Function-Based and Physics-Based Hybrid Modular Neural Network for Radio Wave Propagation Modeling

by

Lee Wai Hung

A Thesis

Submitted in partial fulfillment of the requirements for degree of

Master of Philosophy

Department of Electronic Engineering

The Chinese University of Hong Kong

June 1999

摘要

射線描繪運算法曾以組合式神經網絡作為無線電波傳送模型的執行手段，目的是為了建立一個神經網絡結構以取替傳統計算。它擁有的地域識別能力能在可接受的限制下把不同的環境模擬出來。四個規範模擬以兩個不同方法進行及比較，分別為層遞式射線描繪運算法和組合式神經網絡模型。其中一些模擬更參考標準公式。這些模擬是用以驗証組合式神經網絡模型的精確度。其中一個模擬的環境是中空會堂，天花板上安裝方向性天線。如以層遞式射線描繪運算法的計算結果為依握，組合式神經網絡的平均預測誤差和標準偏差分別為1.57分貝及1.39分貝。在實際測試中，組合式神經網絡亦應用於預測中文大學工程大樓三樓的電磁波傳送情況。在陰影區下，組合式神經網絡的平均預測誤差和標準偏差分別為6.93分貝及6.01分貝。在可見區下，平均預測誤差和標準偏差分別為5.27分貝及4.63分貝。

Abstract

A modular neural network approach was used to implement ray tracing algorithm for radio wave propagation modeling. The goal is to develop a neural network architecture to replace traditional calculations. This method is site-specific so that it can simulate different environments with some acceptable limitation in environment dimensions. Four canonical simulations were preformed in brute force ray tracing method, hybrid modular neural network and deterministic formulation in some case to verify the accuracy of the modular neural network model. In one canonical simulation, empty hall with ceiling mounted directional antenna, 1.57 dB mean prediction error and 1.39 dB standard deviation were obtained by modular neural network with reference to brute force ray tracing method. As an actual test, modular neural network is used to predict propagation inside the third floor of the engineering building of the Chinese University of Hong Kong. The average prediction error of modular neural network is 6.93 dB and 6.01 dB standard deviation for shadow region, and 5.27 dB with 4.63 dB standard deviation for line-of-sight region.

Acknowledgement

I would like to thank my supervisor, Prof. Albert K. Y. LAI for his valuable advice, encouragement and guidance throughout the course of the research program. Prof. Lai's patient reading and indispensable comments contributed much in the progress of the thesis.

It is my pleasure to thank my colleagues, Mr. C. W. Fan, Mr. Y. B. CHOI, Mr. S. C. CHAN, Mr. C., K. NG, Mr. Y. M. CHEN, Mr. W., C. FUNG, Mr. K. K. Fu for sharing their knowledge and experience.

1 Introduction

1.1 Background

Reliable and fast radio wave prediction is a critical factor in the design of mobile communication network. With the help of sophisticated prediction tools, network engineers can design an optimal cellular network at relatively low cost compared to a purely measurement approach. Some service providers in Hong Kong carry out large-scale site measurement every three months to maintain substantial mobile phone service quality in a fast changing environment. Cellular mobile systems require good planning in order to optimize system performance in terms of system coverage and capacity. The number and placement of base stations, as well as the frequency allocation scheme can be optimized. Experience shows that large-scale improvement and adjustment is difficult to carry out just by the measurement approach. With a software simulation tool, network configurations can be tested and evaluated thoroughly before they are put into actual operation. Result can be further improved if the simulation and measurement are combined together as an integrated analysis. Additionally, radio wave prediction tool makes it possible to debug and retrofit the network when problems arise from regular maintenance check or customer complaints.

For propagation in cities, many modeling methods have been proposed in the past. However, most of them are inaccurate particularly when the considered structures are of the same order as the wavelength of signal, and when there is large variation in building heights. This implies that a building or even an advertizement structure may need to be considered in the simulation model. Statistical approximation is not good enough. If one insists on statistical method, then extensive additional measurement needs to be carried out from time to time in order to maintain the network performance. Another type of modeling method, deterministic modeling, has been widely investigated since it provides the fundamental basis for site-specific simulation. Ray tracing is one example of deterministic modeling. In fact, it is an approximated closed form solution for electromagnetic wave problem. Ray tracing

approximates electromagnetic wave by optical rays and traces the propagation of all possible rays by considering reflection, transmission and diffraction. The receiver signal can be obtained by a vector sum of all multi-path signals approaching the receiver antenna.

