Transmission Line Matrix (TLM) modelling of medical ultrasound

Mansour Ahmadian

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
June 2001

Abstract

This thesis introduces TLM as a new method for modelling medical ultrasound wave propagation.

Basic TLM theory is presented and how TLM is related to Huygens principle is discussed. Two dimensional TLM modelling is explained in detail and one dimensional and three dimensional TLM modelling are explained.

Implementing TLM in single CPU computers and parallel computers is discussed and several algorithms are presented together with their advantages and disadvantages. Inverse TLM and modelling non linear wave propagation and different types of mesh are discussed.

A new idea for modelling TLM as a digital filter is presented and removing the boundary effect based on digital filter modelling of TLM is discussed.

Some modelling experiments such as :

- Focusing mirror.
- Circular mirror.
- Array transducers.
- Doppler effect.

are presented and how to use TLM to model these experiments is explained.

A new low sampling rate theory for TLM modelling is proposed and verified. This new theory makes the modelling of a much larger spaces practical on a given hardware platform.

Declaration of originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the Department of Electronics and Electrical Engineering at The University of Edinburgh except where otherwise indicated.

Mansour Ahmadian

Acknowledgements

I wish to acknowledge the assistance given to me by the following people:

Dr. Jimmy Dripps my supervisor, for the many hours given to this project over three years.

Dr. Steve McLaughlin, for his help and suggestion on this project and Professor Alan Murray, for his advice.

Professor Norman McDicken and Dr Warren Frazer Young and Dr. William Henry Nailon for their help on understanding medical ultrasound systems giving me ultrasound data.

I wish to acknowledge financial help that I received from Iranian government during my studying in Edinburgh university.

Finally, I would like to thank Jila, my wife who has supported me fully throughout this project, and has given birth to Ariasun, my first child, whose first recorded likeness can be seen in fig[2.9] on page 18. Thank you Jila for your love and support.

Dedication

This thesis is dedicated to my late father for his help and encouragement over the years. It was him who made it possible.

Contents

Declaration of originality . . . iii
Acknowledgements . . . iv
Dedication . . . v
Contents . . . vi
List of figures . . . x
List of tables . . . xv
Acronyms and abbreviations . . . xvi

1 Introduction **1**
1.1 Why it is necessary to model ultrasound wave propagation . . . 1
1.1.1 Ultrasound tissue characterization . . . 2
1.1.2 Ultrasound transducer design . . . 3
1.1.3 Doppler ultrasound . . . 4
1.2 TLM modelling . . . 4
1.3 The thesis organization . . . 5

2 Medical ultrasound **7**
2.1 Introduction . . . 7
2.2 General ultrasound imaging . . . 8
2.3 Different types of medical ultrasound systems . . . 10
2.3.1 Reflective and transmission based ultrasound . . . 12
2.3.2 Imaging format . . . 14
2.3.3 Ultrasound Transducers . . . 20
2.4 Doppler systems . . . 21
2.5 Conclusion . . . 27

3 Wave and sound theory **29**
3.1 Introduction . . . 29
3.2 Waves . . . 29
3.2.1 Basic wave theory and definitions . . . 29
3.2.2 Sound waves . . . 35
3.2.3 Electromagnetic waves . . . 35
3.2.4 Wave reflection and refraction . . . 36
3.2.5 Wave Scattering . . . 38
3.3 Transmission line . . . 40
3.3.1 Modelling Transmission lines with lumped components . . . 42
3.4 Conclusion . . . 45

4 TLM for modelling wave propagation and its implementation **47**
4.1 Introduction . . . 47
4.2 From the Huygens principle to TLM modelling: . . . 47
4.2.1 The Huygens principle . . . 48
4.2.2 TLM modelling . . . 48

4.3 Two dimensional TLM modelling . . . 51
4.3.1 TLM modelling of a homogeneous medium . . . 51
4.3.2 TLM modelling of a non homogeneous medium . . . 52
4.3.3 Dispersion of velocity of waves in a TLM mesh . . . 52
4.3.4 Wave propagation in a TLM mesh . . . 55
4.3.5 TLM mesh as a digital filter . . . 62
4.3.6 Implementing a TLM model . . . 64
4.4 One dimensional and three dimensional TLM modelling . . . 77
4.4.1 One dimensional TLM modelling . . . 77
4.4.2 Three dimensional TLM modelling . . . 78
4.5 Inverse TLM modelling . . . 79
4.6 Other topics in TLM modelling . . . 80
4.6.1 Modelling non linear waves with TLM . . . 80
4.6.2 Different type of meshes . . . 80
4.7 TLM modelling of ultrasound wave propagation . . . 80
4.8 Conclusions . . . 81

5 Experiments with TLM **83**
5.1 Introduction . . . 83
5.2 Experimental setup, input, output and notation . . . 83
5.2.1 The modelling setup . . . 83
5.2.2 Inputs . . . 84
5.2.3 Modelling output . . . 84
5.3 The speed of a wave in the mesh . . . 88
5.4 Boundary effect . . . 91
5.5 Wave interference . . . 96
5.6 Waves and objects . . . 98
5.6.1 Modelling wave reflection . . . 100
5.6.2 Modelling wave scattering . . . 100
5.6.3 Mixed signal and objects . . . 103
5.6.4 Curved objects . . . 109
5.7 Array transducers . . . 110
5.7.1 Wave propagation without any steering . . . 112
5.7.2 Wave propagation with steering to the left . . . 112
5.7.3 Wave propagation with steering to the right . . . 113
5.7.4 Steering in other angles . . . 113
5.8 Doppler effects . . . 114
5.8.1 Speed $\frac{1}{10}$ . . . 117
5.8.2 Sampling rate 50 . . . 117

6 Sampling rate **121**
6.1 Johns theory for sampling rate . . . 121
6.1.1 Examining wave speed for different sampling rates . . . 123
6.2 Systematic approach to TLM modelling . . . 126
6.2.1 Impulse response of the TLM mesh . . . 128
6.2.2 Transfer function for the TLM mesh . . . 129
6.3 New criteria for sampling rate based on system theory . . . 135

6.3.1 TLM mesh setup . . . 135
6.3.2 Processing . . . 136
6.3.3 Results . . . 136
6.4 New sampling theory . . . 142
6.4.1 Equal error sampling rates . . . 142
6.5 Testing the new sampling rate theory . . . 144
6.5.1 Case1: Sampling rate of 3.333 sample per cycle . . . 144
6.5.2 Case 2: Sampling rate is 2.222 sample per cycle . . . 144
6.6 Using new sampling rate for modelling wide band waves . . . 144
6.6.1 Using new sampling rate for modelling ultrasound wave propagation in medical systems . . . 147
6.6.2 Using new sampling rate for modelling wave propagation in nonlinear medium . . . 147
6.7 Conclusion . . . 148

7 Conclusion and further work **155**
7.1 Introduction . . . 155
7.2 The main results of this thesis . . . 155
7.2.1 TLM for modelling ultrasound wave propagation . . . 155
7.2.2 Using TLM to solve some modelling problems in medical ultrasound wave propagation . . . 155
7.2.3 New sampling rate theory . . . 156
7.3 Further work . . . 156
7.3.1 TLM as a digital filter . . . 156
7.3.2 Removing boundary effect . . . 156
7.3.3 Doppler effect in some complex situations . . . 156
7.3.4 New type of meshes . . . 157
7.4 Conclusion . . . 157

A Original publication **159**

B Developers guide to TLM classes **161**
B.1 Abstract . . . 161
B.2 Introduction . . . 161
B.3 CTlmBasic . . . 161
B.3.1 Definition: . . . 162
B.3.2 Base class . . . 163
B.3.3 Data types . . . 164
B.3.4 Global variables: . . . 164
B.3.5 Macros: . . . 164
B.3.6 Functions: . . . 164
B.3.7 Implementation: . . . 165
B.3.8 Programming: . . . 172
B.4 CTlmPPM: . . . 173
B.4.1 Definition: . . . 173
B.4.2 Base class: . . . 180
B.4.3 Data types . . . 180

B.4.4 Global variables: 180
B.4.5 Functions: 180
B.4.6 Implementation: 180
B.4.7 Programming: 187

References **188**

List of figures

1.1 An artery when it is normal and when it is abnormal [1]. 1
1.2 An ultrasound image from an artery [2]. 2
1.3 A simple method for tissue characterization [1]. 3

2.1 The block diagram of a typical medical ultrasound imaging system. 9
2.2 Some typical transducers that used normally in medical imaging [3]. 9
2.3 The block diagram of the signal processing section of a typical medical ultrasound imaging system (TGC =Time Gain Control). 10
2.4 A typical medical ultrasound system. The display section is a CRT monitor [4]. 11
2.5 A simplified transmission ultrasound system. 13
2.6 The basic principle of a reflective ultrasound system. 14
2.7 A sample A-Mode display. 15
2.8 A linear array transducer can be used to create a B-Mode image [5]. 16
2.9 A typical B-Mode display. 18
2.10 Some typical 3D images [6]. 20
2.11 A linear array transducer used to scan a medium [7]. 22
2.12 A typical linear array transducer [3]. 23
2.13 The steering process in a phased array transducer [7]. 24
2.14 A typical phased array transducer [3]. 25
2.15 The structure of a mechanical scan transducer [1]. 25

3.1 Graphical representation of a simple harmonic oscillation. Note that the X axis is time steps. 30
3.2 Longitudinal waves [8]. 33
3.3 Transverse waves [8]. 34
3.4 Reflection and refraction law. 37
3.5 A transmission line can be modelled as a series of interconnected lumped systems. 42
3.6 Modelling a small section of a transmission line by discrete lumped components. 43
3.7 Breaking down a T model to 3 segments. 44

4.1 Huygens principle. 49
4.2 Wave propagation in a two dimensional TLM mesh 50
4.3 Model for a node in a TLM mesh 52
4.4 Model for a node in a non homogeneous medium ($L \equiv \mu$ and $C \equiv \frac{1}{2} \times \varepsilon$) . . 53
4.5 Model for a node in a non homogeneous medium when the medium has some losses. 53
4.6 Equivalent circuit for a node in a non homogeneous medium. 54
4.7 Dispersion of the velocity of waves in a two-dimensional TLM network. 56
4.8 Wave will reach to point A at time $2\Delta T$ But the distance between point O and A is only $\sqrt{2}\Delta l$. 58
4.9 Wave will reach to point A at time $n\Delta T$ and the distance between point O and A is $n\Delta l$. 59

4.10 The distance between point A and source is $10\Delta l$. 59
4.11 A graphical demonstration of path RRRRRRRURDRR (path number 20 in table 4.2). 61
4.12 The TLM mesh can be modelled as an mesh of interconnected digital filters. . . 63
4.13 The digital filter realization for $\frac{1}{4}$ of a node. 64
4.14 The Digital filter realization for a node. 65
4.15 Node placement for scattering step. 67
4.16 A 3 dimension medium for modelling. 73
4.17 Division of the medium to sub-media. 74
4.18 Algorithm for modelling in a parallel computer. 75
4.19 Dead lock in modelling with a parallel computer. 76
4.20 TLM modelling in one dimension. 77
4.21 TLM modelling in three dimensions [9]. 78
4.22 Node model for scalar TLM modelling. 79

5.1 Medium setup for modelling example. 85
5.2 Signal at the source point and received signal for modelling example. 86
5.3 wave propagation in a medium by using Colour code drawing (Time difference between each two pictures is 5 ΔT). 87
5.4 wave propagation in a medium by plotting data in 3 dimensions (Time difference between each two picturess is 5 ΔT). 89
5.5 When the sampling rate is near 4 sample per cycle, wave propagation isn't the same in different directions. The sampling rate here is 5 sample per cycle.(Time difference between each two pictures is 5 ΔT). The setup for this experiment are as follow:Medium size:100 $\times$ 100 Source Position:(50,50) Receiver position: (60,50) 90
5.6 When the sampling rate is much higher than 4 sample per cycle, waves propagate in diagonal and axial direction with the same speed. The sampling rate here is 20 sample per cycle.(Time difference between each two pictures is 5 ΔT). The setup for this experiment are as follow:Medium size:100 $\times$ 100 Source Position:(50,50) Receiver position: (60,50) 90
5.7 The speed of the wave in the TLM mesh is $\frac{1}{\sqrt{2}}C$. C is the speed of wave in the real medium. The setup for this experiment are as follow:Medium size:100 $\times$ 100 Source Position:(50,50) Receiver position: (60,50) 91
5.8 Wave reflection from the boundary.(Time difference between each two pictures is 10 ΔT).The setup for this experiment are as follow: Medium size:200 $\times$ 200 Source Position:(150,100) Receiver position:(160,100). 92
5.9 To correctly terminating a medium we can terminate it to a matching Multi-input Multi-output IIR filter 93
5.10 Using the filter approximation as shown in equation 5.4 for removing reflection from boundary. (Time difference between each two pictures is 10 ΔT). The setup for this experiment are as follow: Medium size:200 $\times$ 200 Source Position:(150,100) Receiver position:(160,100). 95
5.11 Using the filter approximation as shown in equation 5.7 for removing reflection from boundary. (Time difference between each two pictures is 10 ΔT). The setup for this experiment are as follow: Medium size:200 $\times$ 200 Source Position:(150,100) Receiver position:(160,100). 96

5.12 Here all signals are shown in one graph. The signal between $14\Delta T$ and $114\Delta T$ is the original signal that comes directly from the transmitter. The reflected signal from the boundary arrives at the receiver at time $127\Delta T$. The setup for this experiment are as follow: Medium size:200 $\times$ 200 Source Position:(150,100) Receiver position:(160,100). 97
5.13 When there are two waves in a medium, they interfere with each other. In this modelling example, the frequency and amplitude of the two wave are the same. Time difference between each two pictures is $10\Delta T$. The setup for this experiment are as follow: Medium size:100 $\times$ 100 Source 1 Position:(30,1) Source 2 Position:(70,1). 98
5.14 The received signals at receivers when there isn't any object in the medium. . . 99
5.15 When the size of an object in the medium is bigger than the wavelength, the wave is reflected by the object. There is a shadow at the back of the object and a reflected wave in front of the object. Time difference between each two pictures is $5\Delta T$. The object is an rectangular object from (40,70) to (60, 130). The object size as seem by the wave is $60\Delta T$. 101
5.16 The signals at the receivers when the size of the object is bigger than the wave length. Since There is a shadow at the back of the object, the signals at the receivers behind the object are very weak. The object is an rectangular object from (40,70) to (60, 130). The object size as seem by the wave is $60\Delta T$ 102
5.17 The received signals at point (15,100). The object is an rectangular object from (40,70) to (60, 130). The object size as seem by the wave is $60\Delta T$ 103
5.18 When the size of an object in the medium is smaller than the wavelength, waves are scattered by the object. If the object size is very small compared to the wave length then the object acts as a point source and will generate waves in all directions. The time difference between each two pictures is $5\Delta T$. The object is an rectangular object from (40,99) to (60, 101). The object size as seem by the wave is $2\Delta\ell$ 104
5.19 The signals at receivers when the object in the medium is small compared with the wave length. Since there isn't any shadow, the received signals on all of the objects are strong (Compared with the reflection case). It should be noted that since the scattered signal from the object is very weak, it can't be seen in this figure. The object is an rectangular object from (40,99) to (60, 101). The object size as seem by the wave is $2\Delta\ell$ 105
5.20 The received signals at point (25,100) when there is an object in the medium and when there isn't any object in the medium. The object is an rectangular object from (40,99) to (60, 101). The object size as seem by the wave is $2\Delta\ell$. 106
5.21 The source signal for added signal experiment 106
5.22 The received signal at point (65,100) when the two signal added to each other. . 107
5.23 The wave propagation shape when the two input signals added to each other. . . 108
5.24 The source signal for the amplitude modulation experiment 108
5.25 The received signal at point (65,100) when there is an amplitude modulated signal in the medium. 109
5.26 The wave propagation shape when the there is an amplitude modulated signal in the medium. 110
5.27 Wave propagation shape with a focusing mirror in the medium (time difference between each two pictures is $15\Delta T$). 111

5.28 Wave propagation shape in a circular mirror (time difference between each two pictures is $20\Delta T$). 111
5.29 Wave propagation shape for an array transducer when there isn't any steering. (time difference between each two pictures is $10\Delta T$). 112
5.30 Here the wave is steered to the left. The steering angle is approximately is 45^0. (time difference between each two pictures is $10\Delta T$). 113
5.31 Steering to the right shows here. The steering angle is approximately is -45^0. (time difference between each two pictures is $10\Delta T$). 114
5.32 The setup for modelling the Doppler effect. 115
5.33 The received signals in the time domain when the speed is $\pm\frac{1}{5}$. 116
5.34 The received signals in frequency domain when the speed is $\pm\frac{1}{5}$. 116
5.35 The received signal in time domain when the speed is $\pm\frac{1}{10}$. 117
5.36 The received signal in frequency domain when the speed is $\pm\frac{1}{10}$. 118
5.37 The received signal in frequency domain when the speed is $\pm\frac{1}{5}$ and $\pm\frac{1}{10}$. . . . 118
5.38 The received signal in the time domain when the speed $= \pm\frac{1}{5}$ and the sampling rate is 50 samples per cycle. 119
5.39 The received signal in frequency domain when the speed is $\pm\frac{1}{5}$ and the sampling rate is 50 samples per cycle. 119

6.1 Relationship between sampling rate and the wave propagation speed in a TLM mesh. 123
6.2 A closer look at the relationship between the sampling rate and the wave propagation speed near 4 samples per cycle. 124
6.3 The setup for examining the wave propagation speed in the TLM medium. . . . 125
6.4 Received signal at axial and diagonal receiver when the sampling rate is 20 sample per cycle. 125
6.5 Wave propagation shape when the sampling rate is 20 samples per cycle (time difference between each two pic is $5\Delta T$). 126
6.6 Received signal at axial and diagonal receiver when the sampling rate is 5 sample per cycle. 127
6.7 Wave propagation shape when the sampling rate is 5 samples per cycle (time difference between each two pic is $5\Delta T$). 127
6.8 The impulse function used to find the transfer function of the TLM mesh. . . . 128
6.9 Impulse response when the source is at (50,50) and the receiver is at (55,65). . . 129
6.10 Transfer function when the source is at (50,50) and the receiver is at (55,65). . . 131
6.11 Transfer function when the source is at (50,50) and the receiver is at (55,65). Note that the X axis converted to samples per cycle. 134
6.12 The setup for finding a new criteria for sampling rate 135
6.13 The received signal at the axial receiver. 136
6.14 The received signal at the diagonal receiver. 137
6.15 Frequency domain transfer function when the source is at (50,50) and the receiver is at (50,64). Note that the X axis converted to samples per cycle. 138
6.16 frequency domain transfer function when the source is at (50,50) and the receiver is at (60,605). Note that the X axis converted to samples per cycle. . . . 139
6.17 Amplitude for both transfer functions in frequency domain. 140
6.18 Phase for both transfer functions in frequency domain. 141
6.19 Sampling rate is 3.333. 145

6.20 Wave propagation shape when the sampling rate is 3.333. 145
6.21 Sampling rate is 2.222. 146
6.22 Wave propagation shape when the sampling rate is 2.222. 146

List of tables

4.1 Paths from source to point as shown in fig[4.8] with time delay of $12\Delta T$ 57
4.2 Paths from source to point as shown in fig[4.10] with time delay of $12\Delta T$ . . . 61
4.3 Comparison between received signal with different time delay. 62
4.4 Similarity between Electromagnetic waves and Ultrasound waves 81

6.1 Conversion between bin number in a DFT and sampling rate for a DFT with 128 samples. 133
6.2 Comparison between different sampling rate for modelling system explained in the example. 153

Acronyms and abbreviations

TLM	Transmission Line Matrix
DFT	Digital Fourier Transform
FFT	Fast Fourier Transform
IVUS	Intra Vascular UltraSound
TGC	Time Gain Compensation
AM	Amplitude Modulation

Chapter 1
Introduction

Ultrasound is widely used in many areas in medicine. It provides a safe and efficient means for diagnostics and therapy. Unfortunately our knowledge of how ultrasound waves propagate and interact in a complex medium such as body is limited.

1.1 Why it is necessary to model ultrasound wave propagation

We have enough knowledge of how ultrasound waves propagate in a simple medium, but when the medium becomes complex with several objects of different sizes and shapes, solving the wave propagation formula becomes virtually impossible [10]. Inside the body the case is much more complex since not only each organ has its own shape and size, the ultrasound propagation speed is different for each tissue. Modelling becomes much more complex when one knows that tissues are not a homogenous medium for ultrasound wave propagation. Fig[1.1] shows an artery. As can be seen from this fig, even a specific place in the body has different shapes and properties when it is normal and when it is abnormal. Physicians are very interested to detect abnormalities in arteries, so that they can prescribe the right medicine to the patient. One good

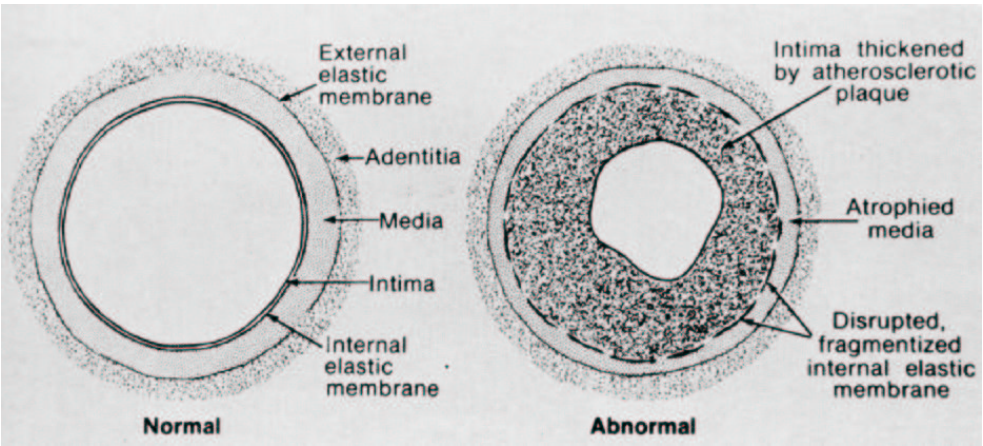


Figure 1.1: *An artery when it is normal and when it is abnormal [1].*

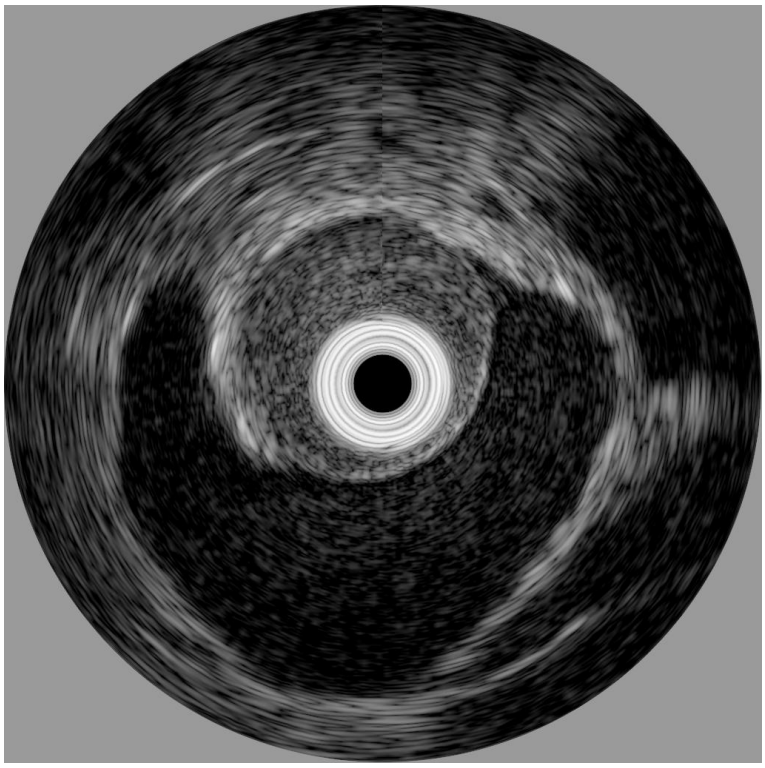


Figure 1.2: *An ultrasound image from an artery [2].*

solution is to use ultrasound imaging[1]. An ultrasound image of an artery is shown in fig[1.2].

1.1.1 Ultrasound tissue characterization

One of the active research topics in medical ultrasound systems is tissue characterization with ultrasound [11], which is determining the type of tissue by using an ultrasound image. The simplest way is to use the gray level of the ultrasound image as an indication of the tissue type. For example, in fig[1.3], a simple method to determine the type of plague in an artery (see fig[1.2]) is shown. This method is very simple but inaccurate.

There are several other techniques for tissue characterization. The most common is spectrum analysis [12], [13] which is based on the fact that the spectrum of the back scattered ultrasound signal depends on the type of tissue being imaged. There is not a clear understanding why and how each tissue changes the spectrum of the received signal. One other problem in tissue characterization with spectrum analysis is that the received signal is affected by the intervening tissue in the propagation path. This effect may change the result dramatically. Unfortunately

[1]This kind of ultrasound imaging called Intra Vascular Ultra Sound (IVUS).

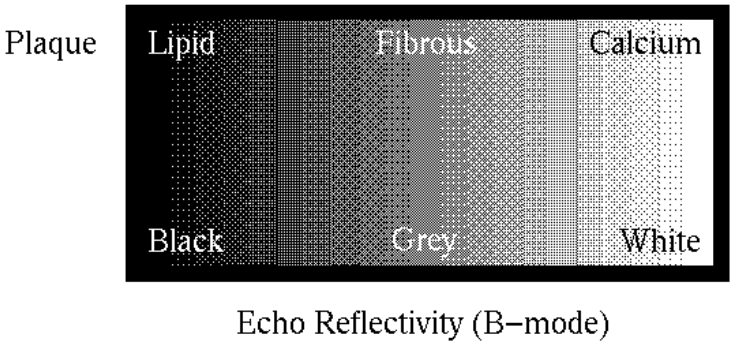


Figure 1.3: *A simple method for tissue characterization [1].*

knowledge of the effect of other tissue in the ultrasound wave propagation path is poor.

Since it is not possible to model the ultrasound wave propagation mathematically[2] , it is not possible to predict the effect of each tissue on the back of the tissue being imaged for the diagnostic purpose [14]. One way to solve this problem is to model the ultrasound wave propagation numerically.

1.1.2 Ultrasound transducer design

The accuracy and performance of an ultrasound system depends on its ultrasound transducer[3]. The shape of an ultrasound transducer determines the shape of its transmitted propagation beam. To understand exactly how an ultrasound image is created one should know how the ultrasound wave is generated and the ultrasound wave beam shape.

There are some expensive pieces of equipment for finding the beam shape of an ultrasound transducer. To use this equipment, one should make the transducer and then test its beam shape.

The situation is more complex for array transducers since for these types of transducers, the beam shape not only depends on each array element but also on when each element is fired compared its neighbours.

The best way to make an ultrasound transducer is to model it to predict the beam shape and

[2]The body structures are so complex that it is not possible to model ultrasound wave propagation mathematically in the body.

[3]The electronic section of an ultrasound system can be modelled in several ways and there are several good commercial software packages available on the market to do this.

then to adjust the transducer until the beam shape is close equal to the desired one. Only then the transducer should be made.

1.1.3 Doppler ultrasound

By using the Doppler effect, some ultrasound systems detect movement[4] and determine the speed of a movement in the body[5]. In all of these examples, the Doppler shift is very complex since the movement may be very complex. To find the exact speed of blood flow, for example, one should know how this complex movement will generate Doppler shifts.

Since the Doppler shift depends on movement and the movement is very complex, modelling the Doppler effect in these cases is very complex. This complexity makes it very difficult or some times impossible to solve the problem mathematically.

1.2 TLM modelling

Transmission line matrix (TLM) modelling is a numerical modelling technique for wave propagation [15]. It is used extensively in modelling electromagnetic wave propagation [16]. As it is a model for wave propagation, it can be used to solve several of the problems[6] in ultrasound wave propagation in medical systems[7].

The work that presented in this thesis is to demonstrate how to use the Transmission Line Matrix (TLM) for modelling ultrasound wave propagation. This modelling can give us a more detailed understanding of how ultrasound waves are reflected and scattered by different tissues in the body.

Although TLM modelling was originally developed for modelling electromagnetic wave propagation, there has been some limited work on using TLM for modelling acoustic waves [17], [18], [19] . Since electromagnetic waves and ultrasound waves are very similar in propagation behaviour, a table to show how these two waves are related to each other is presented (see chapter 4).

[4]For example detect the heart movement of a fetus in a pregnant women.
[5]For example to determine the speed of blood flow in a vessel
[6]In this thesis, TLM will be used to model array transducers and the Doppler effect.
[7]It can be used for other types of ultrasound modelling to, but only medical ultrasound is presented here.

1.3 The thesis organization

This thesis consists of 7 chapters. The first 3 chapters contains background information. In chapter 1 (this chapter), the reason for doing this work is explained.

Chapter 2 is about medical ultrasound systems. It is useful for any engineer who hasn't a background in medical ultrasound systems. It contains some information about how an ultrasound imaging system works and what type of ultrasound imaging systems are there. Some types of ultrasound transducers are also explained in this chapter.

The basic theory of wave propagation and in particular sound/electromagntic waves propagation is presented in Chapter 3. In this chapter, the basic idea of a transmission line is also explained.

In chapter 4 TLM will be introduced. In this chapter, the mathematical theory behind TLM is given and by focusing on 2 dimensional TLM modelling the properties of this modelling technique will be explained. Some algorithms for TLM modelling will be presented in this chapter and advantages and disadvantages of each algorithm are explained. One dimensional and 3 dimensional TLM is also introduced and inverse TLM is discussed. Since TLM was originally used for electromagnetic wave propagation modelling, use for ultrasound systems is explained.

In chapter 5, Some modelling results will be presented. This modelling is achieved with modular software which can be used as the basis for modelling complex problems.

Since the the TLM modelling is a computerized modelling, the amount of memory and computational power that needs depends on the wavelength of the wave in the medium. Since the ultrasound wave propagation speed in the body is not very fast[8] the wavelength of the ultrasound waves in the body is very short. This short wavelength makes the TLM modelling very complex and doing realistic modelling is nearly impossible. To solve this problem a new sampling theory is presented in chapter 6. The result of this sampling theory shows that it is possible to model realistic ultrasound wave propagation with computers available now.

[8]Comparing to the speed of electromagnetic wave in the air.

Chapter 2
Medical ultrasound

2.1 Introduction

Ultrasound waves are widely used in medicine [20],[21]. As diagnostic ultrasound represents no general hazard (but some specific area of hazard) to the health of a patient, its usage is growing rapidly[22] [23] [24]. Ultrasound is well known for imaging but it has several other uses in medicine such as:

- Movement detection
- Medical Therapy
- Lithotripsy

Ultrasound imaging, the most important and widely known application of ultrasound, will be explained in this chapter. It should be noted that the results of this research can be used for modelling ultrasound wave propagation in other applications too.

The principal attractions of medical ultrasound imaging are:

Hazard risk: It is considered by medical experts to represent no hazard to the health of a patient[24].

Price: It is cheaper than other imaging systems [25].

Suitability for soft tissue: It is good for imaging soft tissue structures in detail [2].

Portability: Portable ultrasound systems are available.

Measurement of movement: It is possible to measure movement within the body , for example the movement of blood in the heart or in the arteries.

Ultrasound imaging is considered to be safe[26]. Although some reports have been presented to indicate that ultrasound is hazardous, these reports have proven to be unfounded [24]. Medical experts currently believe that exposing a patient to the clinical level of ultrasound has no danger to the patient[26]. This provides the opportunity for repeated clinical examination of a patient without worry about the side effects.

2.2 General ultrasound imaging

Ultrasound can be used to construct images of the human body due to the interaction between the body tissue and the ultrasound vibration [27] [28] [29]. It is used for imaging many parts of the body including:

- Heart
- Fetus
- Vascular system
- other small parts such as kidneys and liver

In fig[2.1] a typical ultrasound system is shown. As seen from this figure, a typical ultrasound system consists of 3 parts:

Transducer: Transducer is used for producing ultrasound waves and detecting back scattered ultrasound waves. Transducers are piezoelectric crystals which are designed to vibrate at a specific frequency. This frequency is called the natural frequency of the transducer. For medical systems, this frequency is typically 1 MHz to 30 MHz. The ultrasound will be created by applying a short duration voltage pulse to the transducer.The duration of this pulse is selected in such a way that the produced signal is only for a few cycles duration. After the ultrasound wave is created, the transducer changes its mode to act as a receiver. In this mode the transducer detects the ultrasound waves and converts them to corresponding voltages. These voltages will be used by electronic circuits in the data acquisition and processing section to generate an image. Some typical transducers that are normally use in medical imaging are shown is fig[2.2].

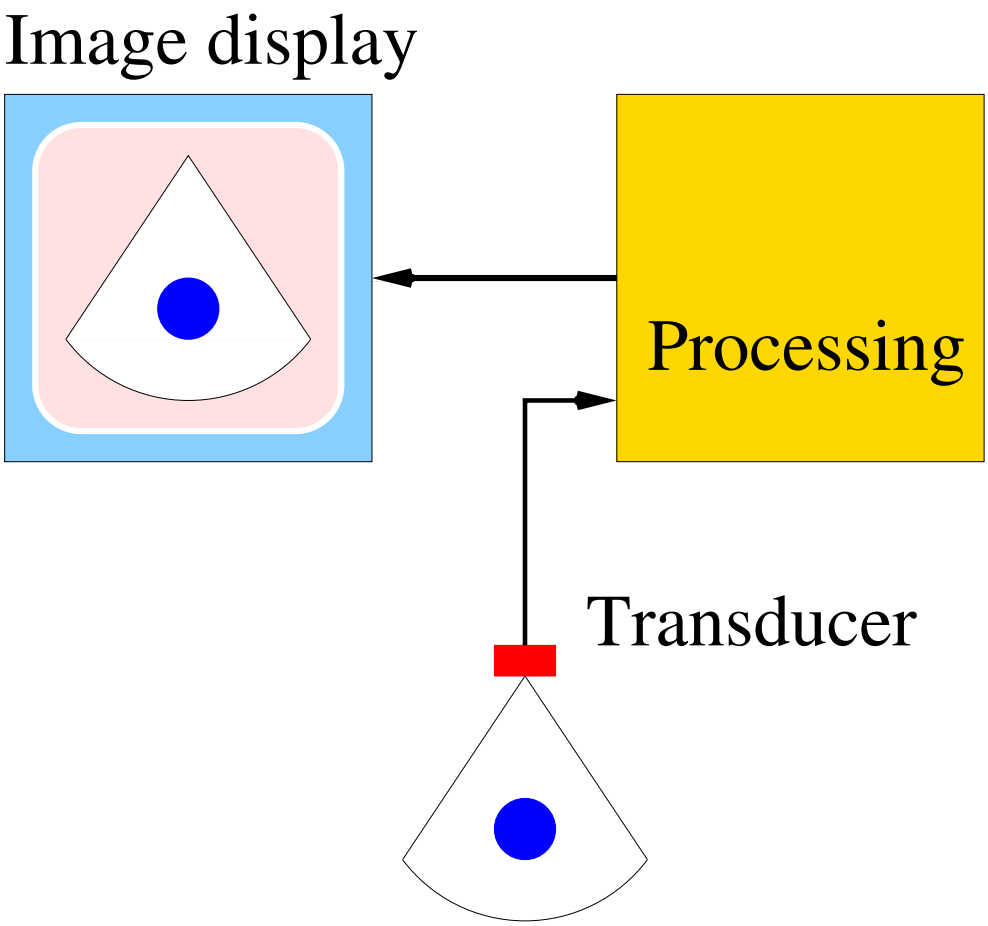


Figure 2.1: *The block diagram of a typical medical ultrasound imaging system.*

Figure 2.2: *Some typical transducers that used normally in medical imaging [3].*

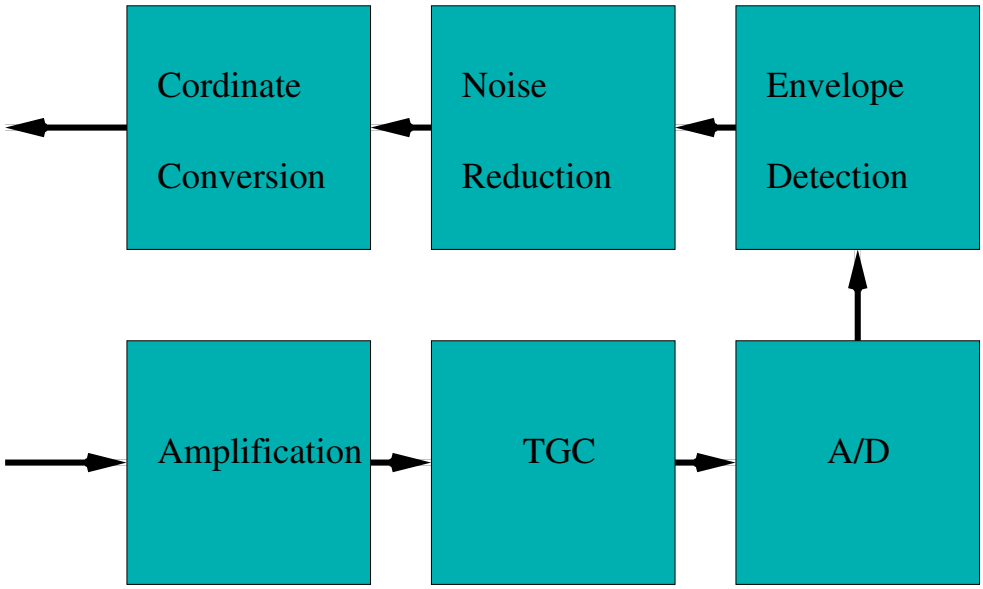


Figure 2.3: *The block diagram of the signal processing section of a typical medical ultrasound imaging system (TGC =Time Gain Control).*

Data acquisition and processing: In this section, the received signal from the transducer is first amplified based on the time that it is received and filtered to remove some additive noise. Then an envelope detector is used to detect the envelope of the amplified signal. The output of the envelope detector is digitised and the digitised data is processed to create the required image. A typical ultrasound system block diagram is shown in fig[2.3]

Display: Typically a CRT monitor is used to display the generated images. Since the ultrasound images are high resolution, these CRT monitors are high resolution monitors. Some ultrasound systems can print the images on a thermal printer and some others can print them on normal paper. Since the physicians use the monitor to see the images and in most cases they do their diagnostics based on their observation, these displays should be very accurate and with good quality. In fig[2.4] a normal ultrasound system is shown.

2.3 Different types of medical ultrasound systems

There are several types of medical ultrasound system. Each of these types is designed to suite a specific usage in medical imaging. For example there are some ultrasound imaging systems

Figure 2.4: *A typical medical ultrasound system. The display section is a CRT monitor [4].*

which are designed to monitor heart activity while some others are designed to monitor foetal and pregnant woman. Different ultrasound systems use different transducers (frequency, shape, number of elements if it is an array, scanning type, etc.) , displaying mode (A-Mode, B-Mode, M-Mode and 3D) and the way that the image is created (Reflection or transmission imaging).

2.3.1 Reflective and transmission based ultrasound

An ultrasound system creates an image by calculating how an ultrasound wave is changed when it propagates through the body. There are two ways that an ultrasound system can calculate the way that an ultrasound wave is propagating in a medium:

Transmission: The image is produced based on the amount of wave that is transmitted thought the body.

Reflection: The reflected wave from different parts of the body is used to create an image.

2.3.1.1 Transmission ultrasound imaging

Each part of the body absorbs a specific amount of ultrasound wave energy. The transmission ultrasound imaging system uses this fact and measure the amount of wave incident to the receiver and creates an image based on it. In fig[2.5], a simplified transmission ultrasound system is shown [30] [31].

In this type of ultrasound systems, there are two transducers, one for generating the ultrasound wave and the other for detecting the ultrasound wave. The patient body will be placed between the transmitter and the receiver transducers. The received signal strength at the receivers is measured and based on these measurements the image is created. The time taken for an ultrasound pulse to pass through the system (from the transmitter to the receiver) is a transmission based system in given by equation[2.1]

$$t = \frac{d}{C} \tag{2.1}$$

Where d is the distance between the transmitter and the receiver and C is the speed of sound in the medium. It is worth noting that the transmitter can generate several pulses before the

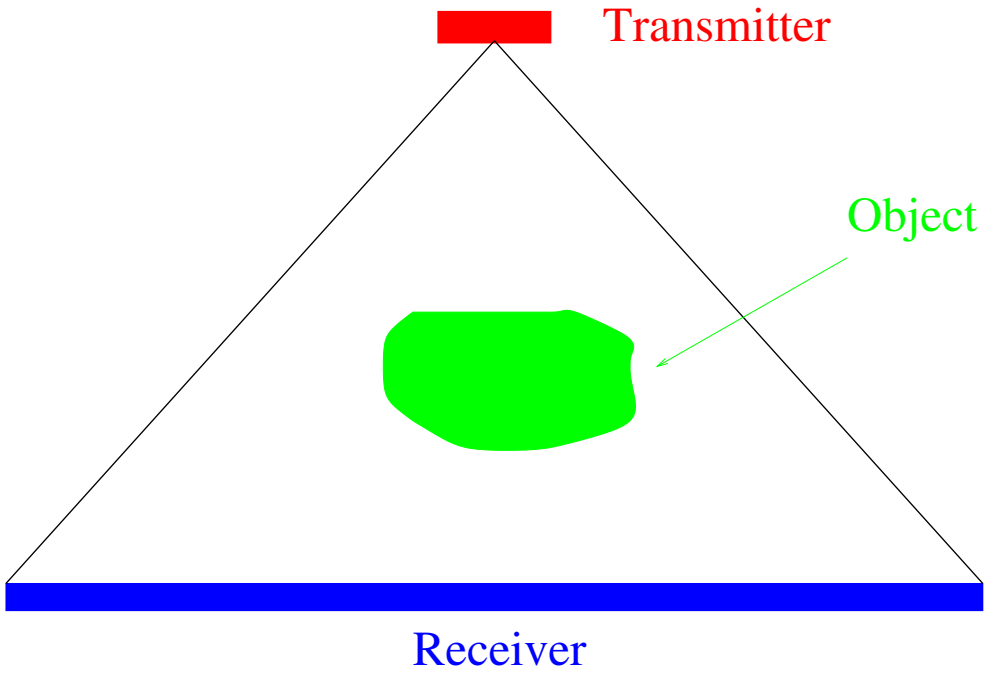


Figure 2.5: *A simplified transmission ultrasound system.*

receiver detect the first one[1]. In this case the frame rates of the ultrasound images are very high.

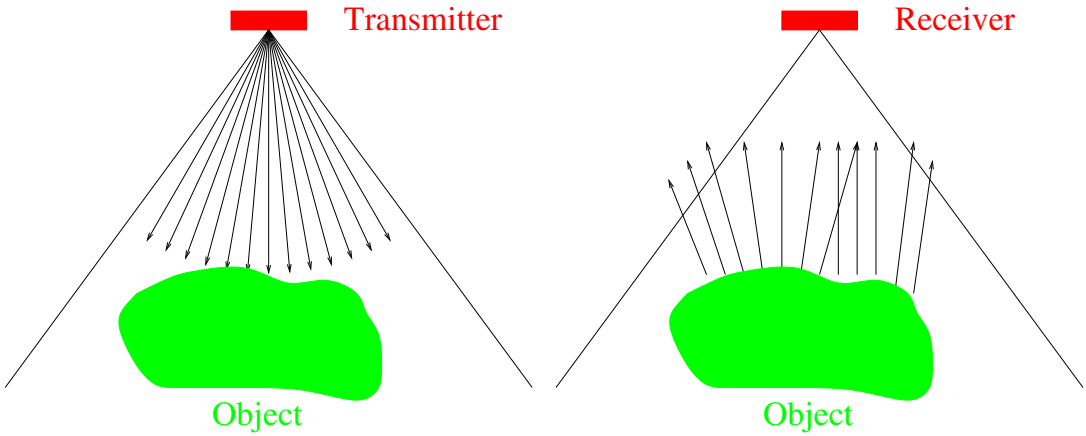
2.3.1.2 Reflective ultrasound imaging

The speed of an ultrasound wave in each part of the body is different from any other part of the body. When an ultrasound wave is reaches the interface of two parts of the body, a small part of the incident energy will be reflected back. The amount of reflected ultrasound depends on the relative impendence of the two part of the body. This amount can be calculated by using equation 2.2.

$$|A_{reflected}| = \left|\frac{Z_2 - Z_1}{Z_1 + Z_2}\right| \times |A_{incident}| \tag{2.2}$$

where Z_2 is the impedance of the body part that sound is trying to enter and Z_1 is the impedance of the material that the sound wave is trying to leave. In a reflective ultrasound system this reflected (or scattered back) signal is measured and an image is created based on it. Erroneous

[1]The time difference between each two pulse should be so that the first pulse would be received by all the receivers before the next pulse received by any of them.



(a) Transmitting phase

(b) Receiving phase

Figure 2.6: *The basic principle of a reflective ultrasound system.*

multiple reflection can occur and be detected. Distinguishing multiple reflections from normal single reflection can be very difficult and some times impossible. In fig[2.6] the basic principle of a reflective ultrasound system is shown.

2.3.2 Imaging format

There are several way to display ultrasound data and hence several imaging formats for ultrasound systems. These imaging formats are:

1. A-Mode imaging.
2. B-Mode imaging.
3. M-Mode imaging.
4. 3D and 4D imaging.

These methods for displaying ultrasound images will be explained in the following sections.

2.3.2.1 A-Mode imaging

A-Mode or amplitude mode is the simplest form of ultrasound imaging. It is a one dimensional imaging technique and has limited applicability. As it is the simplest way to display the

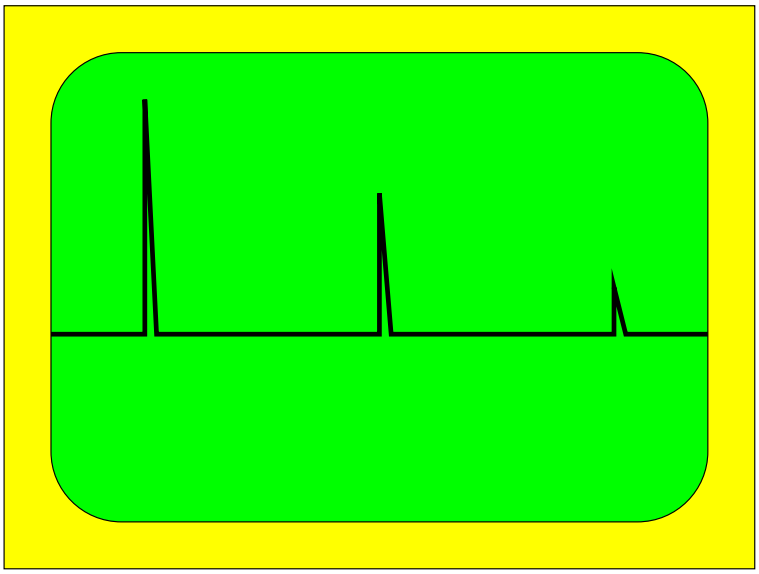


Figure 2.7: *A sample A-Mode display.*

ultrasound data, it will help a user to understand all other types of display modalities.

In fig[2.7] a sample display from an A-Mode ultrasound system is shown. As it can be seen from this figure, the received signal is displayed on the screen in a similar manner to displaying a voltage signal on an oscilloscope. The X axis of the display is time and the Y axis of the display is the amplitude of the received signal after demodulation. One important thing that one should note here is that as the wave is travelling in the body, it will be attenuated and hence this attenuation should be compensated[2]. Time gain compensation (TGC) is a popular way to do this. In TGC, the received signal is amplified based on the time difference between the time that the original pulse is generated and the time that the reflected signal is detected by the transducer. In some new systems, TGC is automated.

2.3.2.2 B-Mode imaging

B-Mode or brightness mode is a successor to the A-Mode imaging. It can be used for displaying 2 dimensional and 1 dimensional images, however, it is mostly used for displaying 2 dimensional images. A 1 dimensional B-Mode image would appear on the screen as a single straight line of dots. The brightness of each dot depend on the intensity of the received signal over time. If there will be more than one transducer[3] (see fig[2.8]) then there will be several

[2]It should be noted that it is only applied to reflective imaging.
[3]This is the linear array transducer.

Figure 2.8: *A linear array transducer can be used to create a B-Mode image [5].*

lines and it will create a 2 dimensional image.

Limitation of computer monitors (and human eye) Modern computer monitors quantize the colours that they display. The colours are constructed from red,green and blue components and each of the read, green and blue are quantised to 256 or shades, making 256^3 (or 16.7 million) colours in all. Gray scales are shown as combinations of red, green and blue in equal measures, so there are 256 available shades or gray. The limited number of available gray scale shades makes the B-mode inferior to A-mode in the case of 1 dimensional images. It is unlikely that computer monitors will be able to display more colours in future, as it is generally accepted that the human eye can't differentiate any more than 256 shades of each red, green and blue.

In order to show greater detail in gray scale images, B-mode images are usually log transformed and scaled so that full scale is between 30 and 60dB. This effectively amplifies small ultrasound signals so that become more clearly visible. For example: At 50dB full scale, a signal 25dB smaller than the strongest signal will appear as $\frac{50}{100}$ grey after log transforming. If the log transform was omitted, then a signal 25dB below the strongest

signal would appear as $\frac{5}{100}$ grey (almost black and barely visible above the black background).

2D image construction in real time The advantage of B-mode imaging is that successive 1-dimensional images of neighbouring regions can be constructed into a composite image that is 2-dimensional (as in fig[2.9]). This scanning and image construction could be done in real time and as the result, the image would be appear as a truly 2-dimensional to the user.

Many commercial ultrasound systems have a frame rate of 25 (or 30) frames per second. This means that every different 1-dimensional scan has been performed 25 times every second and a new 2-dimensional image has been constructed and displayed 25 times every second.

2D image construction: Transducer steering Between successive 1-dimensional scan, the transducer must somehow steer (or shift) so that a different part of the target image area is scanned. There are several common means of doing this, all of which are automated. The steering process and the type of transducer that it uses for steering are explained in section 2.3.3

2D image construction: Raw RF signal processing The processing that is performed between the scan and the image display is very important to any ultrasound imaging system. The signals are typically filtered, demodulated (or envelope detected), log transformed, and geometrically transformed.

In their raw form, the ultrasound signals are not suitable for display in either A-mode or B-mode. In A-mode the signal are processed only one step: demodulation.

Demodulation attempts to provide an envelope detection function. This is commonly performed with Hilbert transform. The Hilbert transform is defined as a filter whose transfer function is given in the following equation:

$$H_{Hilbert}(\omega) = \begin{cases} -j = e^{\frac{-jp\pi}{2}} & \omega > 0 \\ +j = e^{\frac{+jp\pi}{2}} & \omega < 0 \\ 1 & \omega = 0 \end{cases} \tag{2.3}$$

The output of this filter is then called Hilbert[yn)], and is the original signal with the phase of every Fourier component changed by 90^0. Hilbert[y(n)] is defined in the following

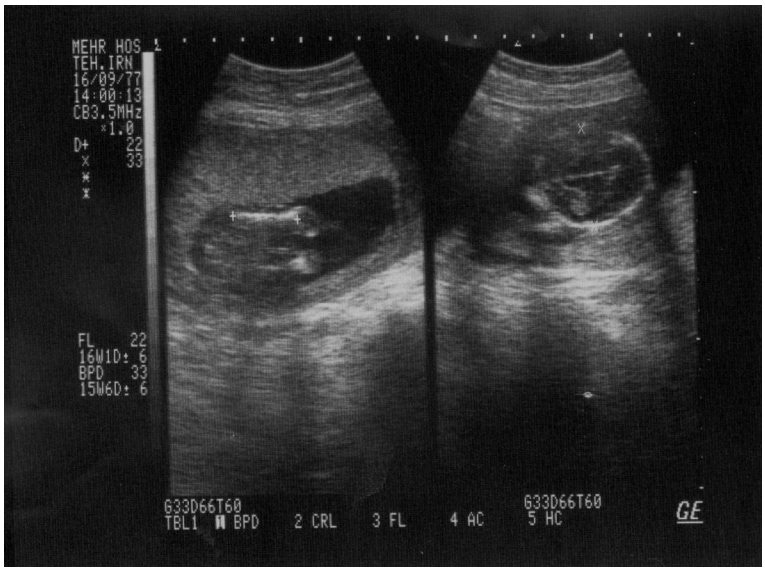


Figure 2.9: *A typical B-Mode display.*

equation:

$$Hilbert[y(n)] = \Im^{-1}\{h_{Hilbert}(\omega).\Im[y(n)]\} \tag{2.4}$$

The final demodulated signal is obtained by applying the following equation to the received signal y(n):

$$y(n)_{demodulated} = \sqrt{(y(n))^2 + (Hilbert[y(n)])^2} \tag{2.5}$$

Since large number of transducers operate by rotating the ultrasound beam (either using rotation or a phased array), consequently it is often necessary to perform a geometric conversion (usually involving a conversion from polar to rectangular co-ordinates). The scan data often uses polar co-ordinates and all display monitors use a rectangular display grid. The co-ordinate conversion is typically performed using bilinear interpolation.

A typical B-Mode image is shown in fig[2.9][4].

[4]This is an ultrasound image from my son: Ariasun, when he was 16 weeks

2.3.2.3 M-Mode Imaging

M-Mode (or motion mode) image results from a 1 dimensional scan, that is scrolled across the screen over time (as a series of 1 dimensional B-Mode lines). The scrolling can be horizontal or vertical. The transducer will generally be held stationary or near stationary, for this type of scan. The frame rate for this type of scan is very high e.g 600 frame per second (fps) [32]

2.3.2.4 Three dimensional and four dimensional imaging

A new development in ultrasound imaging is the advent of three dimensional imaging and the so called four dimensional imaging. The four dimensional imaging is the real time three dimensional imaging with time as the 4^{th} dimension [33]. In the three dimensional imaging, the transducer is physically moved in order to create a 3D image. Since physically moving the transducer is a slow process, currently there are not any real time three dimension ultrasound systems. In a typical three dimensional ultrasound system, the transducer is rotated thought 360^0 in quantized steps while scanning 2D images at each step. There are two common technique used for the three dimensional visualisation (which is called rendering):

Surface rendering: The technique is based on defining a series of connected polygons that combine to form the complete shape of the object being rendered. In some graphic packages this leads to a rendering that appears chunky as a result of curved objects not being well suited to model with flat polygons. Some newer packages attempt to simulate curved objects with curve fitting between the defined points of the polygons. In most cases this technique is successful and provides realistic looking 3D images. The most important advantage of this technique is that it is computationally the most efficient known method for rendering 3D images. One disadvantage of this type of rendering is that if the number of polygons used to render a surface is too few, then the surface detail will be lost. The other disadvantage of this method is that it gives information about surface and nothing about the underlying structural information. Since structural information is important in medical imaging, this is not a good method for medical image rendering. For this reason its use will be restricted to some types of imaging systems such as B-Mode images.

Volume rendering: This technique involves specifying the object to be rendered as a three dimensional set of ”VOXEL” (VOXEL=Volume Element, the 3D equivalent of a 2D pixel). In the simplest case, each voxel is specified as opaque or transparent (that is ,

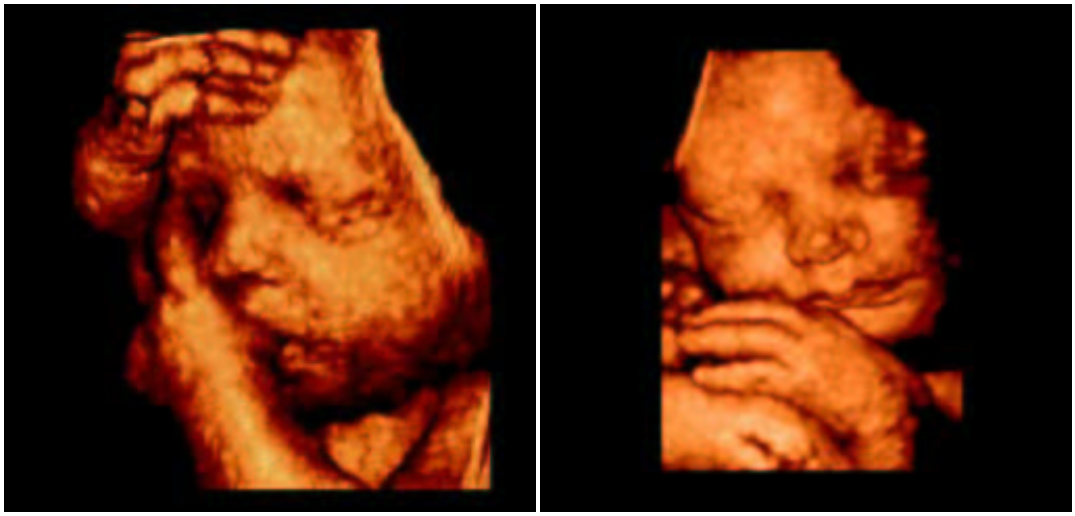


Figure 2.10: *Some typical 3D images [6].*

as solid mass or empty space). Volume rendering is superior to surface rendering for applications requiring tissue characterization. One drawback of the volume rendering is that noise in the signal tends to be enhanced in the final 3D images.

