_dispersion_debye: 1 {} {} medium".format(de, t))

# Set geometry

print("#box: 0 0 0.087 0.300 0.300 0.400 medium")
```

```
print("#cylinder: {} {} {} {} {} {} {} pec".format(x, y, d, x, l, d
    , r))

# Load antenna model

GSSI_2(0.117, 0.110, 0.020)

#end_python:

#geometry_view: 0 0 0 0.300 0.300 0.400 0.001 0.001 0.001
    model_geometry n
```

REFERENCES

Ieee standard definitions of terms for antennas. *IEEE Std 145-1993*, pages 1–32, 1993.

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

S. Abarbanel, D. Gottlieb, and J. Hesthaven. Long time behavior of the perfectly matched layer equations in computational electromagnetics. *Journal of Scientific Computing*, 17:405–422, 2002.

W. Al-Nuaimy, Y. Huang, M. Nakhkash, M. Fang, V. Nguyen, and A. Eriksen. Automatic detection of buried utilities and solid objects with gpr using neural networks and pattern recognition. *Journal of Applied Geophysics*, 43(2):157–165, 2000.

A. M. Alani, M. Aboutalebi, and G. Kilic. Applications of ground penetrating radar (gpr) in bridge deck monitoring and assessment. *Journal of Applied Geophysics*, 97:45–54, 2013. ISSN 0926-9851. Ground Penetrating Radar.

E. E. Altshuler and D. S. Linden. Wire-antenna designs using genetic algorithms. *IEEE Antennas and Propagation Magazine*, 39(2):33–43, 1997.

A. P. Annan. Ground-Penetrating Radar. In *Near-Surface Geophysics*, volume 13. Society of Exploration Geophysicists, 01 2005. ISBN 9781560801306.

A. P. Annan, W. M. Waller, D. W. Strangway, J. Rossiter, J. D. Redman, and R. Watts. The electromagnetic response of a low-loss, 2-layer, dielectric earth for horizontal electric dipole excitation. *Geophysics*, 40(2):285–298, 04 1975. ISSN 0016-8033.

P. Annan. *Ground penetrating radar principles, procedures and applications.* Sensors & Software, 2003.

C. Balanis. *Advanced Engineering Electromagnetics.* Wiley, 2012. ISBN 9781118214763.

C. Balanis. *Antenna Theory: Analysis and Design.* Wiley, 2015. ISBN 9781118585733.

V. Barrile and R. Pucinotti. Application of radar technology to reinforced concrete structures: a case study. *NDT & E International*, 38(7):596–604, 2005. ISSN 0963-8695.

E. Baysal, D. D. Kosloff, and J. W. C. Sherwood. Reverse time migration. *Geophysics*, 48(11):1514–1524, 1983.

A. Benedetto. A three dimensional approach for tracking cracks in bridges using gpr. *Journal of Applied Geophysics*, 97:37–44, 2013. ISSN 0926-9851. Ground Penetrating Radar.

A. Benedetto, F. Tosti, L. B. Ciampoli, and F. D'Amico. Gpr applications across engineering and geosciences disciplines in italy: A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(7): 2952–2965, 2016.

J.-P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(2):185–200, 1994. ISSN 0021-9991.

C. Bishop, P. Bishop, G. Hinton, and O. U. Press. *Neural Networks for Pattern Recognition.* Advanced Texts in Econometrics. Clarendon Press, 1995. ISBN 9780198538646.

P. Bohleber, L. Sold, D. R. Hardy, M. Schwikowski, P. Klenk, A. Fischer, P. Sirguey, N. J. Cullen, M. Potocki, H. Hoffmann, and P. Mayewski. Ground-penetrating radar reveals ice thickness and undisturbed englacial layers at kilimanjaro's northern ice field. *The Cryosphere*, 11(1):469–482, 2017.

T. Bourdi, J. E. Rhazi, F. Boone, and G. Ballivy. Modelling dielectric-constant values of concrete: an aid to shielding effectiveness prediction and ground-penetrating radar wave technique interpretation. *Journal of Physics D: Applied Physics*, 45(40):405401, sep 2012.

J. M. Bourgeois and G. S. Smith. A fully three-dimensional simulation of a ground-penetrating radar: Fdtd theory compared with experiment. *IEEE Transactions on Geoscience and Remote Sensing*, 34(1):36–44, 1996.