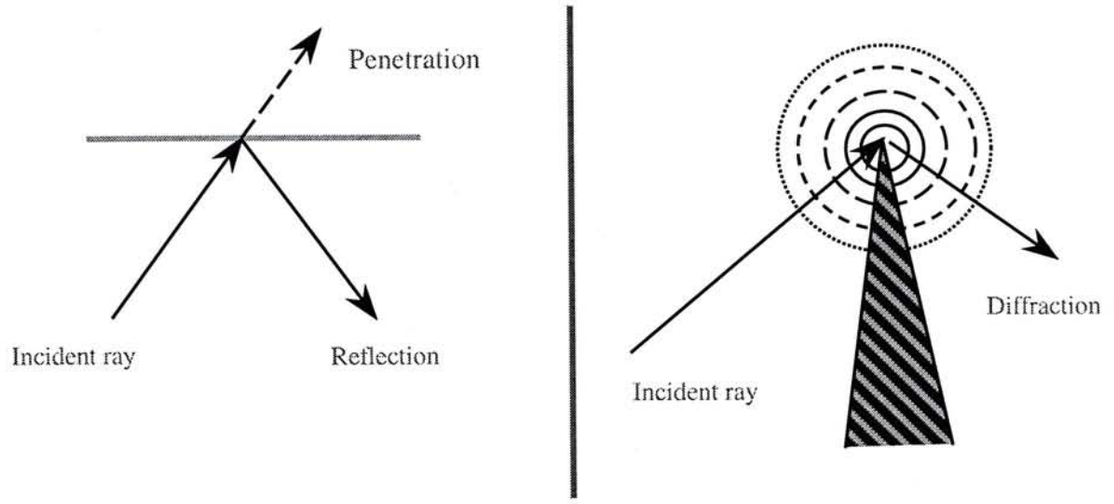


Figure 1-1 : Three typical propagation mechanisms in ray tracing

There is however one major problem of deterministic simulations. Most of these deterministic simulations require long computation time and massive computation resources. This is particularly so for complicated urban environment where multiple bounces from surfaces are expected. In ray tracing method, for example, large reflection depth implies more secondary rays will be produced in the reflection calculation and thus long simulation time is required to complete the process. Theoretically, ray tracing method can be implemented in a parallel processor. The simulation can be accelerated by multi-computer programming or multi-processor programming since the calculation between one ray and the corresponding scattering surface is independent from that of another ray. However, this implementation involves large investment in computer equipment and specific software development. It implies that cost-effective hardware acceleration for ray tracing method is the best way to speed up radio wave modeling tool for cellular network design.