Recently new techniques have emerged , such as ”semi-transparent volume rendering” [34] and ”augmented reality” [35]. Two typical 3D images are shown in fig[2.10].

2.3.3 Ultrasound Transducers

The ultrasound transducers are made from materials that contain piezoelectric crystals. A piezoelectric crystal will vibrate when excited by an AC electric signal of the right frequency (transmit mode). Conversely it could generate a small electric signal when forced to vibrate (received mode).

The transducer is used to convert electronic signals to ultrasound waves which are transmitted to the body and also for converting the received ultrasound waves to electronic signals. A simple transducer may be used in A-mode (one dimensional scanning) ultrasound systems. In 2D and 3D ultrasound systems one part of the body would be scanned and hence a simple one element transducer can’t be used unless it is mechanically scanned. There are several type of transducers that can be used in this type of medical imaging systems [36]:

1. Linear Array transducer

2. Phased array transducer

3. Mechanical scan transducer

2.3.3.1 Linear array transducers

In this type of array transducers, each element of the transducer will be working independent of the others (fig[2.11]). For each groups of elements, a 1 dimensional B-Mode image will be created. By putting all of these one dimensional images next to each other, a 2-dimensional B-Mode image will be created. In a typical ultrasound system, some of these elements may be fired at the same time to make a more powerful beam. A typical linear array transducer is shown in fig[2.12].

2.3.3.2 Phased array transducers

In a phased array transducer, each element will be fired with a time delay compared to other elements in the array. By changing this delay, the wave front could be steered to different places in the body. In fig[2.13] the steering process is shown. In fig[2.14], a typical phased array transducer is shown.

2.3.3.3 Mechanical scan transducer

In a mechanical scan transducer, a single transducer is used for transmitting and receiving. The transducer mechanically rotates to scan the interested media. The structure of a typical mechanical transducer which is used in intravascular imaging system is shown in fig[2.15].

2.4 Doppler systems

The basis of Doppler ultrasound systems is the fact that reflected/scattered ultrasonic waves from a moving interface will undergo a frequency shift [37], [38]. In general the magnitude and the direction of this shift will provide information regarding the motion of this interface. To appreciate this very general fact we need to consider the relationship between the frequency, f_S, of waves produced by a moving source and the frequency, f_R, of the waves received by a moving receiver. For simplicity we shall assume that the source and receiver are moving along

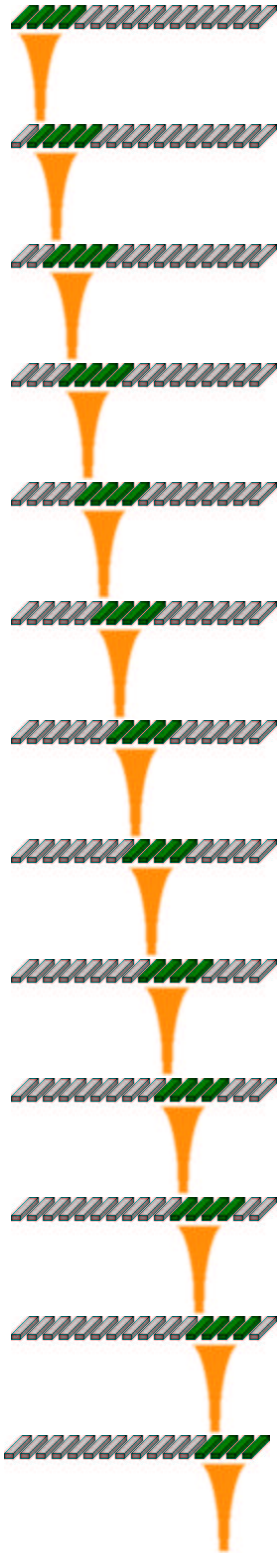
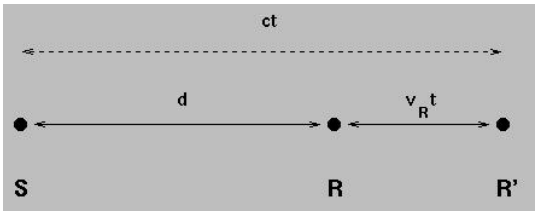


Figure 2.11: *A linear array transducer used to scan a medium [7].*

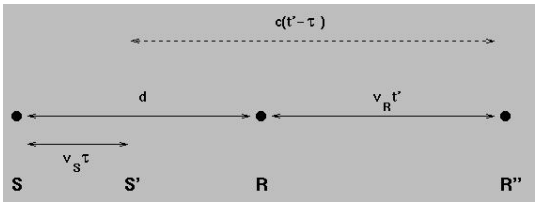
Figure 2.12: *A typical linear array transducer [3].*

the same line. The argument that follows can be generalised to three dimensions if wave speed is isotropic and the source produces spherical waves.

At t=0, let the source, S, and receiver, R, be separated by a distance d.



At t=0 let S emit a wave that reaches R at a time t later as shown in the following fig.



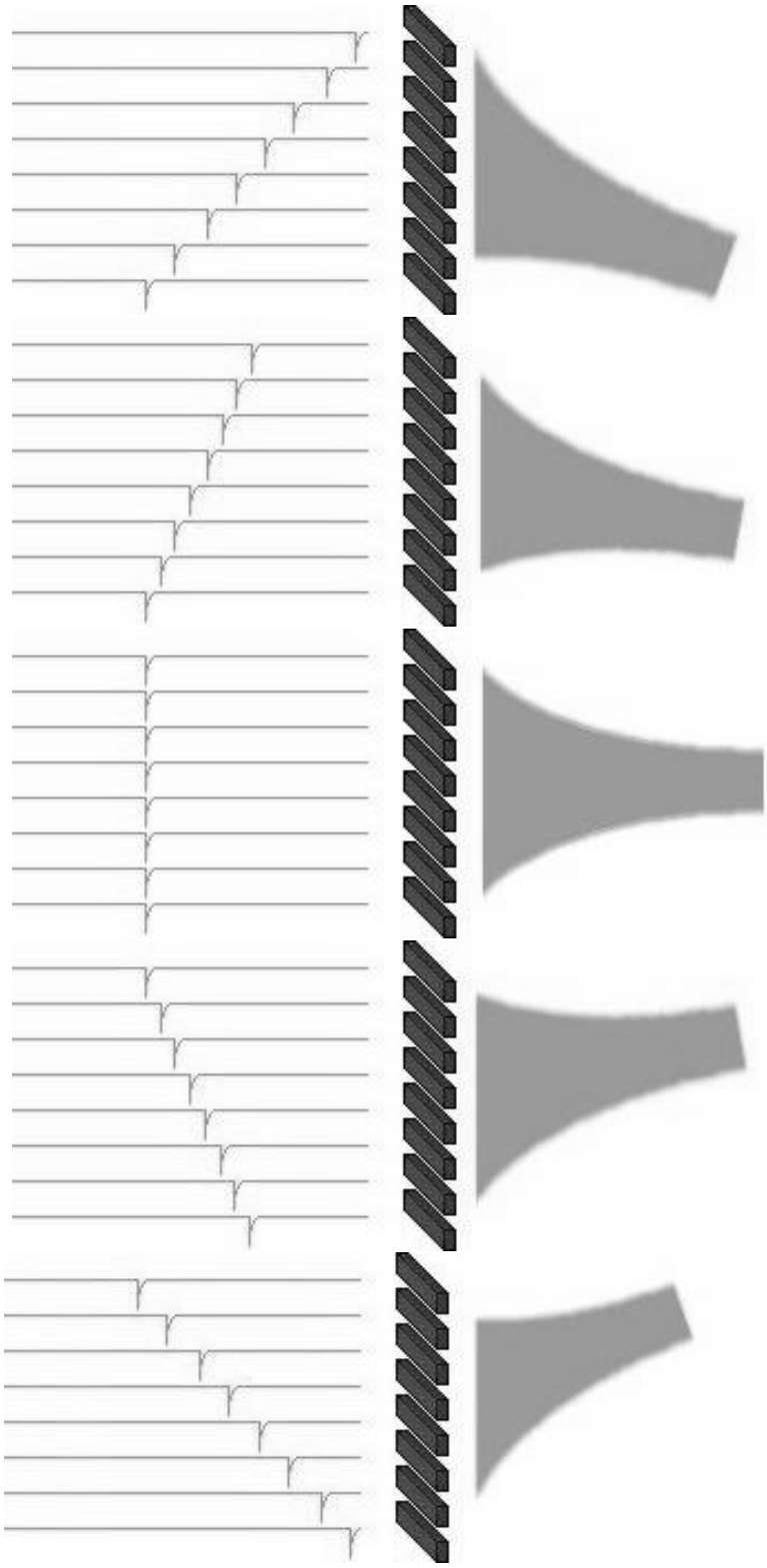


Figure 2.13: *The steering process in a phased array transducer [7].*

Figure 2.14: *A typical phased array transducer [3].*

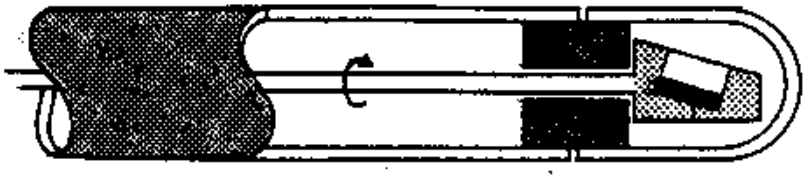
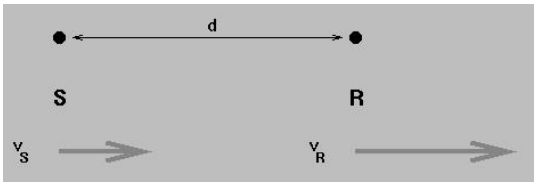


Figure 2.15: *The structure of a mechanical scan transducer [1].*

In this time t the receiver will have moved a distance and the wave, propagating with velocity C will have travelled a distance. Thus

$$\begin{aligned} Ct &= d + V_R t \\ t &= \frac{d}{C - V_R} \end{aligned} \tag{2.6}$$

Now at time τ, the source will have moved a distance τV_R. Let the wave emitted at that instant be received at time t' by R.



In this time R would have travelled a total distance of $V_R t'$, and thus

$$\begin{aligned} C(t' - \tau) &= (d - V_S \tau) + V_R t' \\ t' &= \frac{d + (C - V_S)\tau}{C - V_R} \end{aligned} \tag{2.7}$$

Thus for the receiver the interval between the waves has been

$$\begin{aligned} \tau' &= t' - t \\ \tau' &= \frac{d + (C - V_S)\tau}{C - V_R} - \frac{d}{C - V_R} \\ \tau' &= \frac{d}{C - V_R} + \frac{(C - V_S)\tau}{(C - V_R)} - \frac{d}{C - V_R} \\ \tau' &= \frac{(C - V_S)}{(C - V_R)}\tau \\ \frac{\tau}{\tau'} &= \frac{(C - V_R)}{(C - V_S)} \end{aligned} \tag{2.8}$$

whereas for the source the interval between the waves has been τ . Now the number of waves

emitted in τ' by the source must be equal to the number of waves received by the receiver in i.e

$$\begin{aligned} f_R\tau' &= f_S\tau \\ f_R &= \frac{\tau}{\tau'}f_S \end{aligned} \tag{2.9}$$

Substituting the value of $\frac{\tau}{\tau'}$,

$$f_R = \frac{C - V_R}{C - V_S}f_S \tag{2.10}$$

2.5 Conclusion

Some background information about using ultrasound in medical systems were presented. Since medical ultrasound imaging is widely used by physicians, it was explained in more detail.

Array transducers were introduced and how they can be used to scan a region by firing each element of the array sequentially was shown.

Use of the Doppler effect was examined. For the simple case where the source and/or the receiver was moving the Doppler shift was calculated.

Chapter 3

Wave and sound theory

In this chapter some background information about the theory of ultrasound wave propagation will be presented. This background information should refresh the mind of readers to have a better understanding of the mathematics behind the modelling presented in chapter 5. The mathematics that will be presented here is very short and is only for refreshing the readers mind. There isn't any proof for most of the formulae in this chapter and the interested reader may consult the references for mathematical proof.

3.1 Introduction

Ultrasound is a sound wave whose frequency is higher than the hearing limit of humans which is about 20kHz. The frequency in medical ultrasound systems varies between 2 MHz to about 30 MHz. The frequency of a normal medical imaging ultrasound system is about 3.5 MHz. Some medical ultrasound systems with very high resolution output may use ultrasound waves with higher frequencies.

In the first section of this chapter some background information on sound wave generation and propagation will be presented. Since the TLM is based on transmission line theory, The basic transmission line theory will be presented in the next section of this chapter.

3.2 Waves

3.2.1 Basic wave theory and definitions

In this subsection basic wave theory will be explained. This theory can be extended to sound waves and electromagnetic waves.

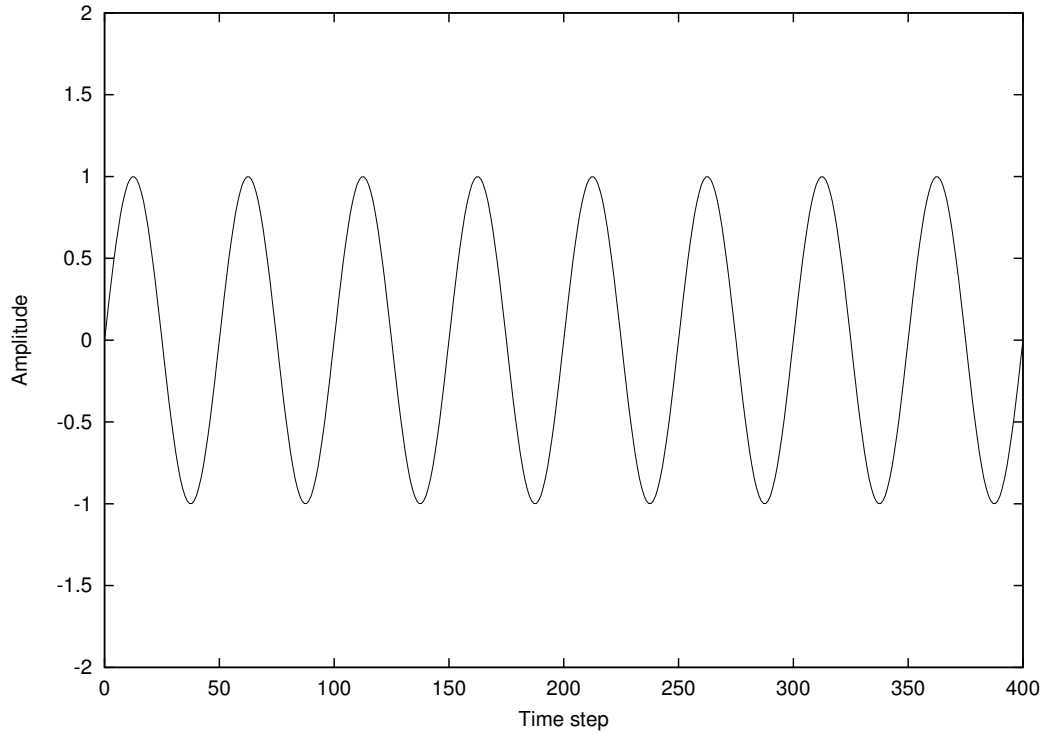


Figure 3.1: *Graphical representation of a simple harmonic oscillation. Note that the X axis is time steps.*

3.2.1.1 Simple harmonic oscillation

It can be shown that a simple harmonic oscillation is represented mathematically as [39] :

$$\Psi(t) = A_0 \sin(\omega t) \tag{3.1}$$

Where ω is the angular velocity of the oscillation and $\Psi(t)$ is the wave amplitude at time t. The graphical representation of a harmonic oscillation is shown in fig[3.1].

The frequency f of the oscillation can be found by using the following equation:

$$f = \frac{\omega}{2\pi} \tag{3.2}$$

The frequency of an oscillation is, by definition, the number of oscillation per second [40]. If the time for one oscillation is T, then the frequency can be found as follow:

$$f = \frac{1}{T} \tag{3.3}$$

In a discretised system in which the time is discretised, the frequency can be found as follow:

$$f = \frac{1}{n\Delta T} \tag{3.4}$$

Where ΔT is the time between adjacent samples and n is the number of samples in one cycle.

3.2.1.2 The wave equation in one dimension

It can be shown that the wave equation in one dimension (direction of movement is z) is:

$$\Psi(z,t) = A_0 \sin(\omega t - Kz) \tag{3.5}$$

Where t is time and z is the distance from the origin (assuming that the wave source is located at $z = 0$). $\Psi(z,t)$ is the wave amplitude at time t at place z. ω and K defined as:

$$\begin{aligned} K &= \frac{2\pi}{\lambda} \\ \omega &= \frac{2\pi C}{\lambda} \end{aligned}$$

where λ is the wavelength[1] and C is the wave speed in the medium.

[1] Wavelength is the distance that wave propagates in one cycle.

3.2.1.3 Wave equations in 3 dimensions

It could be shown that any wave equation should satisfy the following relations [41][42]:

$$\begin{aligned} \frac{\partial^2 \Psi(x,y,z,t)}{\partial t^2} &= -\omega^2 \Psi(x,y,z,t) \\ \frac{\partial^2 \Psi}{\partial x^2} &= -k_x^2 \Psi \\ \frac{\partial^2 \Psi}{\partial y^2} &= -k_y^2 \Psi \\ \frac{\partial^2 \Psi}{\partial z^2} &= -k_z^2 \Psi \end{aligned} \tag{3.6}$$

In one dimension, these equations reduce to :

$$\begin{aligned} \frac{\partial^2 \Psi(z,t)}{\partial t^2} &= -\omega^2 \Psi(z,t) \\ \frac{\partial^2 \Psi}{\partial z^2} &= -k_z^2 \Psi \end{aligned} \tag{3.7}$$

It could be shown that equation 3.5 can satisfy this equation:

$$\begin{aligned} \Psi(z,t) &= A_0 \sin(\omega t - Kz) \\ \frac{\partial \Psi(z,t)}{\partial t} &= -\omega \cos(\omega t - Kz) \\ \frac{\partial^2 \Psi(z,t)}{\partial t^2} &= -\omega^2 \sin(\omega t - Kz) \\ \frac{\partial^2 \Psi(z,t)}{\partial t^2} &= -\omega^2 \Psi(z,t) \end{aligned}$$

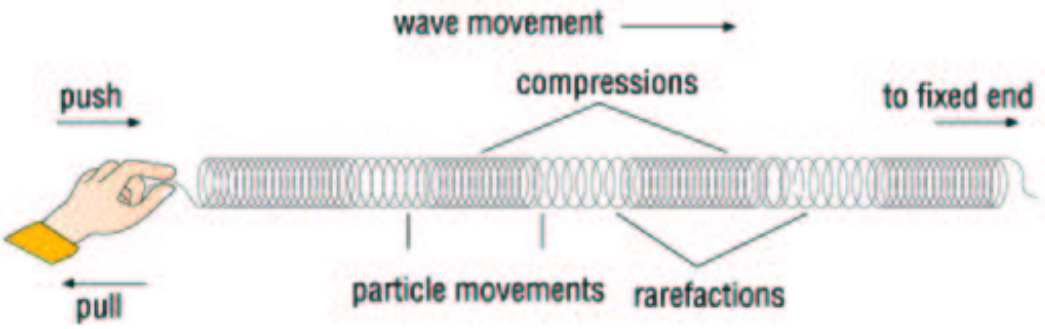


Figure 3.2: *Longitudinal waves [8].*

and:

$$\begin{aligned}
\Psi(z,t) &= A_0 \sin(\omega t - Kz) \\
\frac{\partial \Psi(z,t)}{\partial z} &= K \cos(\omega t - Kz) \\
\frac{\partial^2 \Psi(z,t)}{\partial z^2} &= -K^2 \sin(\omega t - Kz) \\
\frac{\partial^2 \Psi(z,t)}{\partial z^2} &= -K^2 \Psi(z,t)
\end{aligned}$$

3.2.1.4 Longitudinal and transverse waves

If the vibration is parallel to the direction of propagation then the wave is a longitudinal wave [43] (see fig[3.2]).

If the vibration is perpendicular to the direction of propagation then the wave is a transverse wave (see fig[3.3]).

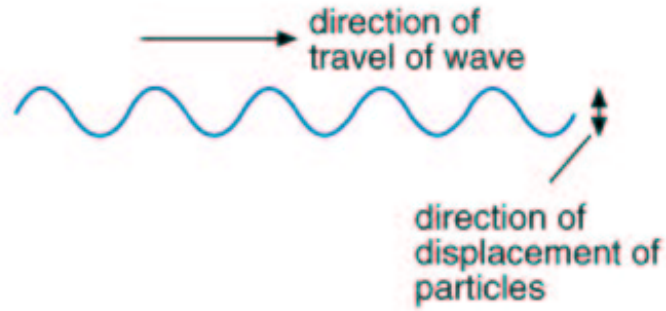


Figure 3.3: *Transverse waves [8].*

3.2.1.5 Scalar and vector waves

If the wave is propagated in the z direction then for a longitudinal wave, only the value[2] of the wave at each point is important since the direction is defined and is parallel to the direction of wave propagation which is z direction. This kind of wave[3] is called a scalar wave since only a number is needed to identify the wave at each point in the medium.

The case for a transverse wave is more complex, since the direction of vibration could be any direction in xy plane. For a transverse wave, not only the value[4] of the wave at each point should be specified, but also its direction[5] should be specified [44]. These kind of waves[6] are called vector waves since to identify the wave at each point, not only a value but also a direction should be specified.

[2] Amplitude
[3] Longitudinal waves
[4] Amplitude
[5] In the plane perpendicular to the direction of wave propagation
[6] Transverse waves

3.2.1.6 Mechanical and non-mechanical waves

Mechanical waves are waves that need a medium for propagation [45], [46]. Non-mechanical waves do not need a medium to propagate. For example sea waves are mechanical since they need a medium (sea water) to propagate. Light is a non-mechanical wave since it can propagate in vacuum.

3.2.2 Sound waves

Sound waves are mechanical waves. They are longitudinal waves and hence are scalar waves [47].

The general sound wave equation in 3 dimension is [45], [40][7] :

$$\frac{\partial^2\Psi(x,y,z,t)}{\partial t^2} = c^2\left(\frac{\partial^2\Psi(x,y,z,t)}{\partial x^2} + \frac{\partial^2\Psi(x,y,z,t)}{\partial y^2} + \frac{\partial^2\Psi(x,y,z,t)}{\partial z^2}\right) \tag{3.8}$$

Where c can be found from the following equation:

$$c = \frac{1}{\sqrt{\rho\beta}} \tag{3.9}$$

In this equation β is the compressibility and ρ is the density.

3.2.3 Electromagnetic waves

Electromagnetic waves are non-mechanical waves. They are transverse waves and hence vector waves.

The electromagnetic wave equation in 3 dimension can be written as:

$$\frac{\partial^2\Psi(x,y,z,t)}{\partial t^2} = c^2\left(\frac{\partial^2\Psi(x,y,z,t)}{\partial x^2} + \frac{\partial^2\Psi(x,y,z,t)}{\partial y^2} + \frac{\partial^2\Psi(x,y,z,t)}{\partial z^2}\right) \tag{3.10}$$

Where c is the wave speed in medium and can be calculated by using the following formula:

$$c = \frac{1}{\sqrt{\mu\epsilon}} \tag{3.11}$$

[7] This is the sound wave equation in air when the wave amplitude is small and so the medium is a linear system.

Where μ is the permeability and ϵ is the permittivity.

3.2.4 Wave reflection and refraction

If a wave arrives at the interface between two media[8], then there would be a reflected (back scattered) wave and a refracted wave. If the new medium is denser than the original medium, then the wave reflects with 180^0 phase difference, other wise the reflected back signal is in phase with the original wave [45]. The wave in the new medium is always in phase with the original wave. The back scattered wave is called the reflected wave and the propagated wave in the new medium is called the refracted wave.

3.2.4.1 The law of reflection and refraction

The reflection law is:

> The reflected and incidence waves are in a plane and the angle of incident and reflection are the same:
>
> $$\theta_1 = \theta_3 \tag{3.12}$$
>
> Where θ_1 is the incident angle and θ_3 is the reflection angle (see fig[3.4]).

The refraction law is:

> The refracted and incidence waves are in a plane and the angles of incidence and reflection are related to each other by the following equation:
>
> $$n_1 \sin\theta_1 = n_2 \sin\theta_2 \tag{3.13}$$
>
> Where θ_1 is the incident angle and θ_2 is the refraction angle (see fig[3.4]). Here n_1 is a dimensionless constant called the index of refraction for medium 1 and n_2 is the index for refraction of medium 2.

The index of refraction of a medium depends on the wave speed in the medium. It is mostly calculated based on the wave speed in a reference medium. For electromagnetic waves this

[8] It is assumed that the size of this interface as seen by the wave is much bigger than the wave length of the wave.

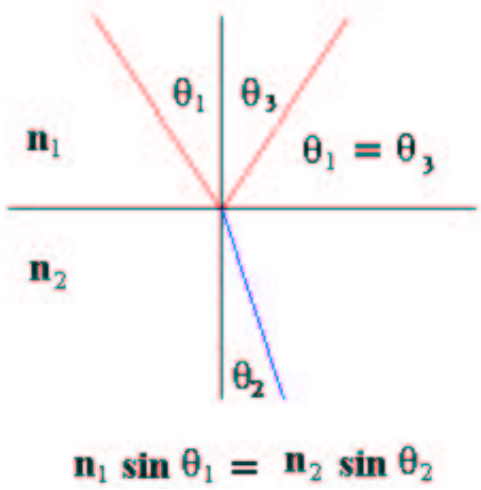


Figure 3.4: *Reflection and refraction law.*

reference medium is the empty space. For sound this reference medium could be air[9]. If the speed of a wave in the medium is v and the speed of a wave in the reference medium is c , then the index of refraction can be calculated as follow:

$$n = \frac{c}{v} \tag{3.14}$$

By substituting the value of the refractive index from this equation to equation 3.13, one can write the following equation:

$$\begin{aligned} n_1 \sin\theta_1 &= n_2 \sin\theta_2 \\ \frac{c}{v_1} \sin\theta_1 &= \frac{c}{v_2} \sin\theta_2 \\ \frac{1}{v_1} \sin\theta_1 &= \frac{1}{v_2} \sin\theta_2 \\ v_2 \sin\theta_1 &= v_1 \sin\theta_2 \end{aligned} \tag{3.15}$$

[9]It doesn't matter which medium is the reference medium as long as the reference medium is the same for calculating both the n_1 and n_2.

3.2.4.2 Total reflection

Equation 3.15 shows the relationship between incident angle and refraction angle. One may try to find the refraction angle and rewrite this equation as follow:

$$\theta_2 = \sin^{-1}(\frac{\upsilon_2}{\upsilon_1}\sin\theta_1) \tag{3.16}$$

As seen from this equation, θ_2 hasn't any value unless $\frac{\upsilon_2}{\upsilon_1}\sin\theta_1$ is between -1 and 1. If the value of $\frac{\upsilon_2}{\upsilon_1}\sin\theta_1$ is greater than 1 or less than -1, then there isn't any refracted wave and all of the wave will be reflected back. The minimum angle that total reflection is occurred can be found as follow[10]:

$$\begin{aligned} |\frac{\upsilon_2}{\upsilon_1}\sin\theta_1| &\leq 1 \\ \frac{\upsilon_2}{\upsilon_1}|\sin\theta_1| &\leq 1 \\ |\sin\theta_1| &\leq \frac{\upsilon_1}{\upsilon_2} \end{aligned} \tag{3.17}$$

3.2.5 Wave Scattering

In the previous section, it was assumed that the interface size is much bigger than the wavelength and the wave speed in the two media changes abruptly. Most interfaces in the human body are not compatible with the reflection interface criteria. In some cases the size of the object is comparable (or smaller)with the wavelength[11] and in other cases the speed is changing gradually from one medium to the next[12]. In most cases the actual interface is a combination of these two cases. The way that wave interact in these cases is called scattering.

Compared to reflection, the case for scattering is much more complex. The scattering pattern not only depends on the object size compared to the wavelength but also to the shape and wave speed in the object[48]. Since this theory is very complex, several authors have tried to find the scattering patterns for some known shapes such as cylindrical scatters [49] and spherical scatters [50]. Even thought there are solutions for a limited range of other shapes, spheres and

[10] Since υ is the wave speed, it is always a positive number and hence $\frac{\upsilon_2}{\upsilon_1}$ and $\frac{\upsilon_1}{\upsilon_2}$ are positive numbers.

[11] For example the interaction of blood cells and ultrasound falls to this category since the size of the blood cells are very small comparing to the wavelength.

[12] For example the interface of skin and the underlying tissue falls to this category

cylinders can approximate a wide range of scatters in biology [51], [52].

3.2.5.1 Basic scattering theory

There are various formulations for scattering, and the one used here considers the medium to have no absorption and small fluctuations in the density ρ_1 and compressibility β_1 about constant values ρ_0 and β_0 inside an inhomogeneous volume V. So inside the object:

$$\begin{aligned} \rho(r) &= \rho_0 + \rho_1 \\ \beta(r) &= \beta_0 + \beta_1 \end{aligned}$$

And outside the object:

$$\begin{aligned} \rho(r) &= \rho_0 \\ \beta(r) &= \beta_0 \end{aligned}$$

It is convenient to define the following parameters:

$$\begin{aligned} \tilde{\rho}(r) &= \frac{\rho_1(r)}{\rho_0} \\ \tilde{\beta}(r) &= \frac{\beta_1(r)}{\beta_0} \end{aligned}$$

Then the wave equation becomes [49]:

$$\nabla^2\Psi(r,t) - \frac{1}{c^2}\frac{\partial^2\Psi}{\partial t^2}(r,t) = \frac{1}{c^2}\frac{\partial^2\Psi}{\partial t^2}(r,t)\tilde{\beta}(r) + div[\tilde{\rho}(r)grad\Psi(r,t)] \tag{3.18}$$

where $c = \frac{1}{\sqrt{\rho_0\beta_0}}$. This equation can be solved by Green's function method[53] in which the right hand side is considered as the source term, and the Green's function is the solution of the

above equation with right hand side equal to a point source radiator. The solution is:

$$\begin{aligned}\Psi(r,t) &= \Psi_i(r,t) + \int_{-\infty}^{\infty} dt_0 \int_V \left\{ \frac{1}{c^2}\frac{\partial^2\Psi}{\partial t^2}(r_0,t_0)\tilde{\beta}(r_0) + \right. \\ &\left. + div[\tilde{\rho}(r_0) grad\Psi(r_0,t_0)] G(r_0,r,t_0,t) d^3 r_0 \right\} \end{aligned} \tag{3.19}$$

Where

$$G(r_0,r,t_0,t) = \frac{\delta(t - t_0 - \frac{|r_0 - r}{c})}{4\pi|r - r_0|} \tag{3.20}$$

Solving this integral equation is only possible for some certain simple objects. In the general case it is possible to solve it only by approximation. The most important approximation is Born's approximation. This approximation is valid only when the scattering is weak and both $\tilde{\rho}(r)$ and $\tilde{\beta}(r)$ are very small. If this approximation is not true (at least one of the $\tilde{\rho}(r)$ or $\tilde{\beta}(r)$ are not small), then the only way that one can solve this equation is by using numerical techniques.

Since in the body this approximation may not be true[13] , it is not possible to solve it analytically. The theory of scattering by human tissue has been reviewed by Chivers [54]. Inverse problem of scattering is another problem which several research groups are working on [52].

3.3 Transmission line

A transmission line is a device for transmitting or guiding energy from one point to another[55],[56]. The energy may be for lighting, heating or performing work, or it may be in the form of signal information (speech, pictures, data, music). Basically a transmission line has two input terminals into which power (or information) is fed and two output terminals from which power (or information) is received. Thus a transmission line may be regarded as a four-terminal device.

Transmission lines are everywhere and there are of infinite variety of them .However, regardless of type, length, or construction, all operate according to the same principles [57].

One easily imaginable transmission line is a deep long straight waterway. If a signal is gener-

[13]Our knowledge about the human body tissue on the scale less than that of the ultrasound wavelength is very limited

ated in one end of the waterway by disturbing the water, this disturbances will propagate in the waterway and eventually will be received at the other side of the waterway.

The received signal at the end of the waterway is weaker than the generated signal at the beginning of the waterway. This is because the energy in the signal converts to heat during transmission. The waterway has a characteristic impedance (Z_0).

If there is a rock or island in the waterway, when the water waves reach the island, they will be reflected back by the island.

If the waterway is infinitely long, then the wave travels in it until it is completely attenuated or absorbed. If the waterway is not infinite, when the wave reaches the end of the waterway, it is reflected back by the wall at the end of the waterway. The only way to prevent this reflection is to have a matching termination at the end of the waterway. This matching termination would have a matching load impedance (Z_L) which matches the propagation characteristics of an infinity long waterway in a tunnel of finite length.

If the waterway length is L and the time that it takes for the disturbances to travel the waterway is t, then the wave speed or propagation speed in the waterway can be calculated:

$$v = \frac{L}{t} \tag{3.21}$$

The time delay could be specified is three ways:

1. Seconds.
2. Periodic time (T)
3. Phase delay

If the wave consists of two or more frequencies, then the behaviour of the waterway for each of these frequencies is different. The wave speed is most likely frequency dependent, so some waves move faster and some waves moves slower in the waterway. This frequency dependent wave speed is called dispersion. Also the attenuation maybe depends on frequency, so at the end of the waterway the different components would be received with different amplitude (related to attenuation) and phase (related on wave speed). This will make the shape of the received signal different from the original one.

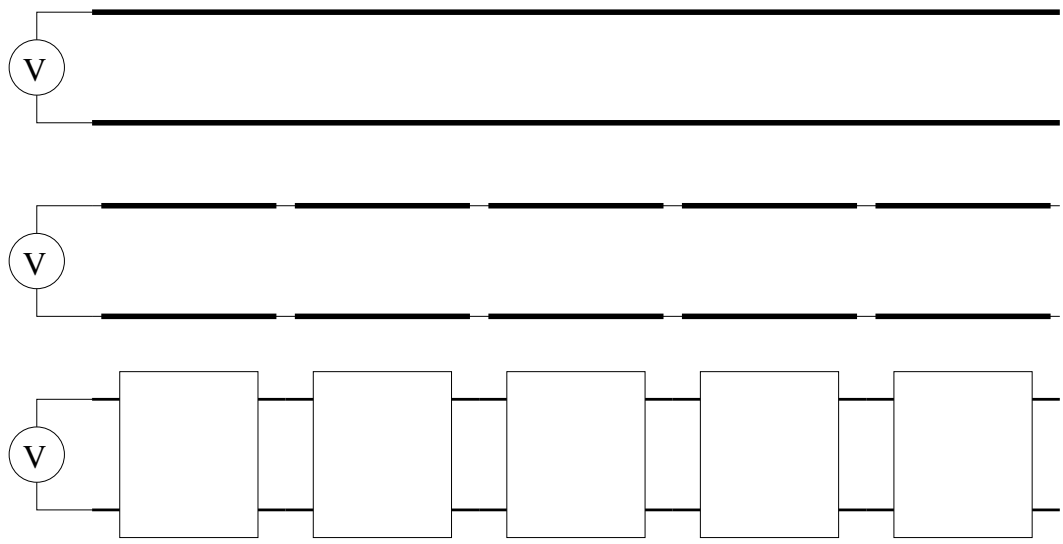


Figure 3.5: *A transmission line can be modelled as a series of interconnected lumped systems.*

3.3.1 Modelling Transmission lines with lumped components

As was explained, a transmission line is a distributed system. To understand the wave propagation property of a distributed system, it is better to model it as a series of lumped systems. Lumped systems are systems whose size is zero, so it takes no time for a wave to propagate through them. There is no real lumped system in the world but when the size of the system is very small comparing to the wavelength, then one can assume that they are lumped systems. A transmission line modelled as a series of lumped systems is shown in fig[3.5].

To do this model, the transmission line is broken to small interconnected transmission lines with the length of Δx. The length of these small transmission line should be selected so that the wavelength of the signal in the medium be much larger than the size of each of these transmission lines. If the wavelength is λ then:

$$\lambda \gg \Delta x \tag{3.22}$$

For a linear transmission line, each of these lumped systems can be modelled as a combinations of inductor, capacitors and resistors. If the transmission line is loss less then there is no resistor in the model.

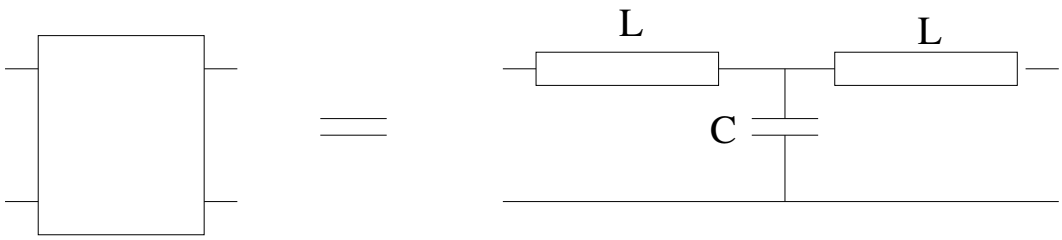


Figure 3.6: *Modelling a small section of a transmission line by discrete lumped components.*

3.3.1.1 Modelling a lumped system with discrete components

If a transmission line is broken into small sections where the size of each section is much smaller than the wavelength for the highest frequency in the transmission line, each section can be modelled as a lumped system. Furthermore this lumped system can be modelled as shown in fig[3.6].

The value of C and L depend on the transmission line characteristics.

3.3.1.2 Dispersion in a transmission line

As was shown in fig[3.5] and fig[3.6], a long transmission line can be modelled by a series of connected C and L sections. To find the transmission line dispersion effect, it is sufficient to find the dispersion effect of each small section, since the transmission line dispersion effect is the sum of the dispersion effect of each of the small sections [58].

The input and output currents and voltages are related to each other as follows:

$$\begin{bmatrix} \bar{V}_i \\ \bar{V}_i \end{bmatrix} = T \begin{bmatrix} \bar{V}_o \\ \bar{V}_o \end{bmatrix} \tag{3.23}$$

Where i and o indicate input and output respectively. The over-bar is used for showing a phasor quantity. The 4×4 matrix T is called transmission matrix and is in the general form of:

$$T = \begin{bmatrix} \cos\beta\ell & jZ\sin\beta\ell \\ j\frac{\sin\beta\ell}{Z} & \cos\beta\ell \end{bmatrix} \tag{3.24}$$

Where β is the phase constant and ℓ is the size of each section.

The T model for each section as shown in fig[3.6], can be broken down to 3 series segments as

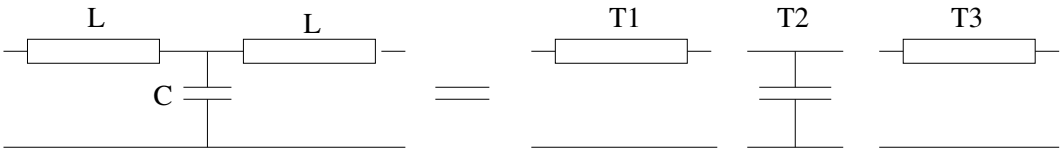


Figure 3.7: *Breaking down a T model to 3 segments.*

shown in fig[3.7]. Since the T section is the cascade of three sections as shown in fig[3.7], T can be found as follows:

$$T = T_1 T_2 T_3 \tag{3.25}$$

Where T_1 and T_2 and T_3 are the transmission matrices for the three sections shown in fig[3.7]. Section 1 and 3 are the same, hence their transmission matrices are the same. The transmission matrix for these two sections is as follow:

$$T_1 = T_3 = \begin{bmatrix} \cos\frac{\omega\Delta t}{2} & jZ_L \sin\frac{\omega\Delta t}{2} \\ j\frac{\sin\frac{\omega\Delta t}{2}}{Z_L} & \cos\frac{\omega\Delta t}{2} \end{bmatrix} \tag{3.26}$$

Where ω is the angular frequency of the wave in the medium and Δt is the time for the wave to propagate through the section.

It can be shown that the transmission matrix for the middle section T_2 is :

$$T_2 = \begin{bmatrix} 1 & 0 \\ j\frac{\tan\frac{\omega\Delta t}{2}}{Z_c} & 1 \end{bmatrix} \tag{3.27}$$

Substituting the values for T_1 and T_2 and T_3 in equation 3.25, and finding the value of T and comparing it with the general form of T as shown in equation 3.24, the transmission parameter for the section can be found:

$$\cos(\beta\ell) = 1 - 2\sin^2(\frac{\omega\Delta T}{2})\left[1 + \frac{Z_L}{2Z_c}\right] \tag{3.28}$$

As seen from this equation, the phase constant for the transmission line, β, is related to the ω, the angular frequency of the wave.

3.4 Conclusion

Sound waves are mechanical, longitudinal, scalar waves. Electromagnetic waves are non-mechanical, transverse, vector waves.

When a wave reaches an object in the medium, then there maybe some reflection or scattering. Reflection is simple to understand and model, but scattering is complex and difficult to model mathematically.

Any device that guides energy from one point to another is a transmission line. The wave speed in a transmission line maybe related to its frequency, This is called dispersion.

Chapter 4
TLM for modelling wave propagation and its implementation

4.1 Introduction

TLM (Transmission Line Matrix Modelling) is a numerical technique for modelling for wave propagation. TLM was originally used for modelling electromagnetic wave propagation[59], [60], [61] but since it is based on Huygens principle (see section 4.2.1 on page 48) it could be used for modelling any phenomena which obeys this principle. Researchers showed that TLM can be used to solve the following problems:

- Diffusion problem [62].
- Vibration [63]
- Heat transfer [64]
- Radar [65]
- Electromagnetic compatibility [66]

In this chapter, Huygens principle is explained and then it is shown how this principle could be used to model wave propagation. Two dimensions TLM modelling will be explained in detailed and one dimensional and three dimensional TLM modelling will be introduced. TLM can be use in time reversal. This capability is explained in section 4.5 on page 79. Since TLM was originally designed for electromagnetic waves, in section 4.7 on page 80 the relationship between these two waves (electromagnetic and ultrasound) will be presented.

4.2 From the Huygens principle to TLM modelling:

TLM modelling is based on the Huygens principle. In this section this principle is explained and it is shown that the TLM is the discrete version of this principle.

4.2.1 The Huygens principle

About 300 years ago, Christian Huygens published his principle which is [67], [68]:

> All points on a wave front serve as point sources of spherical secondary wavelets. After a time T the new position of the wave front will be the surface of tangency to these secondary wavelets.

This principle is shown in fig[4.1]. At time 0 the central point scatters a wave. The wave front at time t_1 is shown in fig[4.1(b)]. At this time (time = t_1) we can assume that all points on the wave front are acting as a point sources (shown in fig[4.1(c)]) and the wave front at any time later (for example t_2) is the wave front from these secondary point sources (fig[4.1(d)])

4.2.2 TLM modelling

Johns [59] modelled this principle by sampling the space and representing it with a mesh of passive transmission line components[1]. He modelled the wave propagation as voltage and current travelling in this mesh. Time was also sampled and the relationship between ΔT, the sample interval and $\Delta \ell$, the sample space, is:

$$\Delta \ell = \Delta T C \tag{4.1}$$

Where C is the wave speed in the medium. In fig[4.2] wave propagation in a two dimensional TLM mesh is shown.

Assume that at time zero, an impulse is incident to the middle node (fig[4.2(a)][2]). This node scatters the wave to its 4 neighbouring nodes. The scattered wave reaches the neighbouring nodes at time = ΔT (fig[4.2(b)][3]). Now these 4 nodes scatter waves to their neighbouring nodes (fig[4.2(c)][4]). At time = $2\Delta T$ the wave front can be found by finding waves scattered from points in fig[4.2(b)] as shown in fig[4.2(d)][5]. At each time step, each node receives an

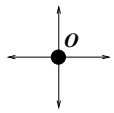
[1] This is two dimension TLM modelling. In section 4.4 (page 77) TLM modelling in one dimension and in three dimensions will be explained.

[2] Compare this with fig[4.1(a)] on page 49.

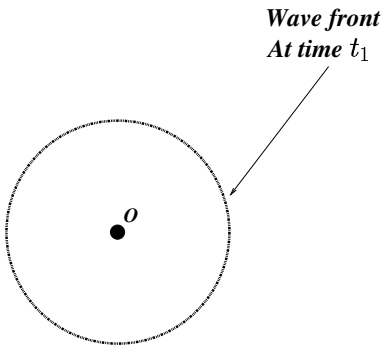
[3] Compare this with fig[4.1(b)] on page 49.

[4] Compare this with fig[4.1(c)] on page 49.

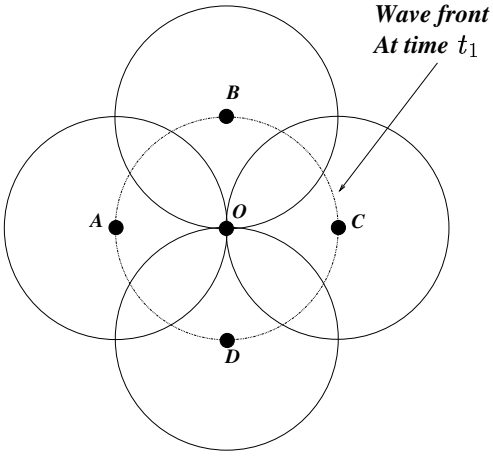
[5] Compare this with fig[4.1(b)] on page 49.



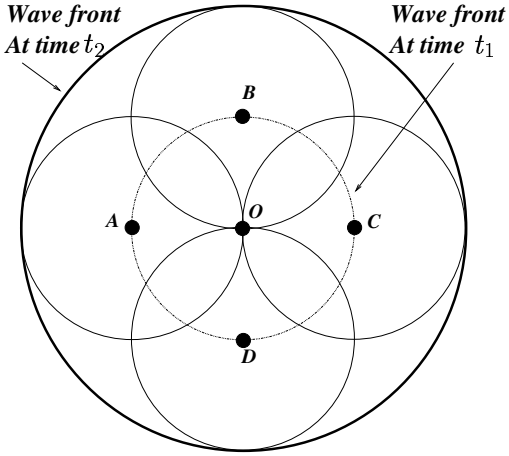
(a) A point source scattering wave



(b) Wave front at time t_1

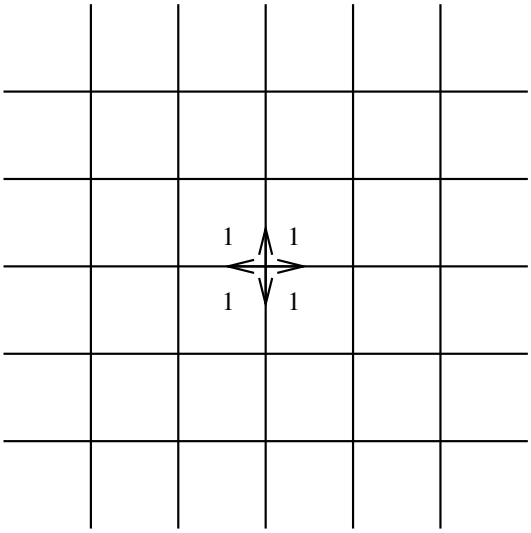


(c) Each point in the wave front act as a point source

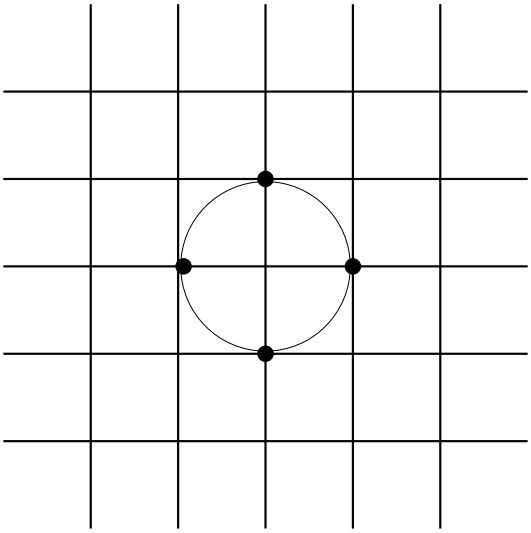


(d) Wave front at time t_2

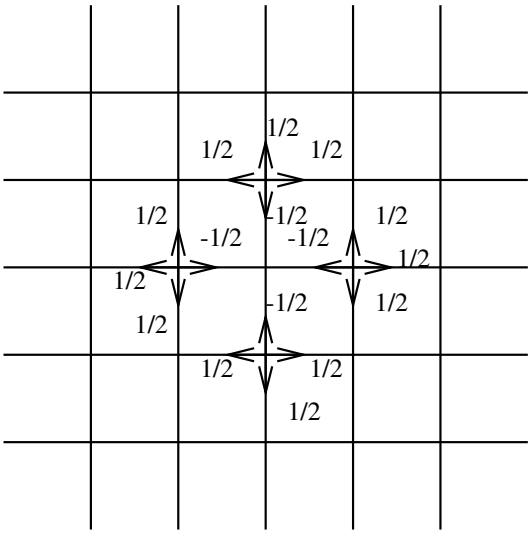
Figure 4.1: *Huygens principle.*



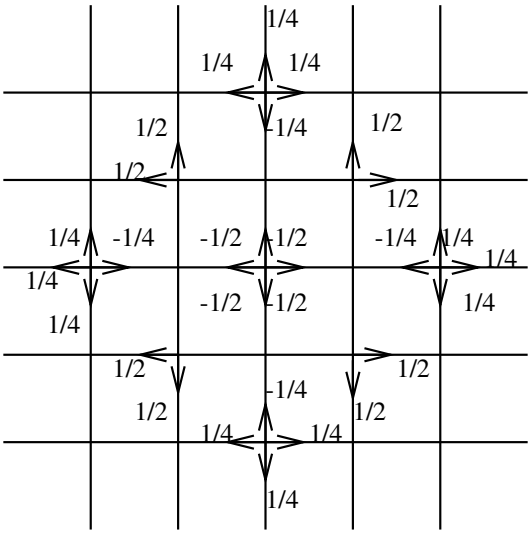
(a) At time $t = 0$ the centre point scatters a wave.



(b) Wave front at time $t = \Delta T$



(c) At this time ($t = \Delta T$) the points that are on wave front scatter waves.



(d) At time $t = 2\Delta T$ the wave front can be found by calculating waves scattering from points in fig4.2(c)

Figure 4.2: *Wave propagation in a two dimensional TLM mesh*

incident wave from its neighbours and scatters it to its neighbours. By repeating the above calculation for each node, the wave distribution on the medium can be calculated[6] .

4.3 Two dimensional TLM modelling

Based on the accuracy and complexity of the modelling medium, one can use 1, 2 or 3 dimensional TLM modelling. Two dimensional TLM modelling is the most popular one as it can model most of the problems and is more efficient compared to 3 dimensional TLM modelling. In this section two dimensional TLM modelling is explained in depth and in the section 4.4 (on page 77) TLM modelling in one and three dimensions will be explain.

4.3.1 TLM modelling of a homogeneous medium

When all sections of a medium have the same properties, the medium is referred as homogeneous. For modelling homogeneous media it does not need to consider the medium properties and hence it is possible to use the simple mesh shown in fig[4.2]. The mesh shown in fig[4.2] consists of several nodes. The model for one node is shown in fig[4.3]. In this fig, V^I represents the incident wave (voltage in the transmission line) and V^S represents the scattered wave.

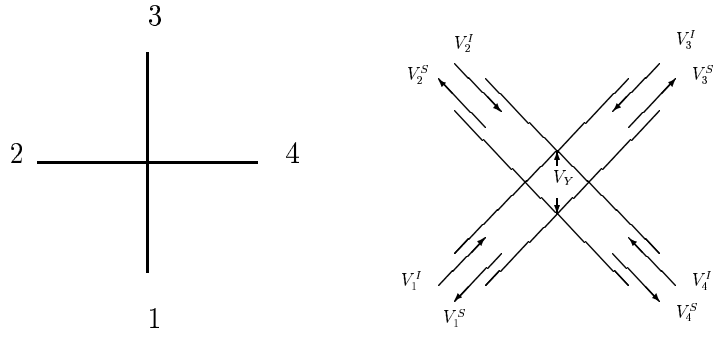
The relationship between the incident wave (voltage in transmission line) and the scattered wave is:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}^S_{K+1} = \frac{1}{2} \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}^I_K \tag{4.2}$$

I stands for the incident wave voltages and S stands for the scattered wave voltages, K and K+1 are arbitrary consecutive time steps separated by the sample interval ΔT. Based on this equation, if the magnitude of the wave (voltage in the TLM modelling) is known at any time $K\Delta T$ then the magnitude of wave in the mesh could be found at time $(K+1)\Delta T$. By repeating this for each time step, wave propagation could be modeled[7].

[6] TLM implementation will be discussed in section 4.3.6 on page 64

[7] TLM implementation will be discussed in sec 4.3.6 on page 64.



(a) Representation in a mesh **(b)** Actual model

Figure 4.3: *Model for a node in a TLM mesh*

4.3.2 TLM modelling of a non homogeneous medium

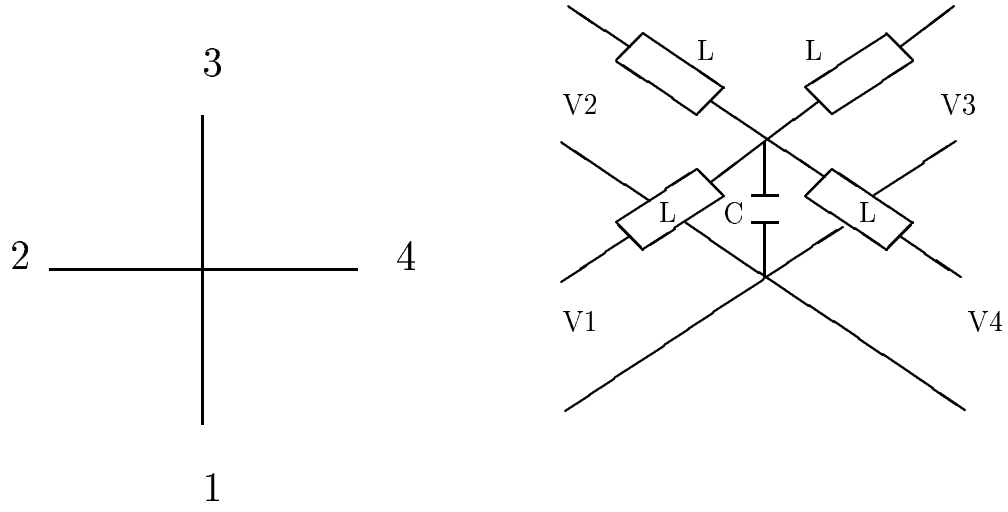
In the TLM model explained in sec 4.2.2 the wave propagates through a homogeneous medium and the properties of medium need not be considered. When modelling a non homogeneous medium, one should consider the properties of the medium in the model. For this reason a new model for a node is created as shown in fig 4.4 [69][70].

This model is valid when the medium is lossless. When there are some loss in the medium, there is a resistor in parallel to the capacitor (fig[4.5]) to model the loss [71].

For calculating scattering, one should calculate the scattering voltage based on equation 4.2 and then apply the impedances shown in fig[4.6].