S. L. Brunton and J. N. Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control.* Cambridge University Press, 2019.

S. Busch, J. van der Kruk, J. Bikowski, and H. Vereecken. Quantitative conductivity and permittivity estimation using full-waveform inversion of on-ground GPR data. *Geophysics*, 77(6):H79–H91, 10 2012.

S. Busch, J. van der Kruk, and H. Vereecken. Improved characterization of fine-texture soils using on-ground gpr full-waveform inversion. *IEEE Transactions on Geoscience and Remote Sensing*, 52(7):3947–3958, 2014.

B. Cagnoli and T. Ulrych. Singular value decomposition and wavy reflections in ground-penetrating radar images of base surge deposits. *Journal of Applied Geophysics*, 48(3):175–182, 2001. ISSN 0926-9851.

M. J. Campbell and J. Ulrichs. Electrical properties of rocks and their significance for lunar radar observations. *Journal of Geophysical Research (1896-1977)*, 74 (25):5867–5881, 1969.

A. C. Cangellaris and D. Wright. Analysis of the numerical error caused by the stair-stepped approximation of a conducting boundary in fdtd simulations of electromagnetic phenomena. *IEEE Transactions on Antennas and Propagation*, 39(10):1518–1525, 1991.

N. Cassidy. Introduction to gpr. In *Workshop at the 12th International Conference on Ground Penetrating Radar*, 2008.

G. Chen and R. C. Liu. A 900mhz shielded bow-tie antenna system for ground penetrating radar. In *Proceedings of the XIII Internarional Conference on Ground Penetrating Radar*, pages 1–6, 2010.

W. C. Chew and W. H. Weedon. A 3d perfectly matched medium from modified maxwell's equations with stretched coordinates. *Microwave and Optical Technology Letters*, 7(13):599–604, 1994.

L. Conyers and D. Goodman. *Ground-Penetrating Radar: An Introduction for Archaeologists*. AltaMira Press, 1997.

D. J. Daniels. *Ground Penetrating Radar*. American Cancer Society, 2005. ISBN 9780471654506.

P. Debye. Polar molecules, ph.d. thesis. *New York: Chemical Catalog Co., p.172*, 48(43):1036–1037, 1929.

N. Diamanti and A. P. Annan. Characterizing the energy distribution around gpr antennas. *Journal of Applied Geophysics*, 99:83 – 90, 2013. ISSN 0926-9851.

N. Diamanti and D. Redman. Field observations and numerical models of gpr response from vertical pavement cracks. *Journal of Applied Geophysics*, 81: 106–116, 2012. ISSN 0926-9851. Recent, Relevant and Advanced GPR Studies in Applied Geophysics.

F. H. Drossaert and A. Giannopoulos. A nonsplit complex frequency-shifted pml based on recursive integration for fdtd modeling of elastic waves. *GEOPHYSICS*, 72(2):T9–T17, 2007.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12(null):2121–2159, jul 2011. ISSN 1532-4435.

M. Elsaadouny, J. Barowski, and I. Rolfes. Extracting the features of the shallowly buried objects using lenet convolutional network. In *2020 14th European Conference on Antennas and Propagation (EuCAP)*, pages 1–4, 2020. doi: 10.23919/EuCAP48036.2020.9135701.

J. R. Ernst, H. Maurer, A. G. Green, and K. Holliger. Full-waveform inversion of crosshole radar data based on 2-d finite-difference time-domain solutions of maxwell's equations. *IEEE Transactions on Geoscience and Remote Sensing*, 45(9):2807–2828, 2007.

E. Fisher, G. A. McMechan, A. P. Annan, and S. W. Cosway. Examples of reverse-time migration of single-channel, ground-penetrating radar profiles. *Geophysics*, 57(4):577–586, 1992.

P. Gamba and S. Lossani. Neural detection of pipe signatures in ground penetrating radar images. *IEEE Transactions on Geoscience and Remote Sensing*, 38(2):790–797, 2000. doi: 10.1109/36.842008.

J. Gazdag. Wave equation migration with the phase-shift method. *Geophysics*, 43(7):1342–1351, 1978.

S. Gedney. An anisotropic perfectly matched layer-absorbing medium for the truncation of fdtd lattices. *IEEE Transactions on Antennas and Propagation*, 44(12):1630–1639, 1996.

I. Giannakis, A. Giannopoulos, C. Warren, and N. Davidson. Numerical modelling and neural networks for landmine detection using ground penetrating radar. In *2015 8th International Workshop on Advanced Ground Penetrating Radar (IWAGPR)*, pages 1–4, 2015. doi: 10.1109/IWAGPR.2015.7292682.