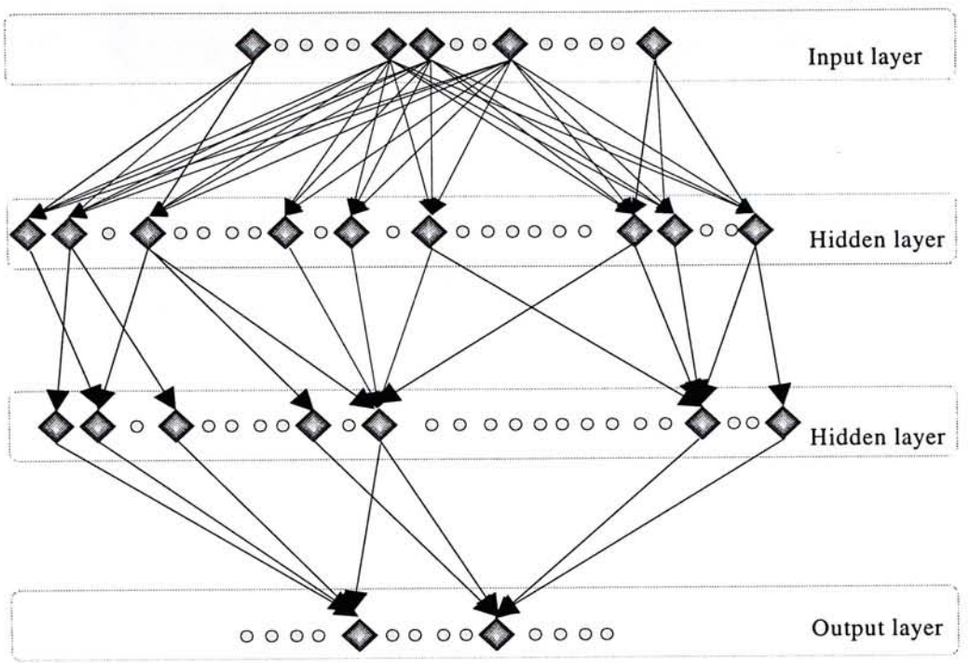


Figure 1-2 : Typical Artifical neural network structure : multiplayer Prepertron

There is one existing technology called artificial neural network that can be applied to realize cost-effective hardware acceleration. Some published work of neural network applications in wireless communication is described here to substantiate the functionality and advantages of neural network. One example is reflector antenna surface error compensation using neural network**[1]**. The authors use parallel neural network to emulate the Constrained Least Squares (CLS) algorithm used previously for the same purpose**[2]**. Computation time requirement on a UNIX machine for CLS method and parallel Multilayer Perceptron are compared, as shown in Table 1-1. Note that the neural network, after training, takes only 0.6 second to do the job, compared to 4 minutes for CLS without constraint and 1 hour 15 minutes for CLS with constraint. One interesting point is the training time of neural network is 4 hours per network, which is quite long. It points out that one is really shifting the computation load to a pre-processing step. The speed in the final computation is earned by carrying out a long training process done beforehand. However, if we have a problem where the calculation has to be repeated many times, or if the problem can be generalized, or if the problem has to be calculated in real time, then the neural network approach can be quite beneficial because once we train it (training is done off-line, or beforehand), it can be used many times at high speed.

Table 1-1 : Computation time requirement for CLS and MLP method

Method	*Computational Time*
CLS method with constraint	1 hour and 15 minutes
CLS method without constraint	4 minutes
Parallel MLP Neural Network	0.6 seconds
Training Time per MLP Network	4 hours per net

The previous example shows the speed of neural network. The next example shows the accuracy of neural network based on training data that comes from measurement. The work is a resonant frequency prediction of triangular patch microstrip antenna**[3]**. The triangular patch requires rather involved mathematics in the design process. Instead of using that, the authors use measurement data to train a neural network to do the same job.

Accuracy of neural network in terms of resonant frequencies predicted is compared with three well known methods and also the moment method. The results of neural network are much closer to the actual measurement then all other methods. The total absolute errors are shown in Table 1-2 for 5 different modes from 1.2GHz to 4.4.GHz. Please refer to **[3]** for more details.

Table 1-2 : The total absolute error for five modeling method

Method	Neural Network	Bahl and Bhartia	Helszafn and James	Garg and Long	Moment Method
Error (MHz)	23	5124	424	326	472

The third example is an inverse problem **[4]** where the shape of an aperture antenna is designed based on a given radiation pattern requirement. Although inverse problem is often a one-to-many relationship, here it does not matter because all we want is one antenna shape that works. In this paper, the author devised a methodology by which the antenna can be designed using a feed-forward neural network. Again, the author pointed out that the training time for this neural network is very long, but once it is trained, the generating of output is extremely fast. This echoes our observation in the first example, where the learning time is also very long. It is effectively moving the computation load to a per-processing step. But again, if

the problem has to be calculated many times, or if the problem can be generalized, or if the problem has to be calculated in real time, then neural network will be very useful. The neural network is well known for performing complex non-linear mappings between different domains. Similar methodologies are also used by researchers elsewhere on radio wave propagation modeling.