4.3.3 Dispersion of velocity of waves in a TLM mesh

Johns and Beurle [59]showed that the speed of wave propagation in the mesh depends on the frequency of the wave. They calculate the following dispersion relation for propagation along



(a) Representation in a mesh **(b)** Actual model

Figure 4.4: *Model for a node in a non homogeneous medium ($L \equiv \mu$ and $C \equiv \frac{1}{2} \times \varepsilon$)*

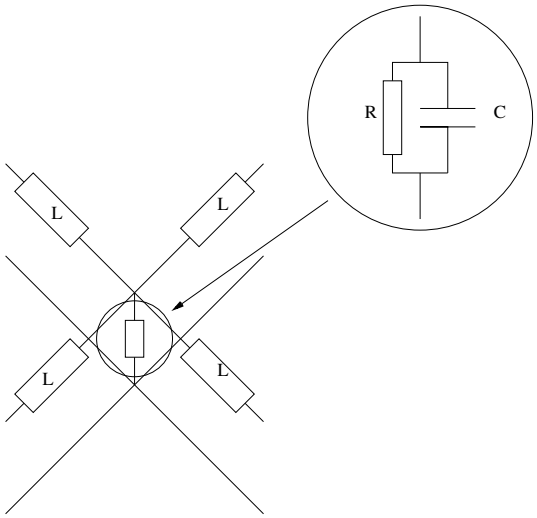


Figure 4.5: *Model for a node in a non homogeneous medium when the medium has some losses.*

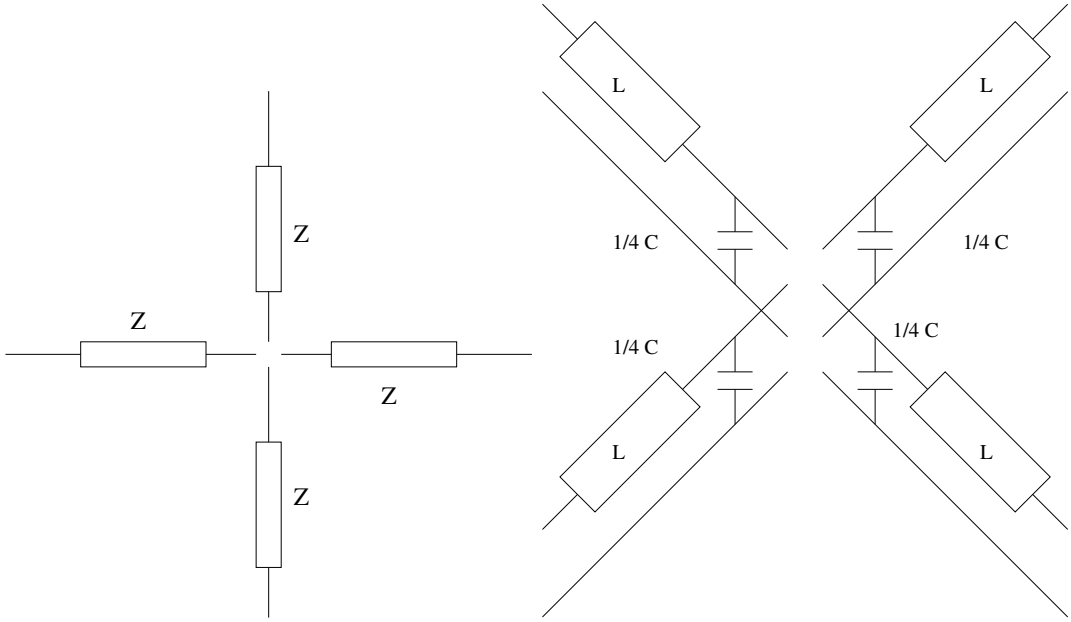


Figure 4.6: *Equivalent circuit for a node in a non homogeneous medium.*

the main mesh axes(X and Y axes in fig4.2):

$$\sin(\beta_n \frac{\Delta\ell}{2}) = \sqrt{2}\sin(\omega \frac{\Delta\ell}{(2C)}) \tag{4.3}$$

β_n is the propagation constant. In fig4.7 the resulting ratio of velocities on the matrix and in free space is shown.

When the wave propagates in axial direction (direction along X or Y axis), the cut-off frequency is at $\frac{\Delta\ell}{\lambda} = \frac{1}{4}$ (λ is the free space wavelength of the wave). However, no cut-off occurs in the diagonal direction, where the velocity of wave is frequency independent as is equal to $\frac{C}{\sqrt{2}}$. In intermediate directions the velocity ratio lies somewhere in between the two curves.

When the wave frequency is so that $4\Delta l \ll \lambda$ then the speed of wave in the axial direction is approximately equal to $\frac{C}{\sqrt{2}}$ and hence the TLM mesh model an isotropic medium[8].

This phenomenon is shown in fig[5.5] on page 90. In chapter 6 on page 121 this phenomenon is explained in more detail.

[8]Since there is a sample from signal on every Δl and the wave length is λ, then the TLM is an isotropic propagating medium when the sample per cycle of all frequencies in the medium be much higher than 4 sample per cycle.

*In conclusion:*The TLM network simulates an isotropic propagating medium only as long as all frequencies are well below the network cutoff frequency, in which case the network propagation velocity may be considered constant and equal to $\frac{C}{\sqrt{2}}$ where C is the wave speed in free space.

4.3.4 Wave propagation in a TLM mesh

As it was seen in the previous section, the speed of the wave in the medium is equal to $\frac{C}{\sqrt{2}}$ (when all frequencies in the medium are well below cutoff frequency). This behaviour is investigate in more detail in this section.

4.3.4.1 Notations

As the wave propagates in the TLM mesh, it passes thought several nodes to reach to it's destination. To show the path that the wave is passing to reach to a point, the following notations are used. The wave should start from the source and if the wave goes to left one would write L and if the waves go up he would write U and so R for right and D for down. At each point in the path the sign of the signal would be calculated (if the signal sign changed at that point the direction would be written in low case letter: r u l d respectively).

4.3.4.2 Diagonal direction

If the source is at point (50,50) and the receiver at point (51,51), the distance between the source and the receiver is $\frac{\Delta\ell}{\sqrt{2}}$. The wave reaches to the receiver at time $2\Delta T$, There are two path for wave to reach to the receiver and they are :

1. RU
2. UR

If the source generate an impulse (Delta) signal as defined:

$$\delta(nT) = \begin{cases} 1 & n = 0 \\ 0 & Otherwise \end{cases} \tag{4.4}$$

The received signal amplitude is 0.5. There isn't any path that the signal can reach to the receiver with time delay of $3\Delta T$. There are 24 paths that the signal can reach to the receiver

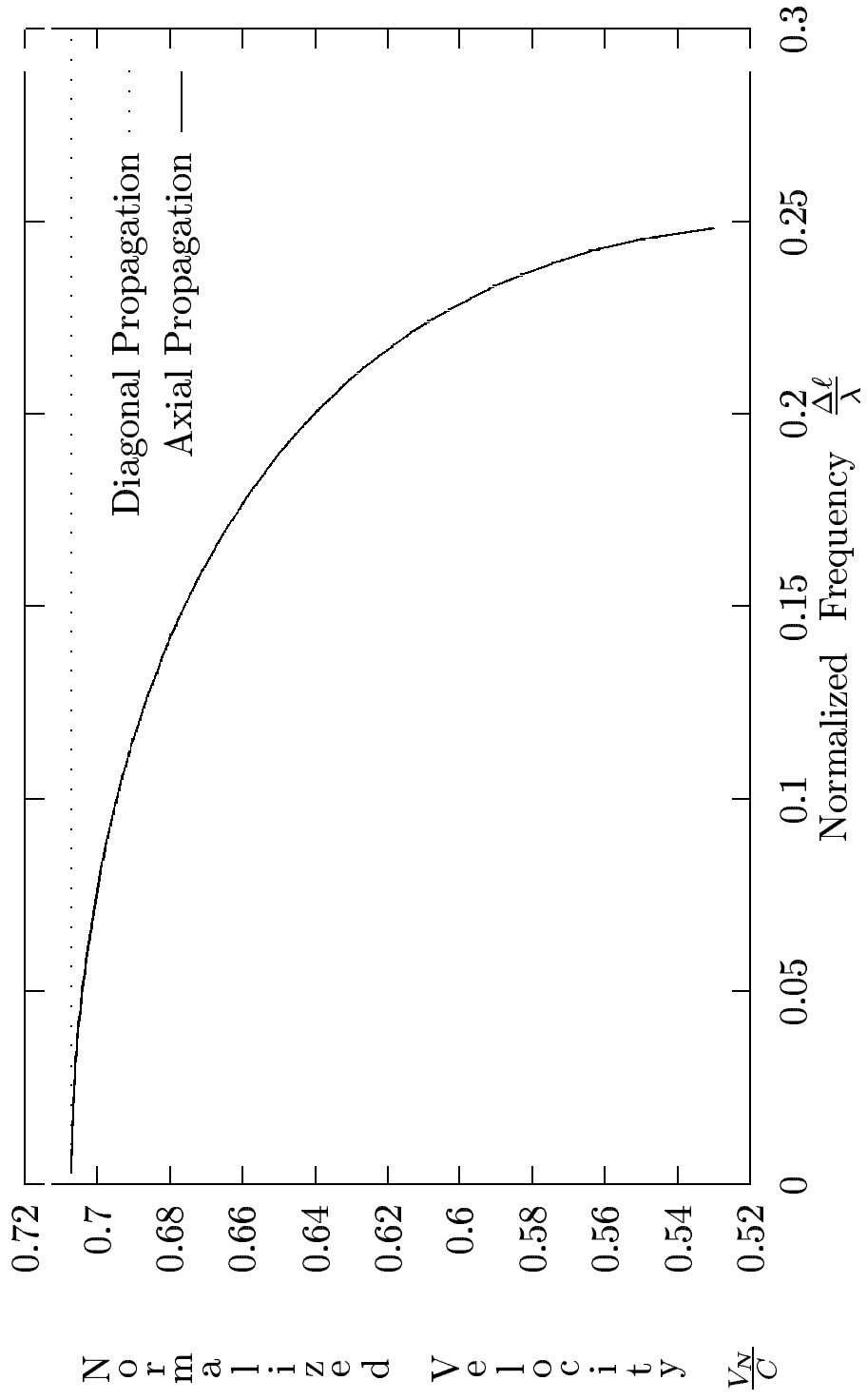


Figure 4.7: *Dispersion of the velocity of waves in a two-dimensional TLM network.*

with time delay of $4\Delta T$. These paths are shown in table[4.1].

#	Path	Amplitude	#	Path	Amplitude	#	Path	Amplitude
1	RRlU	-0.062500	2	RRUL	0.062500	3	RlrU	0.062500
4	RlUR	-0.062500	5	RURl	-0.062500	6	RULr	-0.062500
7	RUUd	-0.062500	8	RUdu	0.062500	9	RDuU	-0.062500
10	LrRU	-0.062500	11	LrUR	-0.062500	12	LURR	0.062500
13	URRl	-0.062500	14	URlr	0.062500	15	URUd	-0.062500
16	URDu	-0.062500	17	ULrR	-0.062500	18	UURD	0.062500
19	UUdR	-0.062500	20	UdRU	-0.062500	21	UduR	0.062500
22	DRUU	0.062500	23	DuRU	-0.062500	24	DuUR	-0.062500

Table 4.1: Paths from source to point as shown in fig[4.8] with time delay of $12\Delta T$

The wave comes with positive sign from 8 of these paths and with negative sign from 16 of these paths, the received signal at the receiver is -0.05. These signal is very weak comparing to the original signal that reaches to the receiver with time delay of 2 and so it has negligible effect on the output. The signals that reaches to this point with time delay of 6, 8, 10 ,.. are much weaker. Since all other signals other than the signal that reaches to the receiver with time delay of 2 are very weak, it is possible to ignore them and assume that the signal reaches to the receiver only with time delay of $2\Delta T$.[9] The reason that the speed in diagonal direction is $\frac{C}{\sqrt{2}}$ can bee seen in fig[4.8]. The wave reaches to point A at time =$2\Delta T$ but its distance from the source (point O) is only $\sqrt{2}\Delta l$. The speed can be calculated as follow:

$$V = \frac{Distance}{Time} \tag{4.5}$$

$$V = \frac{\Delta l \times \sqrt{2}}{\Delta T \times 2} \tag{4.6}$$

$$V = \frac{\Delta l}{\Delta T} \times \frac{1}{\sqrt{2}} \tag{4.7}$$

$$V = \frac{C}{\sqrt{2}} \tag{4.8}$$

[9]It is an approximation and to show how one may calculate the speed in diagonal direction. It would show later that this method doesn't show why the speed in the axial direction is related to speed and hence isn't a good method for studying speed in the TLM mesh.

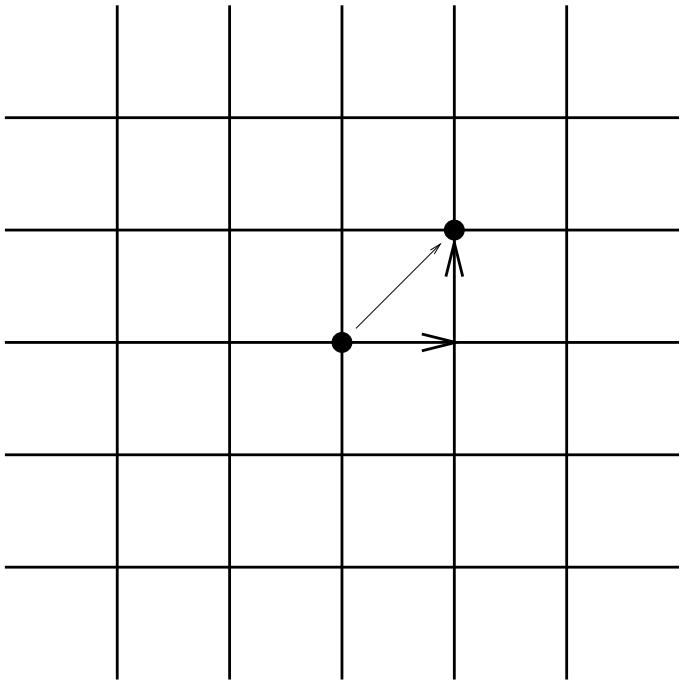


Figure 4.8: *Wave will reach to point A at time $2\Delta T$ But the distance between point O and A is only $\sqrt{2}\Delta l$.*

4.3.4.3 Axial direction

In the axial direction, it is much more complicated since the wave will reach to a point at $n\Delta l$ at time $n\Delta T$, so it seems that the speed of wave in this direction should be C and not $\frac{C}{\sqrt{2}}$ as shown in fig[4.9].

But a more careful investigation of how wave will be propagated in the medium would show that this theory isn't correct. This investigation can be done by finding how the wave will be received at any specific point in the medium. The selected point is shown in fig[4.10]. The distance between this point and the source is $10\Delta l$. The first signal reaches to this point at time $10\Delta T$. This path from source to the point is clearly the shortest one and is the line between these two point. If the source generate an impulse (Delta) signal as defined:

$$\delta(nT) = \begin{cases} 1 & n = 0 \\ 0 & Otherwise \end{cases} \tag{4.9}$$

The received signal at point A at this time ($10\Delta t$) is:

$$(\frac{1}{2})^{10} = 9.765625E^{-004} \tag{4.10}$$

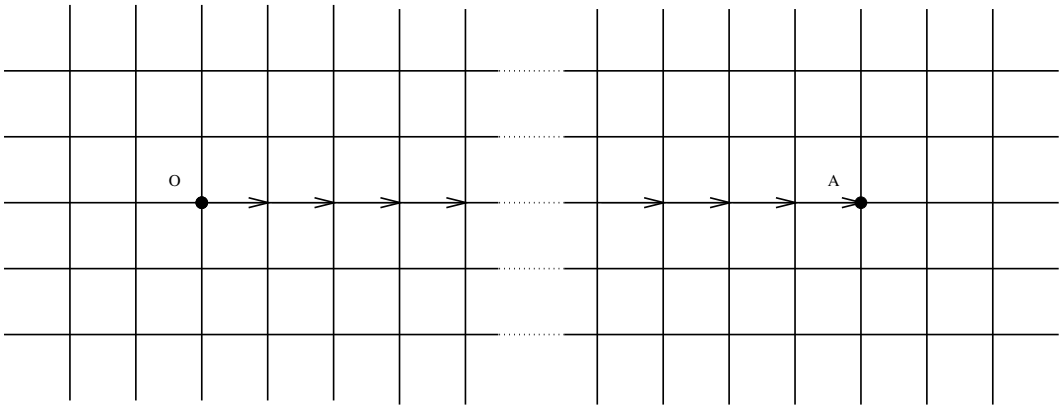


Figure 4.9: *Wave will reach to point A at time $n\Delta T$ and the distance between point O and A is $n\Delta l$.*

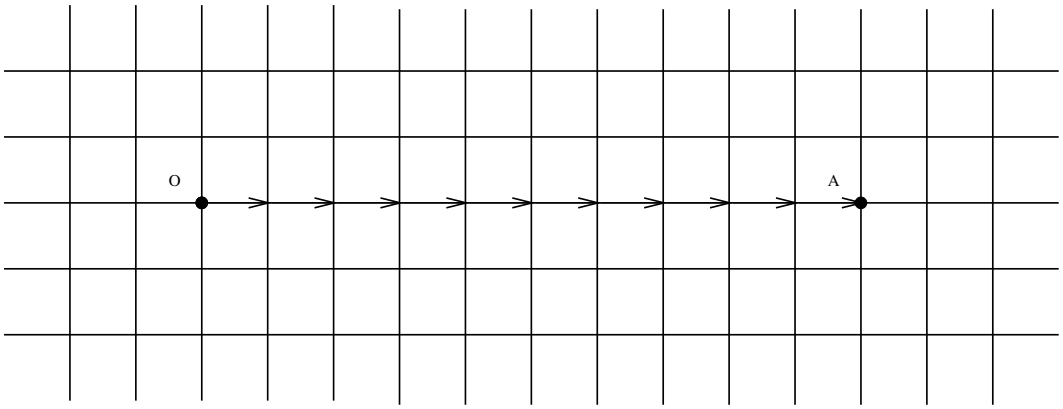


Figure 4.10: *The distance between point A and source is $10\Delta l$.*

There isn't any path that the signal can move along and reach to point A with time delay of $11\Delta T$. When the time delay is $12\Delta T$ there are 144 paths. All of these paths shown in table 4.2.

#	Path	Amplitude	#	Path	Amplitude	#	Path	Amplitude
1	RRRRRRRRRRl	-0.000244	2	RRRRRRRRRRlr	0.000244	3	RRRRRRRRRUd	-0.000244
4	RRRRRRRRRDu	-0.000244	5	RRRRRRRRRlrR	0.000244	6	RRRRRRRRRURD	0.000244
7	RRRRRRRRRUdR	-0.000244	8	RRRRRRRRRDRU	0.000244	9	RRRRRRRRRDuR	-0.000244
10	RRRRRRRRlrRR	0.000244	11	RRRRRRRRURRD	0.000244	12	RRRRRRRRURDR	0.000244
13	RRRRRRRRUdRR	-0.000244	14	RRRRRRRRDRRU	0.000244	15	RRRRRRRRDRUR	0.000244
16	RRRRRRRRDuRR	-0.000244	17	RRRRRRRlrRRR	0.000244	18	RRRRRRRURRRD	0.000244
19	RRRRRRRURRDR	0.000244	20	RRRRRRRURDRR	0.000244	21	RRRRRRRUdRRR	-0.000244
22	RRRRRRRDRRRU	0.000244	23	RRRRRRRDRRUR	0.000244	24	RRRRRRRDRURR	0.000244
25	RRRRRRRDuRRR	-0.000244	26	RRRRRRlrRRRR	0.000244	27	RRRRRRURRRRD	0.000244
28	RRRRRRURRRDR	0.000244	29	RRRRRRURRDRR	0.000244	30	RRRRRRURDRRR	0.000244
31	RRRRRRUdRRRR	-0.000244	32	RRRRRRDRRRRU	0.000244	33	RRRRRRDRRRUR	0.000244
34	RRRRRRDRRURR	0.000244	35	RRRRRRDRURRR	0.000244	36	RRRRRRDuRRRR	-0.000244
37	RRRRRlrRRRRR	0.000244	38	RRRRRURRRRRD	0.000244	39	RRRRRURRRRDR	0.000244
40	RRRRRURRRDRR	0.000244	41	RRRRRURRDRRR	0.000244	42	RRRRRURDRRRR	0.000244
43	RRRRRUdRRRRR	-0.000244	44	RRRRRDRRRRRU	0.000244	45	RRRRRDRRRRUR	0.000244
46	RRRRRDRRRURR	0.000244	47	RRRRRDRRURRR	0.000244	48	RRRRRDRURRRR	0.000244
49	RRRRRDuRRRRR	-0.000244	50	RRRRlrRRRRRR	0.000244	51	RRRRURRRRRRD	0.000244
52	RRRRURRRRRDR	0.000244	53	RRRRURRRRDRR	0.000244	54	RRRRURRRDRRR	0.000244
55	RRRRURRDRRRR	0.000244	56	RRRRURDRRRRR	0.000244	57	RRRRUdRRRRRR	-0.000244
58	RRRRDRRRRRRU	0.000244	59	RRRRDRRRRRUR	0.000244	60	RRRRDRRRRURR	0.000244
61	RRRRDRRRURRR	0.000244	62	RRRRDRRURRRR	0.000244	63	RRRRDRURRRRR	0.000244
64	RRRRDuRRRRRR	-0.000244	65	RRRlrRRRRRRR	0.000244	66	RRRURRRRRRRD	0.000244
67	RRRURRRRRRDR	0.000244	68	RRRURRRRRDRR	0.000244	69	RRRURRRRDRRR	0.000244
70	RRRURRRDRRRR	0.000244	71	RRRURRDRRRRR	0.000244	72	RRRURDRRRRRR	0.000244
73	RRRUdRRRRRRR	-0.000244	74	RRRDRRRRRRRU	0.000244	75	RRRDRRRRRRUR	0.000244
76	RRRDRRRRRURR	0.000244	77	RRRDRRRRURRR	0.000244	78	RRRDRRRURRRR	0.000244
79	RRRDRRURRRRR	0.000244	80	RRRDRURRRRRR	0.000244	81	RRRDuRRRRRRR	-0.000244
82	RRlrRRRRRRRR	0.000244	83	RRURRRRRRRRD	0.000244	84	RRURRRRRRRDR	0.000244
85	RRURRRRRRDRR	0.000244	86	RRURRRRRDRRR	0.000244	87	RRURRRRDRRRR	0.000244
88	RRURRRDRRRRR	0.000244	89	RRURRDRRRRRR	0.000244	90	RRURDRRRRRRR	0.000244
91	RRUdRRRRRRRR	-0.000244	92	RRDRRRRRRRRU	0.000244	93	RRDRRRRRRRUR	0.000244
94	RRDRRRRRRURR	0.000244	95	RRDRRRRRURRR	0.000244	96	RRDRRRRURRRR	0.000244
97	RRDRRRURRRRR	0.000244	98	RRDRRURRRRRR	0.000244	99	RRDRURRRRRRR	0.000244
100	RRDuRRRRRRRR	-0.000244	101	RlrRRRRRRRRR	0.000244	102	RURRRRRRRRRD	0.000244
103	RURRRRRRRRDR	0.000244	104	RURRRRRRRDRR	0.000244	105	RURRRRRRDRRR	0.000244
106	RURRRRRDRRRR	0.000244	107	RURRRRDRRRRR	0.000244	108	RURRRDRRRRRR	0.000244
109	RURRDRRRRRRR	0.000244	110	RURDRRRRRRRR	0.000244	111	RUdRRRRRRRRR	-0.000244
112	RDRRRRRRRRRU	0.000244	113	RDRRRRRRRRUR	0.000244	114	RDRRRRRRRURR	0.000244
115	RDRRRRRRURRR	0.000244	116	RDRRRRRURRRR	0.000244	117	RDRRRRURRRRR	0.000244
118	RDRRRURRRRRR	0.000244	119	RDRRURRRRRRR	0.000244	120	RDRURRRRRRRR	0.000244

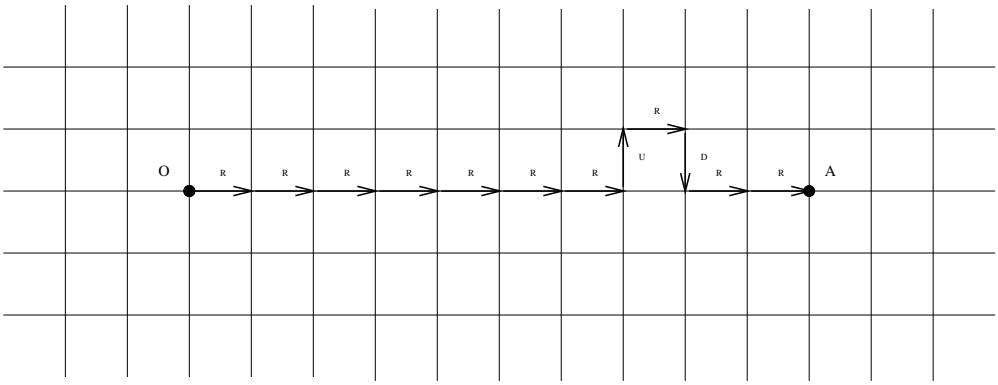


Figure 4.11: *A graphical demonstration of path RRRRRRRURDRR (path number 20 in table 4.2).*

#	Path	Amplitude	#	Path	Amplitude	#	Path	Amplitude
121	RDuRRRRRRRRR	-0.000244	122	LrRRRRRRRRRR	-0.000244	123	URRRRRRRRRRD	0.000244
124	URRRRRRRRRDR	0.000244	125	URRRRRRRRDRR	0.000244	126	URRRRRRRDRRR	0.000244
127	URRRRRRDRRRR	0.000244	128	URRRRRDRRRRR	0.000244	129	URRRRDRRRRRR	0.000244
130	URRRDRRRRRRR	0.000244	131	URRDRRRRRRRR	0.000244	132	URDRRRRRRRRR	0.000244
133	UdRRRRRRRRRR	-0.000244	134	DRRRRRRRRRRU	0.000244	135	DRRRRRRRRRUR	0.000244
136	DRRRRRRRRURR	0.000244	137	DRRRRRRRURRR	0.000244	138	DRRRRRRURRRR	0.000244
139	DRRRRRURRRRR	0.000244	140	DRRRRURRRRRR	0.000244	141	DRRRURRRRRRR	0.000244
142	DRRURRRRRRRR	0.000244	143	DRURRRRRRRRR	0.000244	144	DuRRRRRRRRRR	-0.000244

Table 4.2: Paths from source to point as shown in fig[4.10] with time delay of $12\Delta T$

For example path for RRRRRRRURDRR (path number 20 in table 4.2) shown in fig[4.11].

The received signal with a time delay of $12\Delta T$ can be calculated by adding the amplitudes for all of these paths, which is equal to $2.343750E^{-002}$. Compared to the received signal for time delay $10\Delta T$, even though that the received signal from each path is weaker, there are more paths, so the total signal that will be received is stronger.

If one tries to find paths for time delays $14\Delta T$[10], one would find that there are 8281 paths from the source to receiver at point (10,0). The signal from 4856 paths will be received with a positive sign and from 3416 paths it will be received with a negative sign. The amplitude of the received signal from each path is $6.103516E^{-005}$ and the total received signal at point (10,0) with time delay 14 is:

[10]There isn't any path with time delay 13

$$Amp = 4856 \times 6.103516E^{-005} - 3416 \times 6.103516E^{-005} \quad (4.11)$$

$$Amp = (4856 - 3416) \times 6.103516E^{-005} \quad (4.12)$$

$$Amp = 1449 \times 6.103516E^{-005} \quad (4.13)$$

$$Amp = 8.843994E^{-002} \quad (4.14)$$

Comparing this with time delay $12\Delta T$, it could be seen that the signal amplitude is stronger. The reason is that even if the signal from any one path is weaker, there are more paths than before and this compensates the fact that the signal from each path is weaker. By doing the same procedure for time delay $16\Delta T$ it could be seen that the signal for this time delay ($16\Delta T$) is weaker than signal for time delay $14\Delta T$. The received signals for time delays greater than $14\Delta T$ are weaker than received signal for time delay $14\Delta T$. Table 4.3 show this phenomena in detail. The case for time delay $18\Delta T$ is different. If the received signal for this time delay

Delay	Total paths	+ paths	- paths	Path amplitude	Total amplitude
$10\Delta T$	1	1	0	$9.765625E^{-004}$	$9.765625E^{-004}$
$12\Delta T$	144	120	24	$2.441406E^{-004}$	$2.343750E^{-002}$
$14\Delta T$	8281	4865	3416	$6.103516E^{-005}$	$8.843994E^{-002}$
$16\Delta T$	313600	157152	156448	$1.525879E^{-005}$	$1.074219E^{-002}$
$18\Delta T$	9363600	4666248	4697352	$3.814697E^{-006}$	$-1.186523E^{-001}$

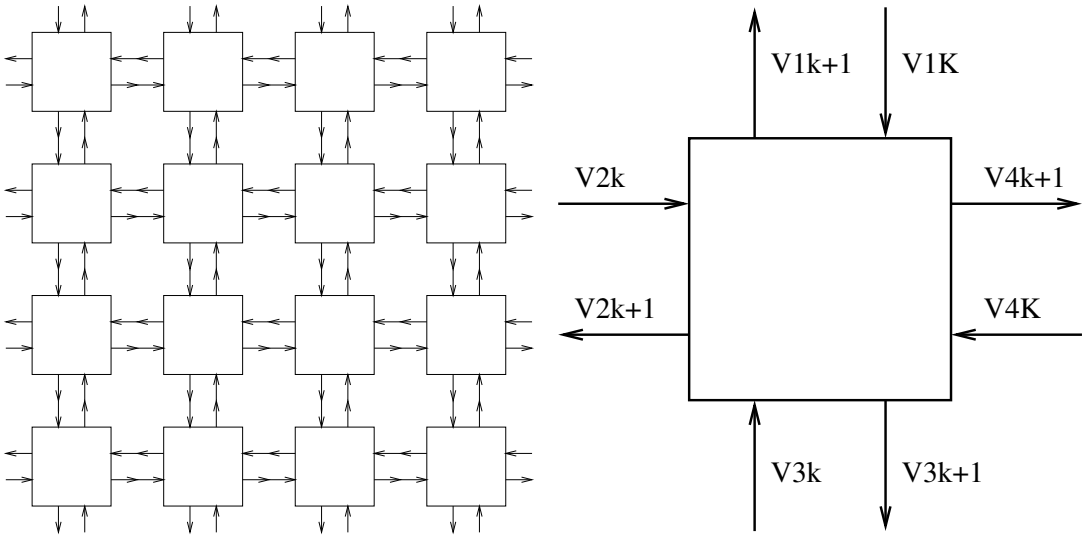
Table 4.3: *Comparison between received signal with different time delay.*

is calculated it is stronger than the received signal for time delay $14\Delta T$ but the sign of the received signal is negative . As the number of paths is increasing rapidly, it is not possible to find a clear understanding of how the received signal is generated based on the source signal.

4.3.5 TLM mesh as a digital filter

Since TLM algorithm is an iterative algorithm, the author suggested to model it as a digital filter[11] . Each node can be modelled as a digital filter with four input and four output. The TLM mesh is an interconnected mesh of these four input / four output digital filters. This

[11]This modelling is very important since when a TLM mesh is modelled as a digital filter, one may think of why the sampling rate theory which tells that it would be possible to recreate a signal with only two sample per cycle does not work here and also one may try to find to a transfer function for the TLM mesh. All of these investigations leads the investigator to find a new sampling rate theory which is explained in chapter 6 on page 121.



(a) The TLM mesh as a interconnected mesh of digital filters.

(b) One node in a TLM mesh as a four input/ four output digital filter.

Figure 4.12: *The TLM mesh can be modelled as an mesh of interconnected digital filters.*

shown in fig[4.12].

For finding the digital filter model for a node, the equation 4.2 which is in matrix format, could be written in the following form:

$$V_{1r}^{k+1} = \frac{1}{2}(-V_{1i}^{k} + V_{2i}^{k} + V_{3i}^{k} + V_{4i}^{k}) \tag{4.15}$$

$$V_{2r}^{k+1} = \frac{1}{2}(V_{1i}^{k} - V_{2i}^{k} + V_{3i}^{k} + V_{4i}^{k}) \tag{4.16}$$

$$V_{3r}^{k+1} = \frac{1}{2}(V_{1i}^{k} + V_{2i}^{k} - V_{3i}^{k} + V_{4i}^{k}) \tag{4.17}$$

$$V_{4r}^{k+1} = \frac{1}{2}(V_{1i}^{k} + V_{2i}^{k} + V_{3i}^{k} - V_{4i}^{k}) \tag{4.18}$$

To make it more like an iterative algorithm it could be written as:

$$V_{1r}(k+1) = \frac{1}{2}(-V_{1i}(k) + V_{2i}(k) + V_{3i}(k) + V_{4i}(k)) \tag{4.19}$$

$$V_{2r}(k+1) = \frac{1}{2}(V_{1i}(k) - V_{2i}(k) + V_{3i}(k) + V_{4i}(k)) \tag{4.20}$$

$$V_{3r}(k+1) = \frac{1}{2}(V_{1i}(k) + V_{2i}(k) - V_{3i}(k) + V_{4i}(k)) \tag{4.21}$$

$$V_{4r}(k+1) = \frac{1}{2}(V_{1i}(k) + V_{2i}(k) + V_{3i}(k) - V_{4i}(k)) \tag{4.22}$$

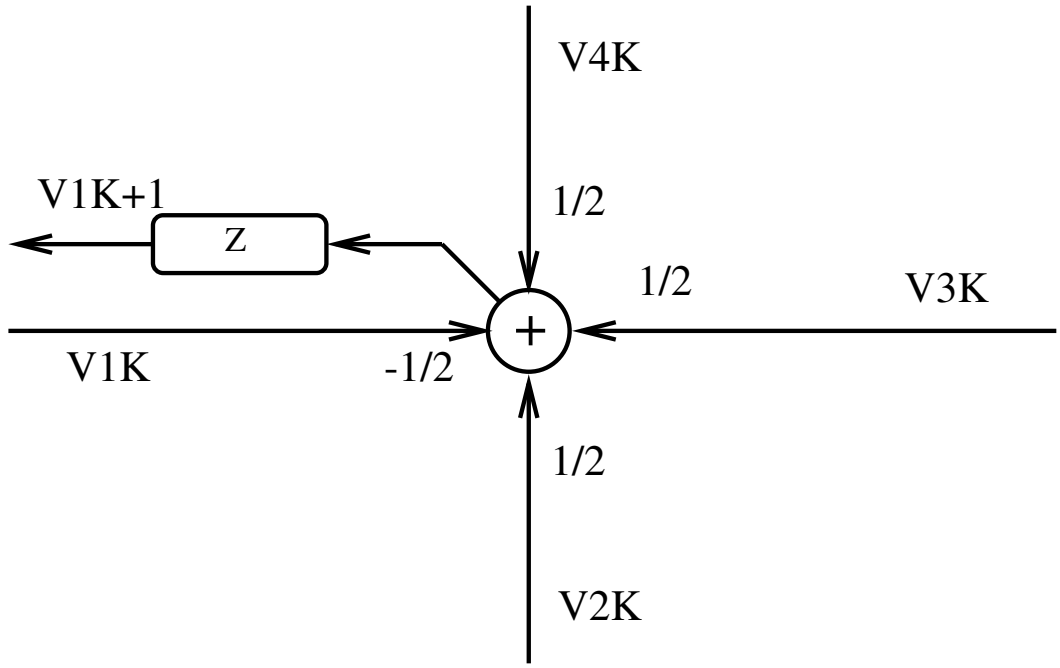


Figure 4.13: *The digital filter realization for $\frac{1}{4}$ of a node.*

Each of these equations can be realized by a digital filter as shown in fig[4.13]. By combining the four section of a node to each other the model for a whole node can be created as shown in fig[4.14].

4.3.6 Implementing a TLM model

Since TLM is a numerical model, it should be implemented in software or hardware. Implementing it in software is the fastest way to create a model but by implementing it in hardware, the fastest way for modelling wave propagation could be obtained.

The algorithm for implementing TLM in software and hardware is approximately the same. It consist of two steps:

Incident: At this step the value of output voltages for next time step will be calculated. It could

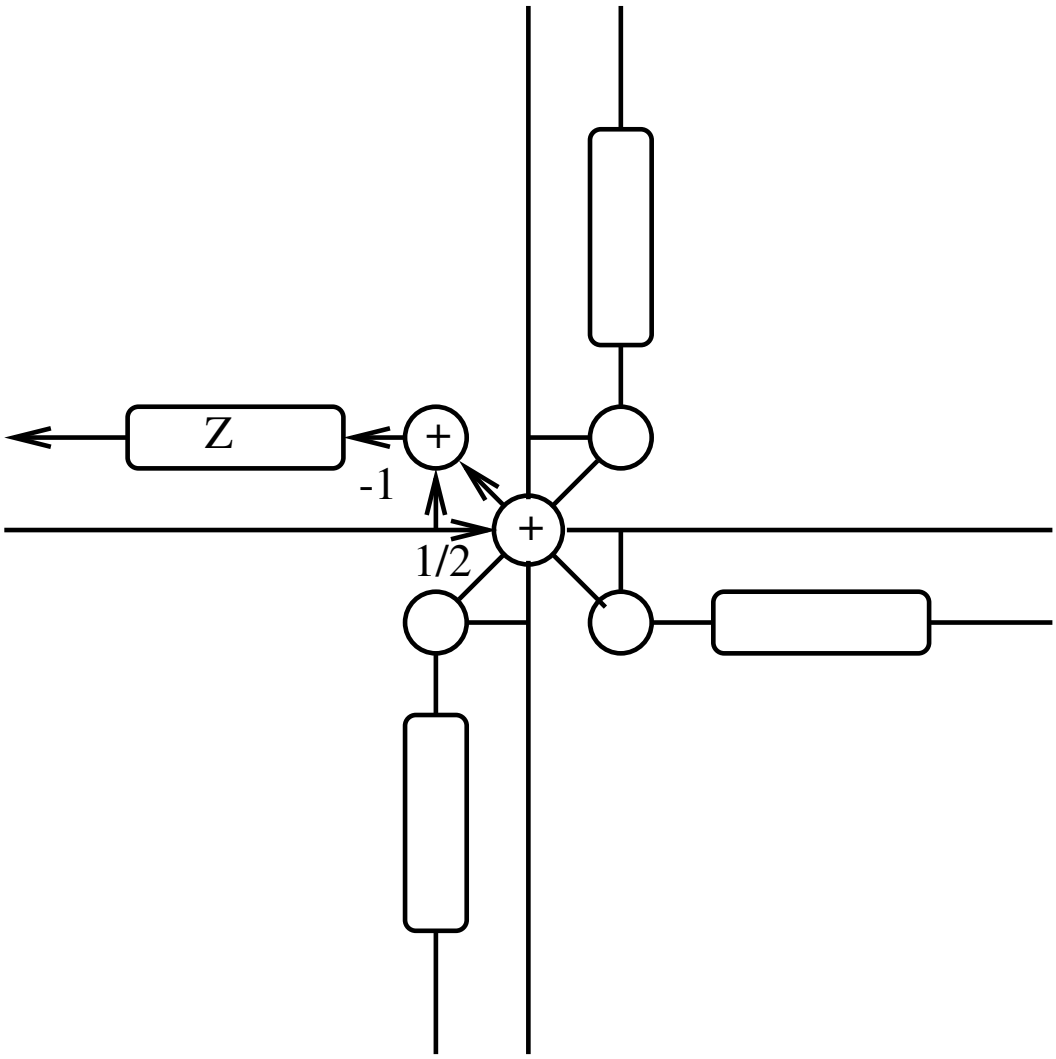


Figure 4.14: *The Digital filter realization for a node.*

be done by using the following set of formula:

$$V_{1r}^{k+1} = \frac{1}{2}(-V_{1i}^{k} + V_{2i}^{k} + V_{3i}^{k} + V_{4i}^{k}) \tag{4.23}$$

$$V_{2r}^{k+1} = \frac{1}{2}(V_{1i}^{k} - V_{2i}^{k} + V_{3i}^{k} + V_{4i}^{k}) \tag{4.24}$$

$$V_{3r}^{k+1} = \frac{1}{2}(V_{1i}^{k} + V_{2i}^{k} - V_{3i}^{k} + V_{4i}^{k}) \tag{4.25}$$

$$V_{4r}^{k+1} = \frac{1}{2}(V_{1i}^{k} + V_{2i}^{k} + V_{3i}^{k} - V_{4i}^{k}) \tag{4.26}$$

Scattering: At this step, the output voltage from each node would be used as the input of the neighbouring nodes. This could be done by using the following set of formulae:

$$V_{3}^{node(x,y+1)} = V_{1}^{node(x,y)} \tag{4.27}$$

$$V_{4}^{node(x+1,y)} = V_{2}^{node(x,y)} \tag{4.28}$$

$$V_{1}^{node(x,y-1)} = V_{3}^{node(x,y)} \tag{4.29}$$

$$V_{2}^{node(x-1,y)} = V_{4}^{node(x,y)} \tag{4.30}$$

This is shown in fig[4.15].

These formulae are for two dimensional TLM. For one dimensional and three dimensional TLM modelling, the steps are the same but the number of input and output voltages is different, depending on the number of dimensions being modelled, for example for one dimensional TLM modelling we have only two voltages and hence the formula will be reduced to two formulae for each step.

4.3.6.1 In a single CPU computer

When working with a single CPU, all of the processing is done in one CPU and hence the same algorithm could be repeated for all nodes in the medium.

There are two type of algorithms for implementation of TLM :

Speed optimized: This model has the fastest possible speed.

Memory optimized: This model uses the least possible memory.

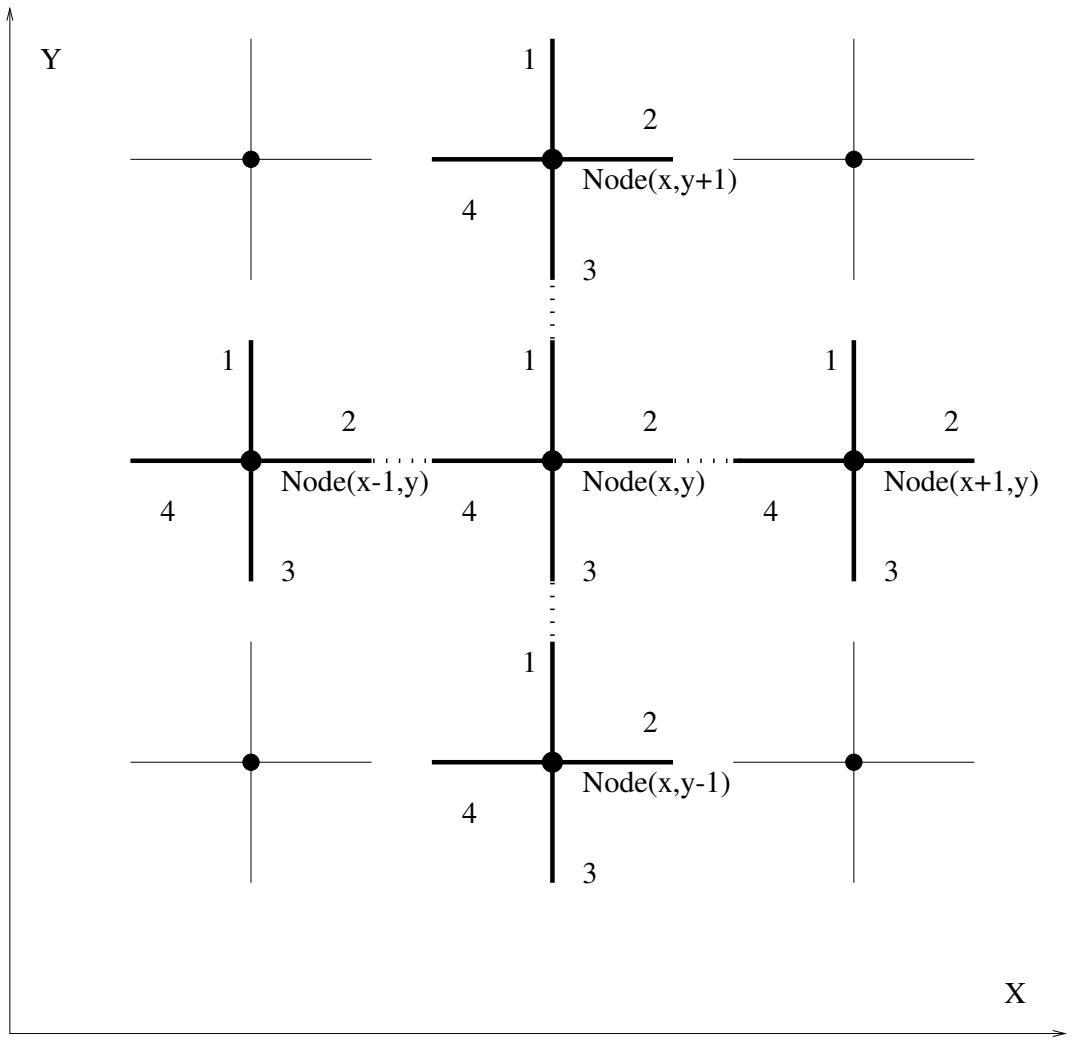


Figure 4.15: *Node placement for scattering step.*

4.3.6.2 Speed optimized implementation

In this type of implementation, more memory will be used but the number of executed instruction is less, so it is faster than the other version of this algorithm.

In this algorithm, there is 8 memory locations for each node. Four of these memory locations are for input voltages and the other four locations are for output voltages. During the incident step, the value of the next output voltages are calculated based on input voltages and put in the memory locations that are reserved for these output voltages. In the scattering step, the output voltages of each node is applied as the input voltages for neighbours nodes. The algorithm is as follow:

Require: 8 memory location for each node.

1: Initialize memory locations.
2: **for all** time steps in simulation time **do**
3: **for all** nodes in the mesh **do**
4: Calculate the output voltages and put them in output voltage memory locations.
5: **for all** nodes in the mesh **do**
6: Exchange voltages between nodes. Put output voltages to the input of neighbouring node.
7: **end for**
8: **end for**
9: **end for**

The total memory needed to model a medium with N nodes and T time steps is:

$$Memory = 8 \times N \tag{4.31}$$

For calculating each output for each node (incident section) and exchange it with neighbouring nodes (scattering section), the following operations should be done[12]:

$$Additions = 20 \tag{4.32}$$

$$Multiplications = 4 \tag{4.33}$$

$$Memory\ movements = 4 \tag{4.34}$$

[12]The additions are 16 for incident calculation and 4 to find neighbouring nodes.

So the number of operations that should be done to finish a TLM modelling with N nodes and T time steps would be:

$$Additions = 20 \times N \times T \tag{4.35}$$

$$Multiplications = 4 \times N \times T \tag{4.36}$$

$$Memory\ movements = 4 \times N \times T \tag{4.37}$$

This algorithm is useful when one wants to model a small mesh for a long time. In this situation, since a small mesh is modelled , hence the increase in memory is not important. It is also good for implementing in hardware as the operations at each node are separated from operations at all other nodes. One way for designing a hardware implementation of TLM model is to design a small CPU for each node with 8 memory location on it and then interconnect these small CPUs to each other to make a mesh.

4.3.6.3 Memory optimized implementation

With this algorithm, less memory will be used but the number of operations is higher (compared with the previous algorithm). There is only need for 4 memory locations for each node. The number of operations is higher since some kind of temporary memory for holding data during operations should be used and then these temporary data should be returned to original memory locations which adds some memory moves to the operation. The algorithm is as follow:

Require: 4 memory locations for each node and 4 temporary memory locations.

1: Initialize memory locations.
2: **for all** time steps **do**
3: **for all** nodes in the mesh **do**
4: Calculate all of the output voltages and put them in temporary memory locations.
5: Transfer the content of the temporary memory locations to the input voltages locations.
6: **for all** nodes in the mesh **do**
7: TMP $\leftarrow V_1^{node(x,y)}$
8: $V_1^{node(x,y)} \leftarrow V_3^{node(x,y+1)}$
9: $V_3^{node(x,y+1)} \leftarrow TMP$

10: TMP $\leftarrow V_2^{node(x,y)}$
11: $V_2^{node(x,y)} \leftarrow V_4^{node(x+1,y)}$
12: $V_4^{node(x+1,y)} \leftarrow TMP$
13: **end for**
14: **end for**
15: **end for**

In this algorithm 4 memory locations is needed for each node, so in the case of modelling a mesh with N node, the total memory needed is:

$$Memory = 8 \times N \tag{4.38}$$

The number of operations needed to process on each node is:

$$Additions = 20 \tag{4.39}$$
$$Multiplications = 4 \tag{4.40}$$
$$Memory\ movements = 10 \tag{4.41}$$

So the number of operations required to finish a TLM modelling with N nodes and T time steps would be:

$$Additions = 20 \times N \times T \tag{4.42}$$
$$Multiplications = 4 \times N \times T \tag{4.43}$$
$$Memory\ movements = 10 \times N \times T \tag{4.44}$$

This algorithm is a good algorithm when one wants to model a big mesh over a short time steps.

4.3.6.4 In a parallel computer

Since TLM needs a large amount of memory and CPU power to model wave propagation, it is impossible to model a real wave propagation problem with normal computers[13]. For example if the size of the medium that we want to model is 1cm by 1cm by 1cm and we are using three dimensional TLM modelling and the speed of wave in the medium is $1500\frac{m}{s}$[14] and the wave frequency be 1.5 MHZ[15] then:

$$Wavelength = \frac{Speed}{Frequency} \tag{4.45}$$

$$Wavelength = \frac{1500}{1.5 \times 10^6} \tag{4.46}$$

$$Wavelength = 10^{-3}(m) \tag{4.47}$$

For creating a homogeneous medium, there is a need to model with at least 50 sample per cycle[16], so we need (SPC = Sample Per Cycle) :

$$Number\ of\ nodes = \frac{size\ X \times SPC}{Wavelength} \times \frac{size\ Y \times SPC}{Wavelength} \times \frac{size\ Z \times SPC}{Wavelength} \tag{4.48}$$

$$Number\ of\ nodes = \frac{10^{-2} \times 50}{10^{-3}} \times \frac{10^{-2} \times 50}{10^{-3}} \times \frac{10^{-2} \times 50}{10^{-3}} \tag{4.49}$$

$$Number\ of\ nodes = 1.25 \times 10^8 \tag{4.50}$$

If for storing each data we use 8 bytes (double precision) then each node needs 32 bytes of memory and the memory needed for modelling is:

[13] At the time of this writing, a good workstation has 512 Mega-byte of RAM.

[14] Approximately the speed of sound in the water [48], [38].

[15] The normal frequency for medical ultrasound imaging.

[16] In chapter 6 on page 121 a new sampling rate theory will be explained which will dramatically reduce this number.

$$Memory\ needed = Number\ of\ nodes \times 32 \tag{4.51}$$

$$Memory\ needed = 1.25 \times 10^8 \times 32 \tag{4.52}$$

$$Memory\ needed = 4 \times 10^9 \tag{4.53}$$

$$Memory\ needed = 4GByte \tag{4.54}$$

This amount of memory is available only in parallel computers[17]. For this reason, especially when the model is 3 dimensional, the modelling should be done in parallel computers.
For modelling in a parallel computer[18], we divide a medium to some sections and then process each section on one CPU, the result of these process would be exchanged with each other. In fig[4.16], the medium that we want to model is shown. Before we can divide the medium we should find out how many CPU's are avaliable[19]. If we have N CPU's we should divide the medium into N sections. The best way is to divide it in a symmetrical way[20]. In fig[4.17] this division for some number of CPU is shown. After dividing the medium to sub-media, each processor models one sub-media. At the boundary, when it is necessary to exchange data with next sub-media, the data is put in a buffer. when the processing for one incident step is finished, the buffer on each CPU is exchanged with the corresponding buffer in the neighbouring CPU. Since it is very difficult to show this procedure in 3D, A 2 dimensional model is shown in fig[4.18]. The modelling starts by setting up the buffer and medium node data to initial conditions (fig[4.18(a)]). During the incident step, when any node at the boundary needs data from a neighbouring node which resides on a different CPU, it should get it from the corresponding buffer (fig[4.18(b)]) After the incident step, the scattering step for all nodes in the sub medium would be done. The scattering data from a node on the boundary that would be written to a node in a medium on another CPU, should be written to the buffer (fig[4.18(c)]). Then the buffer on each CPU would be exchanged with the corresponding buffer in the neighbour CPU[21] (fig[4.18(d)]). This process would be repeated for all time steps.

[17] It should be noted that it is true at the time of writing.

[18] The exact algorithm that should be used in a parallel computer depends of type of parallel computer in use and the language that the software wants to write with it. There are some other types of algorithms for different parallel computer structure [72] [73]. MPI (Message Passing Interface) was used for writing parallel codes that can be run on several platforms [74] and the programming language is C.

[19] In MPI use command: MPI_Comm_size().

[20] In MPI use command: MPI_Dims_create().

[21] In MPI use commands: MPI_Isend() and MPI_Irecv()

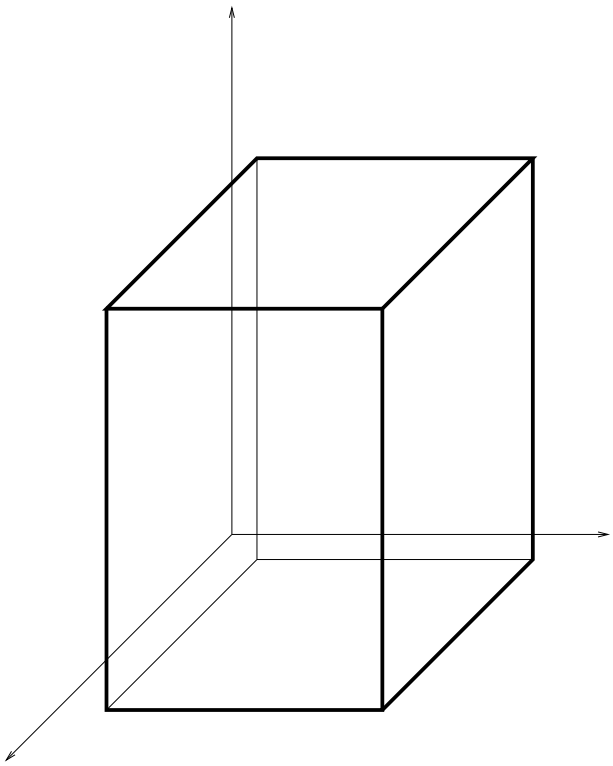


Figure 4.16: *A 3 dimension medium for modelling.*

Dead lock Dead lock happens in parallel programming[22] when one CPU is waiting for another CPU and the other CPU for some reason is waiting for the first CPU (see fig[4.19]). In the simplest form, CPU A is waiting for CPU B (for example to receive data from it) and CPU B is waiting for CPU A as shown in fig[4.19(a)]. In a more complex format of dead lock, CPU A is waiting for CPU B when CPU B is waiting for CPU C and ... CPU X is waiting for CPU A as shown in fig[4.19(b)].