I. Giannakis, A. Giannopoulos, and C. Warren. A machine learning approach for simulating ground penetrating radar. *Proc. 17th Int. Conf. Ground Penetrating Radar, Rapperswil-Jona, Switzerland,*, pages 1–4, 2018.

I. Giannakis, A. Giannopoulos, and C. Warren. Realistic fdtd gpr antenna models optimized using a novel linear/nonlinear full-waveform inversion. *IEEE Transactions on Geoscience and Remote Sensing*, 57(3):1768–1778, 2019a.

I. Giannakis, A. Giannopoulos, and C. Warren. A machine learning-based fast-forward solver for ground penetrating radar with application to full-waveform inversion. *IEEE Transactions on Geoscience and Remote Sensing*, 57(7): 4417–4426, 2019b.

I. Giannakis, F. Tosti, L. Lantini, and A. M. Alani. Diagnosing emerging infectious diseases of trees using ground penetrating radar. *IEEE Transactions on Geoscience and Remote Sensing*, 58(2):1146–1155, 2020.

I. Giannakis, A. Giannopoulos, C. Warren, and A. Sofroniou. Fractal-constrained crosshole/borehole-to-surface full-waveform inversion for hydrogeological applications using ground-penetrating radar. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–10, 2022. doi: 10.1109/TGRS.2021.3054173.

A. Giannopoulos. Modelling ground penetrating radar by gprmax. *Construction and Building Materials*, 19(10):755–762, 2005.

A. Giannopoulos. An improved new implementation of complex frequency shifted pml for the fdtd method. *IEEE Transactions on Antennas and Propagation*, 56(9):2995–3000, 2008. doi: 10.1109/TAP.2008.928789.

A. Giannopoulos. Unsplit implementation of higher order pmls. *IEEE Transactions on Antennas and Propagation*, 60(3):1479–1485, 2012. doi: 10.1109/TAP.2011.2180344.

A. Giannopoulos. Multipole perfectly matched layer for finite-difference time-domain electromagnetic modeling. *IEEE Transactions on Antennas and Propagation*, 66(6):2987–2995, 2018. doi: 10.1109/TAP.2018.2823864.

E. P. Gill, W. Murray, and H. M. Wright. *Practical Optimization.* Academic Press, 1981. ISBN 978-0122839528.

R. González-Drigo, V. Pérez-Gracia, D. Di Capua, and L. G. Pujades. Gpr survey applied to modernista buildings in barcelona: The cultural heritage of the college of industrial engineering. *Journal of Cultural Heritage*, 9(2):196–202, 2008. ISSN 1296-2074.

M. A. González-Huici, I. Catapano, and F. Soldovieri. A comparative study of gpr reconstruction approaches for landmine detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(12): 4869–4878, 2014.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

J. D. Griffiths. *Introduction to Electrodynamics (4th ed.)*. Cambridge University Press, Cambridge, 2017.

N. Gueting, A. Klotzsche, J. van der Kruk, J. Vanderborght, H. Vereecken, and A. Englert. Imaging and characterization of facies heterogeneity in an alluvial aquifer using gpr full-waveform inversion and cone penetration tests. *Journal of Hydrology*, 524:680–695, 2015. ISSN 0022-1694.

S. Hamran, D. Gjessing, J. Hjelmstad, and E. Aarholt. Ground penetrating synthetic pulse radar: dynamic range and modes of operation. *Journal of Applied Geophysics*, 33(1):7–14, 1995. ISSN 0926-9851.

R. L. Haupt. Thinned arrays using genetic algorithms. *IEEE Transactions on Antennas and Propagation*, 42(7):993–999, 1994.

R. L. Higdon. Absorbing boundary conditions for difference approximations to the multi-dimensional wave equation. *Mathematics of Computation*, 47(176): 437–459, 1986.

G. Hogan. Migration of ground penetrating radar data: A technique for locating subsurface targets. *Conference Proceedings, 1st EEGS Symposium on the Application of Geophysics to Engineering and Environmental Problems*, 1988.

K. Holliger, B. Lampe, U. Meier, M. Lambert, and A. G. Green. Realistic modeling of surface ground-penetrating radar antenna systems: where do we stand? In *Proceedings of the 2nd International Workshop onAdvanced Ground Penetrating Radar, 2003.*, pages 45–50, 2003.

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning*. Springer New York, New York, NY, 2013. ISBN 978-1-4614-7138-7.