There is also some published work of neural network applications in radio wave propagation modeling. In early work reported in papers **[5][6]**, authors use empirical formulation and height profiles as the input of the neural network to estimate propagation loss in rural environment. Later, some authors used measured data in similar approaches. Balandier and Caminada **[7]** got 13% gain on the standard deviation (0.8dB) by using neural based hybrid system compared to semi-empirical model in urban environment. Perrault, Rossi, Balandier and Levy **[8]** used neural network as a bias correction of ray tracing model. Their correction can improve the predictions up to 2dB in standard deviation. Gschwendtner, Landstorfer **[9]** also applied neuron-calibration technique to the conventional path loss model, COST. They achieved 0 dB mean error and 5.7 dB standard deviation in testing data. Fraile and Cardona **[10]** proposed a neural network model that can accurately predict urban propagation for the following configuration: antenna height 20 meters lower than surrounding building and 45 meters higher than surrounding building as well as propagation distance up to 2000 m. All these papers showed good prediction ability of neural network for urban environments. Recently, Wolfe and Landstorfer studied three implementation approaches in using neural network to predict indoor propagation loss **[11][12][13]**. They extracted the indoor propagation characteristic as neural network input parameter according to the interaction between antenna and geometry object. Then, they use neural network to relate these parameters with the received power. In summary, their works suggest that the utilization of artificial neural network lead to more accurate radio wave propagation prediction result at much higher speed than other classical models. Among these works, measured data or empirical formulation are used to train the neural network so that it is capable of mapping the propagation loss to basic parameters such as antenna height, separation distance and simplified obstacle information. Some authors even use ray tracing method to describe the propagation channel characteristics between transmitter and receiver, although the application of modeling are limited to certain type of

environment where they collect the training data. When the combination of available input parameters cannot describe a more general environment, the ultimate goal of applying a fast and accurate radio wave propagation prediction tools in real cellular network design system cannot be reached. Therefore, a general-purpose neural network modeling in radio wave propagation needs to be developed for cellular network design system. In fact, this is the goal of the present thesis.

Basically, neural network models the relationship between input and output parameters. In radio wave propagation modeling, received signal is a function of topographical and morphographical data. One may think of a neural network model that accepts all possible parameters related to radio wave propagation including transmitter, receiver and geometry model. When the input data propagate through the network, the channel characteristics will be generated accordingly. However, in Judd's book **[14]**, he pointed out that the time needed to train a network may be exponentially increasing as a function of size for a large neural network system. The author warned that one might transform a complicated problem into a painful neural network training cycle. Moreover, debugging, scaling up and maintenance of the neural network will also be difficult because one can only quantify the model performance in terms of mean squared error between training and testing data, but it is extremely difficult to understand what is going on inside the network. As a result, retraining and network size tuning become the only procedures to improve and scale up the model. In recent years, researches are turning to a new design methodology where a large neural network structure is broken down into smaller sub-nets which is individually trained. Thereafter, a complete radio wave propagation model is constructed by a collection of trained neural networks. This is the modular neural network concept being pursued in this thesis.