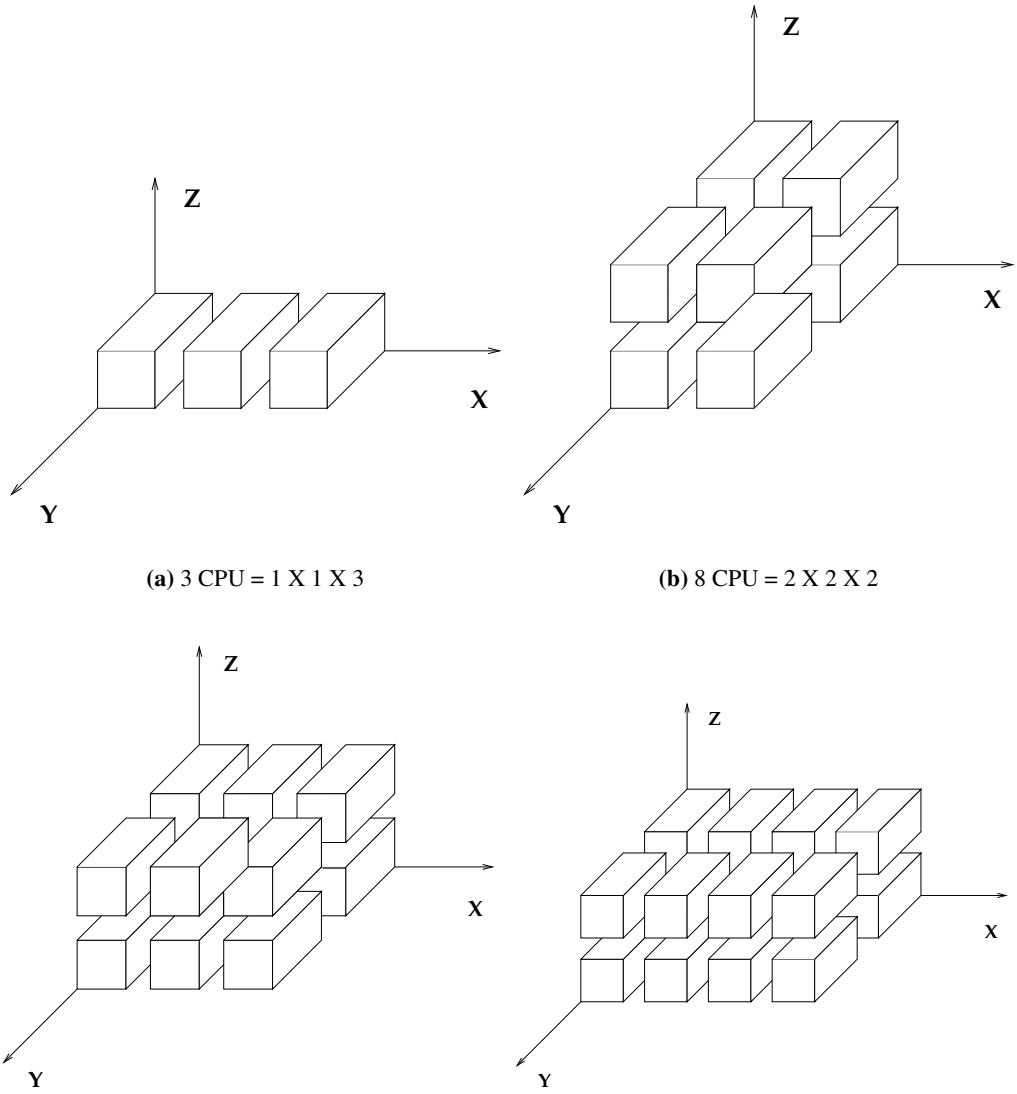
Deadlock should be solved by using some of the parallel computer features[23]. If the parallel computer feature can't help us to solve the problem, the following algorithm can solve it. It will be explained for changing the buffers among the CPUs in the X direction. It can be simply extended to exchange buffers in Y and Z directions. The CPUs should be numbered in X direction. The odd numbered CPUs use the following algorithm:

Require: Odd numbered CPU (CPU number is 2n+1)

1: Send corresponding buffer to CPU number 2(n+1)

[22]Theoretically this problem shouldn't be in a serial program.

[23]In MPI, data would be sent in a non blocking format and then after sending all of the buffers (4 buffer for 2D modelling and 6 buffer for 3D modelling) the program should wait to receive data from the other CPUs.



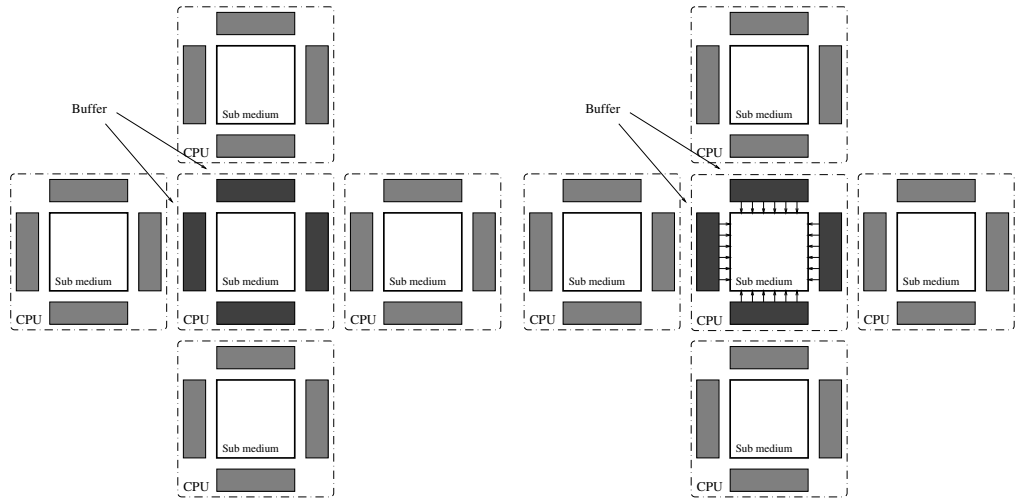
(a) 3 CPU = 1 X 1 X 3

(b) 8 CPU = 2 X 2 X 2

(c) 12 CPU = 2 X 2 X 3

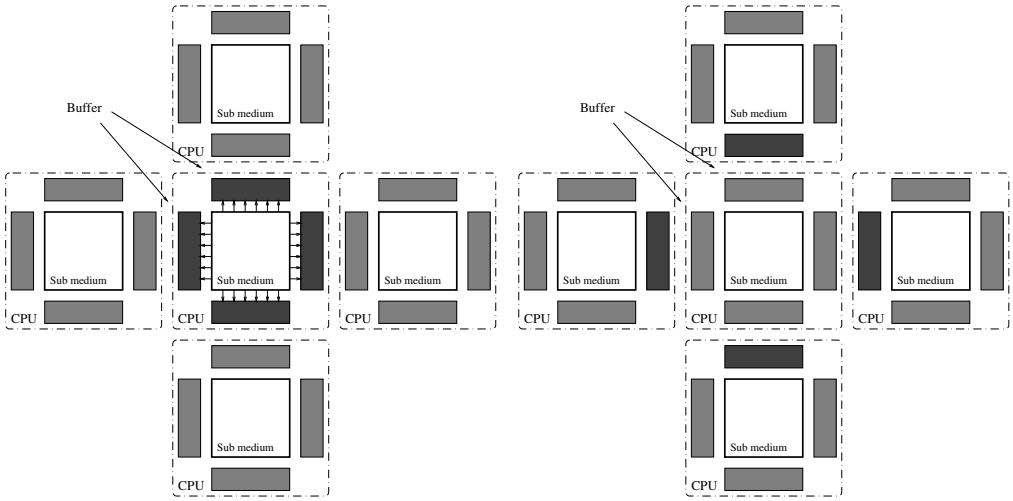
(d) 24 CPU = 2 X 2 X 4

Figure 4.17: *Division of the medium to sub-media.*



(a) Initialize buffer and medium

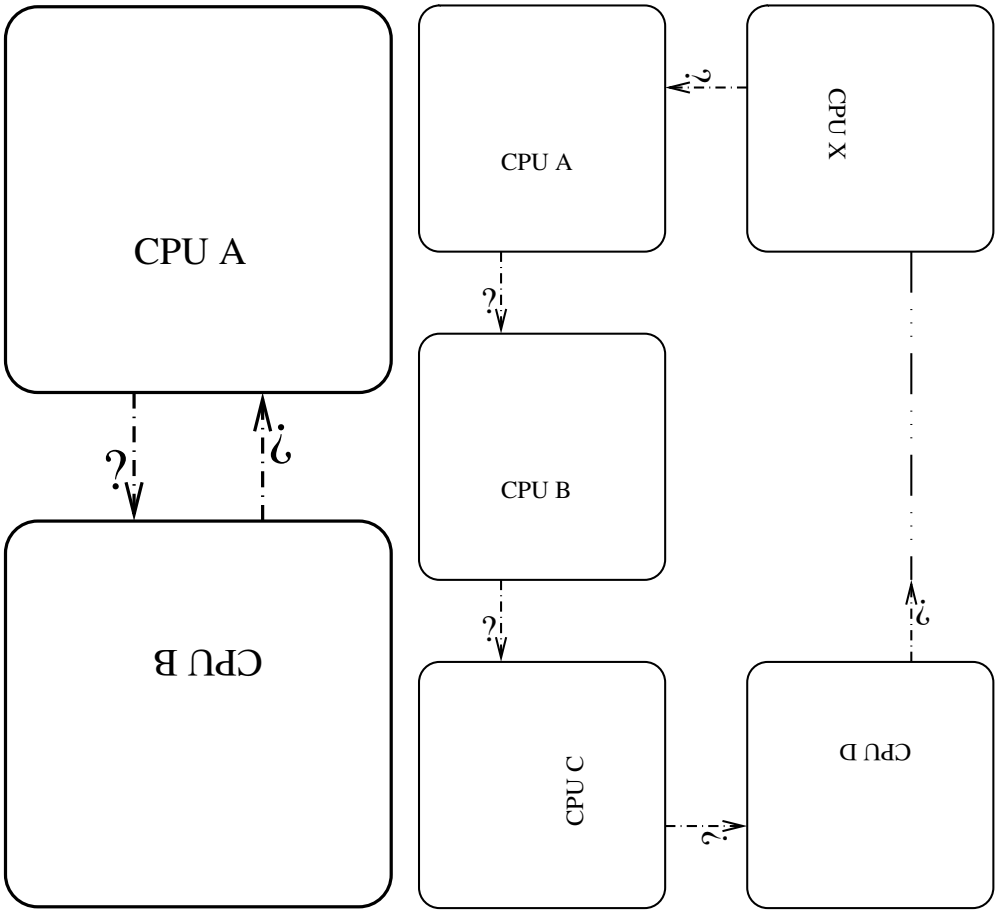
(b) During the incident step when any node at the boundary needs data from a node on a sub-media on a different CPU, it is got from the buffer



(c) During the scattering step, data from a node on boundary which wants to go to a node on a sub-media on a different CPU would be written to the buffer.

(d) At the end of scattering on each sub-media, the buffers between two adjacent sub-media would be exchanged with each other.

Figure 4.18: *Algorithm for modelling in a parallel computer.*



(a) CPU A waits for CPU B while CPU B is waiting for CPU A

(b) CPU A is waits for CPU B while CPU B is waiting for CPU C and ... CPU X is waiting for CPU A

Figure 4.19: *Dead lock in modelling with a parallel computer.*

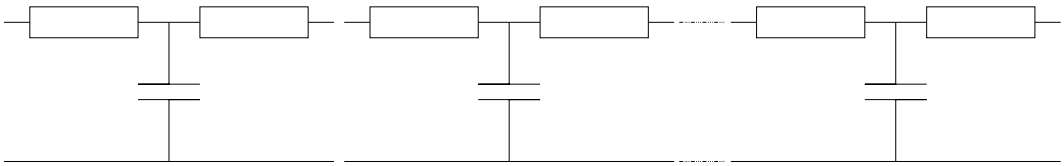


Figure 4.20: *TLM modelling in one dimension.*

2: Receive buffer from CPU number 2(n+1)

3: Send corresponding buffer to CPU number 2n

4: Receive buffer from CPU number 2n

The even numbered CPUs should use the following algorithm:

Require: Even numbered CPU (CPU number is 2n)

1: Receive buffer from CPU number 2n-1 and put it in a temporary buffer.

2: Send corresponding buffer to CPU number 2n-1

3: write the value of temporary buffer to original buffer.

4: Receive buffer from CPU number 2n+1 and put it in a temporary buffer.

5: Send corresponding buffer to CPU number 2n+1

6: write the value of temporary buffer to original buffer.

It should be noted that efficiency of these algorithms depends on the parallel computer hardware and software and is outwith the scope of this writing.

4.4 One dimensional and three dimensional TLM modelling

The two dimensional TLM modelling was explained in detail in the previous section. TLM modelling in one dimension and three dimensions is explained here briefly. Most of the theories for 2 dimensional modelling can easily be extended to one dimensional and three dimensional modelling.

4.4.1 One dimensional TLM modelling

One dimensional TLM modelling is very similar to the transmission lines explained in section 3.3. A one dimensional TLM model is shown in fig[4.20].

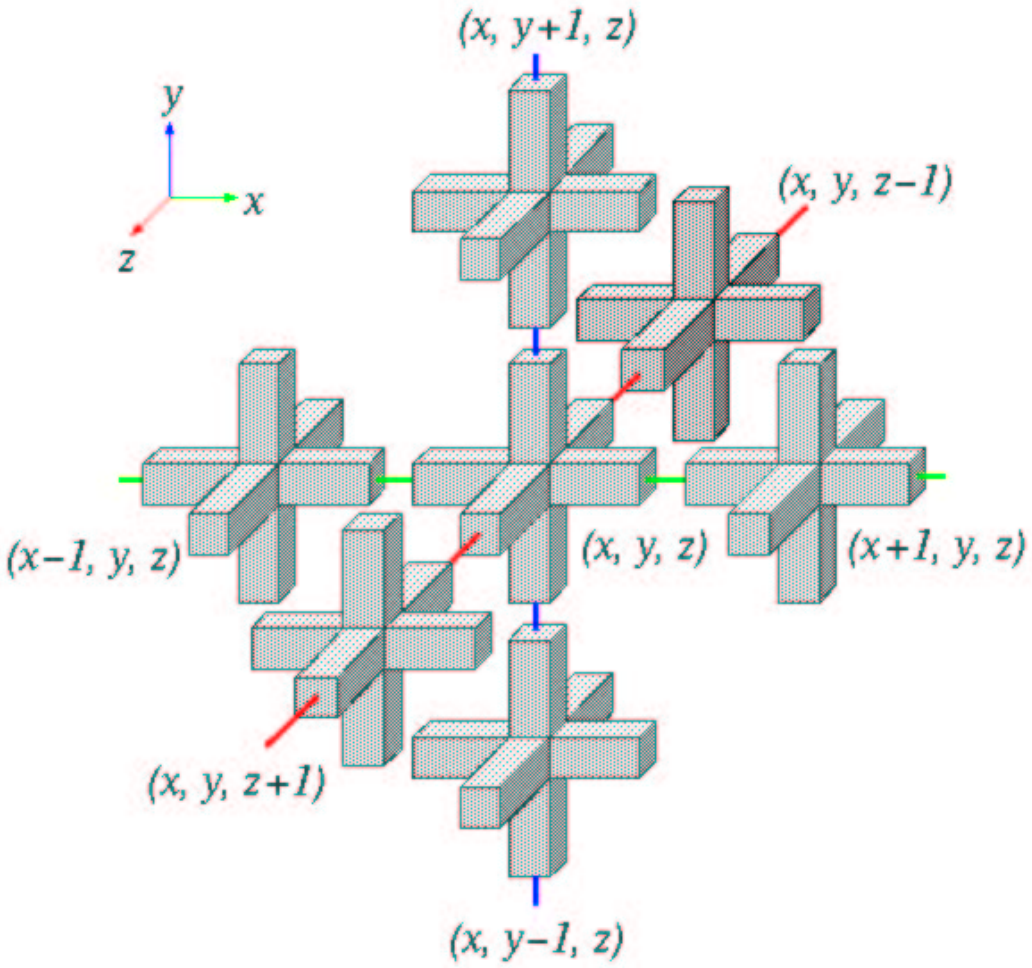


Figure 4.21: *TLM modelling in three dimensions [9].*

4.4.2 Three dimensional TLM modelling

In three dimensional TLM modelling, each node connects to 6 neighbouring nodes as shown in fig[4.21].

The node model depends on the wave type: vector wave or scalar wave.

Since electromagnetic waves are vector waves, the node model for them is very complex. Since the sound waves are scalar waves, it is possible to use a scalar TLM to model them. The node model for a scalar TLM model is shown in fig[4.22].

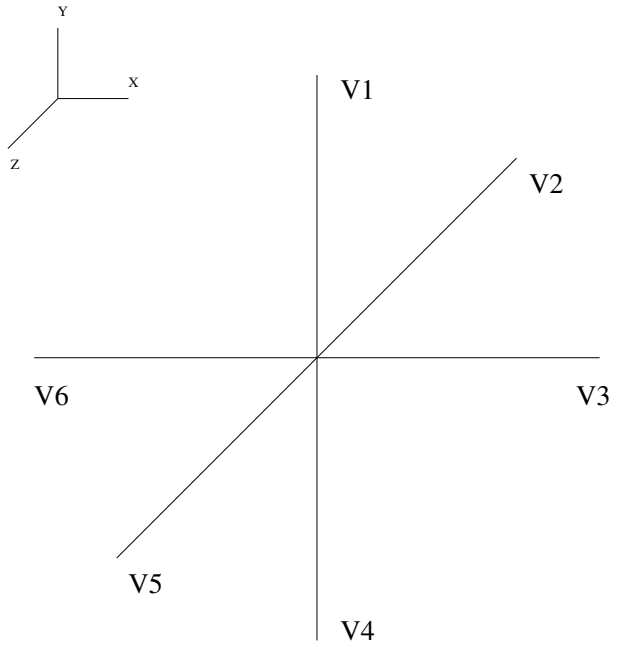


Figure 4.22: *Node model for scalar TLM modelling.*

The relationship between input voltages to each node and scattered voltages from each node is:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix}_{K+1}^{S} = \frac{1}{3} \begin{bmatrix} -2 & 1 & 1 & 1 & 1 & 1 \\ 1 & -2 & 1 & 1 & 1 & 1 \\ 1 & 1 & -2 & 1 & 1 & 1 \\ 1 & 1 & 1 & -2 & 1 & 1 \\ 1 & 1 & 1 & 1 & -2 & 1 \\ 1 & 1 & 1 & 1 & 1 & -2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix}_{K}^{I} \tag{4.55}$$

4.5 Inverse TLM modelling

Sorrentino and Hoefer [75]showed that the TLM can be reversed . They showed that the transmission matrix for each node S and its inverse S^{-1} are equal (This assumes a linear, Homoge-

neous medium) :

$$S = S^{-1} \tag{4.56}$$

$$S = \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}$$

This means that it should be possible to use the TLM modelling algorithm for inverse TLM modelling [76].

When we are working with inverse TLM modelling, we are working in inverse time so that we can work backward in time to find the source from a knowledge of its wave fields [77],[78],[79].

4.6 Other topics in TLM modelling

4.6.1 Modelling non linear waves with TLM

By replacing C and/or L to a nonlinear device, The TLM can be used to model non linear waves [80].

4.6.2 Different type of meshes

The TLM mesh that was explained here was square meshes, There are some active research on other types of meshes [81].

4.7 TLM modelling of ultrasound wave propagation

The TLM was originally used for modelling electromagnetic wave propagation. In electromagnetic wave propagation modelling , the electric and magnetic waves are modelled in the transmission line by voltage and current respectively. For modelling the ultrasound wave propagation we can use the similarity between the electromagnetic waves and ultrasound waves (Table 4.4)[82], [83].

Then the mechanical ultrasound wave is modelled electrically as an equivalent electromagnetic wave. This follows a long establishment practice of modelling mechanical systems as electrical analog.

Electromagnetic Wave	Ultrasound Wave
Electric Field	Pressure Field
Magnetic Field	Displacement Field
Current	Velocity
Permittivity (ε)	Compressibility (β)
Permeability (μ)	Density (ρ)

Table 4.4: *Similarity between Electromagnetic waves and Ultrasound waves*

4.8 Conclusions

TLM is a numerical technique for modelling wave propagation. It is an efficient technique that can be implemented not only in single CPU computers but also in parallel computers.

TLM mesh can be modelled as a digital filter. For a two dimensional TLM modelling, each node can be modelled as a four input- four output digital filter and the TLM mesh can be modelled as a connection between these digital filters.

Since sound waves are scalar waves, scalar three dimensional TLM can be used for modelling wave propagation.

Inverse TLM is TLM in time reversal. Inverse TLM could be used to find the origin of a wave propagation when the wave distribution in the medium is known.

TLM can be used to model ultrasound wave propagation as well as electromagnetic waves.

Chapter 5
Experiments with TLM

5.1 Introduction

In this chapter TLM is used to model some wave propagation phenomena. In most of cases a point source is placed in the medium and the shape of wave propagation is saved frame by frame. Some of these frames will be shown here. By using all of the frames movies were created showing how the waves propagate. Since it is only possible to show a few frames in this thesis animations has been included on a CD that comes with the thesis. This chapter deals with the use of TLM for modelling wave scattering from a small object, wave reflection from a large object, wave interference, array transducers and the Doppler effect. First the set up is explained, since the setup and the notation is the same for all of the modelling examples.

5.2 Experimental setup, input, output and notation

In this section modelling setup is explained, including how input or output is generated and the notation that is used.

5.2.1 The modelling setup

All of the modelling examples in this chapter are done by a TLM modelling library developed by the author in C++. The library is in pure C++ and can be compiled on any platform. Here, a PC computer with an AMD K5 processor running at 166 MHZ was used. The computer has 128 Mbyte of RAM. The library is fast enough and uses the ”Low memory” technique as explained in section 4.3.6.3 on page 69.

All of the modelling is of a 2 dimension TLM. The medium size is always normalized by Δl so when the medium size mentioned as 100×100, it means that the medium size is $100\Delta l \times 100\Delta l$. All of the address in the medium is relative to bottom left corner of the medium. For example point (50,100) in a medium with size 100×200 is exactly in the middle of the medium.

The modelling time is also normalised by ΔT. A modelling time of 100 means that the modelling time is $100\Delta T$.

5.2.2 Inputs

Since this is digital modelling, all of the signals is injected to the medium should be in digital form. The most important signal that is injected into the system is a sinusoidal signal. The peak amplitude of the sinusoidal signal is always 1 and its frequency expressed based on the number of samples per cycle. For example if the number of samples per cycle of the sinusoidal signal is 20, it will be expressed as "a sinusoidal signal with 20 samples per cycle or in the short form as a 20 samples sinusoidal". If the sinusoidal signal isn't continuous the duration of the signal will be expressed in terms of samples too. For example "a 20 sample sinusoidal signal with duration of 100" means that a sinusoidal signal that is sampled at 20 samples per cycle is injected into the system for the duration of 100 samples (There would be 5 cycles of the sinusoidal signal in the injected signal.).

5.2.3 Modelling output

The output of the modelling process will be presented in one of the followings way:

Time amplitude graph This graph shows the received signal at one of the medium points during modelling time.

Frame by frame Here the signal power at each point in the medium is saved at a specific time. These values are used to draw some 2 dimensions or 3 dimensions figures.

5.2.3.1 Time amplitude graph

As an example to this type of output, assume that a receiver is placed at the point (50,60) and a point source at the point (50,50) in a medium with size 100×100 as shown in fig[5.1].

A 20 samples sinusoidal signal with the duration of 100 samples is used as the source signal and the corresponding signal at the receiver is saved. The signal at the source place[1] and received signal are shown in fig[5.2]. The horizontal axis is time in ΔT and the vertical axis is amplitude.

[1]This is the signal as seen by a receiver at the source place and not the generated signal by the source.

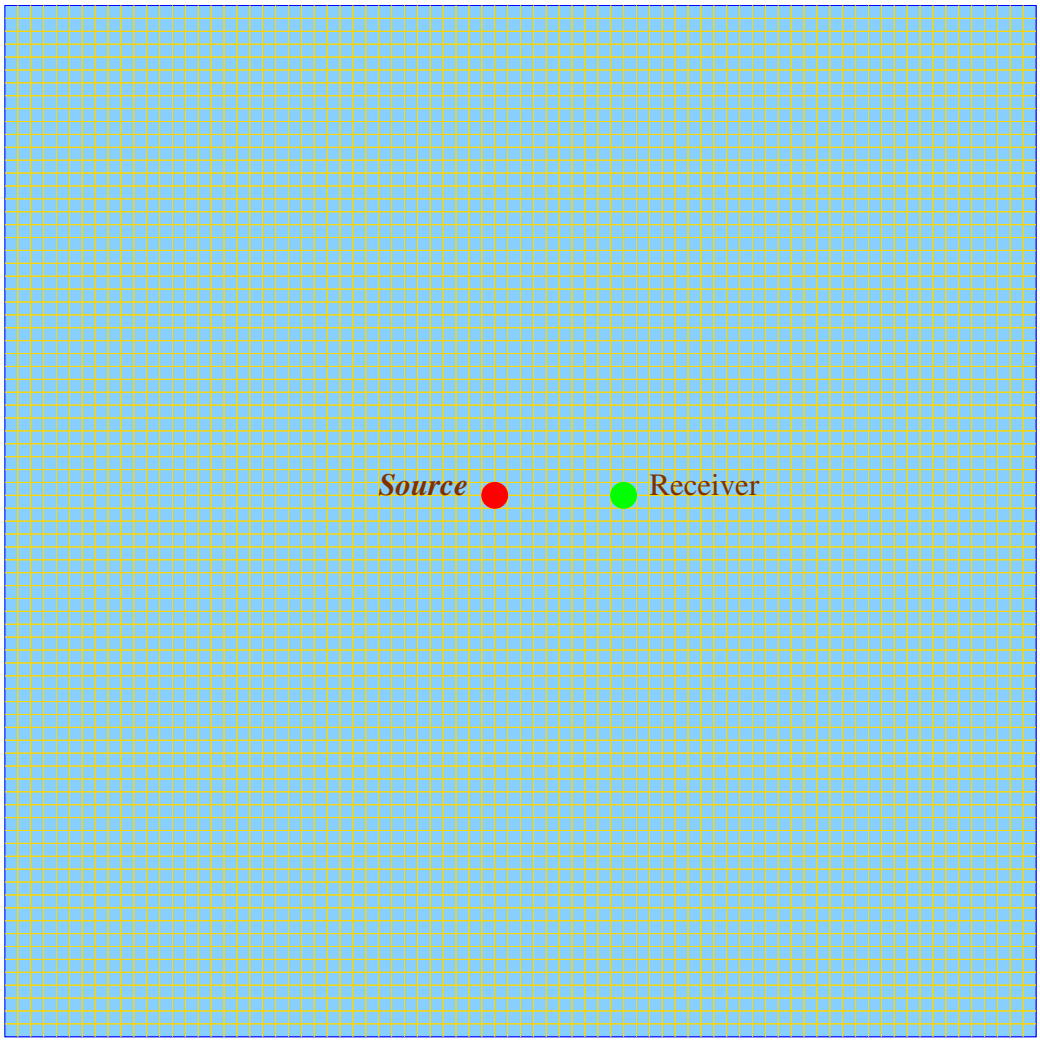


Figure 5.1: *Medium setup for modelling example.*

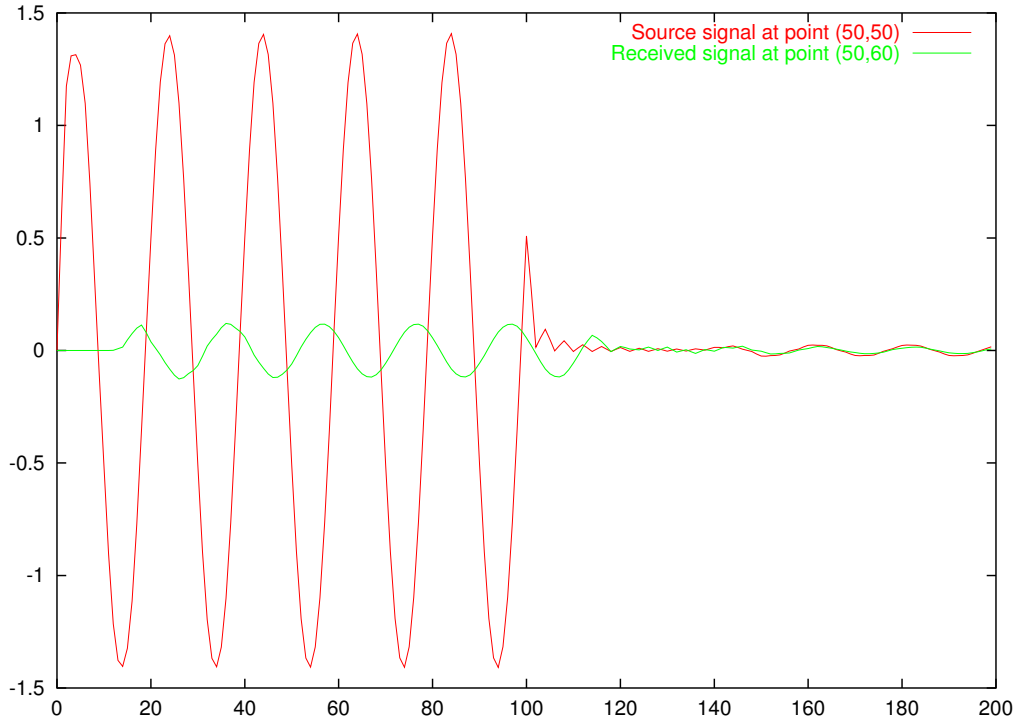


Figure 5.2: *Signal at the source point and received signal for modelling example.*

From this simple example the following properties of TLM modelling can be determined:

1. The time delay between the received signal and source signal is 14, which is equal to $10 \times \sqrt{2}$. So the speed in the medium is $\frac{C}{\sqrt{2}}$. Where C is the speed of wave in the real medium.

2. The received signal is much weaker than the source signal.

3. After the main signal, there is a very weak signal with a frequency equal to the natural frequency of the TLM which is a sinusoidal signal with four sample in a cycle.

5.2.3.2 Frame by frame

In this type of presentation, one image per time step is created. Since there are three variables: X,Y and Amplitude, data can be drawn in several ways. Two ways used here: Colour code and 3 Dimensions.

Colour code: In this type of presentation, the amplitude of signal at each point is converted

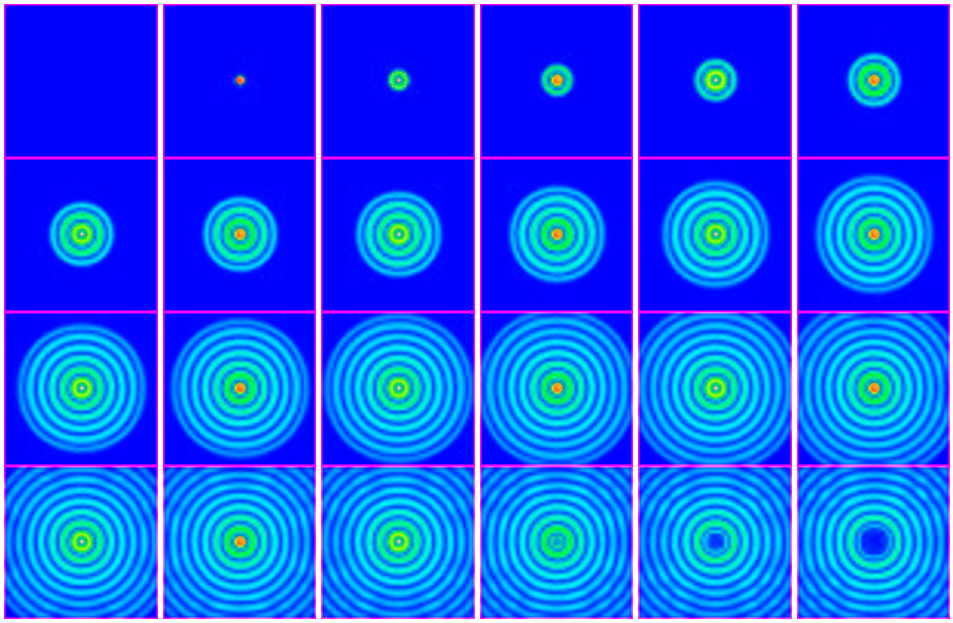


Figure 5.3: *wave propagation in a medium by using Colour code drawing (Time difference between each two pictures is 5 ΔT).*

to a colour and a 2 dimensional picture of the medium is created. It is important to understand how these pictures are created. The following steps were used:

1. At the end of each time step in the TLM modelling, the amplitude in each point is calculated by adding the 4 voltages incident on the node.
2. The absolute value for the amplitude of each node is used and converted to RGB (Red , Green , Blue) colour codes.
3. These values are used to create an image file in PPM format (see http://www.swin.edu.au/astronomy/pbourke/dataformats/ppm/)
4. ImageMagick (http://www.simplesystems.org/ImageMagick/) was used to convert them to EPS format and then adding them to this thesis.

In fig[5.3] a sample output is shown. The time difference between adjacent pictures is $5\Delta T$. This kind of representation is good when the propagation shape needs to be shown.

3 Dimensional: In this type of representation, amplitude is use on the Z axis to create a 3 dimensional image. This type of images is created by using the following steps:

1. At the end of each time step in TLM modelling, the amplitude in each point is calculated by adding the 4 voltages incident on the node.

2. For each frame, the node amplitudes are saved in a file.
3. GNUPLOT (http://www.gnuplot.org/) is used to draw a 3 dimension graph based on the each file (one graph for each frame).
4. The graphs are exported in EPS format for adding to this thesis.

This type of representation is good for representing wave amplitude at different points in the medium.

Movie: It is possible to create a movie based on a series of related frames. A movie was created showing how a wave propagates in the medium. For creating these movies, Mpeg2enc (http://www.mpeg.org/MPEG/MSSG/) is used to convert PPM images to MPEG movies. ImageMagick is used to convert exported images from GNUPLOT to PPM format. The movies are added to the CD that comes with this thesis. All of the movies are in MPEG format. Most of the MEPG players can play these moves. Windows media player is the best one for playing these movies in windows operating system. The movie that shows the wave propagation as represented in fig[5.3] is in file "$sample\backslash color.mpv$" and the movie that shows the wave propagation as represented in fig[5.4] is in file "$sample\backslash 3d.mpv$".

5.3 The speed of a wave in the mesh

As explained in sec 4.3.3 on page 52, When the sampling rate[2] is near 4 sample per cycle, the wave speeds in the axial and the diagonal directions aren't the same. This phenomena is illustrated in fig[5.5]. The sampling rate for the experiment in fig[5.5] is 5 samples per cycle[3].

When the sampling rate is much higher than 4 sample per cycle, the wave propagates in the diagonal and the axial directions with the same speed. This speed is $\frac{C}{\sqrt{2}}$, where C is the speed of wave in the real medium. For example if the transmitter is at point $(10, 10)$ and the receiver is at point $(10, 20)$ then it takes $10 \times \sqrt{2} \times \Delta T$ or about $14\Delta T$ for the wave to reach the to receiver from the transmitter. In fig[5.6], the wave propagation shape is shown for the case of 20 samples per cycle.

In fig[5.7], the received signal at the receiver is shown. Despite the fact that the distance between transmitter and receiver is $10\Delta\ell$ the signal is received at the receiver at time $10\sqrt{2}\Delta T$

[2]The terms "sampling rate" and "sample per cycle" will be used interchangeably.

[3]The related MPEG for this series is shown in $SampligRate\backslash Low\backslash color.mpv$ and $SampligRate\backslash Low\backslash 3d.mpv$ on the CD accompany the thesis.

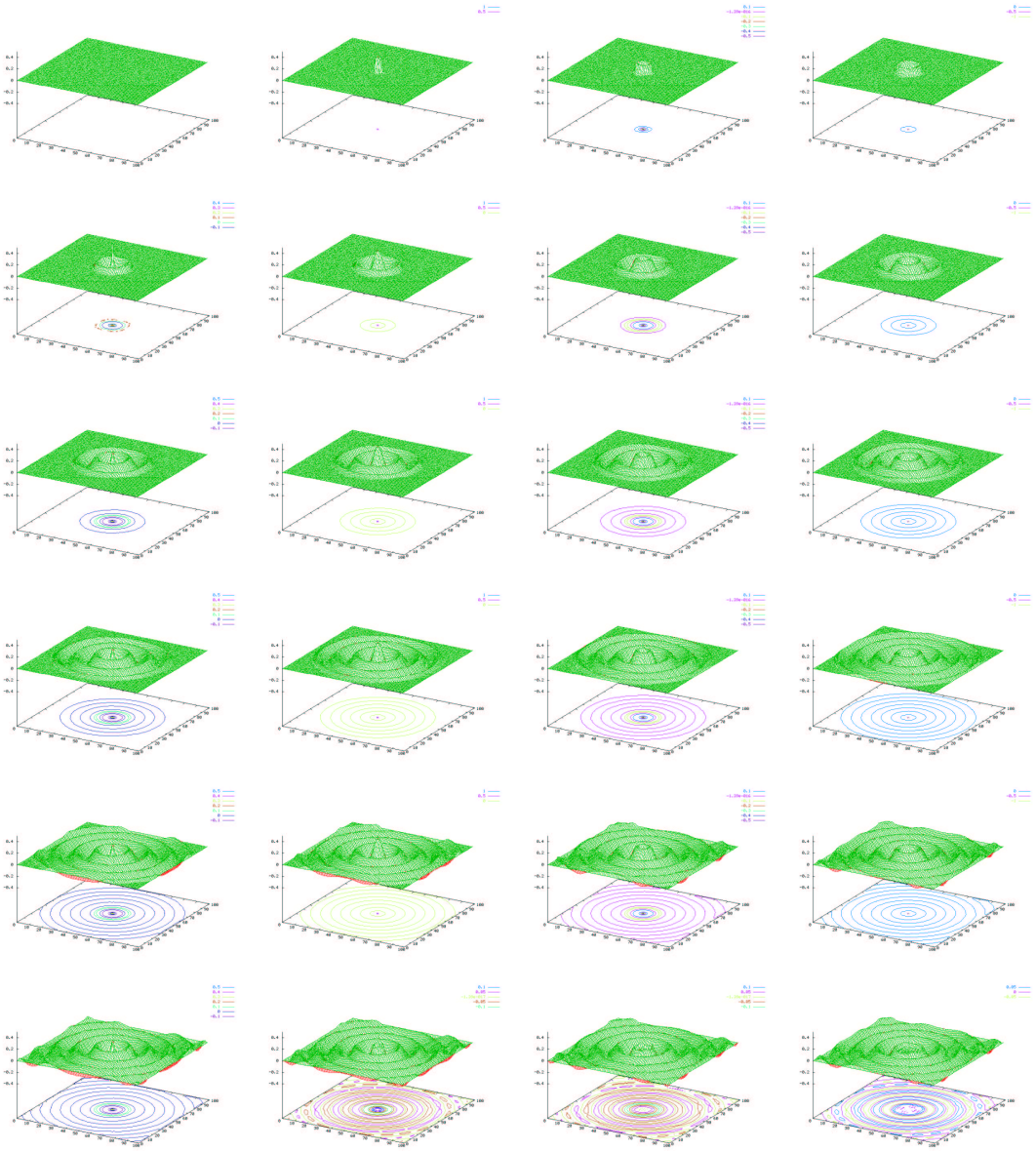


Figure 5.4: *wave propagation in a medium by plotting data in 3 dimensions (Time difference between each two picturess is 5 ΔT).*

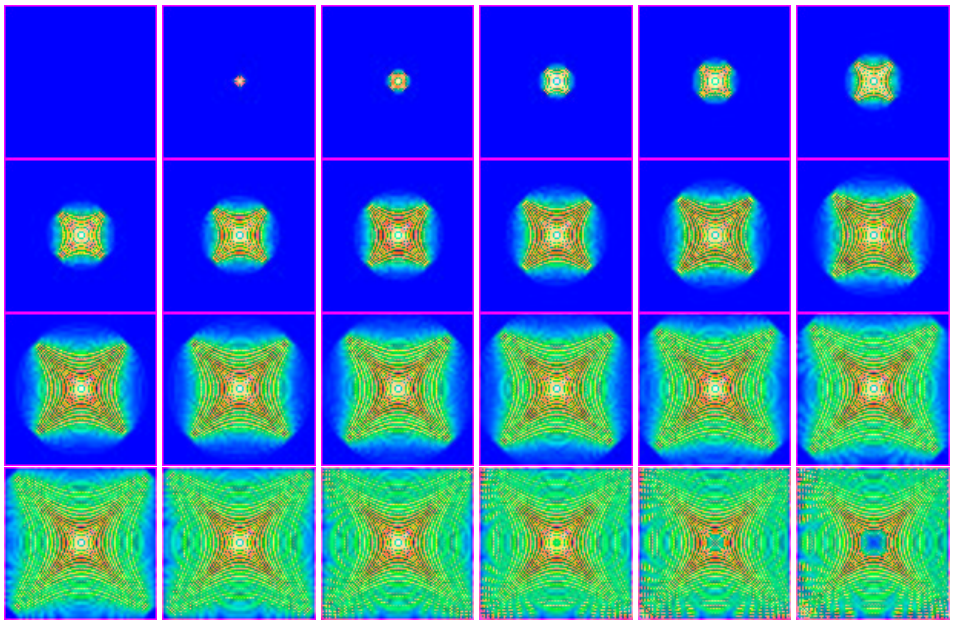


Figure 5.5: *When the sampling rate is near 4 sample per cycle, wave propagation isn't the same in different directions. The sampling rate here is 5 sample per cycle.(Time difference between each two pictures is 5 ΔT). The setup for this experiment are as follow:Medium size:*100×100 *Source Position:(50,50) Receiver position: (60,50)*

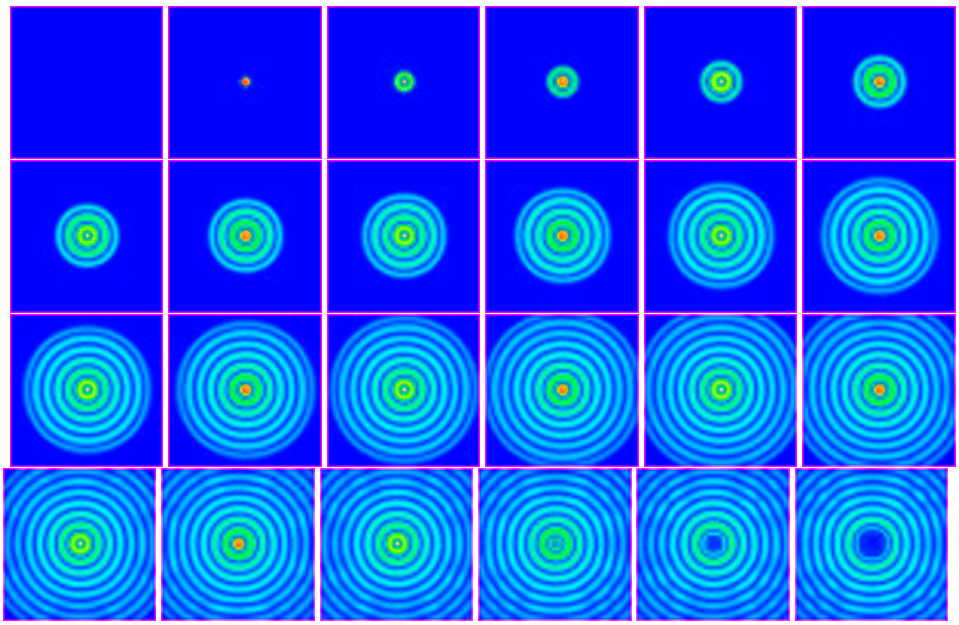


Figure 5.6: *When the sampling rate is much higher than 4 sample per cycle, waves propagate in diagonal and axial direction with the same speed. The sampling rate here is 20 sample per cycle.(Time difference between each two pictures is 5 ΔT). The setup for this experiment are as follow:Medium size:*100×100 *Source Position:(50,50) Receiver position: (60,50)*

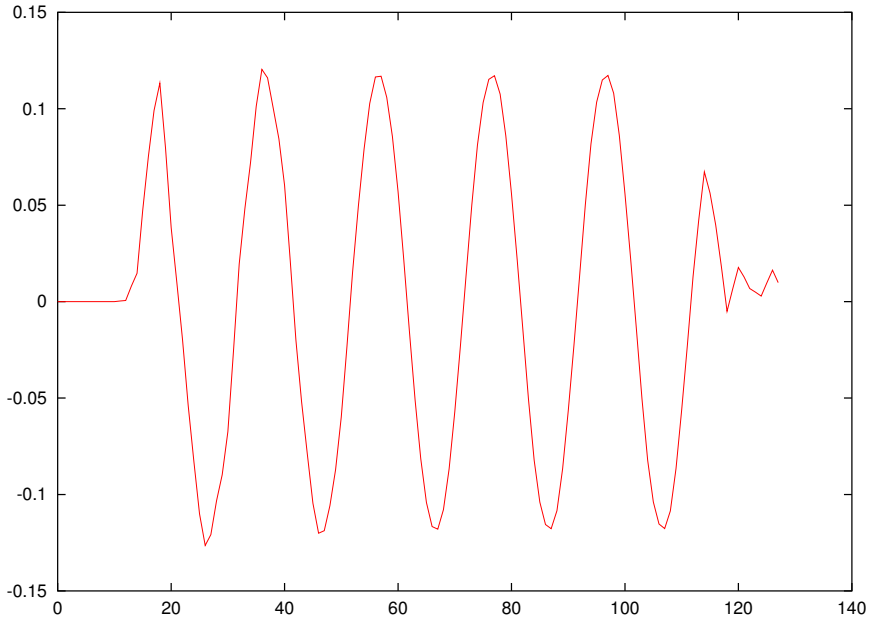


Figure 5.7: *The speed of the wave in the TLM mesh is* $\frac{1}{\sqrt{2}}C$. *C is the speed of wave in the real medium. The setup for this experiment are as follow:Medium size:*100×100 *Source Position:(50,50) Receiver position: (60,50)*

which is $14\Delta T$. This is because the speed of the wave in the TLM mesh is $\frac{C}{\sqrt{2}}$ when C is the speed of wave in the real medium.

5.4 Boundary effect

The TLM is a numerical model that is mostly modelled on computers. Since the amount of RAM and CPU power of a computer is limited, it is not possible to model infinite media. Limited media have boundaries and the medium at these boundaries should be terminated. If this termination is not correct, the wave is reflected back from the boundary and interferes with the original wave[4]. This effect shown in fig[5.8] [5].

Some ways to terminate the medium has been proposed [84], [85], [86], [87]. Most of these models are based on Maxwell's equations. The author has developed a new digital filter (See section 4.3.5 on page 62) model of the TLM to help find a new way for terminating the TLM

[4]This is similar to a wire that should be terminated with matching impedance otherwise the wave in it will be reflected back.

[5]The related MPEG for this series shown in $boundary\backslash Reflecting\backslash color.mpv$ and $boundary\backslash Reflecting\backslash 3d.mpv$.

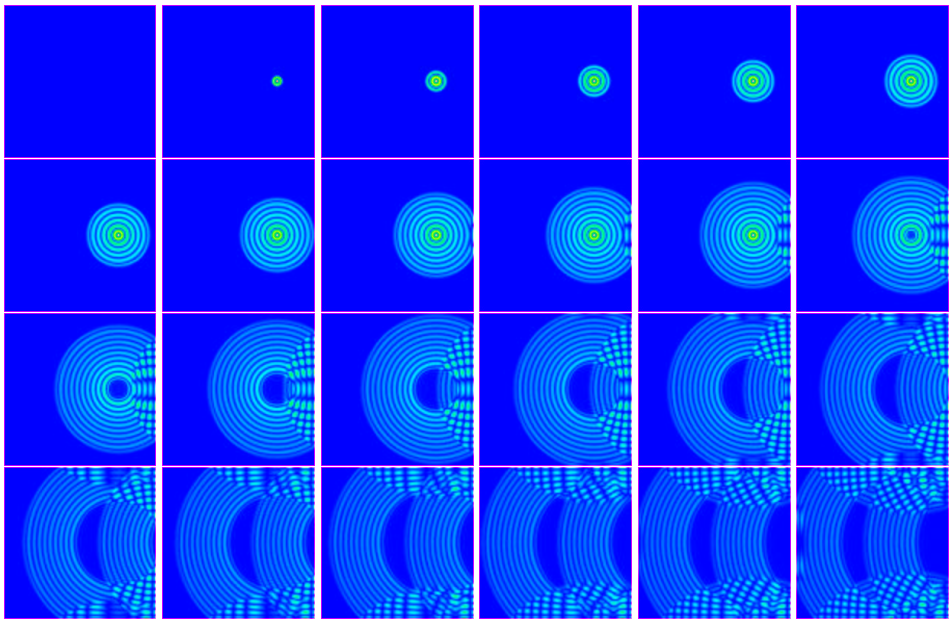


Figure 5.8: *Wave reflection from the boundary.(Time difference between each two pictures is 10 ΔT).The setup for this experiment are as follow: Medium size:*200×200 *Source Position:(150,100) Receiver position:(160,100).*

boundary. As shown in fig[5.9] the terminating medium can be modelled as a Multi-input Multi-output IIR filter.

The output of the filter at each node is related to the inputs at all nodes at different times. This fact is written in the following equation:

$$\begin{aligned} O^{T}_{(x,y)} = & \, a_1 I^{T-1}_{(x,y)} + b_1 I^{T-1}_{(x,y+1)} + c_1 I^{T-1}_{(x,y-1)} + ... \\ & + a_2 I^{T-2}_{(x,y)} + b_2 I^{T-2}_{(x,y+1)} + c_2 I^{T-2}_{(x,y-1)} + ... \\ & + a_3 I^{T-3}_{(x,y)} + b_3 I^{T-3}_{(x,y+1)} + c_3 I^{T-3}_{(x,y-1)} + ... \end{aligned} \tag{5.1}$$

Where $O^{T}_{(x,y)}$ is the output from the node at the point (x, y) at time T and $I^{T-i}_{(x,y)}$ is the input to the node at pint (x, y) at time $T - i$, $i \geq 1$. a_i , b_i and c_i are real or complex[6] weights.

By reviewing the filter model of the TLM we will eventually find that:

[6]For a homogenous medium the weights are real, and for a non-homogenous medium they are complex.

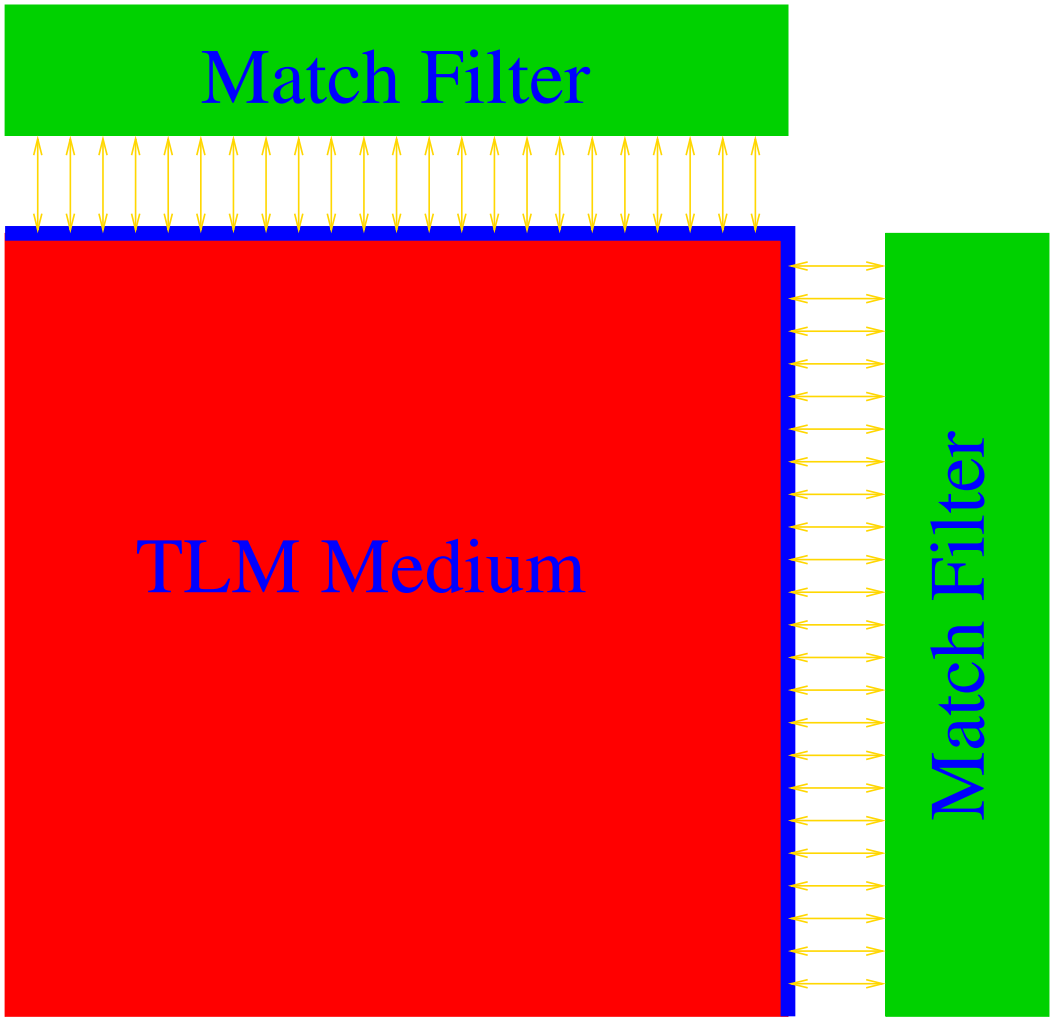


Figure 5.9: *To correctly terminating a medium we can terminate it to a matching Multi-input Multi-output IIR filter*

$$
\begin{aligned}
&a_1 < b_1 < c_1 \quad ... \\
&a_2 < b_2 < c_2 \quad ... \\
&a_3 < b_3 < c_3 \quad ... \\
&\qquad ...
\end{aligned}
\tag{5.2}
$$

and:

$$
\begin{aligned}
&a_1 < a_2 < a_3 \quad ... \\
&b_1 < b_2 < b_3 \quad ... \\
&c_1 < c_2 < c_3 \quad ... \\
&\qquad ...
\end{aligned}
\tag{5.3}
$$

Based on these equations, the first approximation for the IIR filter is when we assume that all coefficients other than a_1 are zero. In this case the IIR filter reduce to a very simple FIR filter with the transfer function of:

$$O^T_{(x,y)} = a_1 I^{T-1}_{(x,y)} \tag{5.4}$$

The value of $a_1 = 0.5$ can be found by inspecting the digital filter model and its transfer function. The result of using this method of removing boundary effect is shown in fig[5.10] [7].

A better filter could be built by approximating the equation 5.1 as shown in the following

[7] The MPEGS for these experiments can be found in:$boundary\backslash Remove1\backslash color.mpv$ and $boundary\backslash Remove1\backslash 3d.mpv$

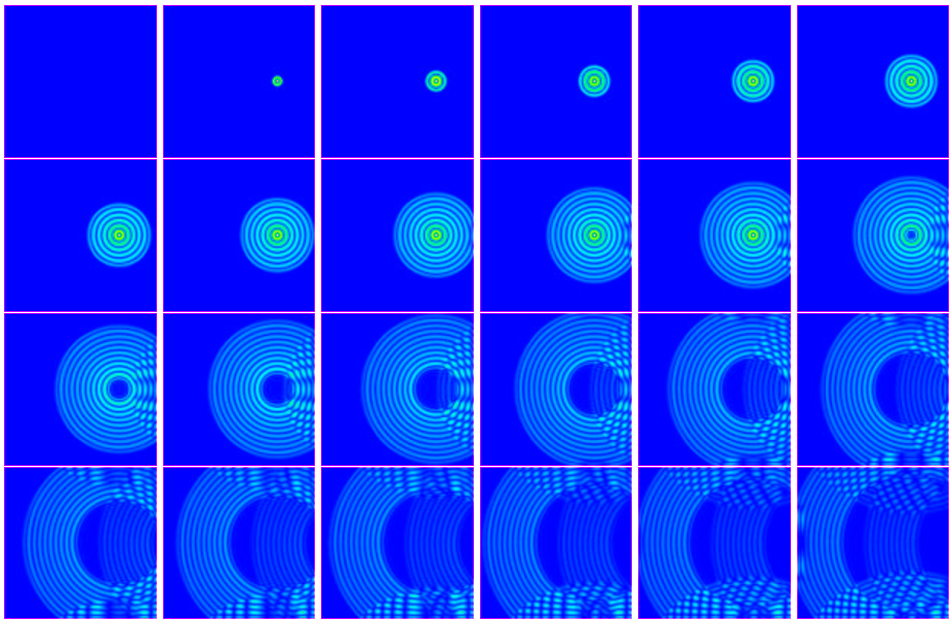


Figure 5.10: *Using the filter approximation as shown in equation 5.4 for removing reflection from boundary. (Time difference between each two pictures is 10 ΔT). The setup for this experiment are as follow: Medium size:*200×200 *Source Position:(150,100) Receiver position:(160,100).*

equation:

$$O^{T}_{(x,y)} = a_1 I^{T-1}_{(x,y)} + b_1 I^{T-1}_{(x,y+1)} + c_1 I^{T-1}_{(x,y-1)} + a_2 I^{T-2}_{(x,y)} \tag{5.5}$$

By assuming that[8]:

$$\begin{aligned} I^{T-1}_{(x,y)} &\simeq I^{T-1}_{(x,y+1)} \\ I^{T-1}_{(x,y)} &\simeq I^{T-1}_{(x,y-1)} \\ I^{T-1}_{(x,y)} &\simeq I^{T-2}_{(x,y)} \end{aligned} \tag{5.6}$$

and finding:

$$\begin{aligned} a_1 &= 0.5 \\ b_1 &= 0.25 \\ c_1 &= 0.25 \\ a_2 &= -0.75 \end{aligned}$$

[8]This approximations could help to save memory when implementing TLM

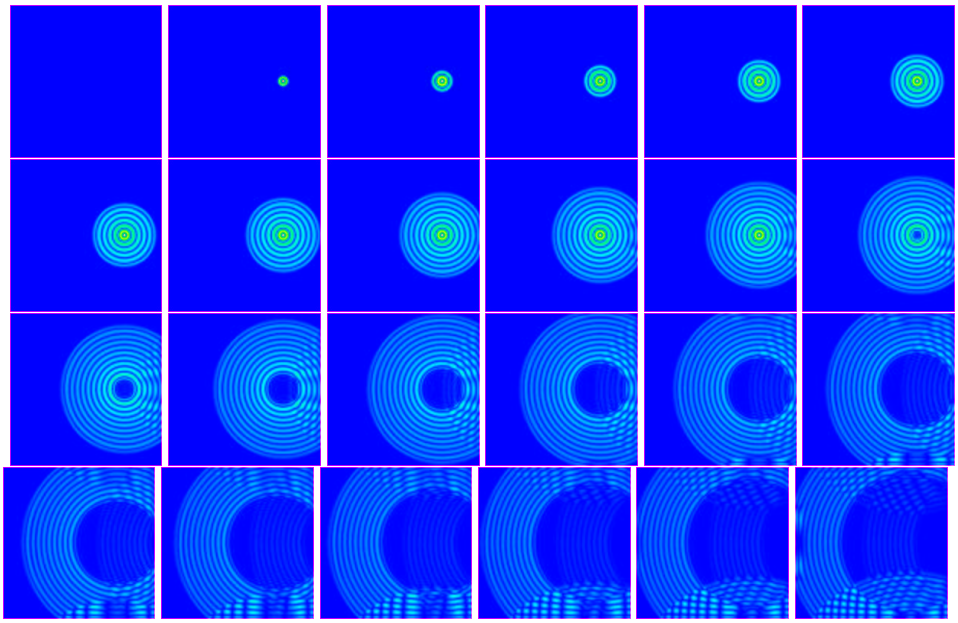


Figure 5.11: *Using the filter approximation as shown in equation 5.7 for removing reflection from boundary. (Time difference between each two pictures is 10 ΔT). The setup for this experiment are as follow: Medium size:*200 $\times$ 200 *Source Position:(150,100) Receiver position:(160,100).*

the filter approximation reduced to:

$$O^{T}_{(x,y)} = 0.25 \times I^{T-1}_{(x,y)} \tag{5.7}$$

The result of using this method of removing the boundary effect is shown in fig[5.11] [9] .