S. W. Jaw and M. Hashim. Locational accuracy of underground utility mapping using ground penetrating radar. *Tunnelling and Underground Space Technology*, 35:20–29, 2013. ISSN 0886-7798.

N. Jin and Y. Rahmat-Samii. Parallel particle swarm optimization and finite-difference time-domain (pso/fdtd) algorithm for multiband and wide-band patch antenna designs. *IEEE Transactions on Antennas and Propagation*, 53 (11):3459–3468, 2005.

Jung-Ho Kim, Myeong-Jong Yi, Jeong-Sul Son, Seong-Jun Cho, and Sam-Gyu Park. Effective 3d gpr survey and its application to the exploration of old remains. volume 1, page 4 pp., 2005.

Jung-Ho Kim, Seong-Jun Cho, and Myeong-Jong Yi. Removal of ringing noise in gpr data by signal processing. volume 11, pages 75–81, 2007.

V. Kafedziski, S. Pecov, and D. Tanevski. Detection and classification of land mines from ground penetrating radar data using faster r-cnn. In *2018 26th Telecommunications Forum (TELFOR)*, pages 1–4, 2018. doi: 10.1109/TELFOR.2018.8612117.

A. A. Keller. *Mathematical Optimization Terminology: A Comprehensive Glossary of Terms*. Academic Press, 2017. ISBN 978-0128051665.

J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

I. Kim and W. Hoefer. A local mesh refinement algorithm for the time domain-finite difference method using maxwell's curl equations. *IEEE Transactions on Microwave Theory and Techniques*, 38(6):812–815, 1990. doi: 10.1109/22.130985.

N. W. Kim, J. Lee, H. Lee, and J. Seo. Accurate segmentation of land regions in historical cadastral maps. *Journal of Visual Communication and Image Representation*, 25(5):1262–1274, 2014. ISSN 1047-3203.

R. W. King. *Theory of Linear Antennas with Charts and Tables for Practical Applications*. Oxford University Press, 1956.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations*, abs/1412.6980, 2014.

A. Klotzsche, J. van der Kruk, G. Angelo Meles, J. Doetsch, H. Maurer, and N. Linde. Full-waveform inversion of cross-hole ground-penetrating radar data to characterize a gravel aquifer close to the thur river, switzerland. *Near Surface Geophysics*, 8(6):635–649, 2010.

A. Klotzsche, J. van der Kruk, N. Linde, J. Doetsch, and H. Vereecken. 3-D characterization of high-permeability zones in a gravel aquifer using 2-D crosshole GPR full-waveform inversion and waveguide detection. *Geophysical Journal International*, 195(2):932–944, 08 2013. ISSN 0956-540X.

G. Klysz, X. Ferrieres, J. Balayssac, and S. Laurens. Simulation of direct wave propagation by numerical fdtd for a gpr coupled antenna. *NDT & E International*, 39(4):338 – 347, 2006. ISSN 0963-8695.

E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, 8th Edition, New York, NY, 1999. ISBN 978-0471333289.

Y. Kuwahara. Multiobjective optimization design of yagi-uda antenna. *IEEE Transactions on Antennas and Propagation*, 53(6):1984–1992, 2005.

M. Kuzuoglu and R. Mittra. Frequency dependence of the constitutive parameters of causal perfectly matched anisotropic absorbers. *IEEE Microwave and Guided Wave Letters*, 6(12):447–449, 1996.

S. Lambot, E. Slob, I. van den Bosch, B. Stockbroeckx, B. Scheers, and M. Vanclooster. Gpr design and modeling for identifying the shallow subsurface dielectric properties. In *Proceedings of the 2nd International Workshop onAdvanced Ground Penetrating Radar, 2003.*, pages 130–135, 2003.

S. Lambot, E. C. Slob, I. van den Bosch, B. Stockbroeckx, B. Scheers, and M. Vanclooster. Estimating soil electric properties from monostatic ground-penetrating radar signal inversion in the frequency domain. *Water Resources Research*, 40(4), 2004.

S. Lameri, F. Lombardi, P. Bestagini, M. Lualdi, and S. Tubaro. Landmine detection from gpr data using convolutional neural networks. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 508–512, 2017. doi: 10.23919/EUSIPCO.2017.8081259.

B. Lampe and K. Holliger. Numerical modeling of a complete ground-penetrating radar system. In C. Nguyen, editor, *Subsurface and Surface Sensing Technologies and Applications III*, volume 4491, pages 99 – 110. International Society for Optics and Photonics, SPIE, 2001.