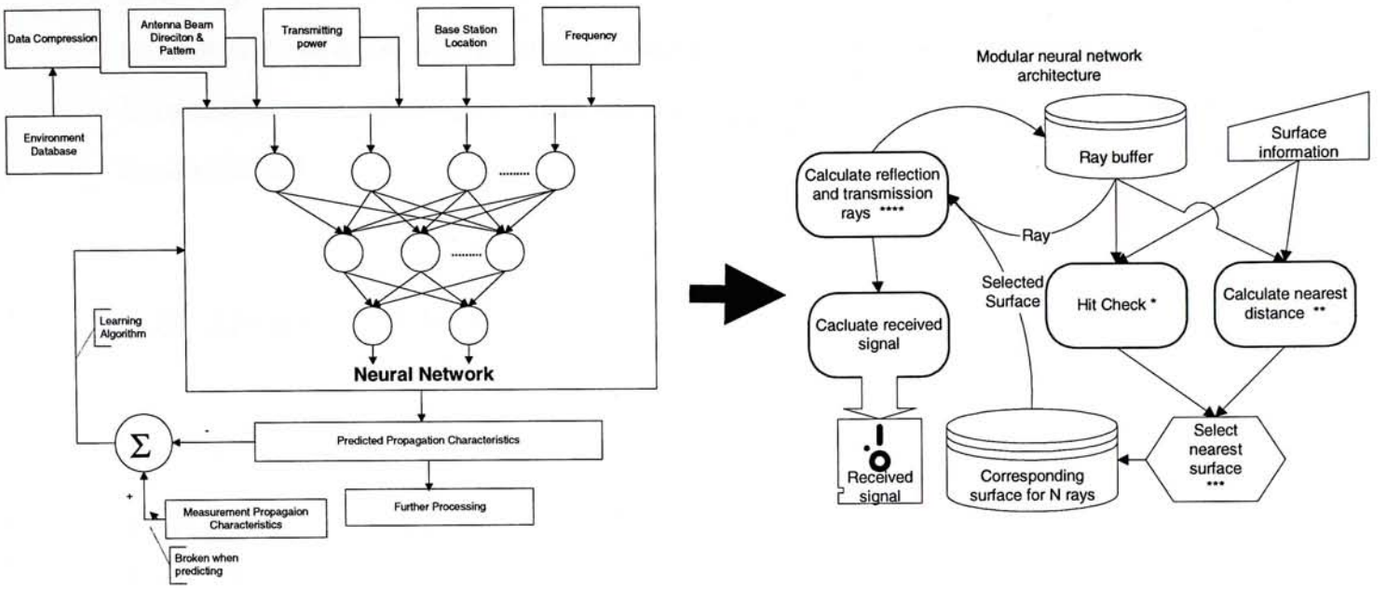


Figure 1-3 : Decomposition of large neural network to a collection of small neural networks

There are two common approaches of breaking down large neural network into smaller sub-nets. The first approach targets the training data set. The large training data set is partitioned into several manageable subsets. These subset training data are used to train individual network and a large neural network model is formed by combining all these smaller modules**[15][16]**. The second approach uses a priori knowledge to define smaller sub-problems that are solved by off-line trained MLP neural networks. Then, these pre-trained neural networks are combined to form a large modular network**[17][18]**. The second approach can be applied to radio wave propagation modeling. (Figure 1-3) Traditionally, deterministic solutions such as ray tracing are used to study radio wave propagation. The problem is that these solutions require large computation resources when large region of city is simulated. If one replaces the ray tracing algorithm by neural network model, then the computation reduces to addition and multiplication that can be easily handled by dedicated accelerator. The ray tracing algorithm is first broken down into small modules. Each module is replaced by a suitable neural network and individually trained. Finally, all these trained neural networks are combined together. This structure is referred to as hybrid modular neural network architecture. This is the modular neural network model proposed. It is used to realize ray tracing algorithm as a fast and accurate radio wave prediction tool. Since neural network is a collection of simple processors working in parallel, the actual calculation can be carried out in many processors to

speed up the simulation. Neural network can also be implemented in hardware accelerator such as digital signal processor board to further speed up the process. Ultimately, a near real-time simulation tool for radio wave propagation modeling can be developed.

1.2 Structure of Thesis

In this thesis, the modular neural network architecture for radio wave propagation model is proposed. In chapter **2**, some background about radio wave propagation modeling, brute force ray tracing techniques and neural network are described. Then, in chapter **3**, the novel modular neural network architecture will be discussed. It explains how to decompose the brute force ray tracing algorithm into small pieces of problem. In chapter **4**, each module is described in details and tested individually. The architecture, parameter definition, network restriction and data preparation will be presented. At the end of each module description, the performance of module is demonstrated. After the model is ready, in chapter **5**, it is used to simulate some canonical geometries to verify its accuracy. In some canonical geometry problem, the deterministic formulation result will also be used to study the accuracy of both models, brute force ray tracing model and modular neural network model. In chapter **6**, this novel modeling tool is used to study an indoor environment where the simulated result is compared with measurement result. Finally, in chapter 7, the work of this thesis is summarized and future work is described.

1.3 Methodology

Modular neural network architecture for radio wave propagation modeling is developed in the following manner. The architecture is shown in Figure 1-4. First, the brute force ray tracing algorithm is divided into different sections according to their functions and physical relations. The original problem now becomes many small problems and these problems can be individually solved. A suitable neural network model is applied to each module. Each neural network is then trained to solve its corresponding problem. Finally, these neural networks are combined to form

a large network. The whole network may need to be trained again in terms of network connection, while the architecture of each module remains fixed. This methodology distributes the large training effect into small pieces. The training process of each module can be done concurrently. After training of each small module is done, the overall architecture is almost ready. Second round training is relatively simple since it consists of minor adjustments in the whole network. The network is highly testable because it is carrying out ray tracing: the physics of that is clearly presented at the output of each module and not lost in the myriad of a massive neural connections.