For comparison of these method, fig[5.12] is drawn. From this fig, it is clear that the filter approximation is working and is reducing the energy in the reflected signal. A better result could be obtained by doing a better approximation. This approximation was considered sufficiently good for the work reported in this thesis.

5.5 Wave interference

When two waves travel in a medium they interfere with each other [44], [88]. As the result of this interference, node and anti-nodes are created (destructive and constructive interference). Nodes are created at places where these two waves subtract from each other. If these two waves

[9] The MPEGS for these experiments can be found in: $boundary\backslash Remove2\backslash color.mpv$ and $boundary\backslash Remove2\backslash 3d.mpv$

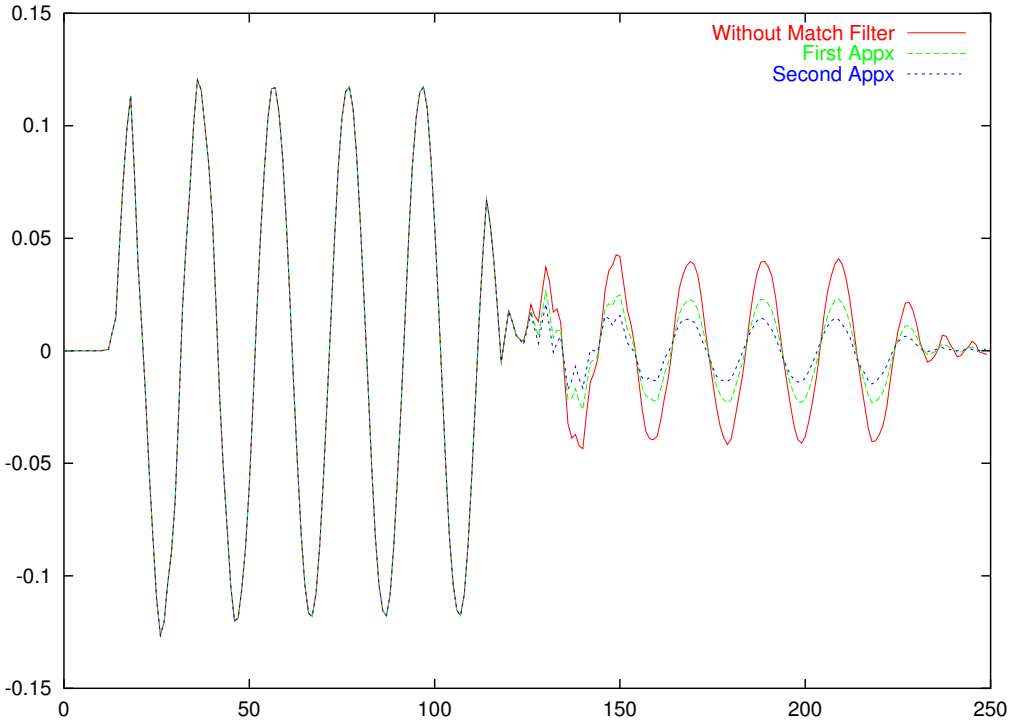


Figure 5.12: *Here all signals are shown in one graph. The signal between* $14\Delta T$ *and* $114\Delta T$ *is the original signal that comes directly from the transmitter. The reflected signal from the boundary arrives at the receiver at time* $127\Delta T$*. The setup for this experiment are as follow: Medium size:*200×200 *Source Position:(150,100) Receiver position:(160,100).*

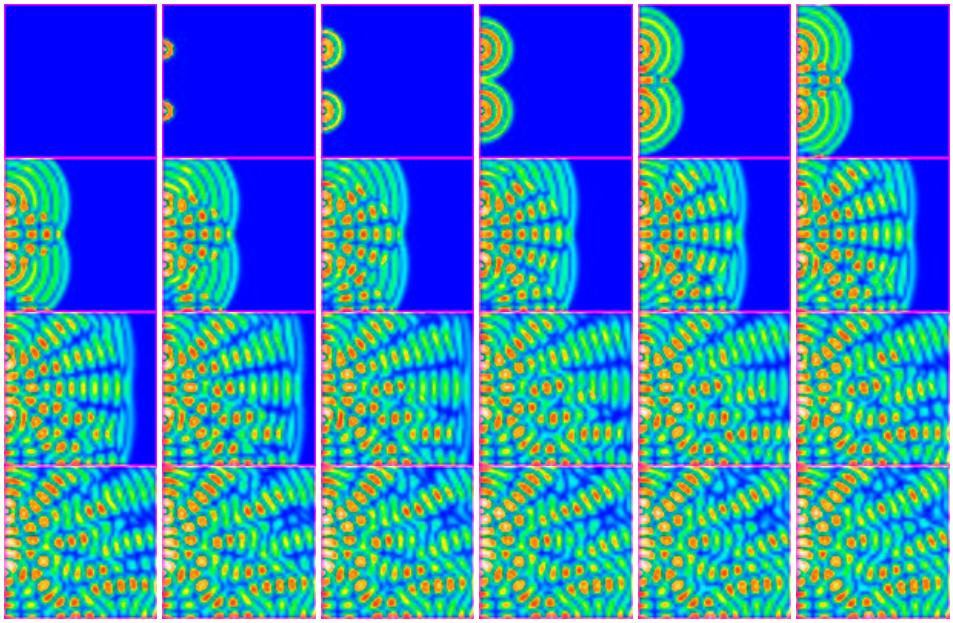


Figure 5.13: *When there are two waves in a medium, they interfere with each other. In this modelling example, the frequency and amplitude of the two wave are the same. Time difference between each two pictures is* $10\Delta T$*. The setup for this experiment are as follow: Medium size:*100×100 *Source 1 Position:(30,1) Source 2 Position:(70,1).*

have the same amplitude then there isn't any energy at the nodes. Anti-nodes are created at places where these two waves are added to each other. The position of nodes and anti-nodes can be calculated by measuring the distance between the point in and the the two source. If the distance is such that the two waves reach the point in phase then that point is a anti-node. If the distance is such that the wave is reaching to the point with 180 degree difference in phase than the point is a node point. This phenomena has been modelled by TLM and is shown in fig[5.13] [10].

5.6 Waves and objects

When there is an object in the medium, it will change the way that wave propagates its vicinity. If the size of object is much bigger than the wave length then reflection takes place. In reflection the wave reflects back from the object and hence we have a new wave in front of the object and a shadow behind the object.

[10]The related MPEG for this series shown in $WaveInterference\backslash color.mpv$ and $WaveInterference\backslash 3d.mpv$.

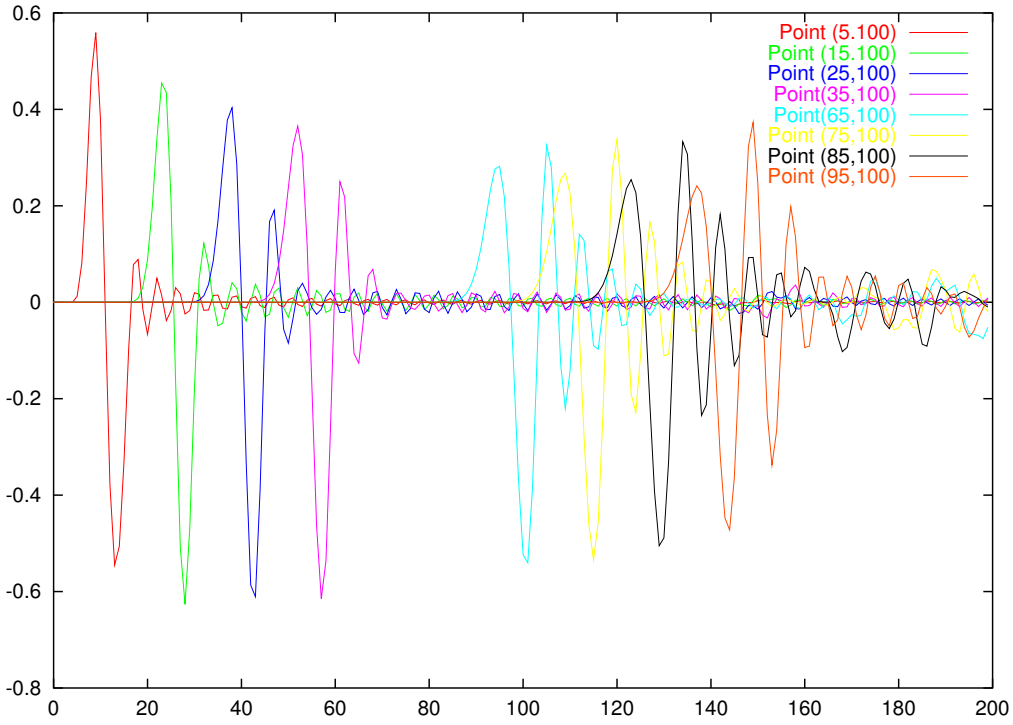


Figure 5.14: *The received signals at receivers when there isn't any object in the medium.*

If the object size as seen by the wave is much smaller than the wave length, then there is scattering. In scattering, objects act as point sources and transmits waves in the medium. In scattering, there isn't any shadow at the back of the object but there is a new wave in the medium which has been made by the object acting as a point source.

For experiments in this section of the thesis a medium with the size of (100,200) is used. There are 196 point sources to create a plain wave and several receivers at the points (5,100), (15,100), (25,100), (35,100), (65,100), (75,100), (85,100), (95,100). The received signal at these points is shown in fig[5.14] when there isn't any object in the medium. These signals should be used as references for comparing with the result when there is an object in the medium. The wave length of the signal is $10\Delta\ell$ and the duration is $10\Delta T$. Thus there is one period of the signal in the medium (see fig [5.14] for the received signal at point (5,100) which is very similar to the generated signal).

5.6.1 Modelling wave reflection

When waves propagate in a medium, containing an object whose size is much bigger than the wave length of signal in the medium then reflection takes place. In this case the waves are reflected back by the object and there is a shadow behind the object. TLM is used to model wave reflection here and the result is shown in fig[5.15][11].

The signals at the receivers are shown in fig[5.16]. Since the signals at the receivers located behind the object are very weak, this experiment shows that there is a shadow behind the object (compare with the received signals shown in fig[5.14]).

To show the reflected signal, the signal at the receiver at point (15,100) is draw against the received signal from the same receiver when there is not an object in the medium (fig[5.17]). The received signals at this point in these two experiments are the same except for the reflected signal.

5.6.2 Modelling wave scattering

When a wave propagates in a medium, if there is an object whose size is smaller than the wave length of signal in the medium then scattering takes place. In this case the incident wave is scattered back by the object and the object act as a point source. The actual wave shape depends on the object shape and its size relative to the wave length. In this modelling the object is assumed to be very small compared with the wave length and hence the object acts as a point source and will generate waves in all directions. The results shown in fig[5.18] [12].

The signals at the receivers shown in fig[5.19].

To show the scattered signal from the object, the received signal at point (25,100) is drawn for two case: when there isn't any object in the medium and when there is a small object in the medium. As seen from fig[5.20], there is a small signal around time $100\Delta T$. This is the scattered signal from the object.

[11] The related MPEG for this series shown in $WaveObject\backslash Reflection\backslash color.mpv$ and $WaveObject\backslash Reflection\backslash 3d.mpv$.

[12] The related MPEG for this series shown in $WaveObject\backslash Scattering\backslash color.mpv$ and $WaveObject\backslash Scattering\backslash 3d.mpv$.

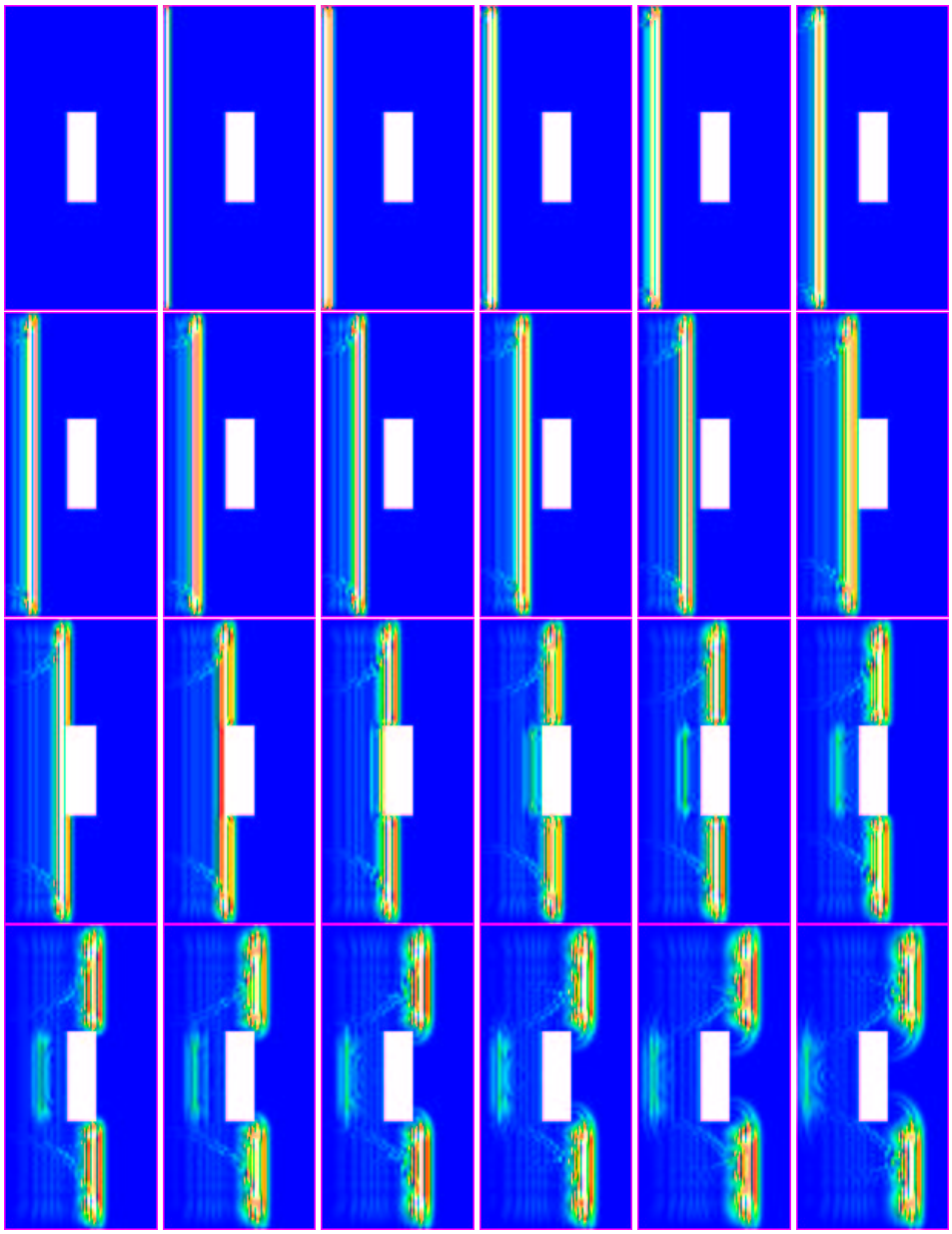


Figure 5.15: *When the size of an object in the medium is bigger than the wavelength, the wave is reflected by the object. There is a shadow at the back of the object and a reflected wave in front of the object. Time difference between each two pictures is* $5\Delta T$*. The object is an rectangular object from (40,70) to (60, 130). The object size as seem by the wave is* $60\Delta T$*.*

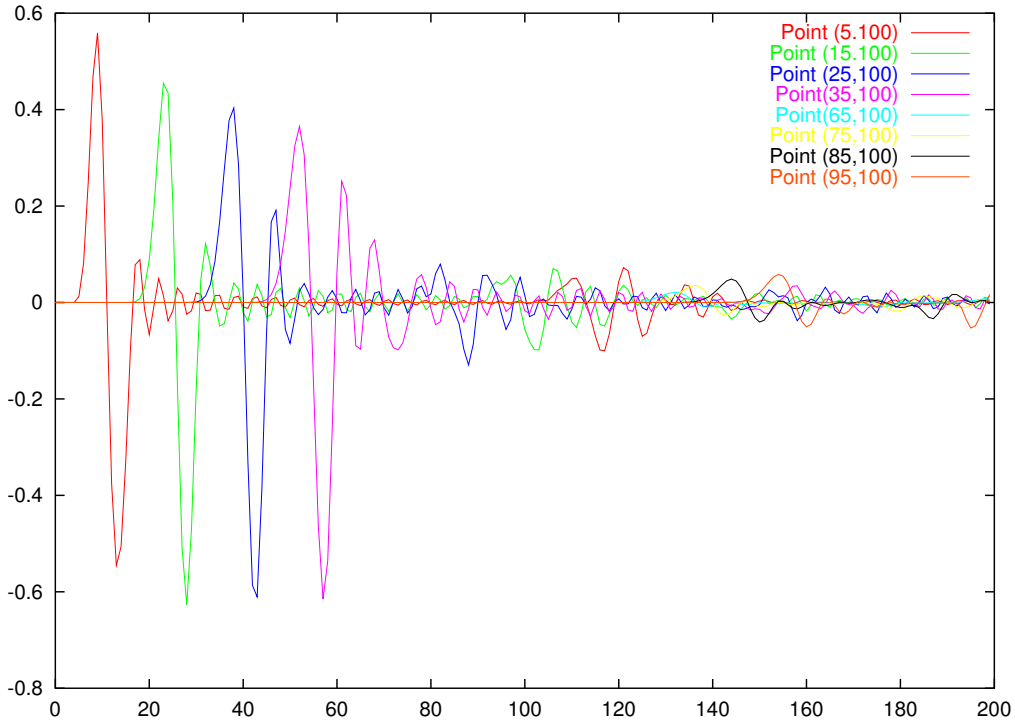


Figure 5.16: *The signals at the receivers when the size of the object is bigger than the wave length. Since There is a shadow at the back of the object, the signals at the receivers behind the object are very weak. The object is an rectangular object from (40,70) to (60, 130). The object size as seem by the wave is* $60\Delta T$

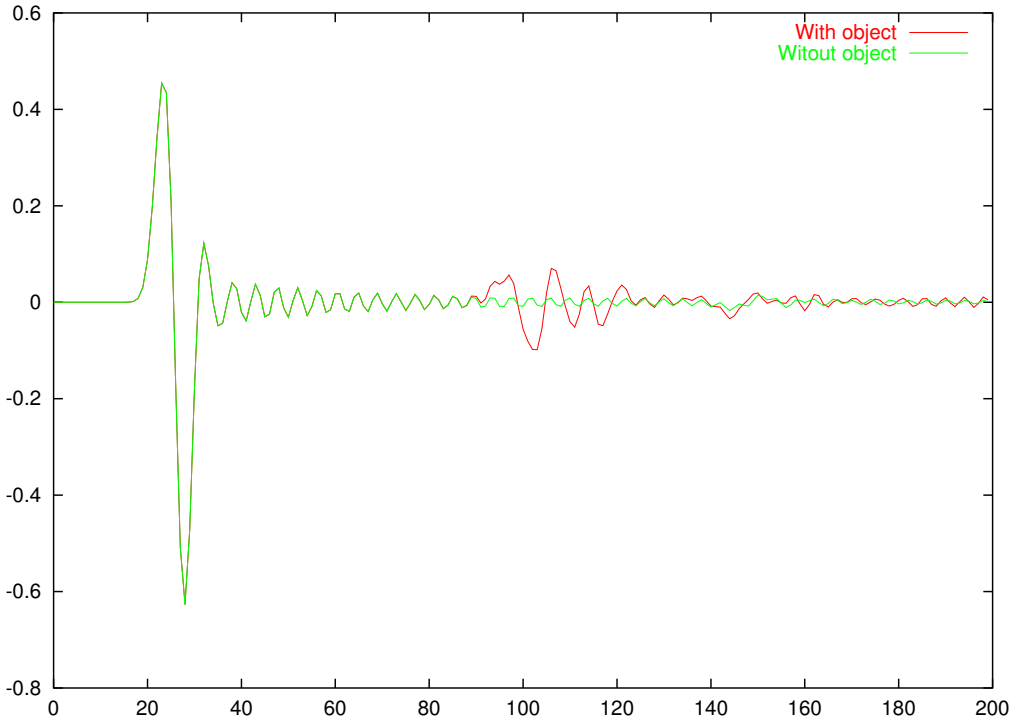


Figure 5.17: *The received signals at point (15,100). The object is an rectangular object from (40,70) to (60, 130). The object size as seem by the wave is* $60\Delta T$

5.6.3 Mixed signal and objects

If there are two or more signals in the medium and one or more object, then there are some interesting effects. Since each signal may react differently with the object, some kind of filtering may occur. The following subsections demonstrate this behaviour.

5.6.3.1 Two added signal

In this experiment the generated wave is the summation of two signals. One signal is high frequency and the other is low frequency. There is an object in the medium. The size of this object is selected in such a way that it is scattering one of the signals and reflecting the other one. The generated signal is shown in fig[5.21].

The wave length of the high frequency signal is $10\Delta\ell$ and the wave length of the low frequency signal is $100\Delta\ell$. The object is a rectangular object from (40,80) to (60,120). The medium size is 200×200. The received signal at point (65,100) is shown in fig[5.22].

As can be seen from fig[5.22], for a receiver placed behind the object, the object filters out the

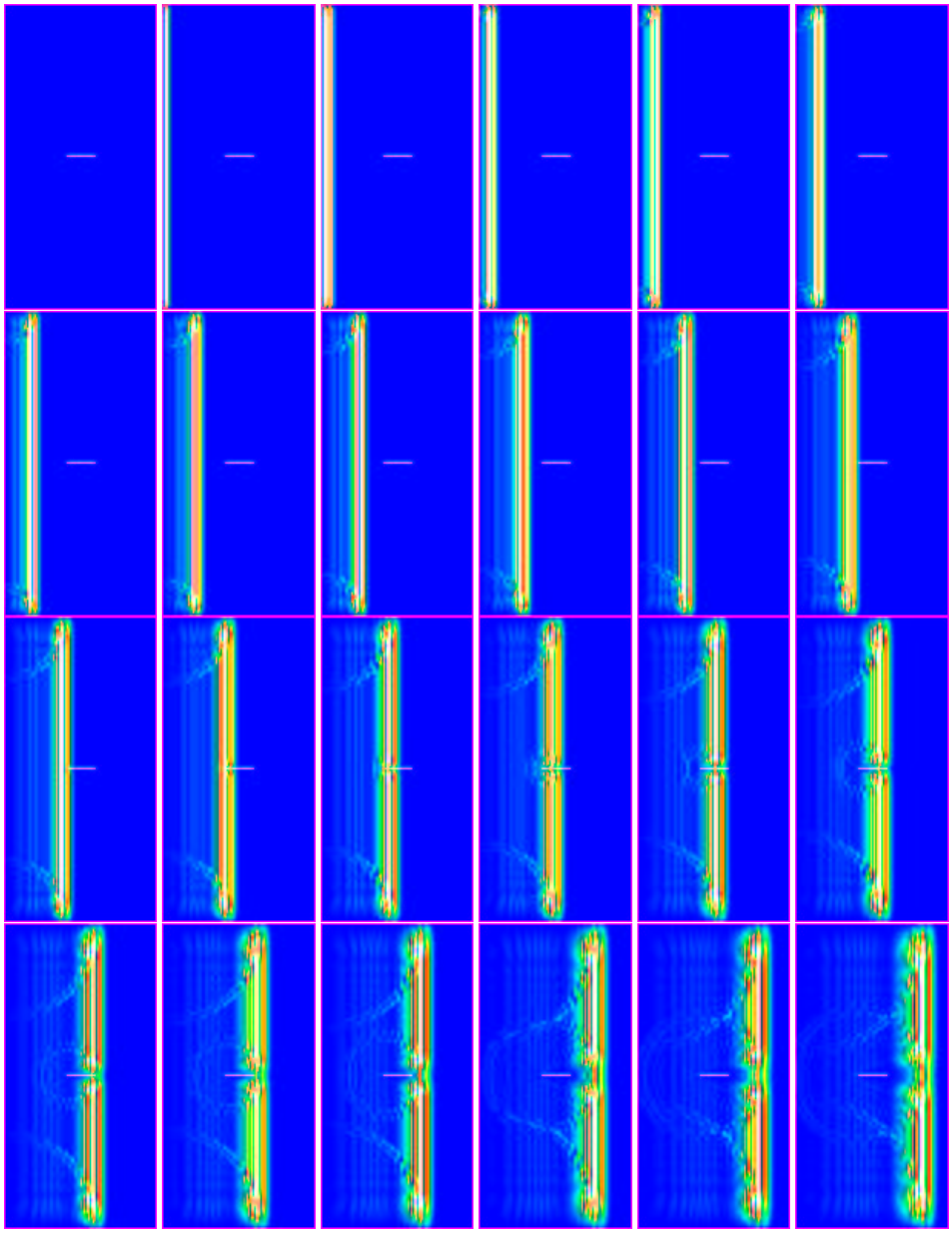


Figure 5.18: *When the size of an object in the medium is smaller than the wavelength, waves are scattered by the object. If the object size is very small compared to the wave length then the object acts as a point source and will generate waves in all directions. The time difference between each two pictures is* $5\Delta T$*. The object is an rectangular object from (40,99) to (60, 101). The object size as seem by the wave is* $2\Delta\ell$

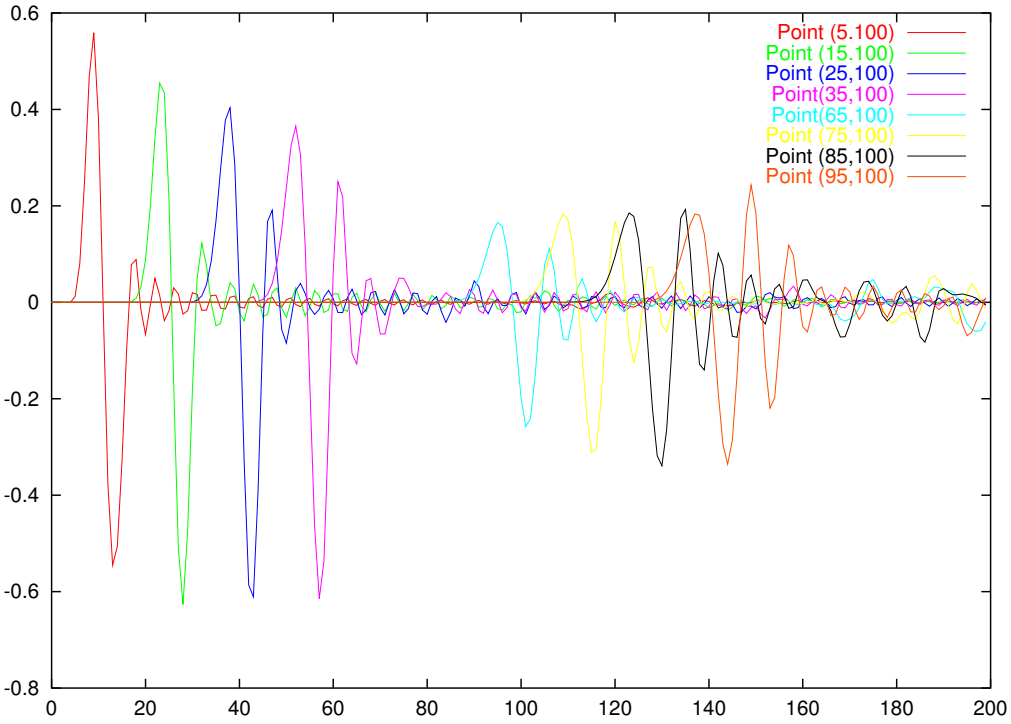


Figure 5.19: *The signals at receivers when the object in the medium is small compared with the wave length. Since there isn't any shadow, the received signals on all of the objects are strong (Compared with the reflection case). It should be noted that since the scattered signal from the object is very weak, it can't be seen in this figure. The object is an rectangular object from (40,99) to (60, 101). The object size as seem by the wave is* $2\Delta\ell$

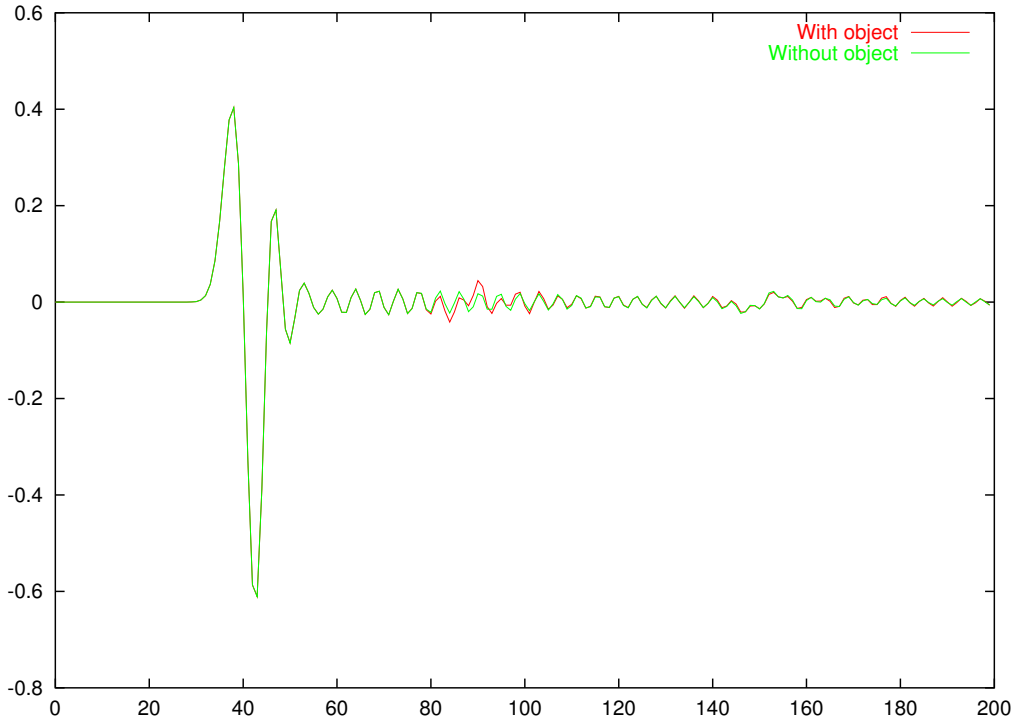


Figure 5.20: *The received signals at point (25,100) when there is an object in the medium and when there isn't any object in the medium. The object is an rectangular object from (40,99) to (60, 101). The object size as seem by the wave is* $2\Delta\ell$

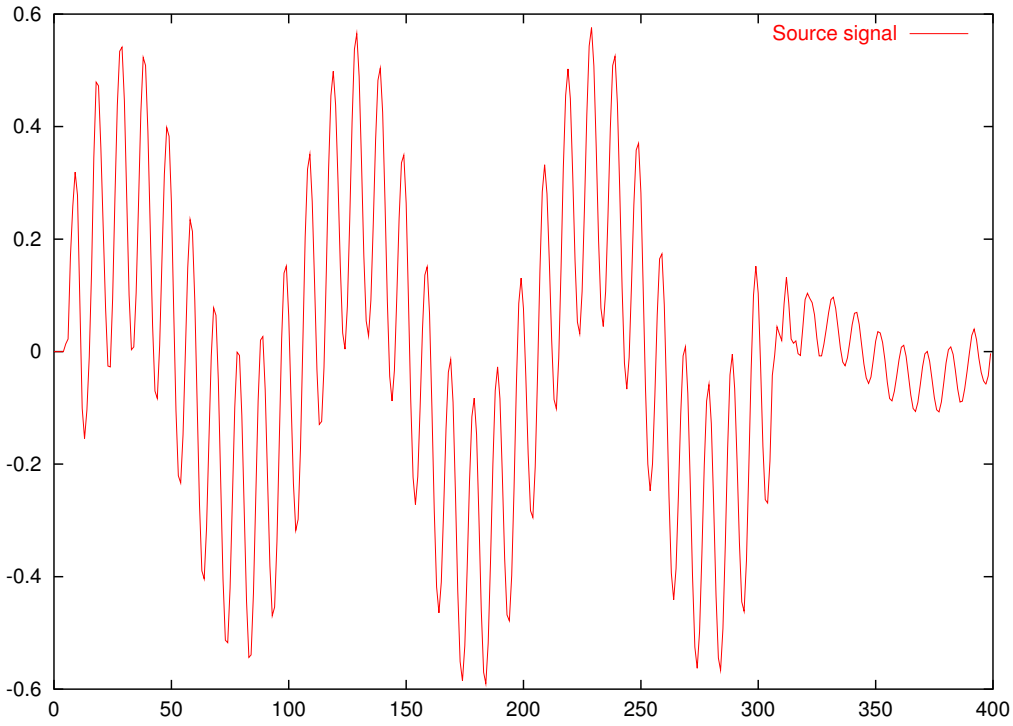


Figure 5.21: *The source signal for added signal experiment .*

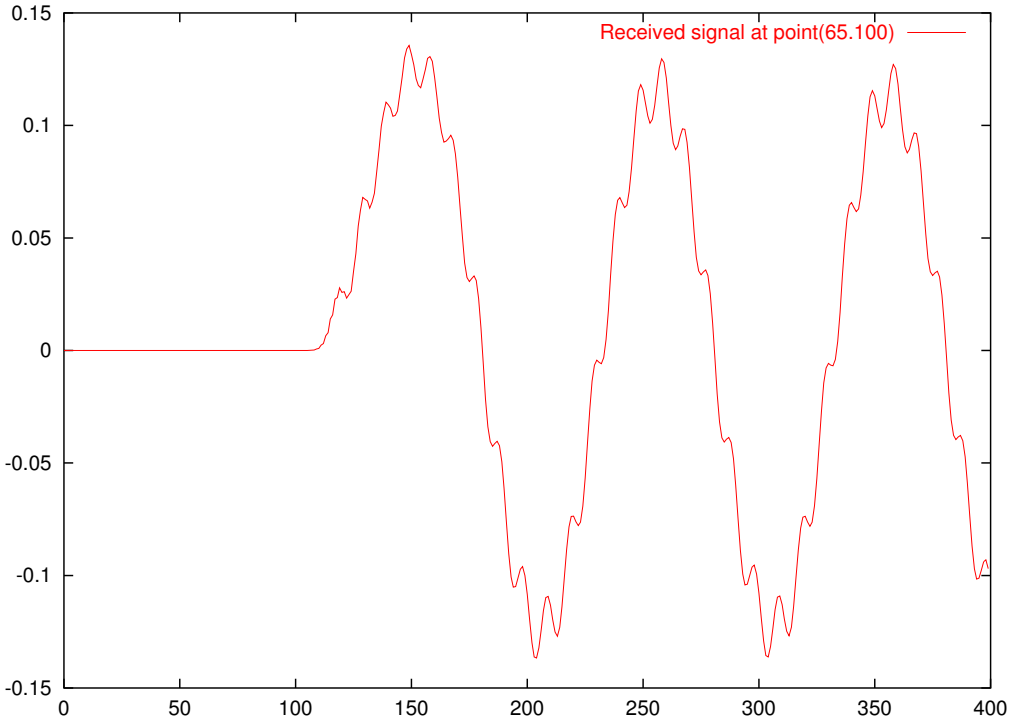


Figure 5.22: *The received signal at point (65,100) when the two signal added to each other.*

high frequency signal as it acts as a reflector for this signal[13]. The wave propagation shape is shown in fig[5.23] [14]..

5.6.3.2 Amplitude modulation signals

In this experiment a wave is generated that is amplitude modulation [89] of two signals. One signal is high frequency (The modulated signal) and the other is low frequency (The modulating signal). There is an object in the medium. The size of this object is selected in such a way that it scatters one of the signals and reflects the other one. The generated signal is shown in fig[5.24].

The wave length of the high frequency signal is $10\Delta\ell$ and the wave length of the low frequency signal is $100\Delta\ell$. The object is a rectangular object from (40,80) to (60,120). The medium size is 200×200. The received signal at point (65,100) is shown in fig[5.25].

As can be seen from fig[5.25], the object doesn't affect in any way the low frequency or high

[13]This is similar to the fact that when there is a load music playing, peoples in other rooms can hear only to the low frequency section of the music.

[14]The related MPEG for this series shown in $WaveObject\backslash Added\backslash color.mpv$.

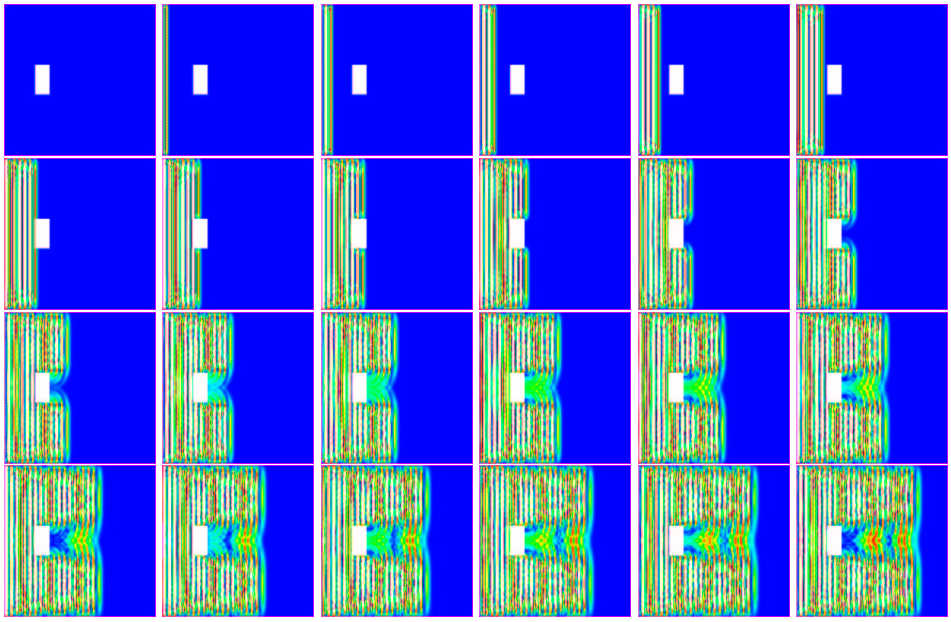


Figure 5.23: *The wave propagation shape when the two input signals added to each other.*

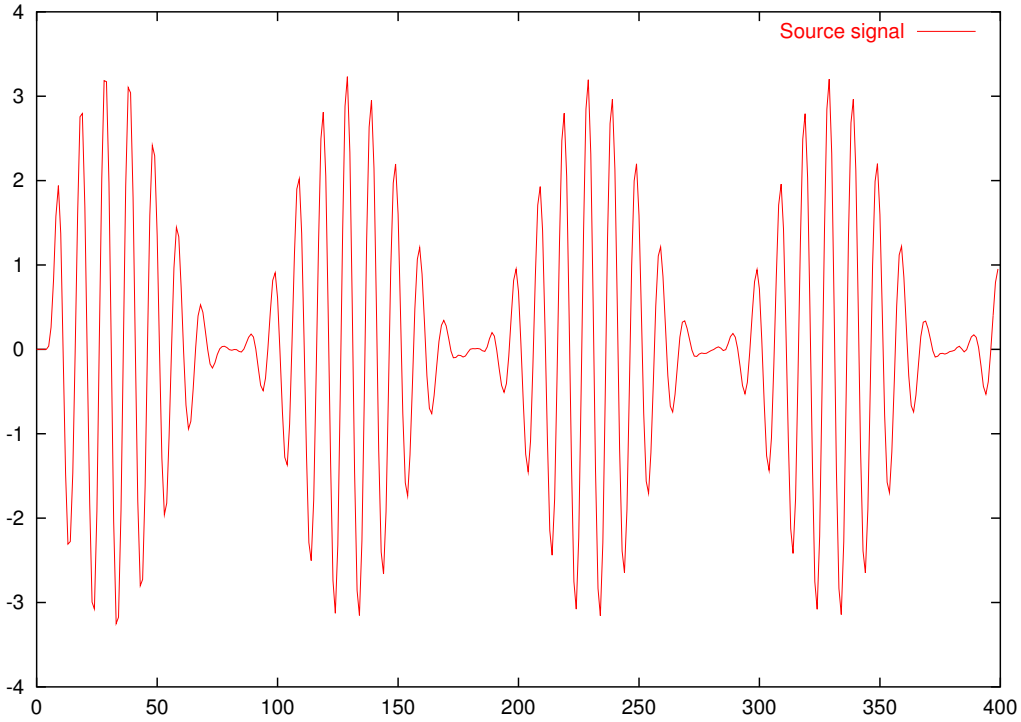


Figure 5.24: *The source signal for the amplitude modulation experiment .*

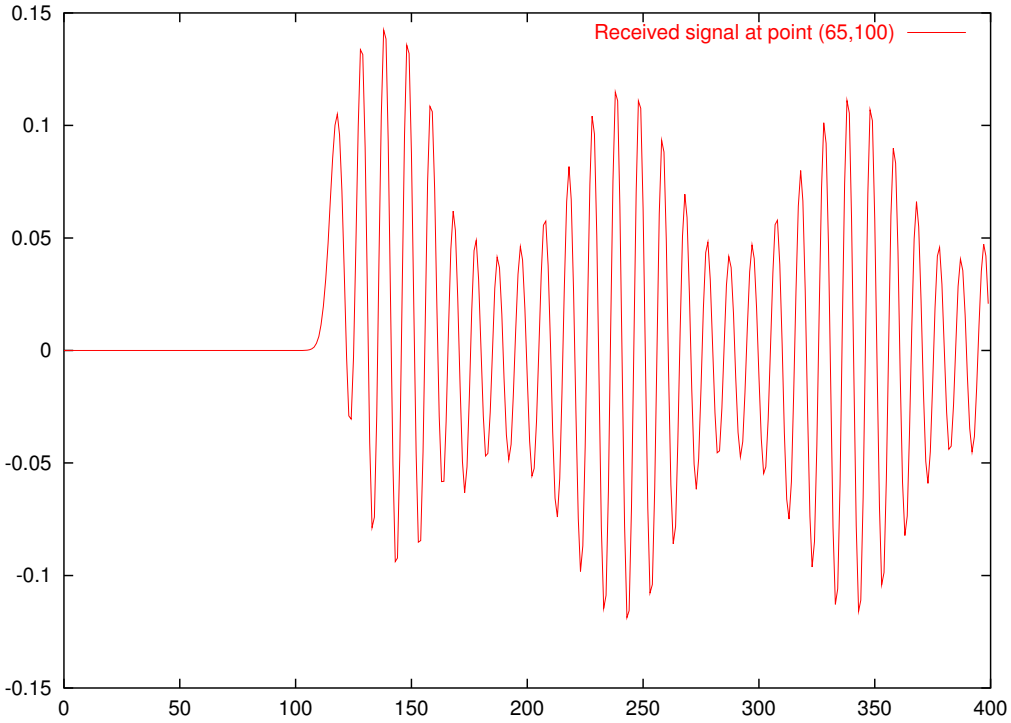


Figure 5.25: *The received signal at point (65,100) when there is an amplitude modulated signal in the medium.*

frequency signal[15][16]. This is because there is no low frequency signal component, only a carrier (high frequency) and two sidebands close to the carrier. The wave propagation shape is shown in fig[5.26] [17]..

5.6.4 Curved objects

In this section an object with a curvy surface is modelled. There are two scenarios: A focusing mirror and a circular mirror.

5.6.4.1 Focusing mirror

In this example a focusing mirror is modelled. In a focusing mirror, the plane waves will be focused on a point which is the focal point for the mirror [40]. This modelling is shown in

[15]We are lucky since if the object filters or alters the signal in any way then we couldn't have AM radios!
[16]The change in modulation index is the result of TLM medium that acted as a filter to different signals.
[17]The related MPEG for this series is shown in $WaveObject\backslash AM\backslash color.mpv$.

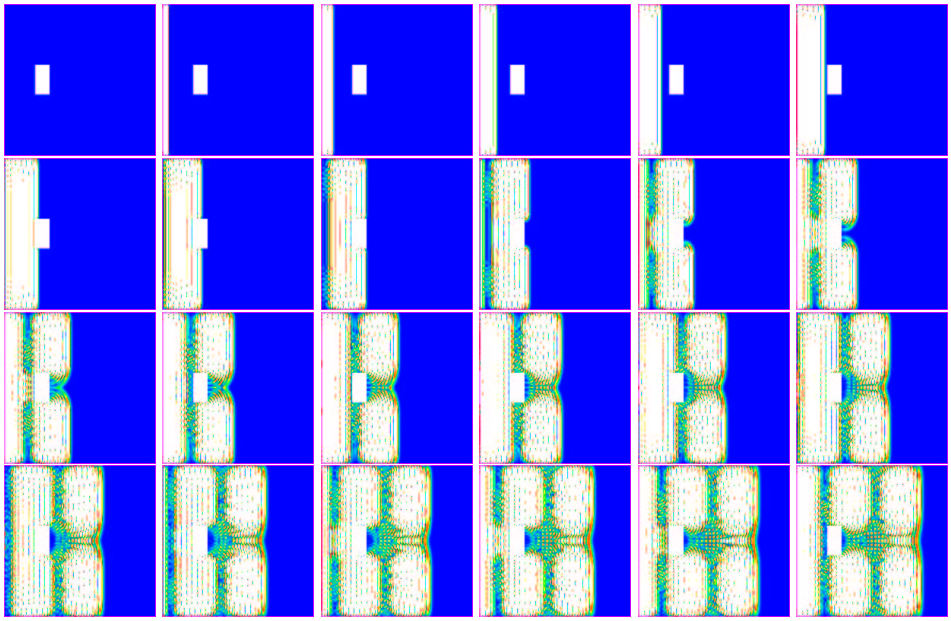


Figure 5.26: *The wave propagation shape when the there is an amplitude modulated signal in the medium.*

fig[5.27][18]..

5.6.4.2 Circular mirror

If a point source is placed in a circular mirror, then when the wave propagates from the source it will be reflected back by the circular mirror [40]. This reflection is such that all of the waves will be collected at another point in the medium. This phenomenon is modelled in fig[5.28] [19].

5.7 Array transducers

In this section TLM is used to model array transducers [90]. At first the setup is explained and then the use of TLM to steer waves in the medium is shown.

5.7.0.3 Setup

The TLM medium for these experiments is a 200 $\times$ 200 and the time step for modelling is $250\Delta T$. There are 20 source at points from (1,90) to (1,110). The sampling rate for the gener-

[18]The related MPEG for this series is shown in $WaveObject\backslash Focusing - mirror\backslash color.mpv$.
[19]The related MPEG for this series is shown in $WaveObject\backslash Circle - mirror\backslash color.mpv$.

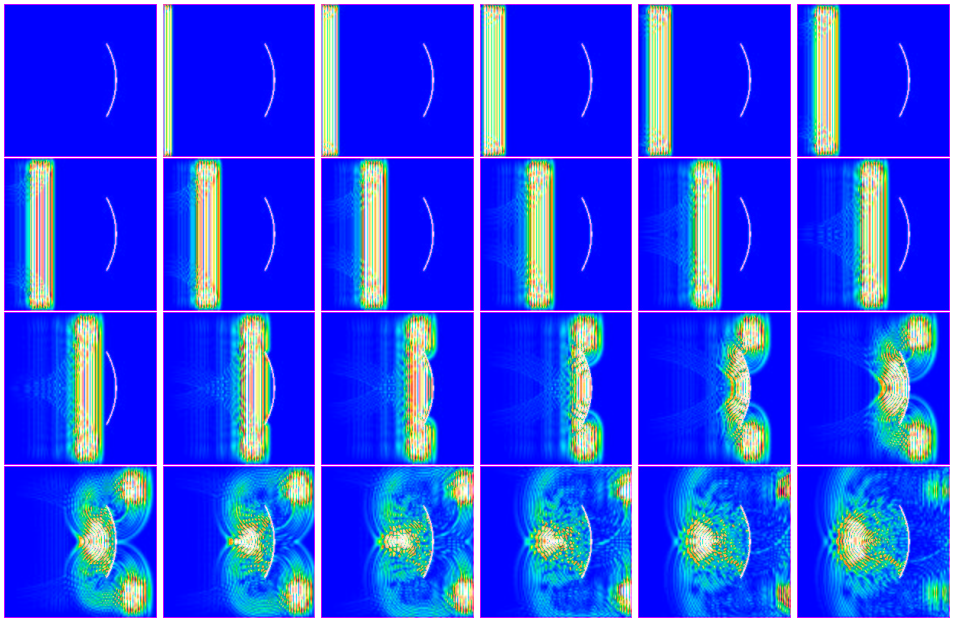


Figure 5.27: *Wave propagation shape with a focusing mirror in the medium (time difference between each two pictures is* $15\Delta T$*).*

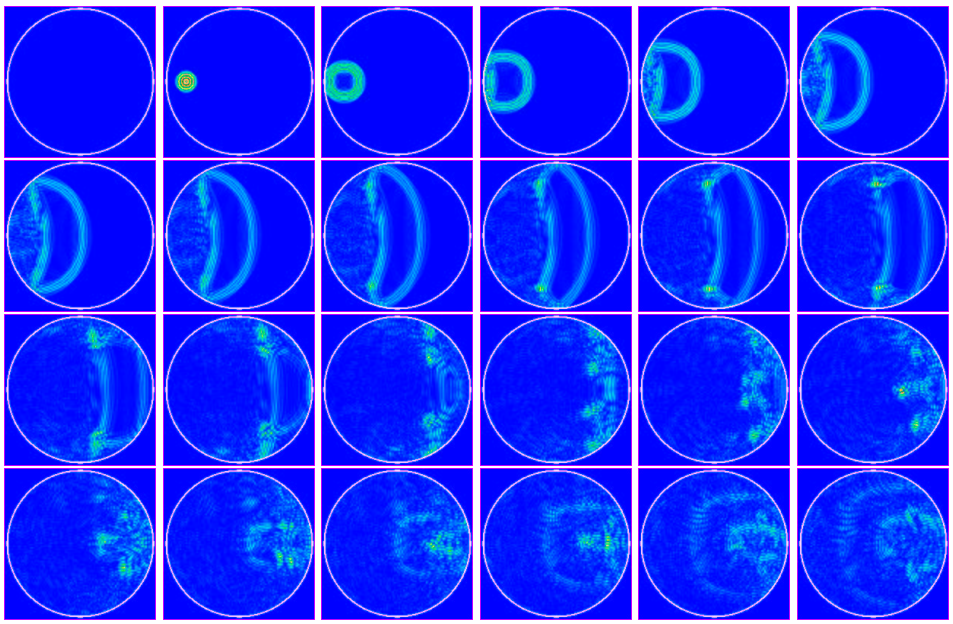


Figure 5.28: *Wave propagation shape in a circular mirror (time difference between each two pictures is* $20\Delta T$*).*

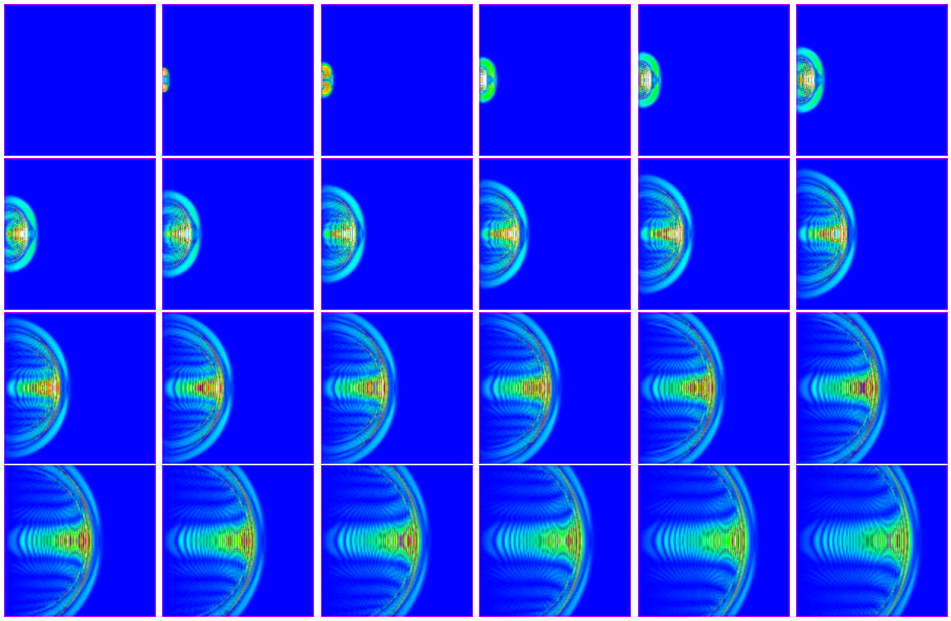


Figure 5.29: *Wave propagation shape for an array transducer when there isn't any steering. (time difference between each two pictures is* $10\Delta T$*).*

ated signal is 2.08[20], thus the new sampling rate theory is being used here.

5.7.1 Wave propagation without any steering

In this experiment no steering is used. This can be used as a reference experiment for the other experiments where the wave is steered. The result shows in fig[5.29] [21].

5.7.2 Wave propagation with steering to the left

Here an array transducer is modelled for the case when the beam is steered to the left[22]. The time difference between firing each two transducer is $1\Delta T$. With this time delay the wave will be steered approximately 45^0. The result shows in fig[5.30][23].

[20] In the chapter 6 on page 121, the use of TLM modelling with less than 4 sample per cycle is shown.
[21] The related MPEG for this series is shown in $ArrayTransducer\backslash NoSteering\backslash color.mpv$.
[22] Left and right directions are based on the wave propagation direction.
[23] The related MPEG for this series is shown in $ArrayTransducer\backslash LeftSteering\backslash color.mpv$.