B. Lampe, K. Holliger, and A. G. Green. A finite-difference time-domain simulation tool for ground-penetrating radar antennas. *Geophysics*, 68(3):971–987, 01 2003. ISSN 0016-8033.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

K.-H. Lee, C.-C. Chen, F. Teixeira, and R. Lee. Modeling and investigation of a geometrically complex uwb gpr antenna using fdtd. *IEEE Transactions on Antennas and Propagation*, 52(8):1983–1991, 2004.

L. Lei, C. Zhong, L. Zhang, Q. Zhao, H. Yan, L. Fu, and S. Liu. Reverse time migration of crosswell gpr data based on wavefield decomposition. In *2018 17th International Conference on Ground Penetrating Radar (GPR)*, pages 1–4, 2018.

A. A. Lestari, A. G. Yarovoy, and L. P. Ligthart. Rc-loaded bow-tie antenna for improved pulse radiation. *IEEE Transactions on Antennas and Propagation*, 52(10):2555–2563, 2004.

A. A. Lestari, E. Bharata, A. B. Suksmono, A. Kurniawan, A. G. Yarovoy, and L. P. Ligthart. A modified bow-tie antenna for improved pulse radiation. *IEEE Transactions on Antennas and Propagation*, 58(7):2184–2192, 2010.

C. Leuschen and R. Plumb. A matched-filter-based reverse-time migration algorithm for ground-penetrating radar data. *IEEE Transactions on Geoscience and Remote Sensing*, 39(5):929–936, 2001.

S. Li, H. Cai, and V. R. Kamat. Uncertainty-aware geospatial system for mapping and visualizing underground utilities. *Automation in Construction*, 53:105–119, 2015. ISSN 0926-5805.

Z. Liao, K. Huang, B.-P. Yang, and Y.-F. Yuan. A transmitting boundary for transient wave analyses. *Science in China Series A-Mathematics, Physics, Astronomy & Technological Science*, 27:1063–1076, 1984.

C. Liu, C. Song, and Q. Lu. Random noise de-noising and direct wave eliminating based on svd method for ground penetrating radar signals. *Journal of Applied Geophysics*, 144:125–133, 2017. ISSN 0926-9851.

T. Liu, A. Klotzsche, M. Pondkule, H. Vereecken, Y. Su, and J. van der Kruk. Radius estimation of subsurface cylindrical objects from ground-penetrating-radar data using full-waveform inversion. *GEOPHYSICS*, 83(6):H43–H54, 2018.

D. Loewenthal, L. Lu, R. Roberson, and J. Sherwood. The wave equation applied to migration*. *Geophysical Prospecting*, 24(2):380–399, 1976.

J. Maloney, G. Smith, and W. Scott. Accurate computation of the radiation from simple antennas using the finite-difference time-domain method. *IEEE Transactions on Antennas and Propagation*, 38(7):1059–1068, 1990.

M. McFadden and W. R. Scott. Numerical modeling of a spiral-antenna gpr system. In *2009 IEEE International Geoscience and Remote Sensing Symposium*, volume 2, pages II–109–II–112, 2009.

G. A. Meles, J. Van der Kruk, S. A. Greenhalgh, J. R. Ernst, H. Maurer, and A. G. Green. A new vector waveform inversion algorithm for simultaneous updating of conductivity and permittivity parameters from combination crosshole/borehole-to-surface gpr data. *IEEE Transactions on Geoscience and Remote Sensing*, 48(9):3391–3407, 2010.

A. Monorchio, A. Bretones, R. Mittra, G. Manara, and R. Martin. A hybrid time-domain technique that combines the finite element, finite difference and method of moment techniques to solve complex electromagnetic problems. *IEEE Transactions on Antennas and Propagation*, 52(10):2666–2674, 2004.

G. Mur. Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations. *IEEE Transactions on Electromagnetic Compatibility*, EMC-23(4):377–382, 1981.

Z.-K. Ni, S. Ye, C. Shi, C. Li, and G. Fang. Clutter suppression in gpr b-scan images using robust autoencoder. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2020.

Y. Nishioka, O. Maeshima, T. Uno, and S. Adachi. Fdtd analysis of resistor-loaded bow-tie antennas covered with ferrite-coated conducting cavity for subsurface radar. *IEEE Transactions on Antennas and Propagation*, 47(6):970–977, 1999.

D. C. Nobes. Geophysical surveys of burial sites: A case study of the oaro urupa. *GEOPHYSICS*, 64(2):357–367, 1999. doi: 10.1190/1.1444540.

L. Nuzzo and T. Quarta. Improvement in gpr coherent noise attenuation using τ-p and wavelet transforms. *GEOPHYSICS*, 69(3):789–802, 2004.

M. L. Oristaglio and G. W. Hohmann. Diffusion of electromagnetic fields into a two-dimensional earth; a finite-difference approach. *Geophysics*, 49(7):870–894, 07 1984. ISSN 0016-8033.

M. F. Pantoja, A. R. Bretones, F. G. Ruiz, S. G. Garcia, and R. G. Martin. Particle-swarm optimization in antenna design: Optimization of log-periodic dipole arrays. *IEEE Antennas and Propagation Magazine*, 49(4):34–47, 2007.

O. Patsia, A. Giannopoulos, and I. Giannakis. A digital twin of the gssi 2000 mhz palm antenna developed using multi-parametric optimisation. In *Proceedings of the 11th International Workshop on Advanced Ground Penetrating Radar (IWAGPR), Valletta, Malta*, 2021a.

O. Patsia, A. Giannopoulos, and I. Giannakis. Full waveform inversion of common offset gpr data using a fast deep learning based forward solver. In *Proceedings of the 11th International Workshop on Advanced Ground Penetrating Radar (IWAGPR), Valletta, Malta*, 2021b.

O. Patsia, A. Giannopoulos, and I. Giannakis. A deep learning framework for ground penetrating radar. In *Proceedings of the 11th International Workshop on Advanced Ground Penetrating Radar (IWAGPR), Valletta, Malta*, 2021c.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

N. Peplinski, F. Ulaby, and M. Dobson. Dielectric properties of soils in the 0.3-1.3-ghz range. *IEEE Transactions on Geoscience and Remote Sensing*, 33 (3):803–807, 1995. doi: 10.1109/36.387598.

E. Pettinelli, B. Cosciotti, S. E. Lauro, and E. Mattei. An overview of GPR subsurface exploration of planets and moons. *The Leading Edge*, 41(10): 672–680, 10 2022.

D. Prescott and N. Shuley. A method for incorporating different sized cells into the finite-difference time-domain analysis technique. *IEEE Microwave and Guided Wave Letters*, 2(11):434–436, 1992. doi: 10.1109/75.165634.

H. W. Press and A. S. Teukolsky. *Numerical recipes in FORTRAN (2nd ed.)*. Cambridge University Press, New York, NY, 1992. ISBN 978-0-521-43064-7.

V. Pérez-Gracia, D. Di Capua, R. González-Drigo, and L. Pujades. Laboratory characterization of a gpr antenna for high-resolution testing: Radiation pattern and vertical resolution. *NDT & E International*, 42(4):336–344, 2009. ISSN 0963-8695.

L. Richardson. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210(459-470):307–357, 1911.

P. Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, 1998. ISBN 9780913478080.

P. J. Roache. Perspective: A Method for Uniform Reporting of Grid Refinement Studies. *Journal of Fluids Engineering*, 116(3):405–413, 09 1994.

R. L. Roberts and J. J. Daniels. Modeling near-field GPR in three dimensions using the FDTD method. *Geophysics*, 62(4):1114–1126, 08 1997. ISSN 0016-8033.

J. A. Roden and S. D. Gedney. Convolution pml (cpml): An efficient fdtd implementation of the cfs–pml for arbitrary media. *Microwave and Optical Technology Letters*, 27(5):334–339, 2000.

F. Sagnard, C. Norgeot, X. Derobert, V. Baltazart, E. Merliot, F. Derkx, and B. Lebental. Utility detection and positioning on the urban site sense-city using ground-penetrating radar systems. *Measurement*, 88:318–330, 2016. ISSN 0263-2241.

B. J. Schneider. *Understanding the Finite-Difference Time-Domain Method.* 2010.

W. A. Schneider. Integral formulation for migration in two and three dimensions. *Geophysics*, 43(1):49–76, 1978.

J. Schofield, D. Daniels, and P. Hammerton. A multiple migration and stacking algorithm designed for land mine detection. *IEEE Transactions on Geoscience and Remote Sensing*, 52(11):6983–6988, 2014.

S. Shihab, W. Al-Nuaimy, and A. Eriksen. Radius estimation for subsurface cylindrical objects detected by ground penetrating radar. In *Proceedings of the Tenth International Conference on Grounds Penetrating Radar, 2004. GPR 2004.*, volume 1, pages 319–322, 2004.