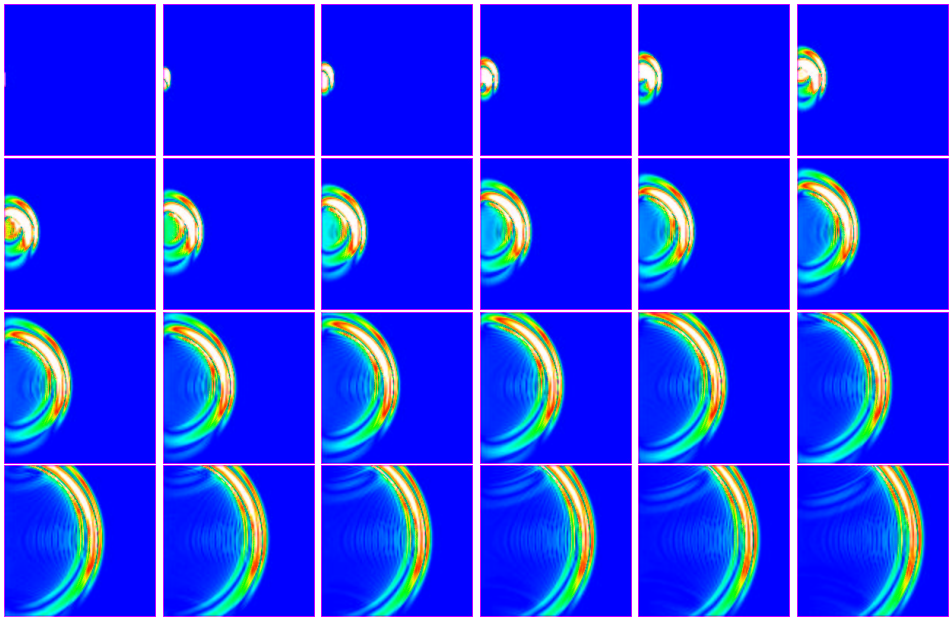


Figure 5.30: *Here the wave is steered to the left. The steering angle is approximately is* 45^0. *(time difference between each two pictures is* $10\Delta T$*).*

5.7.3 Wave propagation with steering to the right

Steering to the right[24] is modelled by TLM in the same way that it was used to model steering to the left. The only difference is that the time delay between firing each two adjacent node is -1 instead of 1 that was for steering to left. With this time delay the wave will be steered approximately -45^0. The result is shown in fig[5.31][25].

5.7.4 Steering in other angles

By changing the time delay between firing each two adjacent node, and also changing the distance between two adjacent node, it is possible to change the steering angle. For example to steer to the left with steering angle of less than 1, one may use an array of transducer where the distance between each two adjacent node in the array is $2\Delta\ell$.

The other choice is to change the sampling rate, as it will change the quantization error of ΔT. To summarize, to model steering with any arbitrary angle, one can change any of the following parameters:

[24] left and right directions are based on the wave propagation direction.
[25] The related MPEG for this series shown in $ArrayTransducer\backslash RightSteering\backslash color.mpv$.

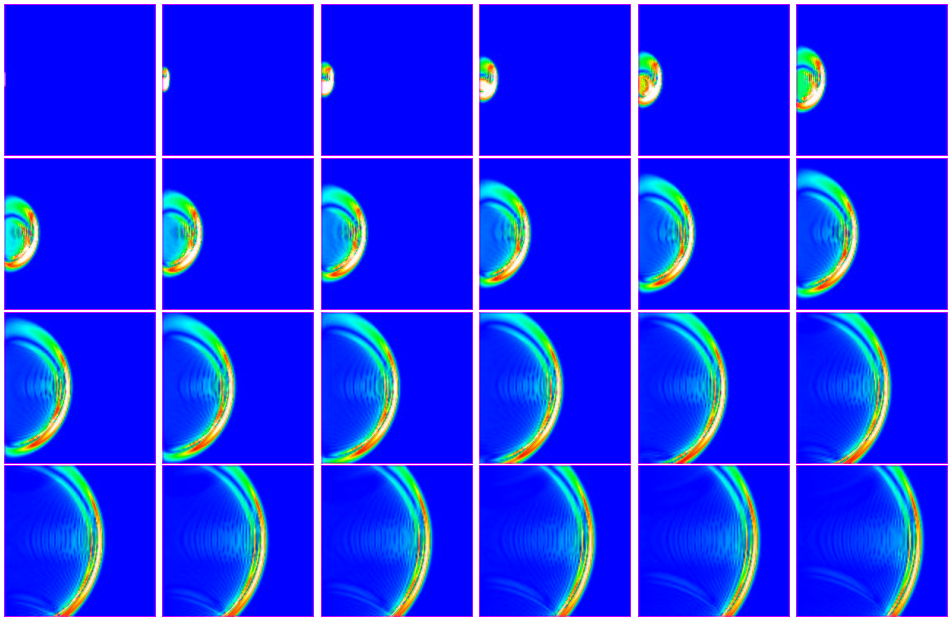


Figure 5.31: *Steering to the right shows here. The steering angle is approximately is* -45^0. *(time difference between each two pictures is* $10\Delta T$*).*

1. Distance between each two adjacent nodes.
2. Time delay between firing each two adjacent nodes.
3. Sampling rate.

5.8 Doppler effects

TLM can be used to model Doppler effects [38]. In these experiments TLM modelling is used to model the Doppler effects caused by the receiver moving toward (or away) of the transmitter. This is the first time that TLM is used to model Doppler effects.

5.8.0.1 Setup

The medium size is 50×200 and the transmitter is located at (25,10). The receiver is initially located at (25,70) and moves toward (or away) of the transmitter. The speed of the receiver movement is normalized to the speed of wave in the medium so that a speed of $\frac{1}{5}$ means that the speed of the receiver in the medium is $\frac{C}{5}$ when C is the speed of wave in the medium. The direction of movement is the direction from the receiver to the source, so a positive speed shows that the receiver is moving toward the source and a negative number for speed shows that the

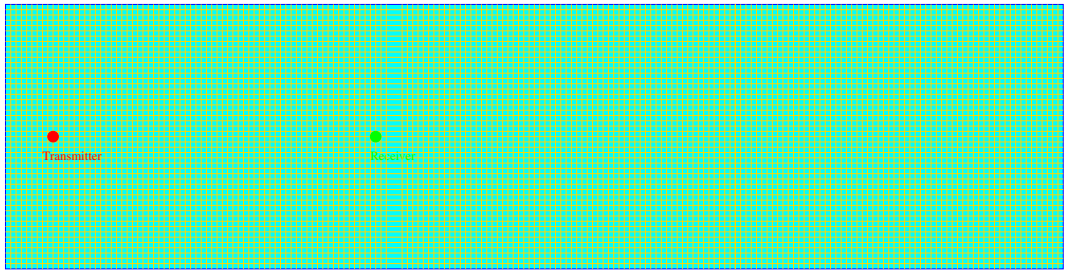


Figure 5.32: *The setup for modelling the Doppler effect.*

receiver is moving away from the source. The setup is shown in fig[5.32]. The sampling rate is 20 sample per cycle unless otherwise stated.

5.8.0.2 Speed $= \frac{1}{5}$

In these experiments , the receiver moves toward (or away) from the source and the speed of receiver is $\frac{1}{5}$. The received signal is shown in fig[5.33]. There are three signals in this fig[26]:

Speed $= 0$ **:** The receiver doesn't move. This signal is added to this figure to have a reference that the other signals can be compared with.

Speed $= \frac{1}{5}$ **:** Here the receiver is moving toward the source with the speed of $\frac{1}{5}$. In this case the frequency of the received signal is increased.

Speed $= -\frac{1}{5}$ **:** Here the receiver is moving away from the source with the speed of $\frac{1}{5}$. The received signal frequency is less than the original transmitted signal.

As can be seen from the fig[5.33], since the movement of the receiver is a discretised, noise accompanies the signal. This noise is smaller when the speed is lower or the sampling rate is higher. The effect of this noise can be seen clearly in the frequency domain representation of the signals. To find the frequency domain representation of the signals, an FFT was used. The result is shown in fig[5.34][27].

As seen from this figure, discretising the movement generates some noise at high frequency. To reduce this noise, we can reduce the speed or increase the sampling rate.

[26]The amplitude for the case when the speed is $\frac{1}{5}$ is higher than the case when the speed is $-\frac{1}{5}$ since in the first case the receiver moves toward source and hence it's amplitude is increasing but in the second case the receiver moves far from source and the amplitude in decreasing

[27]A 64 sample FFT is used and the samples were collected the steady state section of the signals (from sample 180 to sample 244). The figure shows signal magnitude. The phase is not show here.

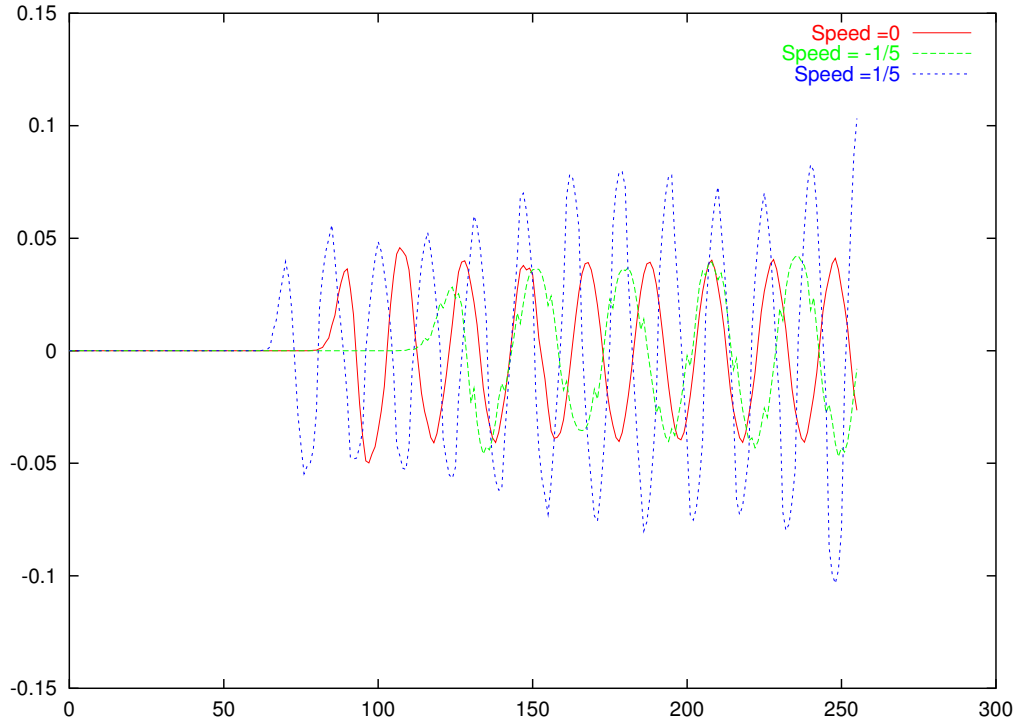


Figure 5.33: *The received signals in the time domain when the speed is* $\pm\frac{1}{5}$.

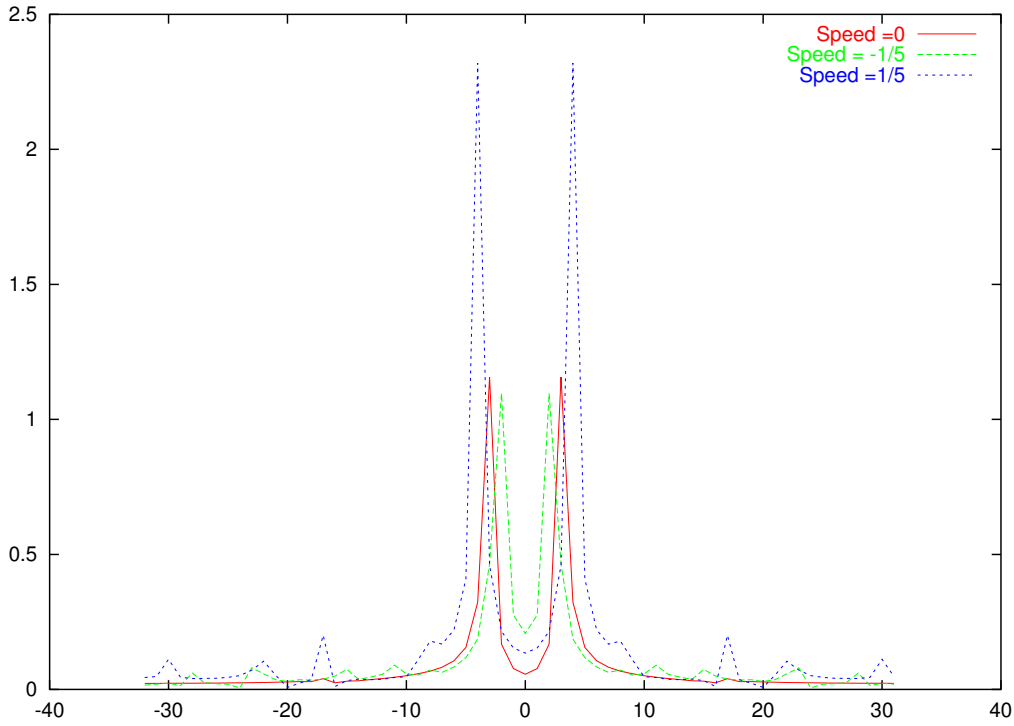


Figure 5.34: *The received signals in frequency domain when the speed is* $\pm\frac{1}{5}$.

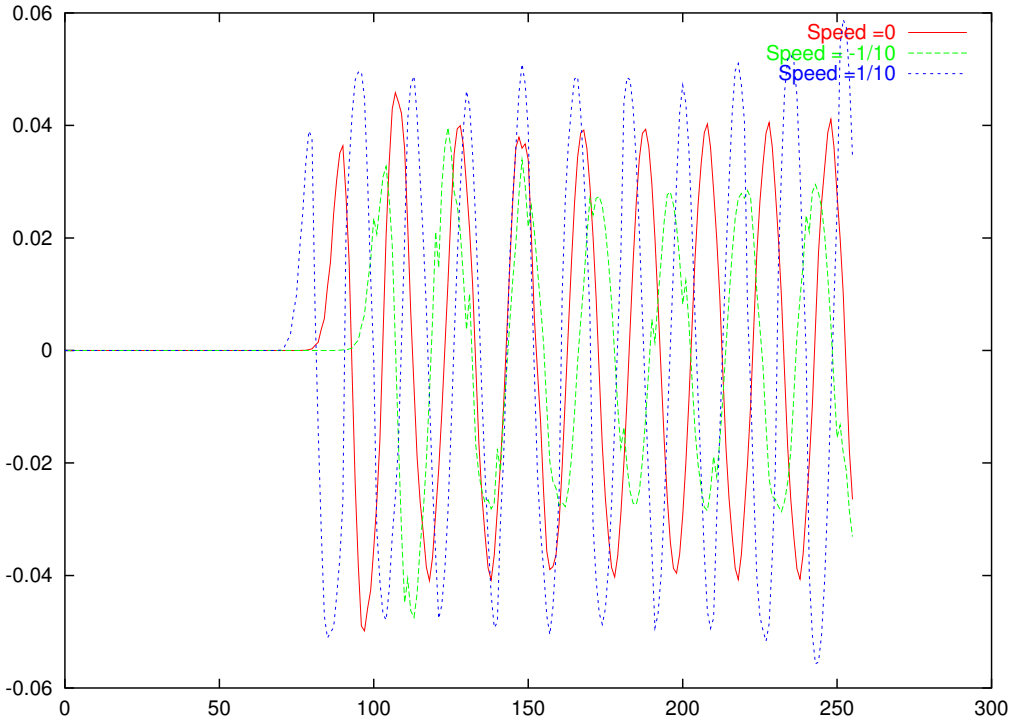


Figure 5.35: *The received signal in time domain when the speed is* $\pm\frac{1}{10}$.

5.8.1 Speed $\frac{1}{10}$

To show how the speed effects the descretising noise, the experiments were repeated with a receiver speed of $\frac{1}{10}$. The results are shown in fig[5.35] and fig[5.36]. As can be seen, the noise level is less here compared with when the speed is $\frac{1}{5}$. In fig[5.37] the signals from both experiments are drawn so that comparing them with each other become simpler.

5.8.2 Sampling rate 50

In this experiment the effect of the sampling rate on the digitising noise is shown. The sampling rate here is 50 sample per cycle. The result is shown in fig[5.38 and fig[5.39]. As seem from these pictures the descretising noise here is less than the previous experiments.

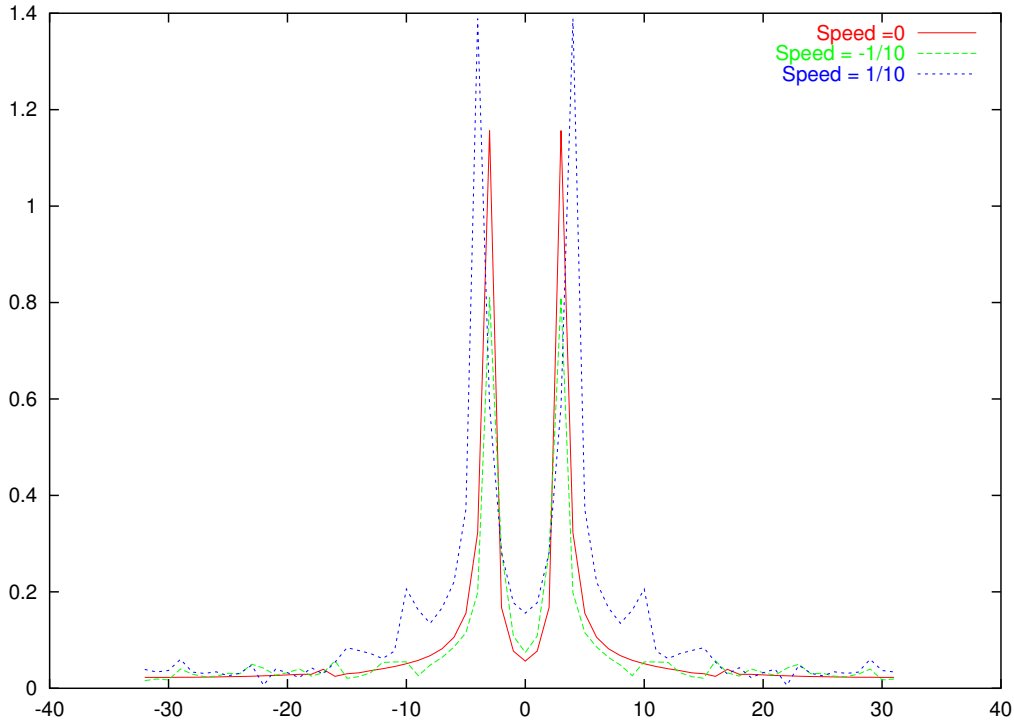


Figure 5.36: *The received signal in frequency domain when the speed is* $\pm\frac{1}{10}$.

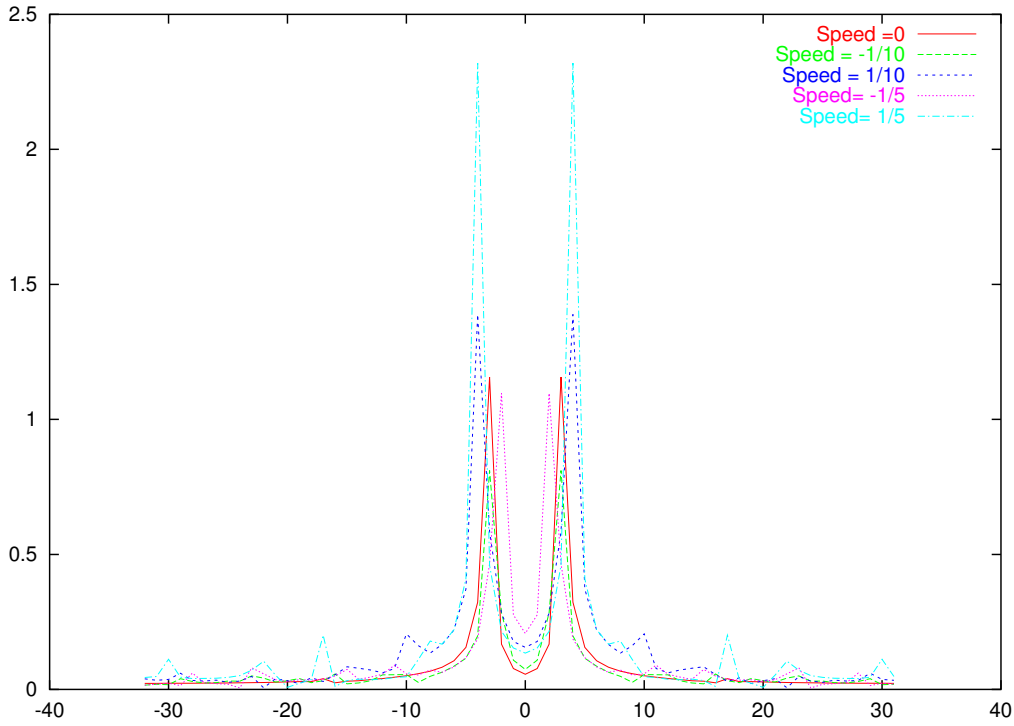


Figure 5.37: *The received signal in frequency domain when the speed is* $\pm\frac{1}{5}$ *and* $\pm\frac{1}{10}$.

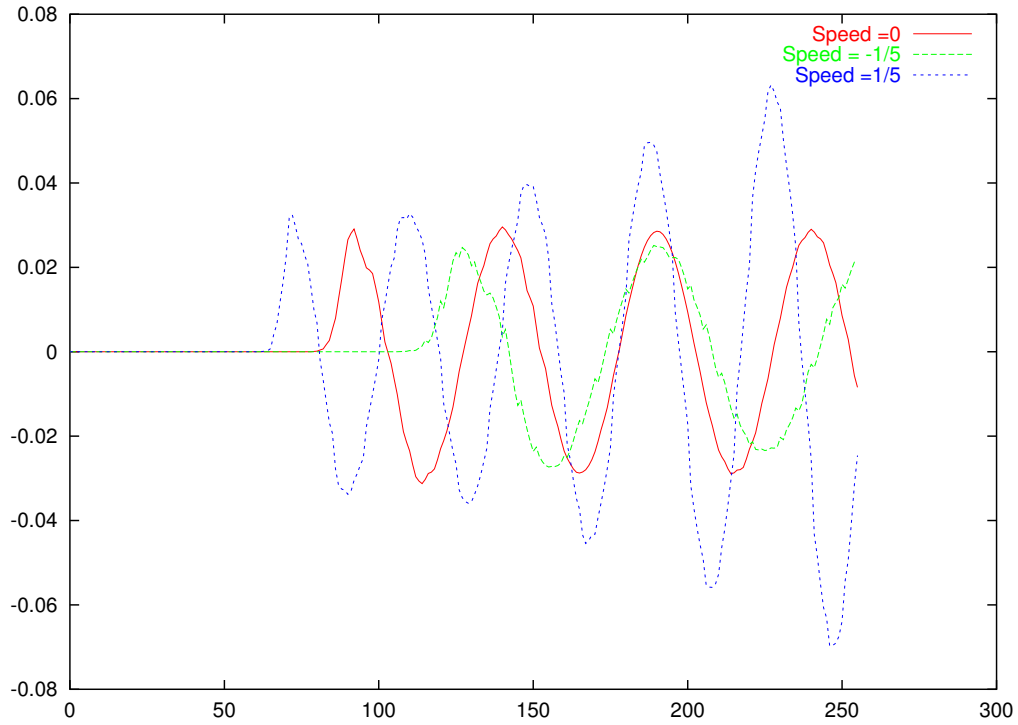


Figure 5.38: *The received signal in the time domain when the speed* $= \pm\frac{1}{5}$ *and the sampling rate is 50 samples per cycle.*

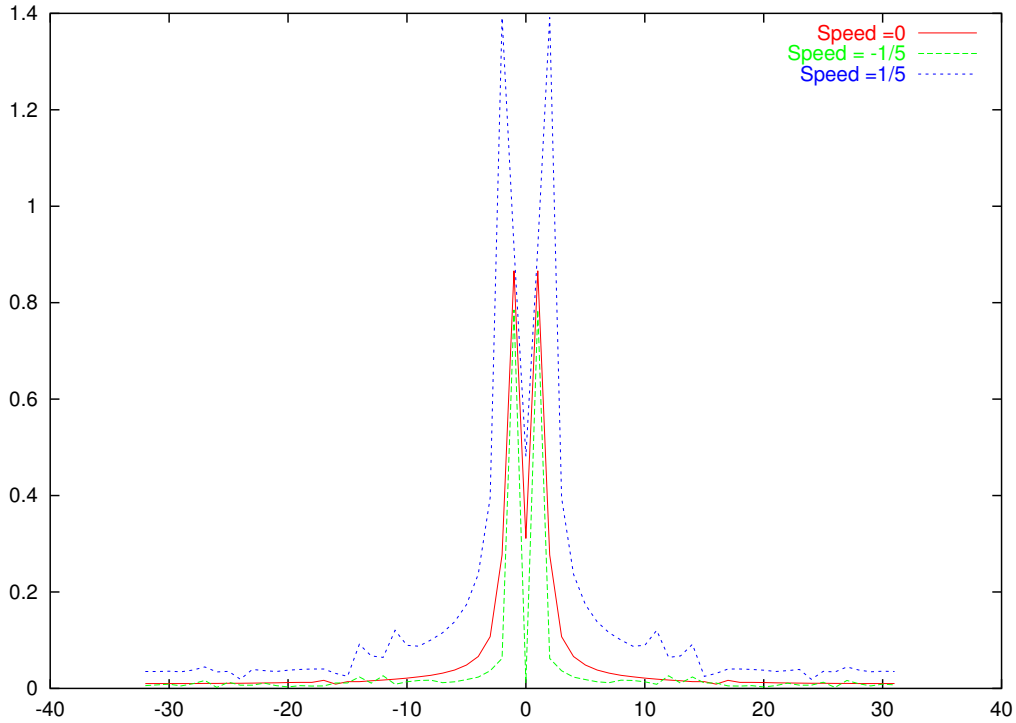


Figure 5.39: *The received signal in frequency domain when the speed is* $\pm\frac{1}{5}$ *and the sampling rate is 50 samples per cycle.*

Chapter 6

Sampling rate

In this chapter a new theory for sampling rate in TLM modelling will be introduced. The original sampling rate that was presented by Johns [59] was based on wave propagation in a transmission line. The new sampling rate is based on system theory and the filtering effect of a TLM mesh.

6.1 Johns theory for sampling rate

In his first paper on TLM modelling for electromagnetic wave propagation [59], Johns shows that the wave propagation in the axial direction[1] is frequency dependent but the speed in the diagonal direction[2] direction is always constant and is equal to $\frac{C}{\sqrt{2}}$ where C is the speed of wave in the real medium. He shows that the speed of the wave propagation in the axial direction can be found from equation [6.1]:

$$\sin\left(\frac{\beta_n \Delta\ell}{2}\right) = \sqrt{2}\sin\left(\frac{\omega\Delta\ell}{2C}\right) \tag{6.1}$$

where β_n is the propagation constant of the medium. β_n is related to the frequency and the speed of the wave propagation in the TLM mesh and the speed of the wave propagation in the real medium by the following equation:

$$\frac{\nu_n}{C} = \frac{\omega}{\beta_n C} \tag{6.2}$$

[1] Axial direction is the direction along the X or Y direction (see fig[6.3])
[2] Diagonal direction is direction along the diagonal of squares in the TLM mesh (see fig[6.3])

where C is the speed of the wave propagation in the real medium and ν_n is the speed of the wave propagation in the TLM medium. The value of β_n can be found from equation 6.2:

$$\begin{aligned} \beta_n &= \frac{\omega}{\frac{\nu_n}{C}C} \\ \beta_n &= \frac{\omega}{\nu_n} \end{aligned}$$

and substituting for β_n in the equation 6.1:

$$\begin{aligned} \sin\left(\frac{\frac{\omega}{\nu_n}\Delta\ell}{2}\right) &= \sqrt{2}\sin\left(\frac{\omega\Delta\ell}{2C}\right) \\ \sin\left(\frac{\omega\Delta\ell}{2\nu_n}\right) &= \sqrt{2}\sin\left(\frac{\omega\Delta\ell}{2C}\right) \end{aligned}$$

Since $\omega = \frac{2\pi C}{\lambda}$:

$$\begin{aligned} \sin\left(\frac{\frac{2\pi C}{\lambda}\Delta\ell}{2\nu_n}\right) &= \sqrt{2}\sin\left(\frac{\frac{2\pi C}{\lambda}\Delta\ell}{2C}\right) \\ \sin\left(\frac{\pi C\Delta\ell}{\nu_n\lambda}\right) &= \sqrt{2}\sin\left(\frac{\pi\Delta\ell}{\lambda}\right) \end{aligned} \tag{6.3}$$

The normalized value for the wave propagation speed and the sampling rate[3] may be defined as:

$$\begin{aligned} V &= \tfrac{\nu_n}{C} \quad \text{Normalized speed} \\ S &= \tfrac{\lambda}{\Delta\ell} \quad \text{Sampling rate} \end{aligned}$$

By substituting the above values in equation 6.2:

$$\sin\left(\frac{\pi}{VS}\right) = \sqrt{2}\sin\left(\frac{\pi}{S}\right) \tag{6.4}$$

[3] In the original Johns work, he uses $\frac{\Delta\ell}{\lambda}$, which is the inverse of sampling rate. Using sampling rate here make it simpler to understand the result.

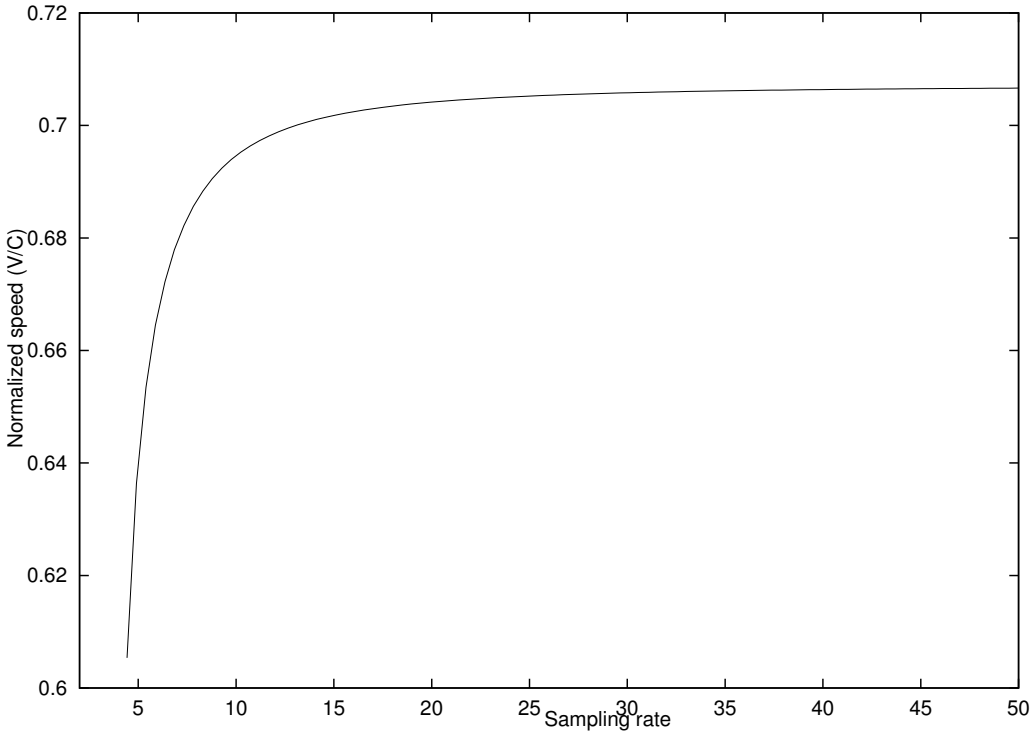


Figure 6.1: *Relationship between sampling rate and the wave propagation speed in a TLM mesh.*

Fig[6.1] shows the speed of wave for different sampling rates. It can be seen from fig[6.1] that when the sampling rate is very high, the speed of the wave propagation in the TLM medium is $\frac{C}{\sqrt{2}}$, where C is the speed of the wave propagation in the real medium. As the sampling rate is reduced [4] then the speed of the wave propagation is reduced. Fig[6.2] shows the wave propagation speed for sampling rates near 4 samples per cycle. As can be seen from fig[6.1] and fig[6.2], the speed of the wave propagation is reduced when the sampling rate is reduced to 4 samples per cycle. Unfortunately this method can't show us what is happening when the sampling rate is less than 4 samples per cycle[5].

6.1.1 Examining wave speed for different sampling rates

This section, shows how the sampling rate will affect the wave propagation speed.

[4] This is equal to increasing the frequency of the signal when the sampling rate is constant.
[5] From the sampling theory it is clear that one can samples a signal with 2 sample per.

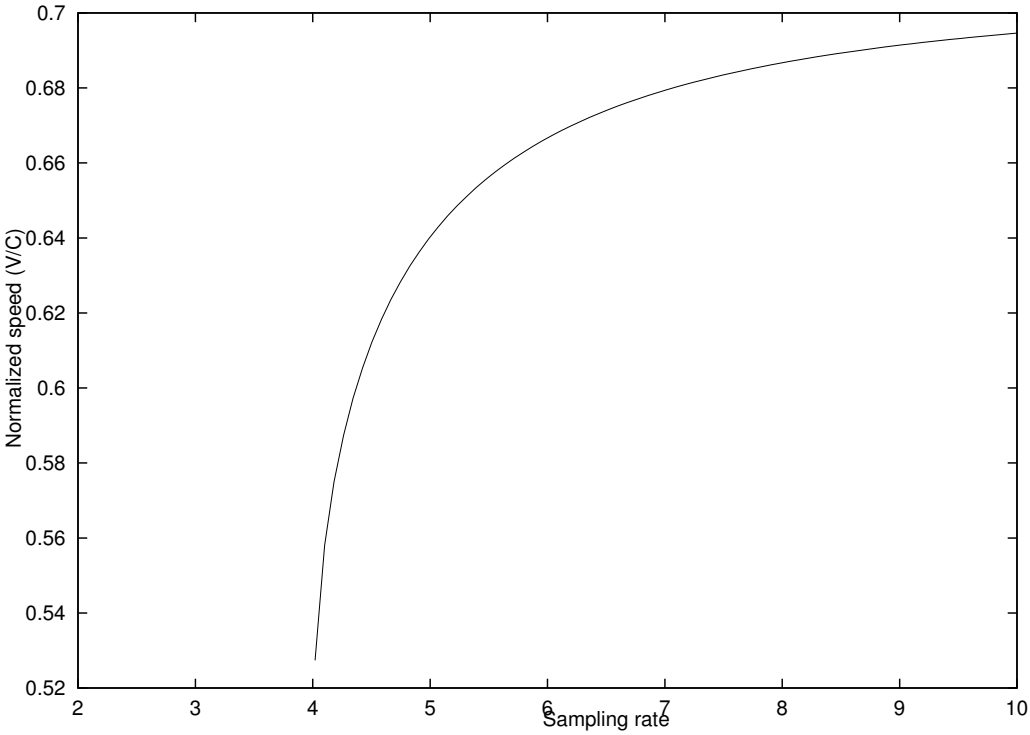


Figure 6.2: *A closer look at the relationship between the sampling rate and the wave propagation speed near 4 samples per cycle.*

6.1.1.1 TLM mesh setup

Fig[6.3] shows the setup for examining the wave propagation speed in the TLM mesh. The TLM medium size is a 100×100. There is a source at (50,50) and there are two receiver at points (64,50),(60,60). It should be noted that each of these two receivers is approximately the same distance from the source. The receiver at (64,50) is on an axial line from the source and the receiver at (60,60) is on a diagonal line from the source. For simplicity these receivers will be called the axial receiver and the diagonal receiver respectively.

6.1.1.2 Case 1: Sampling rate is 20 samples per cycle

In this experiment, a sinusoidal signal with 20 samples per cycle is injected at the source and the resulting signals at the receivers are recorded. These signals are shown in fig[6.4]. As can be seem from this fig, the wave shape and delay for the received signal at these two receivers are the same. It shows that the speeds of waves propagation in these two directions are approximately the same. The classical circular wave propagation pattern is observed indicating correct modelling of the real world. The wave propagation shape is drawn in fig[6.5]

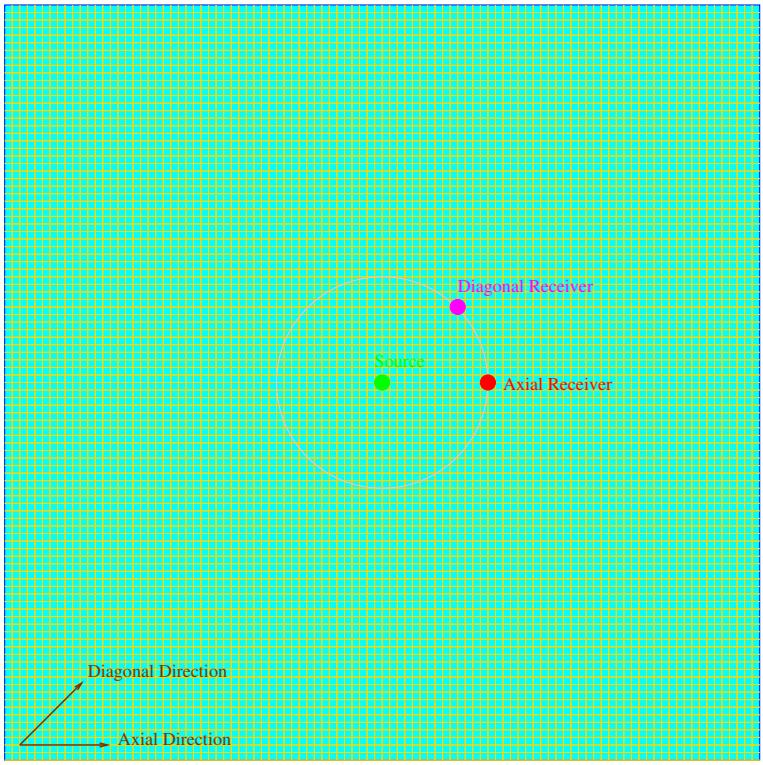


Figure 6.3: *The setup for examining the wave propagation speed in the TLM medium.*

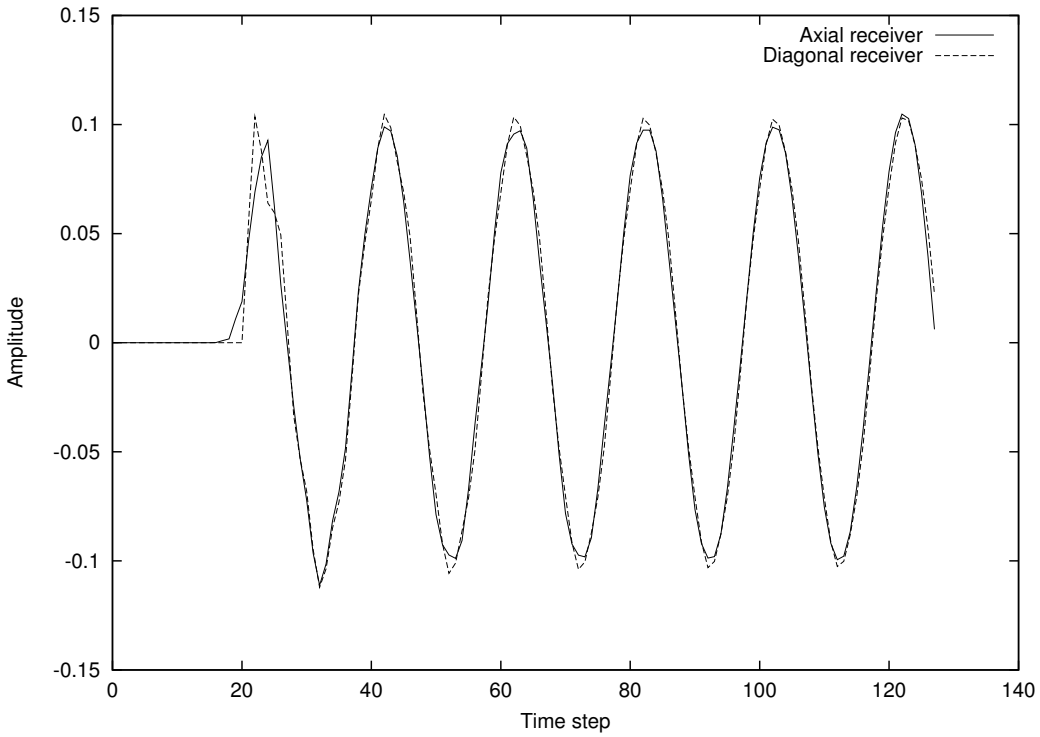


Figure 6.4: *Received signal at axial and diagonal receiver when the sampling rate is 20 sample per cycle.*

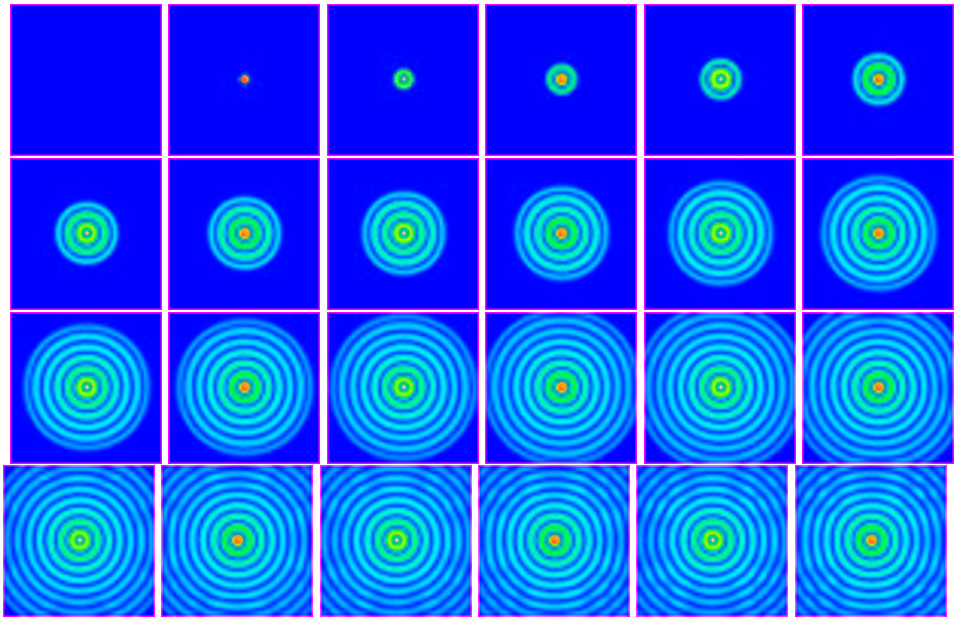


Figure 6.5: *Wave propagation shape when the sampling rate is 20 samples per cycle (time difference between each two pic is* $5\Delta T$*).*

6.1.1.3 Case 2: Sampling rate is 5 samples per cycle

In this experiment, a sinusoidal signal with 5 samples per cycle is injected at the source and the resulting signals at the receivers are recorded. These signal is shown in fig[6.6]. As can be seem from this fig, the wave shapes and delays for the received signals at these two receivers are completely different. It can be easy seen that the wave propagation speed in these two direction (axial and diagonal directions) are not the same. The wave propagation shape is shown in fig[6.7]. This wave propagation modelling is not correct as the wave fronts are not circular and more power is propagated along the diagonal directions than the axial directions.

6.2 Systematic approach to TLM modelling

If we place a source and a receiver in the medium, and assuming the source as input and the receiver as output, we can model TLM as a linear system[6]. When TLM mesh is modelled as a linear system then it is possible to find a transfer function for it [91] [92]. This transfer function depends on the relative positions of the the source and the receiver. As the distance between the source and the receiver increases, there is more attenuation and hence the magnitude of the

[6] A TLM system should be linear since all of its sub systems are linear.

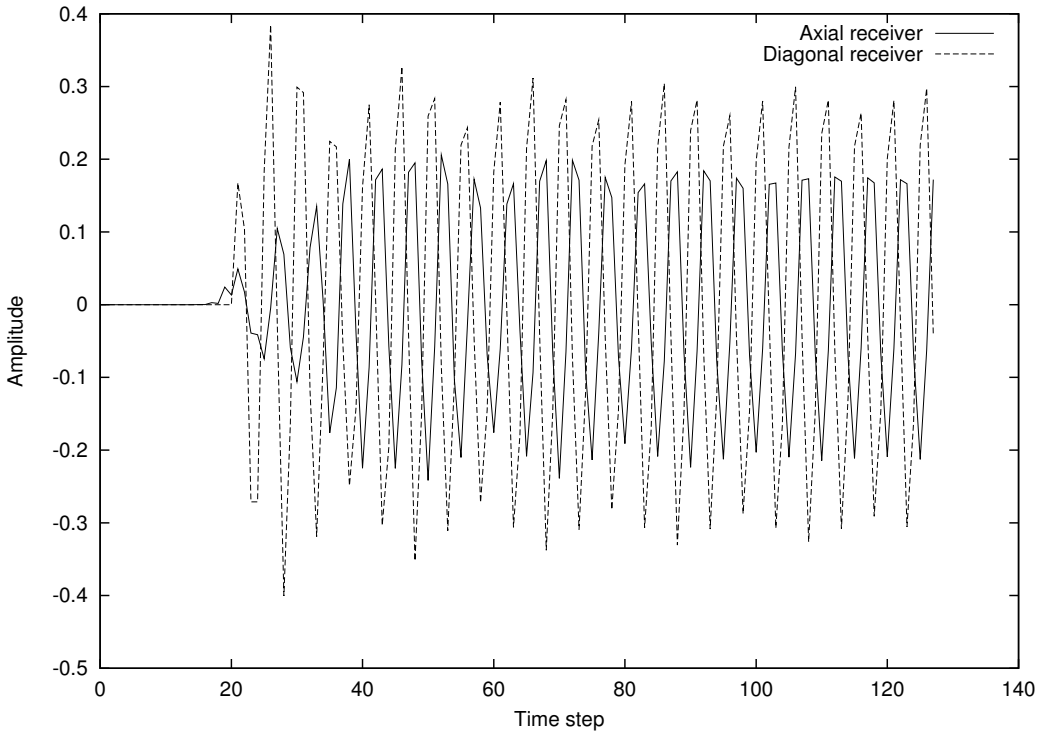


Figure 6.6: *Received signal at axial and diagonal receiver when the sampling rate is 5 sample per cycle.*

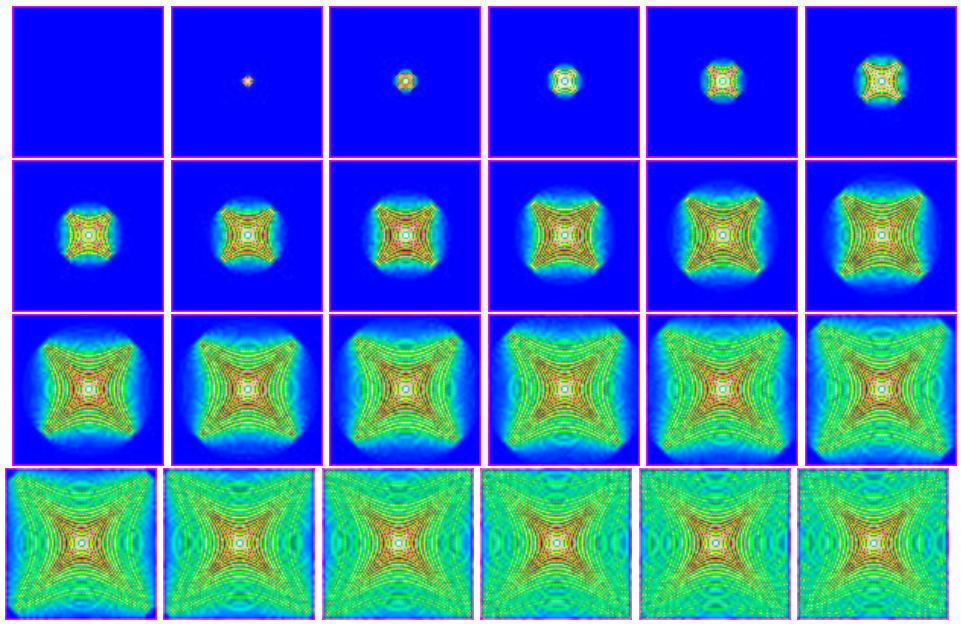


Figure 6.7: *Wave propagation shape when the sampling rate is 5 samples per cycle (time difference between each two pic is* $5\Delta T$*).*

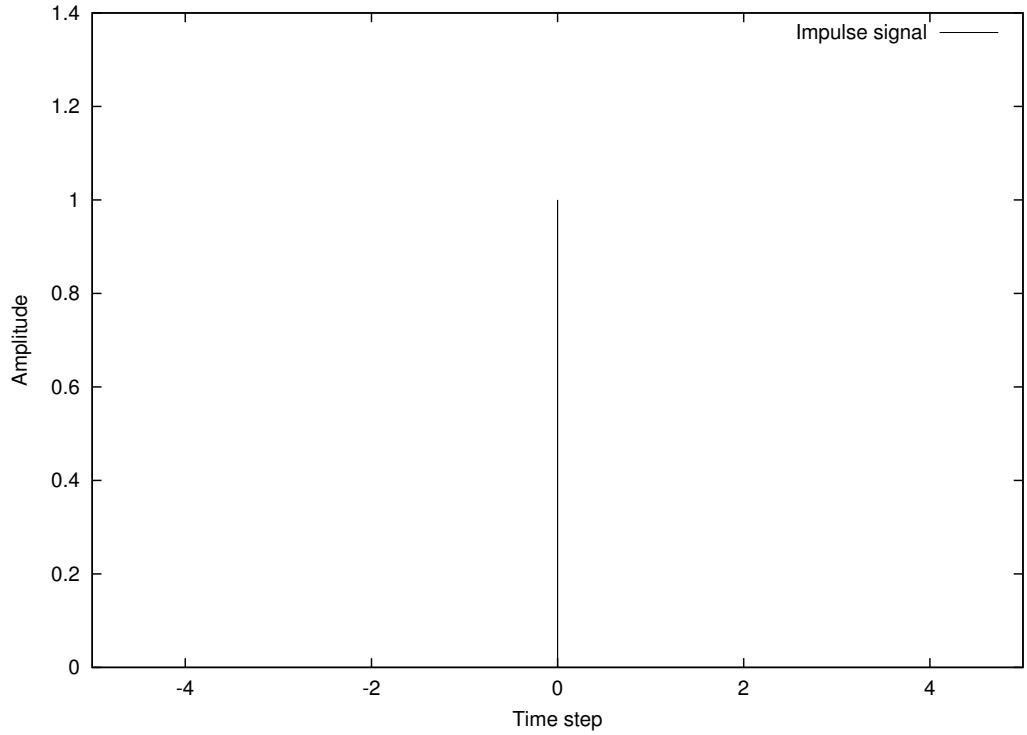


Figure 6.8: *The impulse function used to find the transfer function of the TLM mesh.*

transfer function decreases.

6.2.1 Impulse response of the TLM mesh

As the transfer function[7] of a TLM medium is very complex[8], we can't find it mathematically. One way to find the transfer function experimentally is to inject an impulse signal to the source and find the received signal. The mathematical definition of an impulse signal is shown in equation 6.5 and its graphical representation is shown in fig[6.8][9]

$$\delta[n] = \begin{cases} 1 & \text{n=0,} \\ 0 & \text{otherwise.} \end{cases} \tag{6.5}$$

It should be noted that an impulse signal has a flat power spectrum [94]. One may feel that injecting an impulse signal to the TLM mesh isn't a good idea for calculating transfer function since the TLM is valid only for low frequencies and an impulse signal contain all of the

[7] Since the transfer function is the fourier transform of the impulse response, we use them interchangeably here.
[8] It is a function of source and receiver coordinates.
[9] Since a TLM modelling is a digital model, this is the digital version of impulse signal[93].

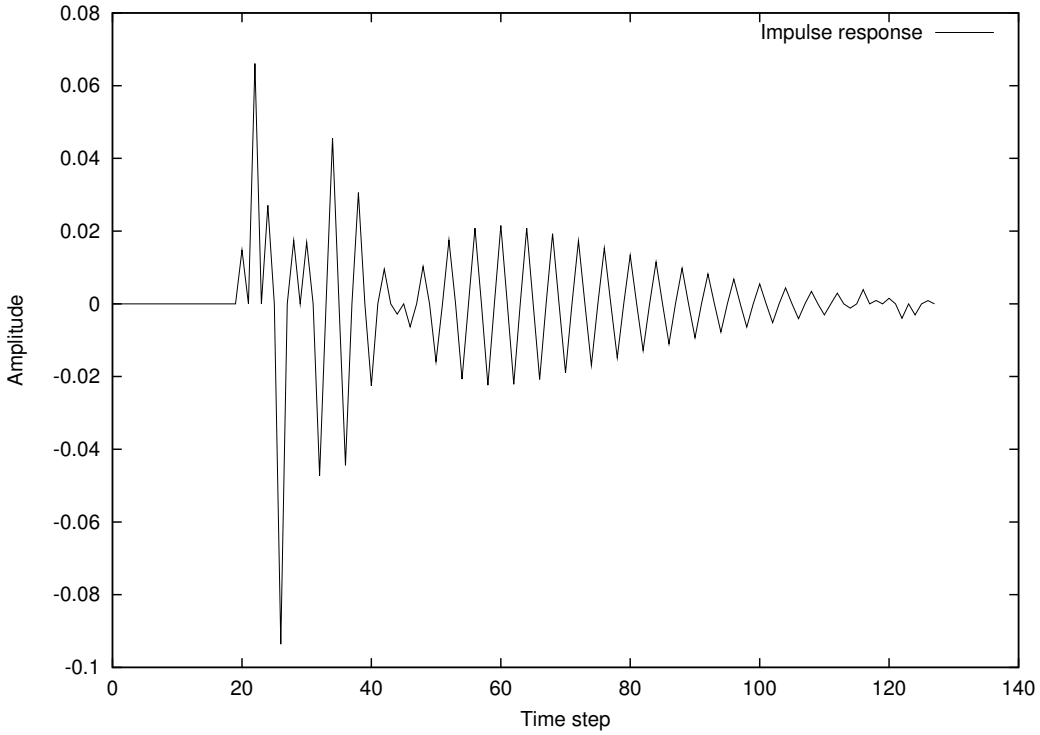


Figure 6.9: *Impulse response when the source is at (50,50) and the receiver is at (55,65).*

frequencies and most importantly high freqencies[10]. The answer is that:

Since the TLM is a linear system, it is always possible to find a transfer function for it. By definition, the transfer function is the system response to the impulse signal. It means that one is always able to find a transfer function for a linear system by injecting an impulse signal to the input of the system and saving the output[11]. In fig[6.9] a sample transfer function is shown. The system input is the source at point (50,50) and system output is the receiver is at point (55,65). The general form of this impulse response will be explained later in the thesis.

6.2.2 Transfer function for the TLM mesh

In the previous section it was explained how one can find the impulse response. In this section it will be explained how to find the transfer function. The best way to find a frequency domain transfer function in is to find the fourier transform [96] of the time domain impulse response.

[10]Corresponding to sampling rate of less or equal 4 sample per cycle.

[11]It should be noted that when the system is a continues system, creating an impulse signal is impossible since its value for t=0 is ∞. In digital systems it is always possible to create an impulse signal and find the transfer function with it [95].

Since the TLM modelling is a digital one, we can use discrete fourier transform (DFT)[12]. By finding the DFT of a time domain impulse response one can find the frequency domain transfer function. The DFT of the sample time domain impulse response that was found in the last section is shown in fig[6.10]. The result of the DFT is a sampled version of the discrete time fourier transform. Each of these samples called a bin number and the distance between each two bin number called bin width. The relationship between sample interval (time difference between each two adjacent sample in the time domain) and bin width (frequency difference between each two adjacent bin in the DFT output) will be shown in the next section.