K. L. Shlager, G. S. Smith, and J. G. Maloney. Optimization of bow-tie antennas for pulse radiation. *IEEE Transactions on Antennas and Propagation*, 42(7): 975–982, 1994.

N. P. Singh and M. J. Nene. Buried object detection and analysis of gpr images: Using neural network and curve fitting. In *2013 Annual International Conference on Emerging Research Areas and 2013 International Conference on Microelectronics, Communications and Renewable Energy*, pages 1–6, 2013. doi: 10.1109/AICERA-ICMiCR.2013.6576024.

W. S. Smith. *Digital Signal Processing: A Practical Guide for Engineers and Scientists.* Newnes; 3rd Revised edition, 2002. ISBN 978-0750674447.

R. Solimene, A. Cuccaro, A. Dell'Aversano, I. Catapano, and F. Soldovieri. Ground clutter removal in gpr surveys. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(3):792–798, 2014. doi: 10.1109/JSTARS.2013.2287016.

M. Solla, H. González-Jorge, H. Lorenzo, and P. Arias. Uncertainty evaluation of the 1ghz gpr antenna for the estimation of concrete asphalt thickness. *Measurement*, 46(9):3032–3040, 2013. ISSN 0263-2241.

X. Song, D. Xiang, K. Zhou, and Y. Su. Improving rpca-based clutter suppression in gpr detection of antipersonnel mines. *IEEE Geoscience and Remote Sensing Letters*, 14(8):1338–1342, 2017.

S. Stadler and J. Igel. A numerical study on using guided gpr waves along metallic cylinders in boreholes for permittivity sounding. In *2018 17th International Conference on Ground Penetrating Radar (GPR)*, pages 1–4, 2018.

R. H. Stolt. Migration by Fourier transform. *Geophysics*, 43(1):23–48, 02 1978.

A. Taflove and S. C. Hagness. *Computational Electrodynamics: The Finite-Difference Time-Domain Method.* Boston: Artech House, 3 edition, 2000.

A. Tarantola. Linearized inversion of seismic reflection data. *Geophysical Prospecting*, 32(6):998–1015, 1984a.

A. Tarantola. Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266, 08 1984b.

E. Tebchrany, F. Sagnard, V. Baltazart, J. Tarel, and X. Dérobert. Assessment of statistical-based clutter reduction techniques on ground-coupled gpr data for the detection of buried objects in soils. In *Proceedings of the 15th International Conference on Ground Penetrating Radar*, pages 604–609, 2014.

E. Temlioglu and I. Erer. Clutter removal in ground-penetrating radar images using morphological component analysis. *IEEE Geoscience and Remote Sensing Letters*, 13(12):1802–1806, 2016.

F. H. C. Tivive, A. Bouzerdoum, and C. Abeynayake. Gpr target detection by joint sparse and low-rank matrix decomposition. *IEEE Transactions on Geoscience and Remote Sensing*, 57(5):2583–2595, 2019. doi: 10.1109/TGRS.2018.2875102.

M. Tong, M. Kuzuoglu, and R. Mittra. A new anisotropic perfectly matched layer medium for mesh truncation in finite difference time domain analysis. *International Journal of Electronics*, 86(9):1085–1091, 1999.

G. Tsokas, A. Giannopoulos, P. Tsourlos, G. Vargemezis, J. Tealby, A. Sarris, C. Papazachos, and T. Savopoulou. A large scale geophysical survey in the archaeological site of europos (northern greece). *Journal of Applied Geophysics*, 32(1):85–98, 1994. ISSN 0926-9851.

D. Uduwawala, M. Norgren, P. Fuks, and A. Gunawardena. A complete fdtd simulation of a real gpr antenna system operating above lossy and dispersive grounds. *Progress In Electromagnetics Research*, 50:209–229, 2005.

S. Valle, L. Zanzi, M. Sgheiz, G. Lenzi, and J. Friborg. Ground penetrating radar antennas: theoretical and experimental directivity functions. *IEEE Transactions on Geoscience and Remote Sensing*, 39(4):749–759, 2001.

C. M. J. van Coevorden, A. R. Bretones, M. F. Pantoja, F. J. G. Ruiz, S. G. Garcia, and R. G. Martin. Ga design of a thin-wire bow-tie antenna for gpr applications. *IEEE Transactions on Geoscience and Remote Sensing*, 44(4):1004–1010, 2006.