6.2.2.1 Sampling interval in the time domain and bin width in frequency domain

The sampling rate in time domain dictates the maximum frequency that one can detect by the DFT in the frequency domain [97]. This is a direct result of sampling theory[13]. If the sample interval in the time domain is ΔT, then the maximum frequency in DFT can be calculated by equation 6.6.

$$MaxF = \frac{1}{2\Delta T} \tag{6.6}$$

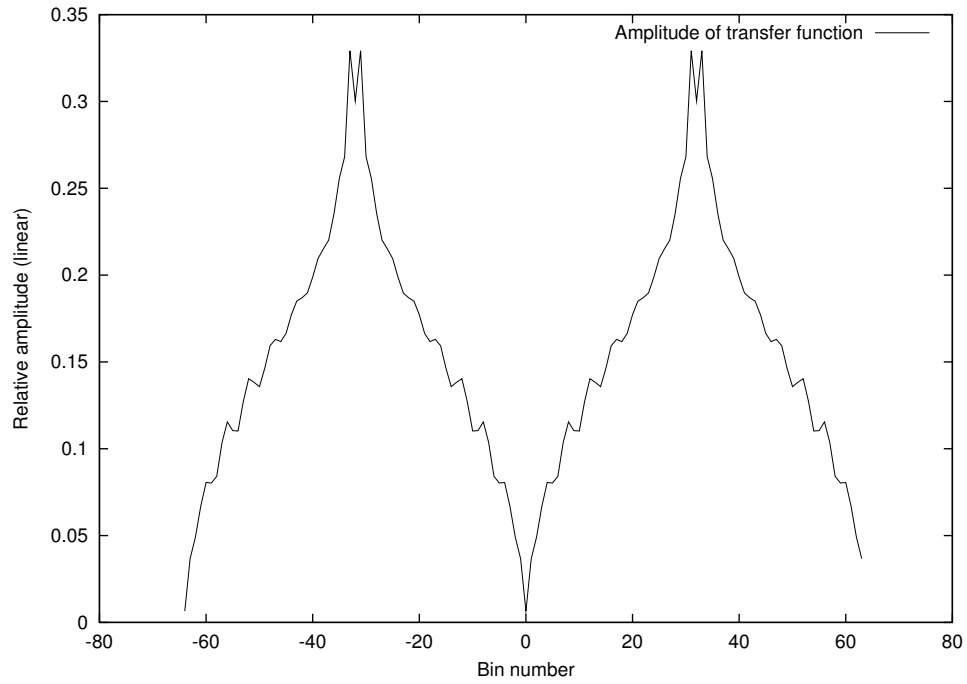
In the same way, the bin width in the DTF output is related to the maximum observation time for the signal in time domain (see equation 6.7).

$$\Delta F = \frac{1}{MaxT} \tag{6.7}$$

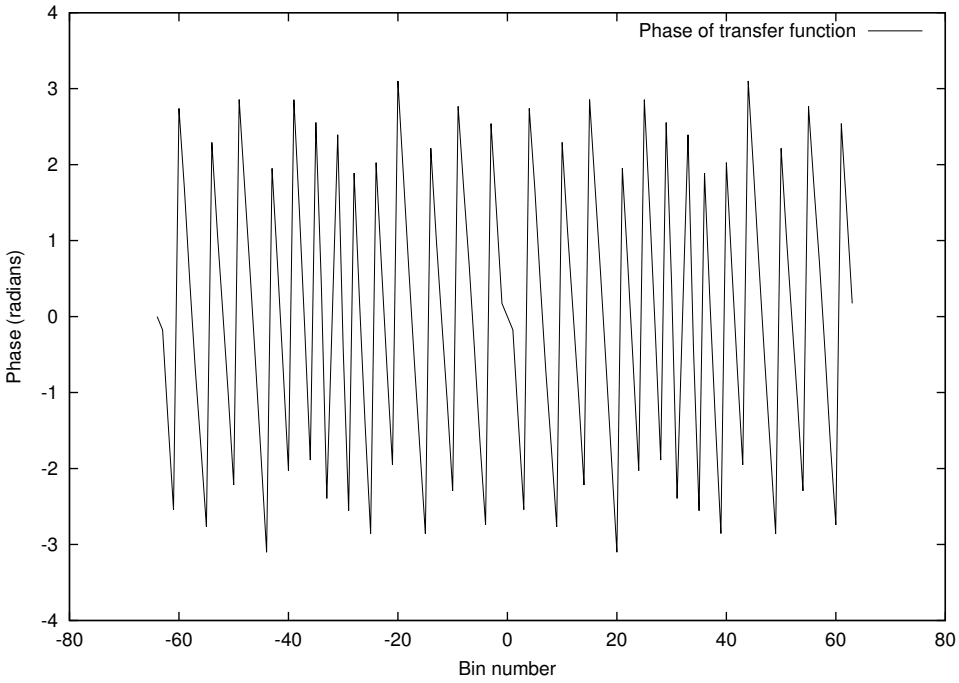
Example: if a signal sampled with ΔT of 1ms (10^3 samples per sec) and there was 128 samples of the signal. The maximum frequency that can be detected in the DFT output can be calculated

[12]DFT is the standard way for finding the fourier transfer of a digital signal. The best way to find it is FFT (Fast Fourier Transfer). There isn't any difference on how to find the fourier transform, the result should always be the same. In my work I use FFT as it is fast but I explain my work here by using DFT as it is standard way and comes directly from the definition. more information can be found in references.

[13]Sampling theory says that for error free reconstruction of a signal, one should sample it at least by two samples per cycle for its largest frequency.



(a) Amplitude



(b) Phase

Figure 6.10: *Transfer function when the source is at (50,50) and the receiver is at (55,65).*

by using equation 6.6:

$$\begin{aligned} MaxF &= \frac{1}{2\Delta T} \\ MaxF &= \frac{1}{2 \times 10^{-3}} \\ MaxF &= \frac{1000}{2} \\ MaxF &= 500Hz \end{aligned}$$

The bin width (ΔF) can be calculated in the same way by using equation 6.7. The $MaxT$ here is $10^{-3} \times 128 = 1.28 \times 10^{-1}$

$$\begin{aligned} \Delta F &= \frac{1}{MaxT} \\ \Delta F &= \frac{1}{1.28 \times 10^{-1}} \\ \Delta F &= \frac{10}{1.28} \\ \Delta F &= 7.8125Hz \end{aligned}$$

So the DFT output is 128 bin from -500 Hz to 500 Hz, the bin width is 7.8125.

As it can be seen from the above example, the interpretation of the DFT output is related to sampling rate of the input signal. It is possible to present the DFT output by using the normalized sampling rate. It is clear that the $MaxF$ in the DFT output represent signal with 2 samples per cycle. The $MaxF$ is bin number N/2 when N is the number of samples in signal. For an arbitrary bin number n, the normalized sampling rate can be found by equation 6.8.

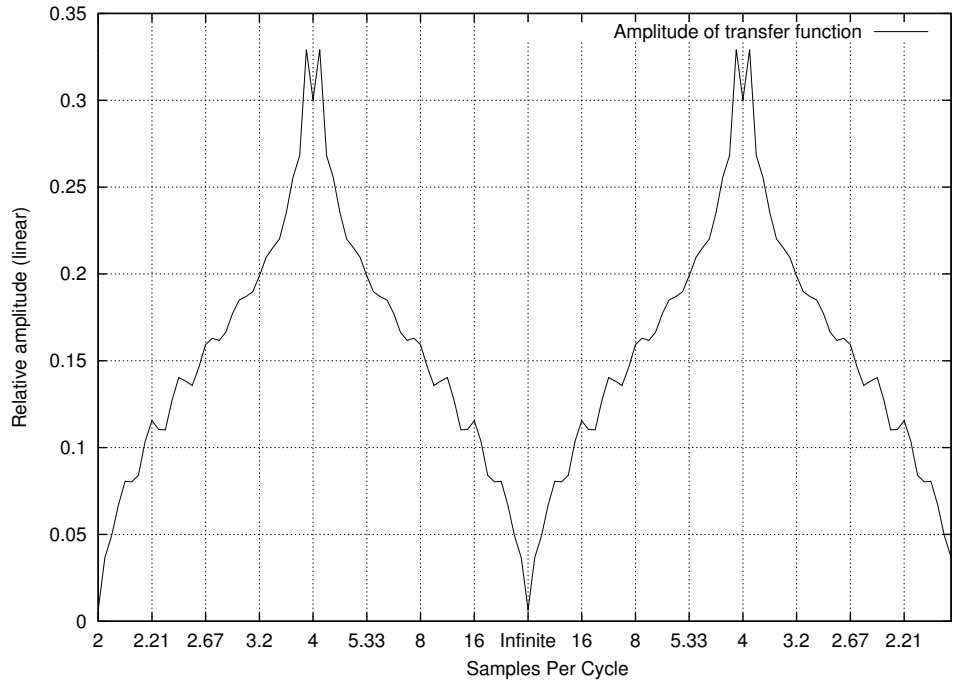
$$S_n = \frac{N}{|n|} \tag{6.8}$$

where S_n is the sampling rate for bin number n. For example the relationship between bin numbers on a 128 point DFT and normalized sampling rate is shows in table 6.1[14]. It is possible to change the bin numbers in fig[6.10] to sample per cycle by using this table. The result shown in fig[6.11].

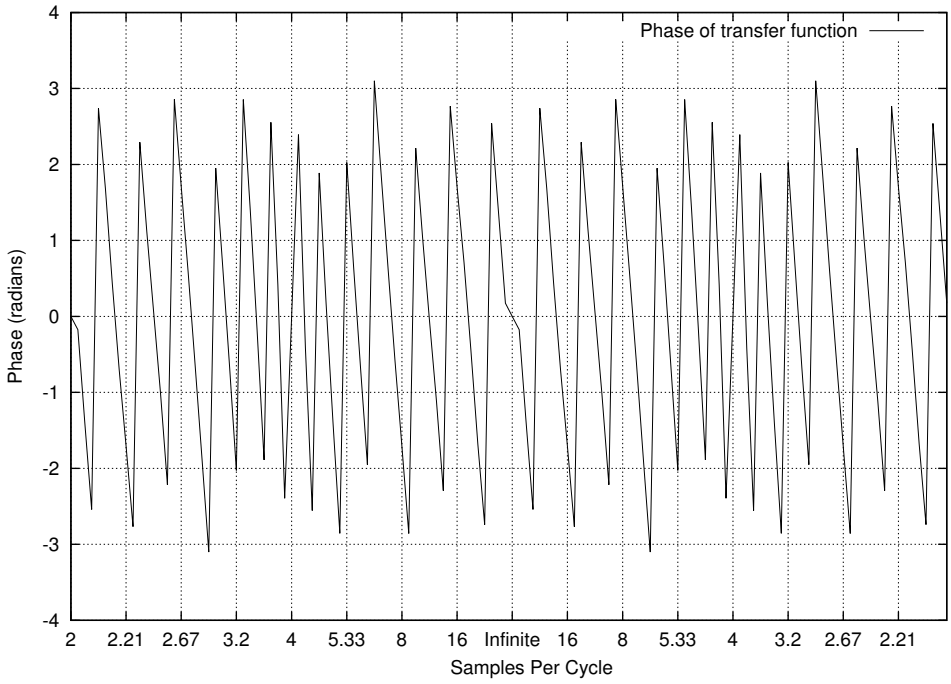
[14]Only positive frequencies are shown here

Bin	Sampling Per Cycle	Bin	Sampling Per Cycle	Bin	Sampling Per Cycle
0	∞	1	128.000000	2	64.000000
3	42.666667	4	32.000000	5	25.600000
6	21.333333	7	18.285714	8	16.000000
9	14.222222	10	12.800000	11	11.636364
12	10.666667	13	9.846154	14	9.142857
15	8.533333	16	8.000000	17	7.529412
18	7.111111	19	6.736842	20	6.400000
21	6.095238	22	5.818182	23	5.565217
24	5.333333	25	5.120000	26	4.923077
27	4.740741	28	4.571429	29	4.413793
30	4.266667	31	4.129032	32	4.000000
33	3.878788	34	3.764706	35	3.657143
36	3.555556	37	3.459459	38	3.368421
39	3.282051	40	3.200000	41	3.121951
42	3.047619	43	2.976744	44	2.909091
45	2.844444	46	2.782609	47	2.723404
48	2.666667	49	2.612245	50	2.560000
51	2.509804	52	2.461538	53	2.415094
54	2.370370	55	2.327273	56	2.285714
57	2.245614	58	2.206897	59	2.169492
60	2.133333	61	2.098361	62	2.064516
63	2.031746	64	2.000000		

Table 6.1: *Conversion between bin number in a DFT and sampling rate for a DFT with 128 samples.*



(a) Amplitude



(b) Phase

Figure 6.11: *Transfer function when the source is at (50,50) and the receiver is at (55,65). Note that the X axis converted to samples per cycle.*

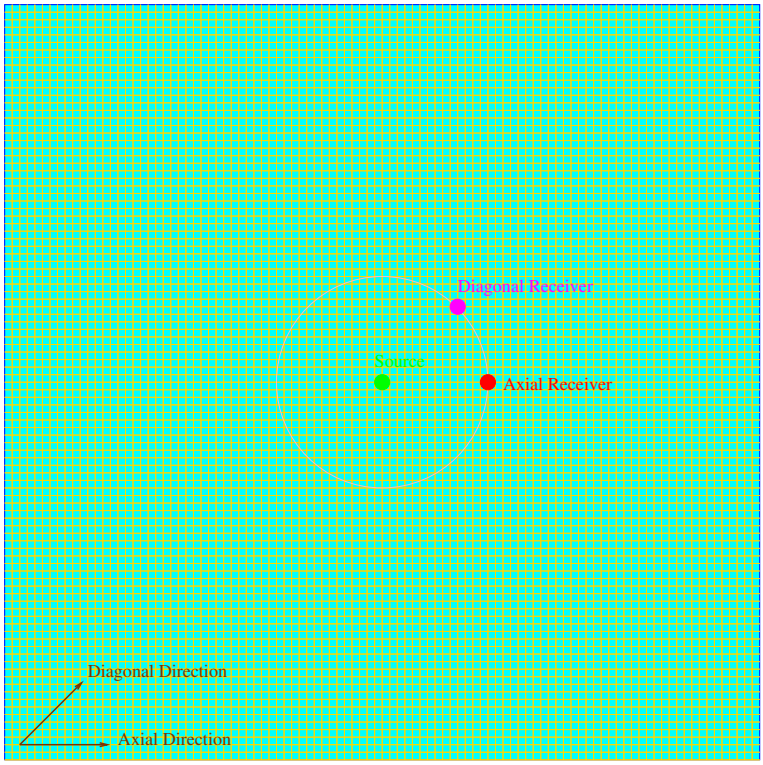


Figure 6.12: *The setup for finding a new criteria for sampling rate*

6.3 New criteria for sampling rate based on system theory

For TLM to be a good model for wave propagation, its transfer function should only depend on the distance between source and receiver and not to the source and receiver positions. This fact can be used to established a new sampling rate theory for TLM modelling.

6.3.1 TLM mesh setup

The experiments in this section is based on a setup that is shown in fig[6.12]. TLM medium size is 100×100. The source is at the point (50,50). There are two receivers at points (60,60) and (64,50). The first receiver (at the point (60,60)) is in a diagonal line with the source and the second one (at the point (64,50)) is in a axial line with the source. These two receivers will called as the diagonal receiver and the axial receiver respectively . It should be noted that the distance between each of these two receivers and the source are approximately equal.

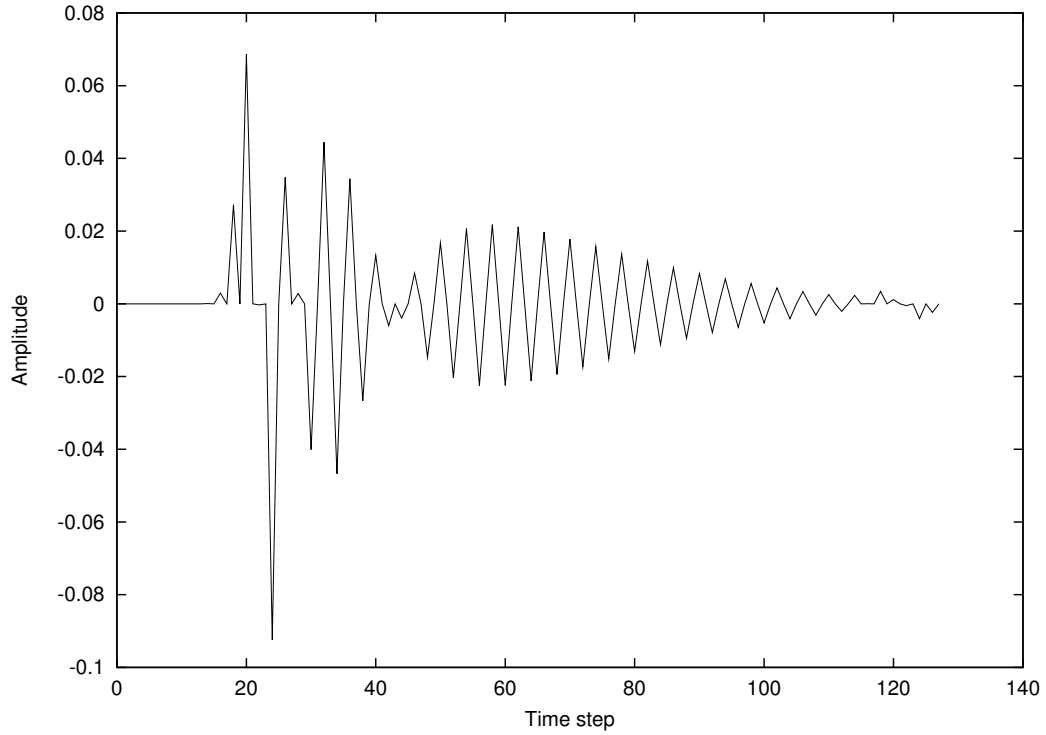


Figure 6.13: *The received signal at the axial receiver.*

6.3.2 Processing

An impulse signal was injected to the source and the resulting signals at the receivers were recorded. The received signals are shown in fig[6.13] and fig[6.14]. These impulse response were transformed into frequency and plotted using sample per cycle for X axis.The results are shown in fig[6.15] and fig[6.16]. From fig[6.15] and fig[6.16], it is clear that these two transfer functions are not the same especially around 4 samples per cycle. At 4 samples per cycle the two transfer function are completely different but at 2 samples per cycle and ∞ samples per cycle (DC signal) they become equal. However, the values of the transfer functions at these two points are zero which means that the received signal have no frequency components at DC or at a frequency corresponding to 2 samples per cycle. This explains the oscillatory nature of the impulse response shown in fig[6.13] and fig[6.14].

6.3.3 Results

In fig[6.17] and fig[6.18], The transfer functions for the cases where the receivers are at point (60,60) (the diagonal receiver) and at point (64,50) (the axial receiver) are shown on the same

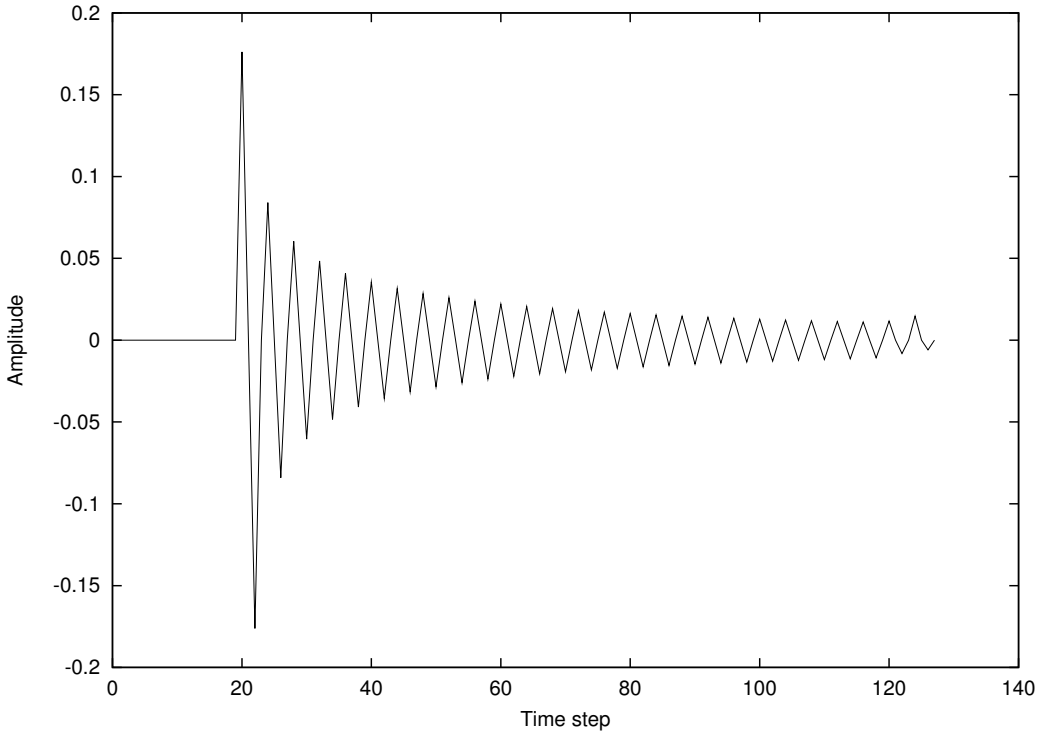


Figure 6.14: *The received signal at the diagonal receiver.*

graph. By studying fig[6.17] and fig[6.18] the following results can be found:

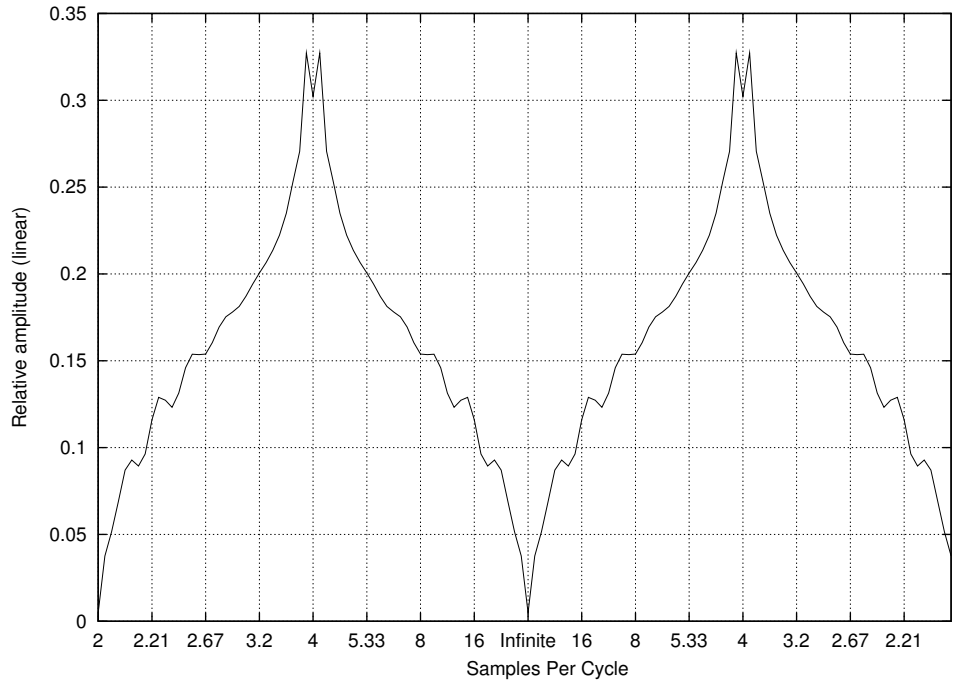
The system response when the sampling rate is 2 samples per cycle: The system response for 2 samples per cycle is 0.

The system response to DC signal: The TLM response to a DC signal is 0. The DC signal corresponds to sampling rate of ∞ samples per cycle.

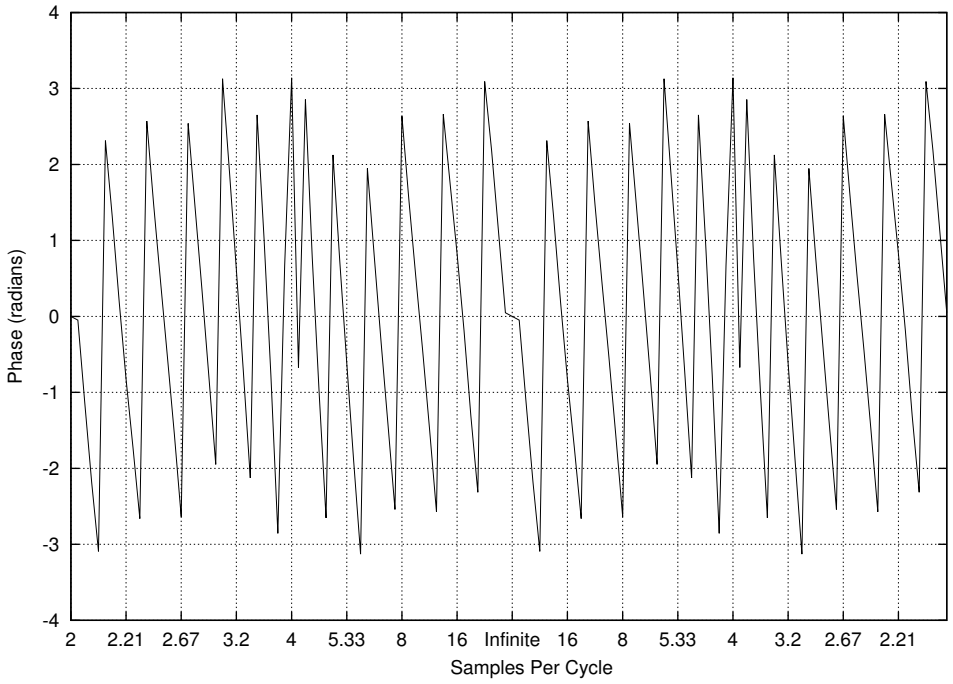
Response to 4 samples per cycle: The transfer functions for 4 samples per cycle are very different.

Sampling rate is higher than 4 samples per cycle: As the sampling rate is increased, the two transfer functions become more similar to each other. This means that for sampling rates much higher than 4 samples per cycle, the TLM mesh is a good model for wave propagation. This result was shown by Johns method too.

Sampling rate is near 2 samples per cycle: When the sampling rate is near 2 samples per cycle, the response of two transfer functions converge to become essentially the same. This mean that we can use TLM modelling with sampling rate approaching 2 samples per cycle .

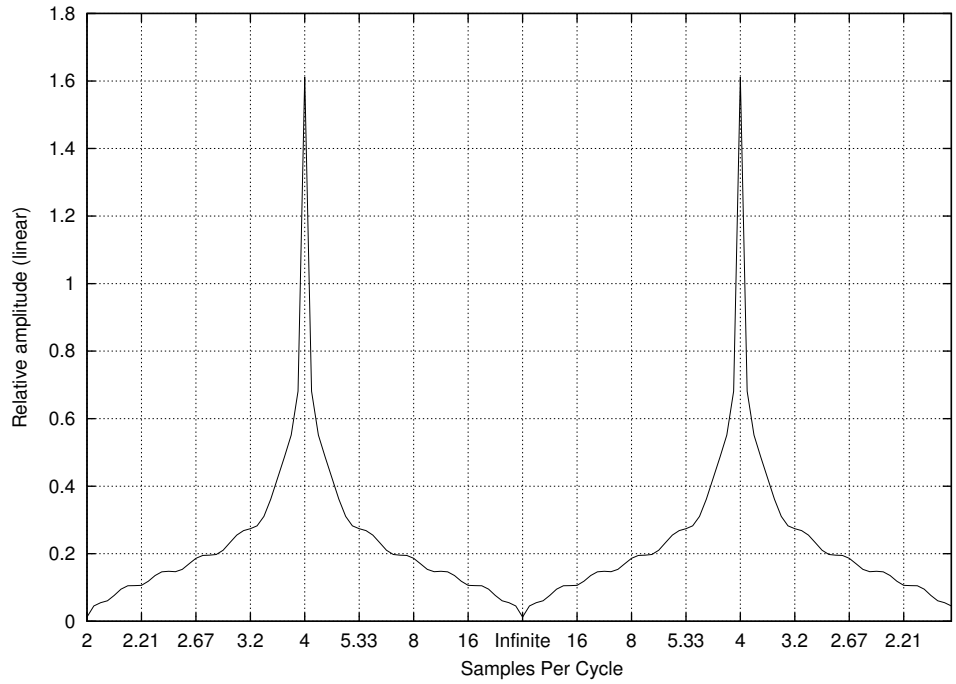


(a) Amplitude

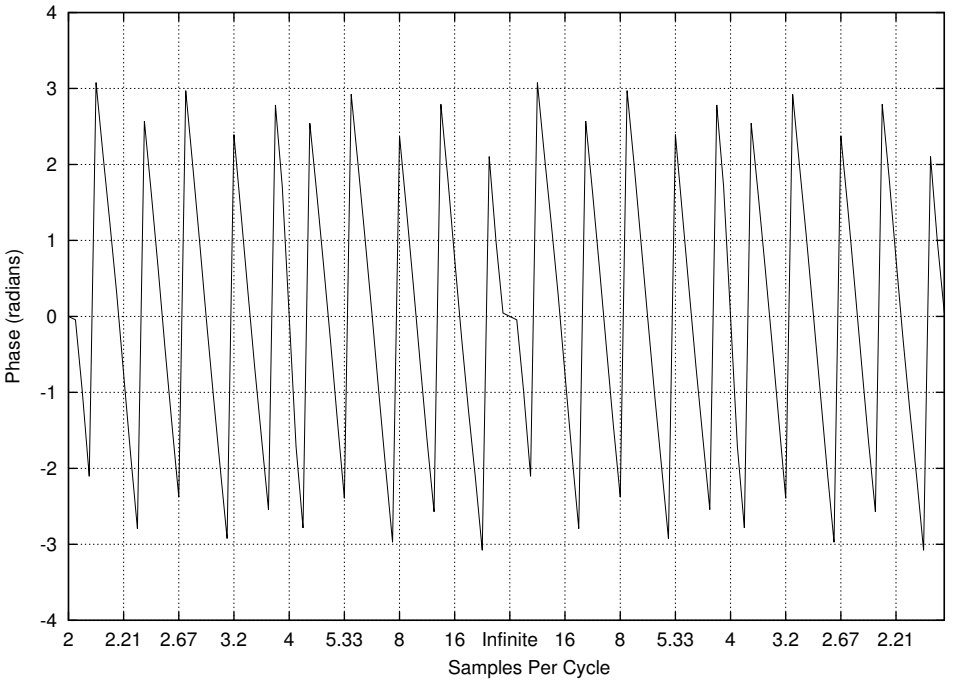


(b) Phase

Figure 6.15: *Frequency domain transfer function when the source is at (50,50) and the receiver is at (50,64). Note that the X axis converted to samples per cycle.*



(a) Amplitude



(b) Phase

Figure 6.16: *frequency domain transfer function when the source is at (50,50) and the receiver is at (60,605). Note that the X axis converted to samples per cycle.*

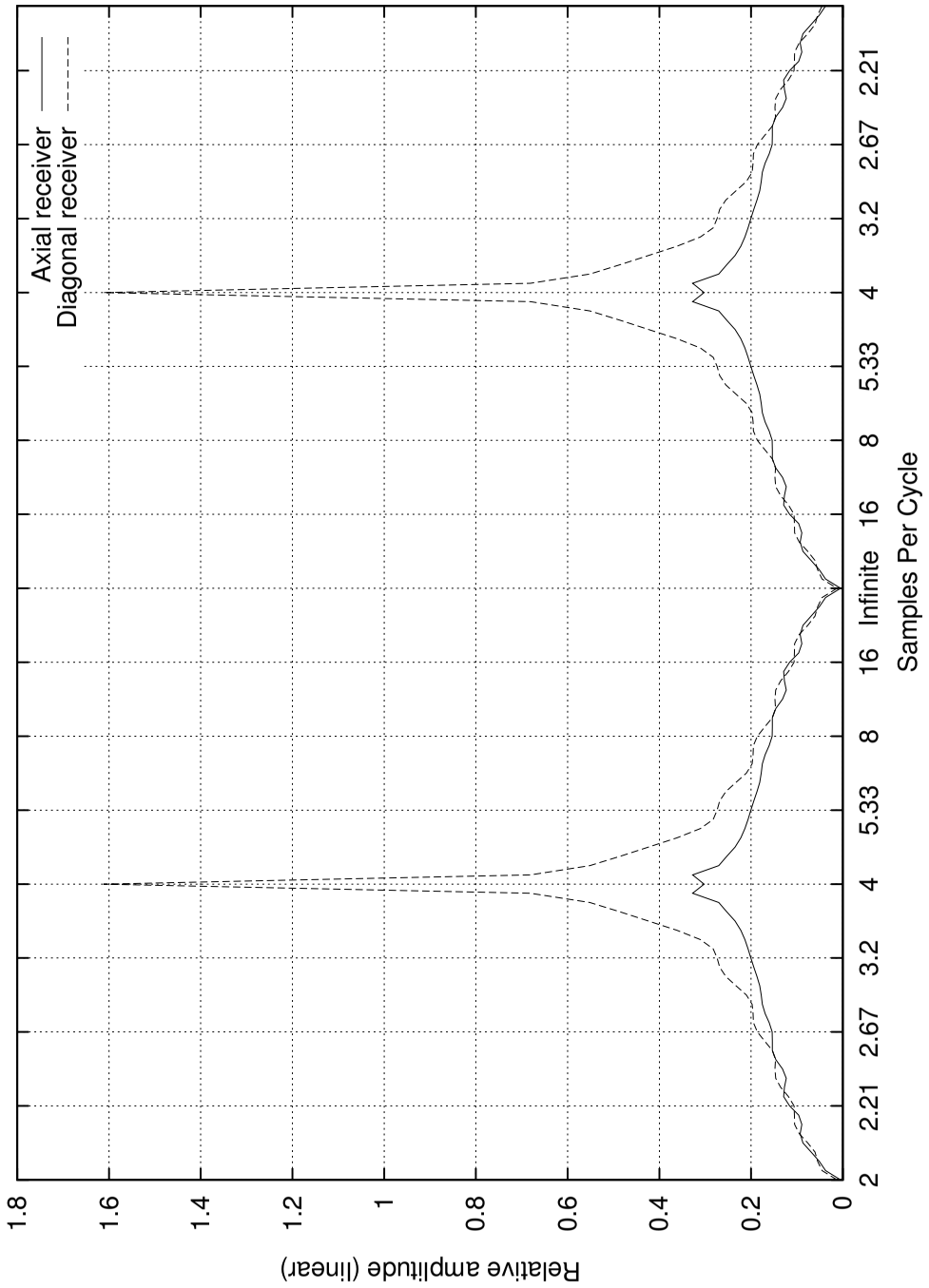


Figure 6.17: *Amplitude for both transfer functions in frequency domain.*

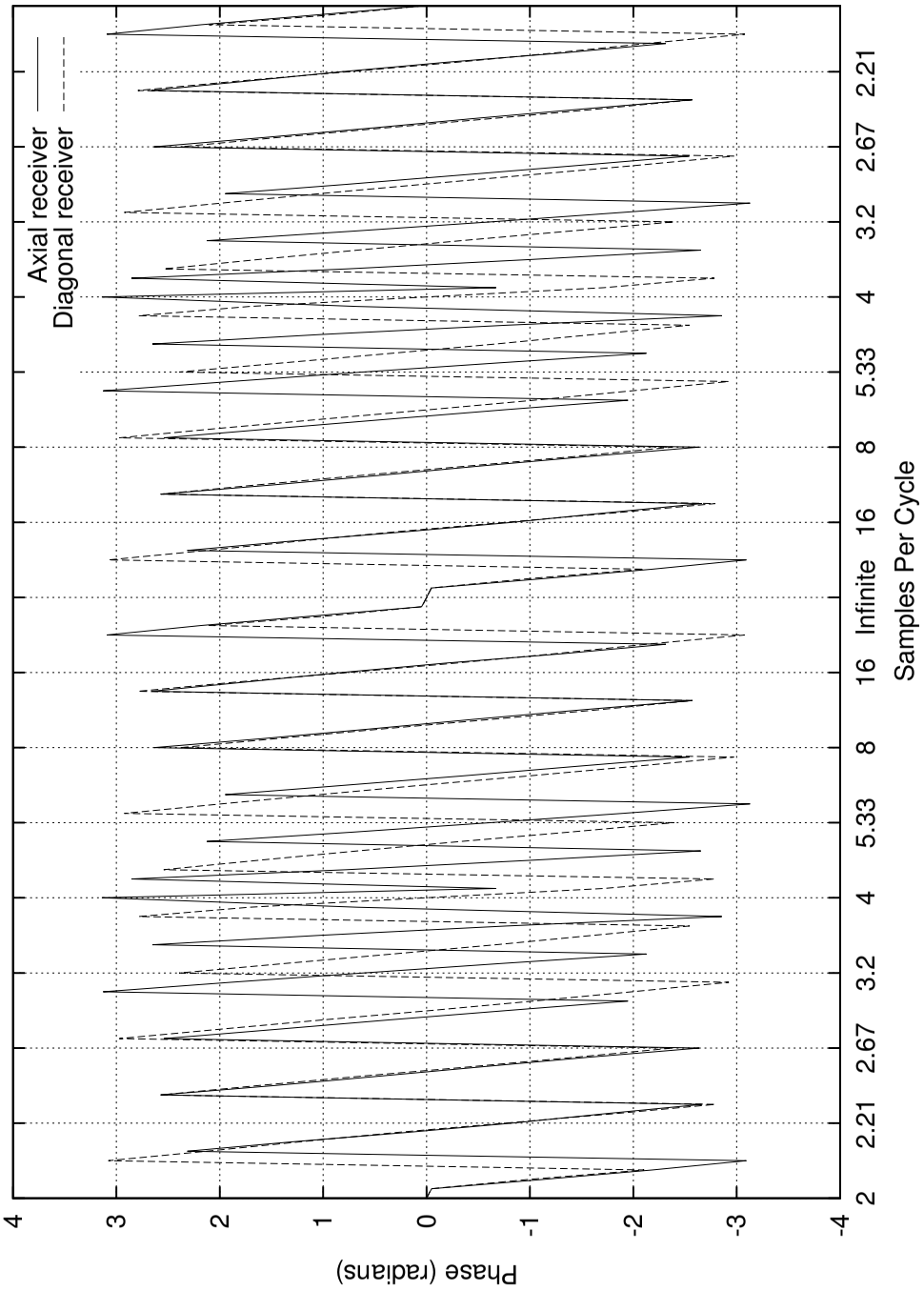


Figure 6.18: *Phase for both transfer functions in frequency domain.*

6.4 New sampling theory

Based on the experiments in the last section, The following sampling theory for TLM mesh can be stated:

> As a digital system, TLM modelling obeys all of the digital signal and system theory. This means that it can work with any signal as long as the sampling rate of the signal obeys sampling rate theory. Based on sampling rate theory, any signal can be regenerated from its samples when the sampling rate is higher than 2 sample per cycle for its higher frequency component.
>
> The TLM mesh behaves as a 2 dimension [15] digital filter. This digital filter has different transfer function which depend on where the receiver is located relative to the source position.
>
> For TLM to be a good model for wave propagation, all of the transfer functions that one can generate by placing receivers at different places should be the same. The transfer functions for these systems [16] are equal only when ΔT and hence $\Delta \ell$[17] the size of the mesh in the TLM, is selected so that one of the following criteria is met:
>
> 1. Sampling rate is much higher than 4 samples per cycle
> 2. Sampling rate is less than 4 sample per cycle and near 2 samples per cycle (One should remember that the TLM response to the system with 2 samples per cycle is 0).

6.4.1 Equal error sampling rates

As can be seem from fig[6.17], for each sampling rate which is higher than 4 samples per cycle, there is a corresponding sampling rate below 4 samples per cycle that has the same amount of error. These two sampling rate would be called "Equal error sampling rates". To find the relation between these two sampling rates, it should be noted that these two sampling rates are symmetrically placed on the frequency axis with respect to the 4 samples per cycle point. If the frequency related to 4 samples per cycle will be called as f_4 and the frequencies that related to these two sampling rates will be called as f_L and f_H[18], we have:

$$f_H - f_4 = f_4 - f_L \tag{6.9}$$

[15] 2 dimension TLM modelling is supposed here. This theory can be extended to 3 dimension simply.

[16] When the receiver places at different points in the mesh.

[17] remember that $C \times \Delta T = \Delta \ell$ when C is the speed of wave in the real medium.

[18] f_h is the high frequency with low sampling rate which is near 2 sample per cycle and f_L is low frequency with high sampling rate (more than 4 sample per cycle)

For any frequency we can write :

$$f = \frac{1}{S \times \Delta T} \tag{6.10}$$

where S is the sampling rate of the frequency f in sample per cycle. Placing the value of different frequencies from equation 6.10 into equation 6.9,and noting that the sampling rate for f_4 is 4 samples per cycle, we have:

$$\begin{aligned}
\frac{1}{S_H \times \Delta T} - \frac{1}{4 \times \Delta T} &= \frac{1}{4 \times \Delta T} - \frac{1}{S_L \times \Delta T} \\
\frac{1}{S_H} - \frac{1}{4} &= \frac{1}{4} - \frac{1}{S_L} \\
\frac{4 - S_H}{4 \times S_H} &= \frac{S_L - 4}{4 \times S_L} \\
4S_L(4 - S_H) &= 4S_H(S_L - 4) \\
S_L(4 - S_H) &= S_H(S_L - 4) \\
4S_L - S_H S_L &= S_H S_L - 4S_H \\
4S_L &= 2S_H S_L - 4S_H \\
2S_L &= S_H S_L - 2S_H \\
2S_L &= S_H(S_L - 2)
\end{aligned} \tag{6.11}$$

Solving equation 6.11 to find S_H, equation 6.12 could be found.

$$S_H = \frac{2S_L}{S_L - 2} \tag{6.12}$$

In the same one can find:

$$S_L = \frac{2S_H}{S_H - 2} \tag{6.13}$$

With this equation (6.13) one can calculate the sampling rate near 2 samples per cycle which generates the same amount of error as from the equal error sample rate above 4 samples per cycle. For example, one can use a TLM for modelling wave propagation for a signal with 2.08

sample per cycle instead of 50 samples per cycle, since[19]:

$$\begin{aligned} S_L &= \frac{2S_H}{S_H - 2} \\ S_L &= \frac{2 \times 50}{50 - 2} \\ S_L &= \frac{100}{48} \\ S_L &= 2.08333 \end{aligned} \tag{6.14}$$

6.5 Testing the new sampling rate theory

To test the new sampling rate theory, sample modelling in sec 6.1.1 was repeated with the appropriate sampling rate less than 4 samples per cycle.

6.5.1 Case1: Sampling rate of 3.333 sample per cycle

The Sampling rate 3.333 has the same amount of error as the sampling rate of 5 samples per cycle. In fig[6.19] the received signals at diagonal and axial receivers are shown. The corresponding wave propagation shape is shown in fig[6.20]

6.5.2 Case 2: Sampling rate is 2.222 sample per cycle

The Sampling rate 2.222 has the same amount of error as the sampling rate of 20 samples per cycle. In fig[6.21] the received signal at diagonal and axial receivers is shown. The corresponding wave propagation shapes is shown in fig[6.22]

6.6 Using new sampling rate for modelling wide band waves

If the injected signal to the TLM is not single tone signal, then making sure that the sampling rate for all frequency in the signal is near 2 samples per cycle is difficult and sometimes impossible. There are three important case:

[19]Since the subscript can be interchanged in equation 6.12 and 6.13 to go either one of these equations to the other one, then it doesn't matter one regards S_H as the sampling rate per cycle for higher frequency or as denoting a high sampling rate. Since the later definition is clearer it will be used here.

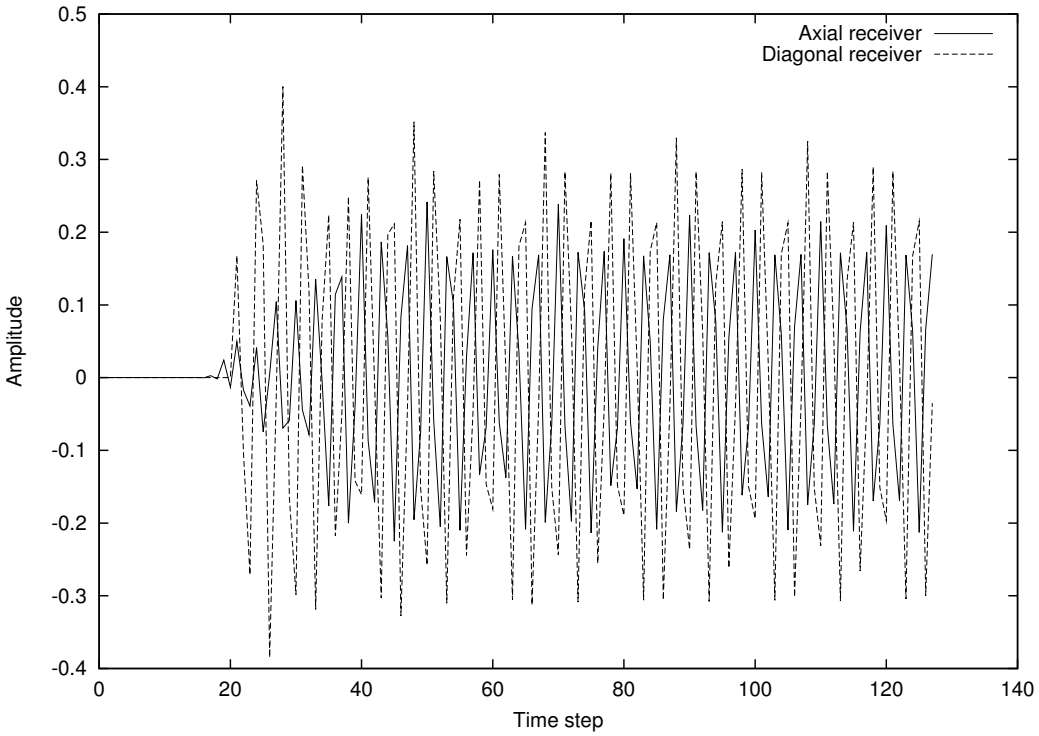


Figure 6.19: *Sampling rate is 3.333.*

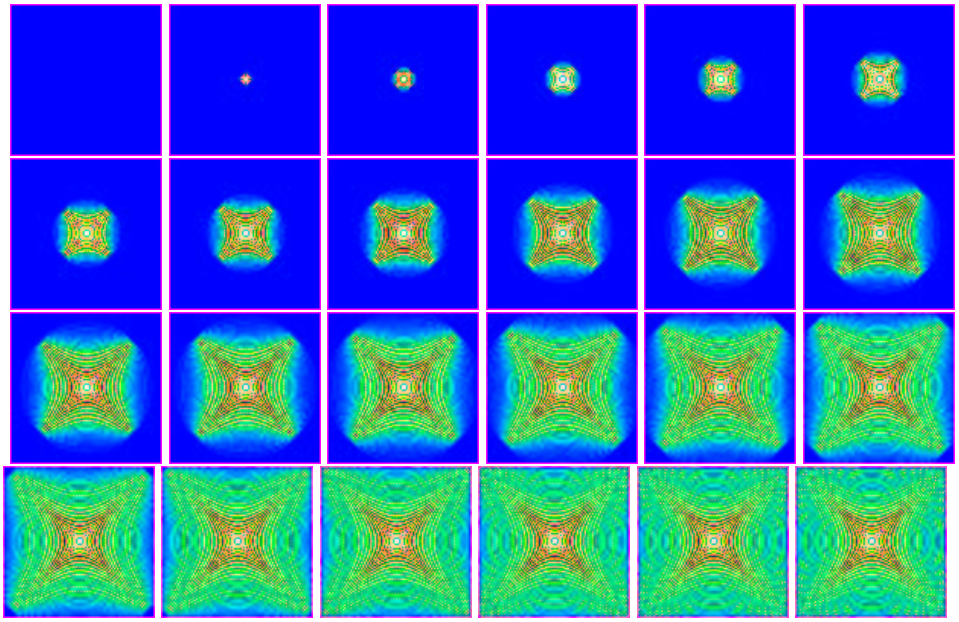


Figure 6.20: *Wave propagation shape when the sampling rate is 3.333.*

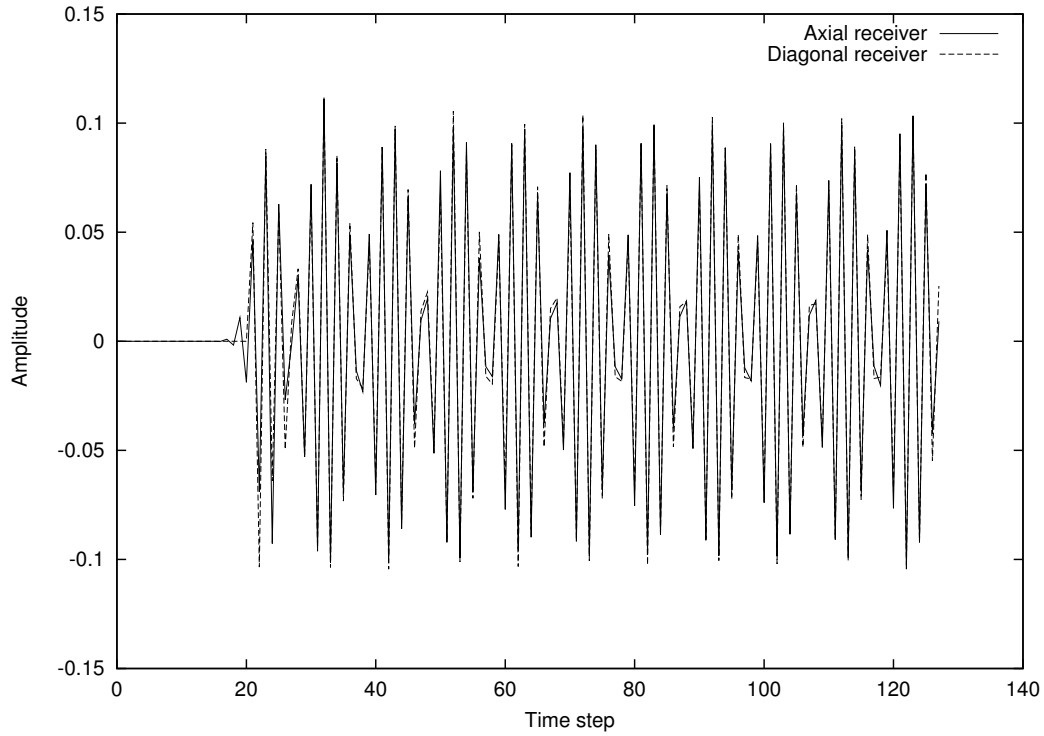


Figure 6.21: *Sampling rate is 2.222.*

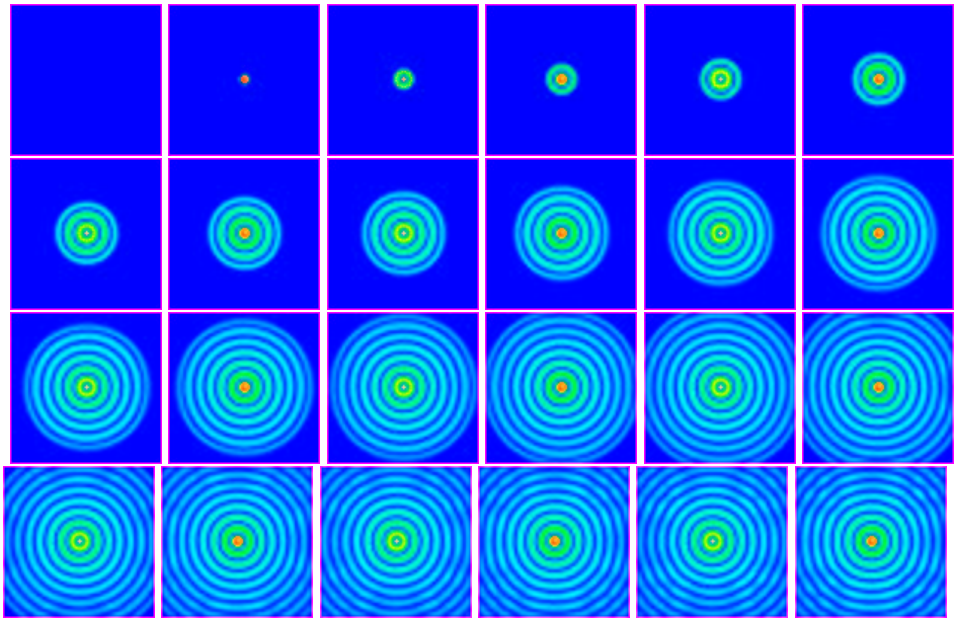


Figure 6.22: *Wave propagation shape when the sampling rate is 2.222.*

1. Signal band is than one tenth of the minimum frequency in the signal.

   In this case TLM can be used with low sample rate.

2. Signal band is between one tenth of the minimum frequency and 1000 times of the minimum frequency.

   In this case the best way to use TLM modelling is to break the signal to some smaller bands and model each band in a different TLM mesh with low sampling rate. The result would be the sum of these modelling processes [20]. In this case, there is some reduction in the processing that would be done since for example if the signal be modelled by using 50 samples per cycle, instead of 2.08 samples per cycle, there are $(\frac{50}{2.08})^3 \simeq 13891$ times more calculations compared to when it is modelled by 2.08 samples per cycle. But if it is break down to 10000 different TLM modelling[21], the saving in breaking down different TLM meshes are $\frac{13891}{10000} \simeq 1.4$ times less computation.

3. Signal band is more than 1000 times of the minimum frequency.

   The best way to model such signals is to use higher sampling rate and one TLM mesh.

6.6.1 Using new sampling rate for modelling ultrasound wave propagation in medical systems

Since most of the applications of ultrasound in medical systems falls to the second category in the previous section then the best way to model the ultrasound wave propagation in medical systems is to break the signal to some narrow band signals and model each of them with a different TLM and add the result to each other.

6.6.2 Using new sampling rate for modelling wave propagation in nonlinear medium

Since breaking the signal to some narrow band signals and doing different modelling and adding the results to each other is only acceptable for linear systems, for nonlinear systems, if the signal band width is larger than 0.1 of minimum frequency in the signal, the only way that one can model them is to use high frequency sampling rate.

[20] It is based on the superposition concept and is only acceptable for linear medium

[21] when the original signal band is 1000 times of the minimum frequency and it is breaks down to signals that the signal bands are 0.1 of the minimum frequency in the signal

6.7 Conclusion

In this chapter it was shown that it is possible to use TLM with less than 4 samples per cycle. The Johns theory on sampling rate for TLM modelling isn't complete and it doesn't show the possibility of modelling the TLM with about 2 samples per cycle. By using 2 samples per cycle for modelling wave propagation, it is possible to model more realistic ultrasound problems with TLM in today's computers. The next example clearly illustrate this:

Example: We have a medium where the speed of sound is 1500 m/s (metre per second)[22]. We want to model ultrasound wave propagation in the medium. The frequency of ultrasound is 1MHz and the size of the medium is $10 \times 10 \times 10$ cm. The modelling time[23] is 7×10^{-5} . The wavelength of the signal in the medium is :

$$\begin{aligned} \lambda &= \frac{C}{f} \\ \lambda &= \frac{1500}{10^6} \\ \lambda &= 1.5 \times 10^{-4} m \end{aligned} \tag{6.15}$$

case 1: Sampling rate of 50 samples per cycle The number of nodes (NON) in each X,Y,Z axis should be:

$$\begin{aligned} \Delta\ell &= \frac{\lambda}{50} \\ \Delta\ell &= \frac{1.5 \times 10^{-4}}{50} \\ \Delta\ell &= 3 \times 10^{-6} \end{aligned}$$

$$\begin{aligned} NON &= \frac{L}{\Delta\ell} \\ NON &= \frac{.1}{3 \times 10^{-6}} \\ NON &= 3.3 \times 10^4 \end{aligned} \tag{6.16}$$

[22] Approximately the speed of sound in body (water) [48], [38]
[23] Time for wave to propagate 10 cm.