J. van der Kruk, T. Liu, A. Mozaffari, N. Gueting, A. Klotzsche, H. Vereecken, C. Warren, and A. Giannopoulos. Gpr full-waveform inversion, recent developments, and future opportunities. In *2018 17th International Conference on Ground Penetrating Radar (GPR)*, pages 1–6, 2018.

A. van der Merwe and I. J. Gupta. A novel signal processing technique for clutter reduction in gpr measurements of small, shallow land mines. *IEEE Transactions on Geoscience and Remote Sensing*, 38(6):2627–2637, 2000.

M. Varela-González, M. Solla, J. Martínez-Sánchez, and P. Arias. A semi-automatic processing and visualisation tool for ground-penetrating radar pavement thickness data. *Automation in Construction*, 45:42–49, 2014. ISSN 0926-5805.

P. K. Verma, A. N. Gaikwad, D. Singh, and M. J. Nigam. Analysis of clutter reduction techniques for through wall imaging. In *in UWB range," Progress In Electromagnetics Research B*, 2009.

A. Von Hippel. *Dielectrics and waves.* Cambridge, Mass, MIT Press, 3 edition, 1954.

W. Wai-Lok Lai, X. Dérobert, and P. Annan. A review of ground penetrating radar application in civil engineering: A 30-year journey from locating and testing to imaging and diagnosis. *NDT & E International*, 96:58–78, 2018. ISSN 0963-8695.

C. Warren. *Numerical modelling of high-frequency ground-penetrating radar antennas.* PhD thesis, The University of Edinburgh, 2009.

C. Warren and A. Giannopoulos. Creating finite-difference time-domain models of commercial ground-penetrating radar antennas using taguchi's optimization method. *GEOPHYSICS*, 76(2):G37–G47, 2011.

C. Warren and A. Giannopoulos. Characterisation of a ground penetrating radar antenna in lossless homogeneous and lossy heterogeneous environments. *Signal Processing*, 132:221 – 226, 2017. ISSN 0165-1684.

C. Warren, A. Giannopoulos, A. Gray, I. Giannakis, A. Patterson, L. Wetter, and A. Hamrah. A cuda-based gpu engine for gprmax: Open source fdtd electromagnetic simulation software. *Computer Physics Communications*, 237: 208 – 218, 2019. ISSN 0010-4655.

W. Warren, A. Giannopoulos, and I. Giannakis. gprmax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar. *Computer Physics Communications*, 209:163 – 170, 2016. ISSN 0010-4655.

W. A. Wenskink, G. Greeuw, J. Hofman, and J. K. VAN DEEN. Measured underwater near-field e-patterns of a pulsed, horizontal dipole antenna in air: Comparison with the theory of the continuous wave, infinitesimal electric dipole1. *Geophysical Prospecting*, 38(7):805–830, 1990. ISSN 1365-2478.

W. A. Wenskink, J. Hofman, and J. K. Van Deen. Measured reflection strengths of underwater pipes irradiated by a pulsed horizontal dipole in air: Comparison with continuous plane-wave scattering theory1. *Geophysical Prospecting*, 39(4): 543–566, 1991. ISSN 1365-2478.

K. Yee. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, 14(3):302–307, 1966.

R. Yelf. Where is true time zero ? In *Proceedings of the Tenth International Conference on Grounds Penetrating Radar, 2004. GPR 2004.*, volume 1, pages 279–282, 2004.

O. Yilmaz. *Migration*, chapter 4, pages 463–654. Society of Exploration Geophysicists, 2001.

R. Young and J. Sun. Revealing stratigraphy in ground-penetrating radar data using domain filtering. *Geophysics*, 64(2):435–442, 1999. cited By 31.

X. Zhang, L. Han, M. Robinson, and A. Gallagher. A gans-based deep learning framework for automatic subsurface object recognition from ground penetrating radar data. *IEEE Access*, 9:39009–39018, 2021. doi: 10.1109/ACCESS.2021.3064205.

S. Zhao and I. L. Al-Qadi. Development of an analytic approach utilizing the extended common midpoint method to estimate asphalt pavement thickness with 3-d ground-penetrating radar. *NDT & E International*, 78:29–36, 2016. ISSN 0963-8695.

S. Zivanovic, K. Yee, and K. Mei. A subgridding method for the time-domain finite-difference method to solve maxwell's equations. *IEEE Transactions on Microwave Theory and Techniques*, 39(3):471–479, 1991. doi: 10.1109/22.75289.