The total number of nodes (TNON) is

$$\begin{aligned} TNON &= NON^3 \\ TNON &= (3.3 \times 10^4)^3 \\ TNON &= 3.5937 \times 10^{13} \end{aligned} \tag{6.17}$$

We need 32 byte of memory for each node, then the total amount of memory that need to model this system is:

$$\begin{aligned} MEM &= 32 \times TNON \\ MEM &= 32 \times 3.5937 \times 10^{13} \\ MEM &= 1.149984 \times 10^{16} \end{aligned} \tag{6.18}$$

The memory that is needed to model this system with 50 samples per cycle 1.149984×10^7 Giga Bytes! The time steps (TS) for modelling is:

$$\begin{aligned} \Delta T &= \frac{\Delta \ell}{C} \\ \Delta T &= \frac{3 \times 10^{-6}}{1500} \\ \Delta T &= 2 \times 10^{-9} \end{aligned}$$

$$\begin{aligned} TS &= \frac{Time}{\Delta T} \\ TS &= \frac{7 \times 10^{-5}}{2 \times 10{-}9} \\ TS &= 3.5 \times 10^4 \end{aligned} \tag{6.19}$$

The processing for each node consist of 4 additions and 4 multiplications. For simplicity we assume that the time for multiplication and addition are the same, it takes 8 instruction

to process a node. The total precessing for a time step (TPTS) is:

$$\begin{aligned} TPTS &= 8 \times TNON \\ TPTS &= 8 \times 3.5937 \times 10^{13} \\ TPTS &= 2.87496 \times 10^{13} \end{aligned} \tag{6.20}$$

The total processing (TP) is:

$$\begin{aligned} TP &= TPTS \times TS \\ TP &= 2.87496 \times 10^{13} \times 3.5 \times 10^{4} \\ TP &= 1.006236 \times 10^{19} \end{aligned} \tag{6.21}$$

Today PCs can deliver more than 1 giga instructions per sec[24]. Time request (TR) to model this system is:

$$\begin{aligned} TR &= \frac{TP}{10^9} \\ TP &= \frac{1.006236 \times 10^{19}}{10^9} \\ TP &= 1.006236 \times 10^{10} \end{aligned} \tag{6.22}$$

This time is around 100 year!

Case 2: Sampling rate of 2.08 sample per cycle The number of nodes (NON) in each X,Y,Z

[24]This number is an approximate value.

axis should be:

$$\begin{aligned} \Delta\ell &= \frac{\lambda}{2.08} \\ \Delta\ell &= \frac{1.5 \times 10^{-4}}{2.08} \\ \Delta\ell &= 7.2 \times 10^{-5} \end{aligned}$$

$$\begin{aligned} NON &= \frac{L}{\Delta\ell} \\ NON &= \frac{.1}{7.2 \times 10^{-5}} \\ NON &= 1.4 \times 10^{3} \end{aligned} \tag{6.23}$$

The total number of nodes (TNON) is

$$\begin{aligned} TNON &= NON^3 \\ TNON &= (1.4 \times 10^3)^3 \\ TNON &= 2.744 \times 10^9 \end{aligned} \tag{6.24}$$

We need 32 byte of memory for each node, then the total amount of memory that needed to model this system is:

$$\begin{aligned} MEM &= 32 \times TNON \\ MEM &= 32 \times 2.744 \times 10^9 \\ MEM &= 8.7808 \times 10^{10} \end{aligned} \tag{6.25}$$

The memory that needed to model this system with 2.08 samples per cycle is about 87 Giga Byte. This amount of memory can be found on supercomputers today and should

be in the Pcs with in a few years. The time steps (TS) for modelling is:

$$\begin{aligned} \Delta T &= \frac{\Delta \ell}{C} \\ \Delta T &= \frac{7.2 \times 10^{-5}}{1500} \\ \Delta T &= 4.8 \times 10^{-8} \end{aligned}$$

$$\begin{aligned} TS &= \frac{Time}{\Delta T} \\ TS &= \frac{7 \times 10^{-5}}{4.8 \times 10{-}8} \\ TS &= 1.45 \times 10^{3} \end{aligned} \tag{6.26}$$

The processing for each node consist of 4 additions and 4 multiplications. For simplicity we assume that the time for multiplication and addition are the same, it takes 8 instructions to process a node. The total precessing for a time step (TPTS) is:

$$\begin{aligned} TPTS &= 8 \times TNON \\ TPTS &= 8 \times 2.744 \times 10^{9} \\ TPTS &= 2.1952 \times 10^{10} \end{aligned} \tag{6.27}$$

The total processing (TP) is:

$$\begin{aligned} TP &= TPTS \times TS \\ TP &= 2.1952 \times 10^{10} \times 1.45 \times 10^{3} \\ TP &= 3.18304 \times 10^{13} \end{aligned} \tag{6.28}$$

Today PCs can deliver up to 10^9 instruction per sec[25]. Time request (TR) to model this

[25]This number is an approximate value.

	Sampling rate is 50	Sampling rate is 2.08
Memory	1.149984×10^7 Giga Byte	87 Giga Byte
Time	About 100 years	About 8 hours

Table 6.2: *Comparison between different sampling rate for modelling system explained in the example.*

system is:

$$\begin{aligned} TR &= \frac{TP}{10^9} \\ TP &= \frac{3.18304 \times 10^{13}}{10^9} \\ TP &= 3.18304 \times 10^4 \end{aligned} \tag{6.29}$$

This time is about 8 hours.

This result shows in table[6.2].

Chapter 7
Conclusion and further work

7.1 Introduction

This chapter summarises the main findings of this thesis and the work that is reported herein. The aim of this work was to use TLM for modelling medical ultrasound wave propagation.

It was shown that when TLM is used for modelling medical ultrasound wave propagation, there is a need for a huge amount of computational power. This problem is the direct result of the existing sampling theory for TLM modelling. To solve the problem, a new sampling theory was found for TLM modelling.

In the following sections, the main results of this work are presented and some ideas for further work are explained.

7.2 The main results of this thesis

The main results of this thesis could be explained as follows:

7.2.1 TLM for modelling ultrasound wave propagation

TLM was originally designed for modelling electromagnetic waves propagation. In this thesis, the use of it for modelling ultrasound waves propagation was shown.

7.2.2 Using TLM to solve some modelling problems in medical ultrasound wave propagation

TLM was used to model ultrasound wave propagation and, in particular, array transducers and Doppler effect.

7.2.3 New sampling rate theory

It was shown that the original sampling rate that was presented by Johns, is not complete and a new theory for sampling rate is explained. The new sampling rate theory shows the Johns sampling rate theory as one part of a more complete solution. The new sampling rate can be used in modelling ultrasound wave propagation as well as modelling electromagnetic wave propagation. By using this new sampling rate theory, the computational complexity of TLM modelling can be reduced dramatically.

7.3 Further work

There are several area that further research can be done.

7.3.1 TLM as a digital filter

It was shown that a TLM mesh can be modelled as two dimensional digital filter. The effect of this two dimensional filter should be investigated more detail.

7.3.2 Removing boundary effect

It was shown that by modelling TLM as a digital filter, it is possible to design a match filter to terminate the TLM mesh and remove boundary effects. Some simple FIR filters were used to approximate the match filter. Some more complex (FIR and IIR) filters could be designed to terminate the TLM medium.

7.3.3 Doppler effect in some complex situations

The Doppler effect in some situations was presented in this thesis. Further research can be done on using this technique on more complex situations (For example when source and receiver are stationary but there are several objects in the medium that are moving[1]).

[1]The result of this experiment can be used for modelling ultrasound systems that use Doppler effect to detect the blood velocity.

7.3.4 New type of meshes

New type of meshes would generate other types of digital filters. These new types of digital filters could have better transfer functions from the point of modelling wave propagation.

7.4 Conclusion

It was shown that TLM is a good numerical technique for modelling medical ultrasound wave propagation. A new digital filter model for TLM mesh was proposed and explained. Based on this digital filter model, a new way to terminate the TLM mesh to a matching media is presented.

Since TLM was originally designed for modelling electromagnetic wave propagation, the relationship between electromagnetic waves and sound waves was presented so the TLM can be used for modelling ultrasound wave propagation too.

Several modelling results were presented to show how TLM could be use to model some interesting wave propagation problems.

It was shown how TLM can be used to model curved surfaces, such as a focusing mirror and a circular mirror.

TLM was used to model wave steering in an array transducer. It was also shown how to use TLM to model the Doppler effect.

It was also shown that the sampling rate for TLM modelling is not complete and a new sampling rate for TLM modelling was presented.

Appendix A
Original publication

This paper published in the proceeding of :
”First International Conference on Advances in Medical Signal and Information Processing” organized by IEEE and held in Bristol, UK on September 2000 [98].

Appendix B
Developers guide to TLM classes

B.1 Abstract

In this document it is explained how to use classes in TLM library to write a TLM model.

B.2 Introduction

TLM library is a collection of classes and functions for writing TLM modelling in C++. The classes in this library consist of the following classes:

CTlmBasic:The basic class which take care of memory management and processing nodes and changing data between nodes.This class by it self can't be used.

CTlmPPM:This class is based on CTlmBasic and add the functionality of saving each frame in a PPM files to the CTlmBasic. This class can't be used by itself.

CTlmSimpleThe simplest TLM model. This class is the simplest functional TLM model.

B.3 CTlmBasic

This is the base for all other TLM classes.The main tasks for this class are:

- Allocate the necessary memory for storing node data
- De-allocating memory at exit.
- Do the iteration and call necessary function for updating node data and getting data from sources.
- Managing Node data

B.3.1 Definition:

```
// TlmBasic.h: interface for the CTlmBasic class.
//
//////////////////////////////////////////////////////////////////////

#if !defined(AFX_TLMBASIC_H__C45F5544_8065_11D2_837D_0000B4731445__INCLUDED_)
#define AFX_TLMBASIC_H__C45F5544_8065_11D2_837D_0000B4731445__INCLUDED_

#if _MSC_VER >= 1000
#pragma once
#endif // _MSC_VER >= 1000
#include <math.h>
#include <dsp.h>


typedef enum {
DIR_PX=0,// positive X
DIR_NX,// negative X
DIR_PY,// positive Y
DIR_NY // negative Y
}TDIR;




class CTlmBasic
{

public:
TNODE NodeValue(int X,int Y);
virtual TNODE NodeIntensity(int X,int Y);
virtual TNODE MaxFrameIntensity();
```

```
virtual bool Init(int XDim,int YDim,int Time);
CTlmBasic();
virtual ~CTlmBasic();
virtual bool DoModel();

protected:
int m_Time;
virtual bool HandleBoundry(int X,int Y);
virtual bool ChangeData();
bool m_Aborted;
virtual bool PreProcessFrame (int Time);
virtual bool PostProcessFrame (int Time);

virtual bool PreProcessNode (int X,int Y,int Time);
virtual bool ProcessNode (int X,int Y,int Time);
virtual bool PostProcessNode(int X,int Y,int Time);


virtual bool InitMesh();
int m_TotalTime;
int m_YDim;
int m_XDim;
TNODE NodeData(int X,int Y,TDIR Dir);
TNODE NodeData(int X,int Y,TDIR Dir, TNODE Data);
private:
TNODE * m_NodeData;
};

#endif // !defined(AFX_TLMBASIC_H__C45F5544_8065_11D2_837D_0000B4731445__I
```

B.3.2 Base class

CTlmBasic hasn't any base class.

B.3.3 Data types

TNODE:This is the type of node data. if more accuracy is needed, TNODE maybe changed to double.

TDIR: This enum type represent the 4 direction around a 2D node.

B.3.4 Global variables:

CTlmBasic hasn't any global variable which it's user can change or use it.

B.3.5 Macros:

PI : PI defined as the value of π (3.14159265358979323846264338327 95).

B.3.6 Functions:

virtual TNODE NodeIntensity(int X,int Y): Return the value for intensity of a node. In it's simplest format it return the absolute value for sum of four voltage in a 2D node.

virtual TNODE MaxFrameIntensity(): Return the maximum intensity for a node in a frame. Thsi data maybe used for normalizing each frame.

virtual bool DoModeling(bool Flag): This function do the actual modeling. This function will not return until modeling finished or abort function called. if Flag set to true, then Do Modeling will create a new thread and start to run on the new thread, otherwise (Flag=false) there isn't any new thread and the modeling will be done in the current thread.

virtual bool Init(int XDim,int YDim,int Time): Initialize a TLM mesh. XDim and YDim are mesh size and Time is the total time that the modeling will be run.

CTlmBasic(): Constructor, do nothing.

virtual CTlmBasic(): Destructor, delete memory allocated for modeling.

virtual bool HandleBoundry(int X,int Y): Fter calculating each node and on changing data with other nodes, Nodes which are on boundary must be processed in a espicial way.

When the ChangeData() function find such nodes, this function will be called to process them.When investigating boundary effects, you may override this function.

virtual bool ChangeData(): For changing data between nodes. After calculating voltages for each node, DoModeling() call this function to change data with neighboring nodes. Normally there isn't any need to override this function.

virtual bool PreProcessFrame (int Time): This function is called by DoModeling() each time processing for a new frame is started. This function do nothing by itself.

virtual bool PostProcessFrame (int Time): This function is called by DoModeling()after the processing for each frame finished.This function do nothing by itself.

virtual bool PreProcessNode (int X,int Y,int Time): This function is called before processing for each node started. This function do nothing by itself.

virtual bool ProcessNode (int X,int Y,int Time): Processing for each node must be done in this function.Since CTlmBasic is the base for all TLM classes, Nothing is done in this function.[1]

virtual bool PostProcessNode(int X,int Y,int Time); This function is called after the processing for each node finished.

virtual bool DoModeling(): This is the actual modeling function. DoModeling(bool Flag) call this function directly or indirectly based on the value of Flag.There is no need that you call this function.

virtual bool InitMesh(): Initialise TLM mesh to zero before starting modeling. If you want to initialize the mesh to something else other than zero you may override it.

TNODE NodeData(int X,int Y,TDIR Dir): For reading specific voltage from a node.

TNODE NodeData(int X,int Y,TDIR Dir, TNODE Data): For writing a specific voltage to a node.

B.3.7 Implementation:

[1] CTlmSimple use this function to implement a simple medium.

```
// TlmBasic.cpp: implementation of the CTlmBasic class.
//
//////////////////////////////////////////////////////////////////////
#include <malloc.h>
#include <stdio.h>


#include "TlmBasic.h"

#define RETURN_ON_FAIL(x) { if(x==false) return false;}

//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

CTlmBasic::CTlmBasic()
{

}

CTlmBasic::~CTlmBasic()
{
delete [] m_NodeData;
}

bool CTlmBasic::Init(int XDim, int YDim,int Time)
{
m_XDim=XDim;
m_YDim=YDim;
m_NodeData=new TNODE[m_XDim*m_YDim*4];
m_TotalTime=Time;
return true;
}
```

```
TNODE CTlmBasic::NodeData(int X, int Y,TDIR Dir)
{
return m_NodeData[(((X*m_YDim)+Y)*4)+Dir];
}


TNODE CTlmBasic::NodeData(int X, int Y,TDIR Dir, TNODE Data)
{
TNODE Buffer=m_NodeData[(((X*m_YDim)+Y)*4)+Dir];
m_NodeData[(((X*m_YDim)+Y)*4)+Dir]=Data;
return Buffer ;
}



bool CTlmBasic::InitMesh()
{
for(int X=0;X<m_XDim;X++)
{
for(int Y=0;Y<m_YDim;Y++)
{
NodeData(X,Y,DIR_PX,0);
NodeData(X,Y,DIR_NX,0);
NodeData(X,Y,DIR_PY,0);
NodeData(X,Y,DIR_NY,0);
}
}
return true;
}
```

```
bool CTlmBasic::DoModel()
{
m_Aborted=false;
RETURN_ON_FAIL(InitMesh());
for(m_Time=0;m_Time<m_TotalTime;m_Time++)
{
printf("Processing frame %d\n",m_Time);
RETURN_ON_FAIL(PreProcessFrame(m_Time));
for(int X=0;X<m_XDim;X++)
{
for(int Y=0;Y<m_YDim;Y++)
{
RETURN_ON_FAIL(PreProcessNode(X,Y,m_Time));
RETURN_ON_FAIL(ProcessNode(X,Y,m_Time));
RETURN_ON_FAIL(PostProcessNode(X,Y,m_Time));
}
}
RETURN_ON_FAIL(PostProcessFrame(m_Time));
ChangeData();
if(m_Aborted==true) return false;
}

return true;
}

bool CTlmBasic::PreProcessFrame(int Time)
{
return true;

}
bool CTlmBasic::PostProcessFrame(int Time)
```

```
{
return true;

}
bool CTlmBasic::PreProcessNode(int X,int Y,int Time)
{
return true;

}
bool CTlmBasic::ProcessNode(int X,int Y,int Time)
{
return true;

}
bool CTlmBasic::PostProcessNode(int X,int Y, int Time)
{
return true;

}



TNODE CTlmBasic::MaxFrameIntensity()
{
TNODE Buffer,MaxData;
for(int Y=0;Y<m_YDim;Y++){
for(int X=0;X<m_XDim;X++){
Buffer=NodeIntensity(X,Y);
if (MaxData<Buffer){
MaxData=Buffer;
}
}
}
```

```
return MaxData;
}

TNODE CTlmBasic::NodeIntensity(int X, int Y)
{
TNODE Buffer=(NodeData(X,Y,DIR_PX)+NodeData(X,Y,DIR_NX)+NodeData(X,Y,DIR_PY)+Nod
return (TNODE)fabs((double)Buffer);
}

bool CTlmBasic::ChangeData()
{
TNODE Buffer;
for(int X=0;X<m_XDim;X++)
{
for(int Y=0;Y<m_YDim;Y++)
{
if((X!=0) & (Y!=0) & (X!=m_XDim-1) & (Y!=m_YDim-1)) // not on boundry
{
Buffer=NodeData(X,Y,DIR_PX,NodeData(X+1,Y,DIR_NX));
NodeData(X+1,Y,DIR_NX,Buffer);

// Buffer=NodeData(X,Y,DIR_NX,NodeData(X-1,Y,DIR_PX));
// NodeData(X-1,Y,DIR_PX,Buffer);

Buffer=NodeData(X,Y,DIR_PY,NodeData(X,Y+1,DIR_NY));
NodeData(X,Y+1,DIR_NY,Buffer);

// Buffer=NodeData(X,Y,DIR_NY,NodeData(X,Y-1,DIR_PY));
// NodeData(X,Y-1,DIR_PY,Buffer);
}
else
{
HandleBoundry(X,Y);
```

```
}
}
}
return true;
}


bool CTlmBasic::HandleBoundry(int X, int Y)
{
if(X==0)
{
NodeData(X+1,Y,DIR_NX,NodeData(X,Y,DIR_PX));
// NodeData(X,Y,DIR_PX, 0.25 * NodeData(X,Y,DIR_PX));
}
else
if(X==m_XDim-1)
{
NodeData(X,Y,DIR_NX,NodeData(X-1,Y,DIR_PX));
// NodeData(X,Y,DIR_NX, 0.25 * NodeData(X,Y,DIR_NX));
}
else{
NodeData(X,Y,DIR_PX,NodeData(X+1,Y,DIR_NX));
NodeData(X,Y,DIR_NX,NodeData(X-1,Y,DIR_PX));

}

if(Y==0)
{
NodeData(X,Y+1,DIR_NY,NodeData(X,Y,DIR_PY));
// NodeData(X,Y,DIR_PY, 0.25 * NodeData(X,Y,DIR_PY));
}
else
if(Y==m_YDim-1)
```

```
{
NodeData(X,Y,DIR_NY,NodeData(X,Y-1,DIR_PY));
// NodeData(X,Y,DIR_NY,  0.25 * NodeData(X,Y,DIR_NY));
}
else
{
NodeData(X,Y,DIR_PY,NodeData(X,Y+1,DIR_NY));
NodeData(X,Y,DIR_NY,NodeData(X,Y-1,DIR_PY));
}

return true;
}

TNODE CTlmBasic::NodeValue(int X, int Y)
{
if(X==0 && Y==99)
{
return (NodeData(X,Y,DIR_PX)+NodeData(X,Y,DIR_NX)+NodeData(X,Y,DIR_PY)+NodeData(
}

return (NodeData(X,Y,DIR_PX)+NodeData(X,Y,DIR_NX)+NodeData(X,Y,DIR_PY)+NodeData(
}
```

B.3.8 Programming:

You must inherited from this class before you can use it.In inherited class do the followings:

1. Override ProcessNode() to implement the medium properties.
2. Override PreProcessFrame() or PreProcessNode() to add source signal to the medium.[2]

[2]if you have one source signal it is simpler to override PreProcessNode() and if you have several source signal it is simpler to override PreProcessFrame()

3. Override PostProcessFrame() or PostProcessNode() to save receiver signal.[3]

4. Add any other capability that you want to your class.

In the main program do the followings:

1. Crate an instance of your class.

2. Initialize the TLM mesh by calling Init().

3. Do any initialization you class needed.

4. Start to model by calling DoModelling().

5. On exiting from your program, CTlmBasic destructor will clean memories that it allocated for modeling.

B.4 CTlmPPM:

CTlmPPM add the functionality needed to write frames in PPM format. CTlmPPM doesn't save each frame by itself and the inherited function must call SaveFarem() for saving frame[4].

B.4.1 Definition:

```
// TlmPPM.cpp: implementation of the CTlmPPM class.
//
//////////////////////////////////////////////////////////////////////
#include <stdio.h>
#include <string.h>

#include "TlmPPM.h"

//////////////////////////////////////////////////////////////////////
```

[3]if you have only one receiver it is simpler to use PostProcessNode() but if you have several receiver or you want to save the data for a complete frame then you must use PostProcessFrame().

[4]This function maybe called in PostProcessFrame()

```
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

CTlmPPM::CTlmPPM()
{
m_FileNameTemplate[0]=0;

}

CTlmPPM::~CTlmPPM()
{

}
bool CTlmPPM::SaveFrame(int FrameNumber)
{
if(m_FileNameTemplate[0]==0) return true;
char FileName[255];
int Red,Blue,Green;
sprintf(FileName,m_FileNameTemplate,FrameNumber);
FILE * FPointer=fopen(FileName,"wb");
fprintf(FPointer,"P6\n");
fprintf(FPointer,"%d %d\n",m_XDim,m_YDim);
fprintf(FPointer,"255\n");
//
TNODE MaxIntensity=MaxSignal();
for(int Y=m_YDim-1;Y>=0;Y--){
for(int X=0;X<m_XDim;X++){
if(IsOnAnObject(X,Y)>=0)
{
Red=255;
Blue=255;
Green=255;
}
```

```
else
{
RGBData(X,Y,Red,Blue,Green,MaxIntensity);
// draw boundry
if(X==0 || X==m_XDim-1 || Y==0 || Y==m_YDim-1){
// Blue=255;
// Green=255;
Red=255;

}
}
fwrite(&Red,1,1,FPointer);
fwrite(&Green,1,1,FPointer);
fwrite(&Blue,1,1,FPointer);

}

}
fclose(FPointer);
return true;
}
bool CTlmPPM::SetFileTemplate(char * Template)
{
strcpy(m_FileNameTemplate,Template);
return true;

}

#define MAXGREY (255)
#define MAXRANGE (100)
#define DAVE (1)
#define PETER (2)
#define COLORSETUP DAVE
```

```
bool CTlmPPM::RGBData(int X, int Y, int & Red, int & Blue, int & Green,TNODE Max
{
#if (COLORSETUP==PETER)
TNODE Value=NodeIntensity(X,Y);
if(Value> .5* MaxFrameIntensity) Value=MaxFrameIntensity;
Green = 0;
Red =   Value * 255/MaxFrameIntensity;
Blue = (MaxFrameIntensity-Value)*255/MaxFrameIntensity;

#endif


#if (COLORSETUP == DAVE)
int Colour = (int)(MAXGREY*(NodeIntensity(X,Y)/MaxFrameIntensity));
if(Colour >MAXRANGE )
Colour=MAXRANGE;
Colour*=255.0/MAXRANGE;
// if(Colour>0)
// {
// fprintf(fp,"%d %d \t%d\t%d\n",m_Time,X,Y,Colour);
// }
    if (Colour>MAXGREY) Colour=MAXGREY;
    if (Colour<0) Colour=0;
    if (Colour<(MAXGREY/6))
{
Red = 0;  Green = Colour*6;  Blue = MAXGREY;
    } else if (Colour<(MAXGREY/3))
{
Red = 0;  Green = MAXGREY;  Blue = MAXGREY - 6*(Colour - MAXGREY/6);
    } else if (Colour<(MAXGREY*11/24))
{
Red = (Colour - MAXGREY/3)*8;  Green = MAXGREY;  Blue = 0;
```

```
} else if (Colour<(MAXGREY*7/12))
{
Red = MAXGREY;  Green = MAXGREY - 8*(Colour - MAXGREY*11/24);  Blue = 0;
    } else if (Colour<(MAXGREY*5/6))
{
Red = MAXGREY;  Green = 4*(Colour - MAXGREY*7/12);  Blue = 0;
    } else {
Red = MAXGREY;  Green = MAXGREY;  Blue = 12*(Colour - MAXGREY*5/6);
    }
    if (Blue < 0) Blue = 0;
    if (Blue > MAXGREY) Blue = MAXGREY;
    if (Red < 0) Red = 0;
    if (Red > MAXGREY) Red = MAXGREY;
    if (Green < 0) Green = 0;
    if (Green > MAXGREY) Green = MAXGREY;
#endif
return true;
}

/*bool CTlmPPM::RGBData(int X, int Y, int & Red, int & Blue, int & Green,T
{
int NodeInt=(int)(255*(NodeIntensity(X,Y)/MaxFrameIntensity));
unsigned int Palate[256]={ 0x000000,  0x003300,  0x006600,  0x009900,  0x0
0x000033,  0x003333,  0x006633,  0x009933,  0x00CC33,  0x00FF33,
0x000066,  0x003366,  0x006666,  0x009966,  0x00CC66,  0x00FF66,
0x000099,  0x003399,  0x006699,  0x009999,  0x00CC99,  0x00FF99,
0x0000CC,  0x0033CC,  0x0066CC,  0x0099CC,  0x00CCCC,  0x00FFCC,
0x0000FF,  0x0033FF,  0x0066FF,  0x0099FF,  0x00CCFF,  0x00FFFF,


0x330000,  0x333300,  0x336600,  0x339900,  0x33CC00,  0x33FF00,
0x330033,  0x333333,  0x336633,  0x339933,  0x33CC33,  0x33FF33,
0x330066,  0x333366,  0x336666,  0x339966,  0x33CC66,  0x33FF66,
```

```
0x330099,  0x333399,  0x336699,  0x339999,  0x33CC99,  0x33FF99,
0x3300CC,  0x3333CC,  0x3366CC,  0x3399CC,  0x33CCCC,  0x33FFCC,
0x3300FF,  0x3333FF,  0x3366FF,  0x3399FF,  0x33CCFF,  0x33FFFF,


0x660000,  0x663300,  0x666600,  0x669900,  0x66CC00,  0x66FF00,
0x660033,  0x663333,  0x666633,  0x669933,  0x66CC33,  0x66FF33,
0x660066,  0x663366,  0x666666,  0x669966,  0x66CC66,  0x66FF66,
0x660099,  0x663399,  0x666699,  0x669999,  0x66CC99,  0x66FF99,
0x6600CC,  0x6633CC,  0x6666CC,  0x6699CC,  0x66CCCC,  0x66FFCC,
0x6600FF,  0x6633FF,  0x6666FF,  0x6699FF,  0x66CCFF,  0x66FFFF,


0x990000,  0x993300,  0x996600,  0x999900,  0x99CC00,  0x99FF00,
0x990033,  0x993333,  0x996633,  0x999933,  0x99CC33,  0x99FF33,
0x990066,  0x993366,  0x996666,  0x999966,  0x99CC66,  0x99FF66,
0x990099,  0x993399,  0x996699,  0x999999,  0x99CC99,  0x99FF99,
0x9900CC,  0x9933CC,  0x9966CC,  0x9999CC,  0x99CCCC,  0x99FFCC,
0x9900FF,  0x9933FF,  0x9966FF,  0x9999FF,  0x99CCFF,  0x99FFFF,


0xCC0000,  0xCC3300,  0xCC6600,  0xCC9900,  0xCCCC00,  0xCCFF00,
0xCC0033,  0xCC3333,  0xCC6633,  0xCC9933,  0xCCCC33,  0xCCFF33,
0xCC0066,  0xCC3366,  0xCC6666,  0xCC9966,  0xCCCC66,  0xCCFF66,
0xCC0099,  0xCC3399,  0xCC6699,  0xCC9999,  0xCCCC99,  0xCCFF99,
0xCC00CC,  0xCC33CC,  0xCC66CC,  0xCC99CC,  0xCCCCCC,  0xCCFFCC,
0xCC00FF,  0xCC33FF,  0xCC66FF,  0xCC99FF,  0xCCCCFF,  0xCCFFFF,


0xFF0000,  0xFF3300,  0xFF6600,  0xFF9900,  0xFFCC00,  0xFFFF00,
0xFF0033,  0xFF3333,  0xFF6633,  0xFF9933,  0xFFCC33,  0xFFFF33,
0xFF0066,  0xFF3366,  0xFF6666,  0xFF9966,  0xFFCC66,  0xFFFF66,
0xFF0099,  0xFF3399,  0xFF6699,  0xFF9999,  0xFFCC99,  0xFFFF99,
```

```
0xFF00CC,  0xFF33CC,  0xFF66CC,  0xFF99CC,  0xFFCCCC,  0xFFFFCC,
0xFF00FF,  0xFF33FF,  0xFF66FF,  0xFF99FF,  0xFFCCFF,  0xFFFFFF  };




// int NodeInt=(int)(0xffffff*(NodeIntensity(X,Y)/MaxFrameIntensity));
Red  =((unsigned char)(Palate[NodeInt]>>16));
Green=((unsigned char)(Palate[NodeInt]>>8));
Blue =((unsigned char)(Palate[NodeInt]));
// if(NodeInt!=0)
// {
//
// Red=  ((unsigned char)(NodeInt>>16));
// Green=((unsigned char)(NodeInt>>8));
// Blue= ((unsigned char)(NodeInt));
// }


// Red=(int)(255*(NodeIntensity(X,Y)/MaxFrameIntensity));
// Blue=Red;
// Green=Red;
return true;
}
*/


TNODE CTlmPPM::MaxSignal()
{
return 1;
}
```

B.4.2 Base class:

CTlmPPM is based on CTlmBasic.

B.4.3 Data types

CTlmPPM hasn't any internal data types.

B.4.4 Global variables:

This class hasn't any global variable that it's user can read or change them.

B.4.5 Functions:

virtual bool SetFileTemplate(char * Template): Set the file template for writing each frame with that name. for example if the template set to "frame%d" then the first frame (frame 0) will be save as "frame0.ppm" and frame 1 as "frame1.ppm". File template use the same rules as printf format string.

CTlmPPM(): Constructor.

virtual CTlmPPM(): Destructor.

bool SaveFrame(int FrameNumber): Save a frame in PPM format. File name for saving is created based on two parameter: FileTemplate and frame Number.For more information see SetFileTemplate().

virtual bool RGBData(int X,int Y,int & Red,int & Blue,int & Green,TNODE MaxFrameIntensity): This function return the RGB value for each node.It maybe overridden to create a colorful image based on properties of the medium or wave properties.

B.4.6 Implementation:

```
// TlmPPM.cpp: implementation of the CTlmPPM class.
//
//////////////////////////////////////////////////////////////////////
#include <stdio.h>
```

```
#include <string.h>

#include "TlmPPM.h"

//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

CTlmPPM::CTlmPPM()
{
m_FileNameTemplate[0]=0;

}

CTlmPPM::~CTlmPPM()
{

}
bool CTlmPPM::SaveFrame(int FrameNumber)
{
if(m_FileNameTemplate[0]==0) return true;
char FileName[255];
int Red,Blue,Green;
sprintf(FileName,m_FileNameTemplate,FrameNumber);
FILE * FPointer=fopen(FileName,"wb");
fprintf(FPointer,"P6\n");
fprintf(FPointer,"%d %d\n",m_XDim,m_YDim);
fprintf(FPointer,"255\n");
//
TNODE MaxIntensity=MaxSignal();
for(int Y=m_YDim-1;Y>=0;Y--){
for(int X=0;X<m_XDim;X++){
if(IsOnAnObject(X,Y)>=0)
```

```
{
Red=255;
Blue=255;
Green=255;
}
else
{
RGBData(X,Y,Red,Blue,Green,MaxIntensity);
// draw boundry
if(X==0 || X==m_XDim-1 || Y==0 || Y==m_YDim-1){
// Blue=255;
// Green=255;
Red=255;

}
}
fwrite(&Red,1,1,FPointer);
fwrite(&Green,1,1,FPointer);
fwrite(&Blue,1,1,FPointer);

}

}
fclose(FPointer);
return true;
}
bool CTlmPPM::SetFileTemplate(char * Template)
{
strcpy(m_FileNameTemplate,Template);
return true;

}
```

```
#define MAXGREY (255)
#define MAXRANGE (100)
#define DAVE (1)
#define PETER (2)
#define COLORSETUP DAVE

bool CTlmPPM::RGBData(int X, int Y, int & Red, int & Blue, int & Green,TNO
{
#if (COLORSETUP==PETER)
TNODE Value=NodeIntensity(X,Y);
if(Value> .5* MaxFrameIntensity) Value=MaxFrameIntensity;
Green = 0;
Red =   Value * 255/MaxFrameIntensity;
Blue = (MaxFrameIntensity-Value)*255/MaxFrameIntensity;

#endif


#if (COLORSETUP == DAVE)
int Colour = (int)(MAXGREY*(NodeIntensity(X,Y)/MaxFrameIntensity));
if(Colour >MAXRANGE )
Colour=MAXRANGE;
Colour*=255.0/MAXRANGE;
// if(Colour>0)
// {
// fprintf(fp,"%d %d \t%d\t%d\n",m_Time,X,Y,Colour);
// }
    if (Colour>MAXGREY) Colour=MAXGREY;
    if (Colour<0) Colour=0;
    if (Colour<(MAXGREY/6))
{
Red = 0;  Green = Colour*6;  Blue = MAXGREY;
    } else if (Colour<(MAXGREY/3))
```

```
{
Red = 0;  Green = MAXGREY;  Blue = MAXGREY - 6*(Colour - MAXGREY/6);
    } else if (Colour<(MAXGREY*11/24))
{
Red = (Colour - MAXGREY/3)*8;  Green = MAXGREY;  Blue = 0;
} else if (Colour<(MAXGREY*7/12))
{
Red = MAXGREY;  Green = MAXGREY - 8*(Colour - MAXGREY*11/24);  Blue = 0;
    } else if (Colour<(MAXGREY*5/6))
{
Red = MAXGREY;  Green = 4*(Colour - MAXGREY*7/12);  Blue = 0;
    } else {
Red = MAXGREY;  Green = MAXGREY;  Blue = 12*(Colour - MAXGREY*5/6);
    }
    if (Blue < 0) Blue = 0;
    if (Blue > MAXGREY) Blue = MAXGREY;
    if (Red < 0) Red = 0;
    if (Red > MAXGREY) Red = MAXGREY;
    if (Green < 0) Green = 0;
    if (Green > MAXGREY) Green = MAXGREY;
#endif
return true;
}

/*bool CTlmPPM::RGBData(int X, int Y, int & Red, int & Blue, int & Green,TNODE M
{
int NodeInt=(int)(255*(NodeIntensity(X,Y)/MaxFrameIntensity));
unsigned int Palate[256]={ 0x000000,  0x003300,  0x006600,  0x009900,  0x00CC00,
0x000033,  0x003333,  0x006633,  0x009933,  0x00CC33,  0x00FF33,
0x000066,  0x003366,  0x006666,  0x009966,  0x00CC66,  0x00FF66,
0x000099,  0x003399,  0x006699,  0x009999,  0x00CC99,  0x00FF99,
0x0000CC,  0x0033CC,  0x0066CC,  0x0099CC,  0x00CCCC,  0x00FFCC,
0x0000FF,  0x0033FF,  0x0066FF,  0x0099FF,  0x00CCFF,  0x00FFFF,
```

```
0x330000,  0x333300,  0x336600,  0x339900,  0x33CC00,  0x33FF00,
0x330033,  0x333333,  0x336633,  0x339933,  0x33CC33,  0x33FF33,
0x330066,  0x333366,  0x336666,  0x339966,  0x33CC66,  0x33FF66,
0x330099,  0x333399,  0x336699,  0x339999,  0x33CC99,  0x33FF99,
0x3300CC,  0x3333CC,  0x3366CC,  0x3399CC,  0x33CCCC,  0x33FFCC,
0x3300FF,  0x3333FF,  0x3366FF,  0x3399FF,  0x33CCFF,  0x33FFFF,

0x660000,  0x663300,  0x666600,  0x669900,  0x66CC00,  0x66FF00,
0x660033,  0x663333,  0x666633,  0x669933,  0x66CC33,  0x66FF33,
0x660066,  0x663366,  0x666666,  0x669966,  0x66CC66,  0x66FF66,
0x660099,  0x663399,  0x666699,  0x669999,  0x66CC99,  0x66FF99,
0x6600CC,  0x6633CC,  0x6666CC,  0x6699CC,  0x66CCCC,  0x66FFCC,
0x6600FF,  0x6633FF,  0x6666FF,  0x6699FF,  0x66CCFF,  0x66FFFF,

0x990000,  0x993300,  0x996600,  0x999900,  0x99CC00,  0x99FF00,
0x990033,  0x993333,  0x996633,  0x999933,  0x99CC33,  0x99FF33,
0x990066,  0x993366,  0x996666,  0x999966,  0x99CC66,  0x99FF66,
0x990099,  0x993399,  0x996699,  0x999999,  0x99CC99,  0x99FF99,
0x9900CC,  0x9933CC,  0x9966CC,  0x9999CC,  0x99CCCC,  0x99FFCC,
0x9900FF,  0x9933FF,  0x9966FF,  0x9999FF,  0x99CCFF,  0x99FFFF,

0xCC0000,  0xCC3300,  0xCC6600,  0xCC9900,  0xCCCC00,  0xCCFF00,
0xCC0033,  0xCC3333,  0xCC6633,  0xCC9933,  0xCCCC33,  0xCCFF33,
0xCC0066,  0xCC3366,  0xCC6666,  0xCC9966,  0xCCCC66,  0xCCFF66,
0xCC0099,  0xCC3399,  0xCC6699,  0xCC9999,  0xCCCC99,  0xCCFF99,
0xCC00CC,  0xCC33CC,  0xCC66CC,  0xCC99CC,  0xCCCCCC,  0xCCFFCC,
0xCC00FF,  0xCC33FF,  0xCC66FF,  0xCC99FF,  0xCCCCFF,  0xCCFFFF,
```

```
0xFF0000,  0xFF3300,  0xFF6600,  0xFF9900,  0xFFCC00,  0xFFFF00,
0xFF0033,  0xFF3333,  0xFF6633,  0xFF9933,  0xFFCC33,  0xFFFF33,
0xFF0066,  0xFF3366,  0xFF6666,  0xFF9966,  0xFFCC66,  0xFFFF66,
0xFF0099,  0xFF3399,  0xFF6699,  0xFF9999,  0xFFCC99,  0xFFFF99,
0xFF00CC,  0xFF33CC,  0xFF66CC,  0xFF99CC,  0xFFCCCC,  0xFFFFCC,
0xFF00FF,  0xFF33FF,  0xFF66FF,  0xFF99FF,  0xFFCCFF,  0xFFFFFF  };
```

```
// int NodeInt=(int)(0xffffff*(NodeIntensity(X,Y)/MaxFrameIntensity));
Red  =((unsigned char)(Palate[NodeInt]>>16));
Green=((unsigned char)(Palate[NodeInt]>>8));
Blue =((unsigned char)(Palate[NodeInt]));
// if(NodeInt!=0)
// {
//
// Red=  ((unsigned char)(NodeInt>>16));
// Green=((unsigned char)(NodeInt>>8));
// Blue= ((unsigned char)(NodeInt));
// }


// Red=(int)(255*(NodeIntensity(X,Y)/MaxFrameIntensity));
// Blue=Red;
// Green=Red;
return true;
}
*/
```

```
TNODE CTlmPPM::MaxSignal()
{
return 1;
}
```

B.4.7 Programming:

you must inherited from this class before you can use it.In inherited class do the followings:

1. Override ProcessNode() to implement the medium properties.
2. Override PreProcessFrame() or PreProcessNode() to add source signal to the medium.[5]
3. Override PostProcessFrame() or PostProcessNode() to save receiver signal.[6]
4. When you want to save a frame data[7], call SaveFrame() with appropriate frame number (usually time stamp of the frame).
5. If you want to add color to image based on medium property or wave property, override RGBData().
6. Add any other capability that you want to your class.

In the main program do the followings:

1. Crate an instance of your class.
2. Initialize the TLM mesh by calling Init().
3. Initialize PPM file name creation by calling SetFileTemplate().
4. Do any initialization your class needed.
5. Start to model by calling DoModelling().

[5] if you have one source signal it is simpler to override PreProcessNode() and if you have several source signal it is simpler to override PreProcessFrame()

[6] if you have only one receiver it is simpler to use PostProcessNode() but if you have several receiver or you want to save the data for a complete frame then you must use PostProcessFrame().

[7] usually this function is called in PostProcessFrame()

References

[1] W. H. Nailon, *Tissue characterisation from intravascular ultrasound using texture analysis*. Phd, Edinburgh university, Electronic and electrical department edinburgh university edinburgh UK, 1997.

[2] W. F. Young, *Pseudo inverse filter design for improving the axial resolution of ultrasound imaging*. Phd, Edinburgh university, Electronic and electrical department. edinburgh university edinburgh UK, August 1998.

[3] "Published at URL:*http://sti-ultrasound.com/*."

[4] "Published at URL:*http://www.healthcare.agilent.com/ultrasound/*."

[5] "Published at URL:*http://www.ob-ultrasound.net/*."

[6] "Published at URL:*http://www.3dsono.org/*."

[7] "Published at URL:*http://www.imasonic.com/*."

[8] "Published at URL:*http://www.ieee-uffc.org/*."

[9] "Published at URL:*http://www.nml.eee.nott.ac.uk/research/tlm.html*."

[10] P. Pedersen and D. Orofino, "Modelling of received signal from finite reflectors in pulse-echo ultrasound," in *Ultrasonic symposium*, pp. 1177–1181, IEEE, 1994.

[11] J. F. Greenleaf, ed., *Tissue characterization with ultrasound*, vol. 2- Result and applications. CRC press Inc., 1986.

[12] F. L. Lizzi, M. Ostromogilsky, E. J. Feleppa, M. C. Rorke, and M. M. Yaremko, "Relationship of ultrasound spectral parameters to features of tissue micro structure," *IEEE transactions on ultrasonics, ferroelectric and frequency control*, vol. UFFC-33, pp. 319 – 329, May 1986.

[13] F. L. Lizzi, M. Greenebaum, E. J. Feleppa, and M. Elbaum, "Theoretical framework for spectrum analysis in ultrasound tissue characterization," *Journal of acoustic society of america*, vol. 73, pp. 1366–1373, April 1983.

[14] G. Trahey, D. Zhao, J. A.Miglin, and S. W. Smith, "Experimental result with a real-time adaptive ultrasonic imaging system for viewing through distorting media," *IEEE transaction on ultrasonics, ferroelectric and frequency control*, vol. 37, pp. 418–427, September 1990.

[15] P.B.Johns, "A new mathematical model to describe the physics of propagation," *Radio electron eng.*, vol. 44, pp. 657 – 666, December 1974.

[16] T. Itoh, "Recent advanced in numerical methods for microwave and millimetre-wave passive structures," *IEEE transactions on magnetics*, vol. 25, p. 29312934, july 1989.

[17] Y. Kagawa, "Computational acoustics theories of numerical analysis in acoustics with emphasis on transmission line matrix modelling," in *ASVA 97*, pp. 19 – 26, 1999.

[18] D.K.Lewis, J.F.DeFord, K.S.Kunz, C.S.Landrum, D.J.Steich, S.E.Bumpas, M.A.hriston, T.R.Donich, J.D.Foch, G.L.Goudreau, J.Jones-Olivera, and S.J.Wineman, "Acoustic modelling code development for time domain broad bandwidth computations," in *Ultrasonic symposium* (IEEE, ed.), 1994.

[19] G. Zhang, S. Pomeroy, and C. Wykes, "A hybrid model for propagation and scattering in a robotic acoustic imaging system," in *Acoustic sensing and imaging* (IEE, ed.), pp. 233–238, IEE, IEE, March 1993.

[20] W. J. Zwiebel and R. Sohaey, *Introduction to ultrasound*. WB Saunders, 1997.

[21] J. Bltz, *Fundamentals of ultrasound*. Butterworths, 1963.

[22] Scottish home and health department, *Obstetric ultrasound in scotland*. Edinburgh: Her Majesty's stationery office, 1988.

[23] J. Janssen, *Clinical application of video image processing in cardiac angiology*. Assen: Van Gorcum, 1989.

[24] P. N. T. Wells, "The safety of diagnostic ultrasound," tech. rep., The british institute of radiology, London, 1987.

[25] The Scottish home and health department and scottish service advisory council, *Reports on vascular surgery service in scotland*. Edinburgh: Her Majesty's stationery office, 1994.

[26] "Published at URL:*http://www.hc−sc.gc.ca/ehp/ehd/catalogue/rpb_pubs/88ehd59.pdf*."

[27] F. Kremkau, *Diagnostic ultrasound: principles, instruments and exercises.* W.B. Sunders, 3^{rd} ed., 1989.

[28] S. Bushong and B. Archer, *Diagnostic ultrasound: Physics, biology and instrumentation*. Mosby-year Book Inc., 1991.

[29] E. Barnet and P. Morley, *Clinical diagnostic ultrasound*. Blackwell scientific publications, 1985.

[30] H. Ermert, O. Keitmann, R. Oppelt, B. Granz, A. Pesavento, M. Vester, B. Tillig, and V. Sander, "A new concept for a real-time ultrasound transmission camera," *IEEE ultrasonics symposium*, vol. 2, pp. 1611 –1614, 2000.

[31] S. Sanei, "Characterisation of fat and malignancy in transmissive ultrasound breast tomographs applying fuzzy logic," *Proceedings of the 20th annual international conference of the IEEE on engineering in medicine and biology society*, vol. 3, pp. 1367 –1370, 1998.

[32] W.N. McDicken, *Diagnostic ultrasound: Principles and use of instruments*. Livingston: Churchill, 3^{rd} ed., 1991.

[33] N. Thune and B. Olstad, "Visualizing 4-d medical ultrasound data," *IEEE visualization proceeding*, pp. 210–5, 1991.

[34] J. Hokland and T. Hausken, "Robust semi-transparent volume rendering of abdominal 3d ultrasound data," *IEEE ultrasound symposium proceedings*, vol. 3, pp. 567 –71, November 1994.

[35] A. State, D.T. Chen, C. Tector, A. Brandt, H. Chen, R. Ohbuchi, M. Bajura, and H. Fuchs, "Case study: Observing a volumes rendered fetus within a pregnant patient," *IEEE visualization proceeding*, pp. 364–8, 1994.

[36] G. Wade, ed., *Acoustic imaging*. Plenum Press, 1976.

[37] D. Evans and W. McDicken, *Doppler ultrasound : Physics, instrumentation and signal processing*. John Wiley and Sons, 2nd ed., 1999.

[38] A. Wood, *A textbook of sound*. G.Bell and sons LTD, 1944.

[39] A. Walshaw, *Mechanical vibrations with applications*. Ellis horwood limited, 1984.

[40] E. Catchpool and J. Satterly, *Textbook of Sound*. University Tutorial Press LTD, 1949.

[41] I. G. Main, *Vibrations and waves in physics*. Cambridge university press, 1978.

[42] R.J. Grade, *Turbulent Flow*. John willy and sons, 1994.

[43] P. Filippi, D. habault, jean Pierre Leafebvre, and A. Bergassoli, *Acoustics: Basic physics, theory and methods*. Academic press, 1999.

[44] F. S. Crawford, *Waves*, vol. 3 of *Berkeley physics course*. McGraw hill, 1968.

[45] E. H. Barton, *A Textbook on Sound*. Macmillan and Co Limited, 1919.

[46] M. Lalanne, P. Berthier, and J. D. Hagopian, *Mechanical vibrations for engineers*. John Wiley and Sons, 1983.

[47] C.R.Hill, *The generation and structure of acoustic fields*, ch. 2, pp. 68 – 92. In [48], 1986.

[48] C.R.Hill, *Physical principles of medical ultrasonics*. Ellis Horwood series in applied physics: Ellis Horwood Limited, 1986.

[49] P. M. Morse and K. N. Ingard, *Theoretical acoustics*. McGraw hill, 1968.

[50] M. KerKer, *The scattering of light*. Academic press, 1969.

[51] C.R.Hill, *Reflection and scattering*, ch. 6, pp. 225 – 260. In [48], 1986.

[52] S. J. Norton and M. Linzer, "Ultrasonic reflectivity imaging in three dimensions: Exact inverse scattering solutions for plane cylindrical and spherical apertures," *IEEE transactions on biomedical engineering*, vol. BME-28, pp. 202 – 220, February 1981.

[53] P. M. Morse and H. Feshbach, *Method of theoretical physics*. McGraw Hill, 1953.

[54] R. C. Chivers, "The scattering of ultrasound by human tissue: Some theoretical model," *Ultrasound in medical and biology*, no. 3, pp. 1 – 13, 1977.

[55] K. D. John, *Electromagnetics*. McGraw hill, $3^r d$ ed., 1984.

[56] kraus John D., *Electromagnetics*. McGraw hill, 4^{th} ed., 1992.

[57] E. da Silva, *High frequency and microwave engineering*. Butterworth Heinemann, 2001.

[58] C. Chistopoulos, *The transmission-line modelling method TLM*. IEEE, 1995.

[59] P.B.Johns and R.L.Beurle, "Numerical solution of 2-dimensional scattering problems using a transmission-line matrix," *Proc of IEE*, vol. 118, pp. 1203–1208, Sept 1971.

[60] W.J.R.Hoefer, "The transmission-line matrix method: Theory and application," *IEEE Trans Microwave Theory Tech*, vol. MTT-33, pp. 882–893, Oct 1985.

[61] W. J. Hoefer, "Huygens and the computer - a powerful alliance in numerical electromagnetic," *Proceeding of the IEEE*, vol. 79, pp. 1459 – 1471, October 1991.

[62] P.B.Johns, "A simple explicit and unconditionally stable numerical routine for the solution of the diffusion equation," *Int. J. Num. Meth. Eng.*, vol. 11, pp. 1307 – 1328, 1977.

[63] Partridge, C. Christopoulos, and P.B.Johns, "Transmission line modelling of shaft dynamic systems," *Proceedings of the institute of mechanical engineers*, no. 201, pp. 271 – 278, 1987.

[64] V. Trenkic, C. Christopoulos, and J.G.P.Binner, "The application of the transmission line modelling (tlm) method in combined thermal and electromagnetic problem," *Proceedings of the international conference on numerical methods for thermal problems*, pp. 1263 – 1274, 1993.

[65] F.J.German, G.K.Gothard, L.S.Riggs, and P.M.Goggans, "The calculation of radar cross-section (rcs) using the tlm method," *Int. J. Numerical Modelling*, vol. 2, pp. 267 – 278, 1989.

[66] K. Umashankar and A. Taflove, "A novel method to analyze electromagnetic scattering of complex objects," *IEEE transactions on electromagnetic compatibility*, vol. EMC-24, pp. 397 – 405, November 1982.

[67] D. Halliday and R. Resnick, eds., *Fundamental of Physics*. John Wiley and Sons, third ed., 1988.

[68] E. H. Barton, *Huygence Principle*, ch. 2, pp. 71–74. In [45], 1919.

[69] P.B.Johns, "Application of the transmission-line matrix method to homogeneous waveguides of arbitrary cross-section," *Proc . inst . elec. eng.*, vol. 119, pp. 1086 – 1091, Aug. 1972.

[70] P.B.Johns, "The solution of inhomogeneous waveguide problems using a transmission-line matrix," *IEEE trans. Microwave Theory Tech*, vol. MTT-22, pp. 209–215, March 1974.

[71] S. Akhtarzad and P. Johns, "Transmission-line matrix solution of waveguides with wall losses," *Electron. Lett*, vol. 9, pp. 335 – 336, July 1973.

[72] P.P.M.SO, C. Eswarappa, and W.J.R.Hoefer, "Parallel and distributed tlm computation with signal processing for electromagnetic field modelling," *International journal of numerical modelling: Electronic network, device and fields*, vol. 8, pp. 169 – 185, 1995.

[73] P.J.Parsons, S.R.Jaques, S.H.Pulko, and F.A.Rabhi, "Tlm modelling using distributed computing," *IEEE microwave and guided wave letters*, vol. 6, pp. 141 – 142, march 1996.

[74] MPI committee, "Mpi: A message passing interface standard," tech. rep., Message passing interface forum, University of tennessee, Knoxville, Tennessee, 1995.

[75] P. So, W. Hoefer, and R. Sorrentino, "Numerical microwave synthesis by inverse of the "tlm" process," in 21^st *European microwave conference digest*, pp. 1273–1277, Sept 1991.

[76] M. Forest and W. J. Hoefer, "Tlm synthesis of microwave structure using time reversal," *MTT-S digest*, pp. 779 –782, 1992.

[77] M. Forest and W. J. Hoefer, "Novel synthesis technique for conducting scatters using tlm time reversal," *IEEE transaction on microwave theory and techniques*, vol. 43, pp. 1371 – 1378, june 1995.

[78] D. de Cogan and A. Soulos, "Inverse thermal modelling using tlm," *Numerical Heat Transfer*, no. 29, pp. 125 – 135, 1996.

[79] K. Y, T.Tsuchiya, and K.Fujioka, "Acoustic imaging using transmission-line matrix model in time reversal," in *Education, practice and promotion of computational methods in engineering using small computers*, pp. 1429 – 1434, August 1995.

[80] S. Leibovich and A. R. Seebass, *Nonlinear Waves*. Cornell University Press, 1974.

[81] D. Meliani and a. P. De Cogan, "The use of orthogonal curvilinear meshes in tlm models.," *Int. j. numerical modelling*, vol. 1, pp. 221–238, 1988.

[82] A. Saleh, "The application of tlm in sonar and underwater acoustics," *IEE Colloquium on TLM*.

[83] G. Zhang, C. Wykes, and S. C. Pomeroy, "The application of tlm to the modelling of airborne ultrasonics," *IEE colloquium on TLM*.

[84] Z. Chen, M. M. Ney, and W. J. Hoefer, "Absorbing and connecting boundary condition for the tlm method," *IEEE Transaction on microwave theory and techniques*, vol. 41, pp. 2016 – 2024, November 1993.

[85] J. A. Morente, J. A. Porti, and M. Khalladi, "Absorbing boundary conditions for the tlm method," *IEEE transaction on microwave theory and techniques*, vol. 40, pp. 2095 – 2099, November 1992.

[86] N.R.S.Simon and E. Bridges, "Method for modelling free space boundaries in tlm situations," *Electronic letter*, vol. 26, pp. 453–455, 1990.

[87] I. G. Gosling, "Reduction of edge reflection in the tlm model," *IEEE transaction on microwave theory and techniques*, vol. 41, pp. 1057 – 1064, june/july 1993.

[88] D. Halliday and R. Resnick, eds., *Interference*, ch. 40, pp. 900 – 921. In Halliday and Resnick [67], third ed., 1988.

[89] A. Carlson, *Communication systems*. McGrow-Hill, second ed., 1968.

[90] A. Saleh and P. Blanchfield, "Analysis radiation patterns of array transducers using the tlm method," *Int. j. numer. modelling*, vol. 3, pp. 39 – 56, 1990.

[91] J. G. Proakis and D. G. Manolakis, *Introduction to digital signal processing*. Macmillan Publishing Company, 1989.

[92] A. V. Oppenheim, A. S. Willsky, and I. T. Young, *Signal and systems*. Prentice Hall Signal Processing Series, Prentice Hall International Inc., 1983.

[93] J. G. Proakis, C. M. Rader, F. Ling, and C. L. Nikias, *Advanced digital signal processing*. Macmillan Publishing Company, 1992.

[94] R. Ziemer and W. Tranter, *Principles of communications*. Houhton Mifflin company, 2^{nd} ed., 1985.

[95] P. Z. peebles, *Digital communication systems*. Printice hall international, 1987.

[96] J. Fourier, *The analytical theory of heat*. Cambridge, 1878. Translated to english by A.Freeman.

[97] C. S. Burrus and T. W. Parks, *DFT/FFT and Convolution Algorithms*. John wiley & sons, 1985.

[98] M. Ahmadian and J. Dripps, "Introducing transmission line matrix (tlm) modelling and its application in medical ultrasound," *Proceeding of the First International Conference on Advances in Medical Signal and Information Processing*, pp. 212 –221, September 2000.