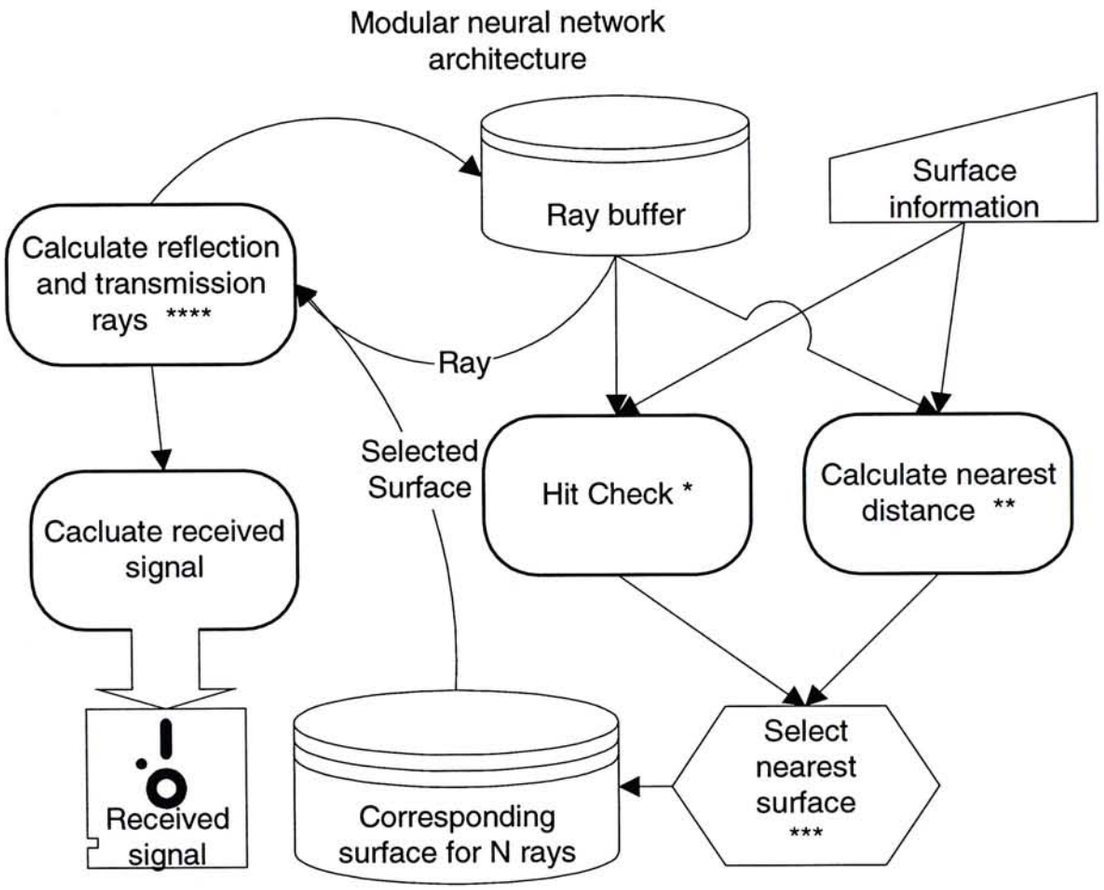


Figure 1-4 : Modular neural network architecture

2 Background Theory

2.1 Radio Wave Propagation Modeling

In this chapter, radio wave propagation modeling methods for indoor and outdoor environment are reviewed. There are two categories of modeling method: deterministic methods and statistical methods. They are different in scale, speed, accuracy and applicable area.

2.1.1 Basic Propagation Phenomena

2.1.1.1 Propagation in Free Space

Consider a transmitting antenna in free space (remote from earth or any other obstructions). If G_T is its gain in the direction to a receiving antenna and P_T is the transmitted power, then the power density (power per unit area) at a distance d in that direction is given by

$$W = \frac{P_T G_T}{4\pi d^2}$$

The power received by the receiving antenna is given by the power density multiplied by the effective area A.

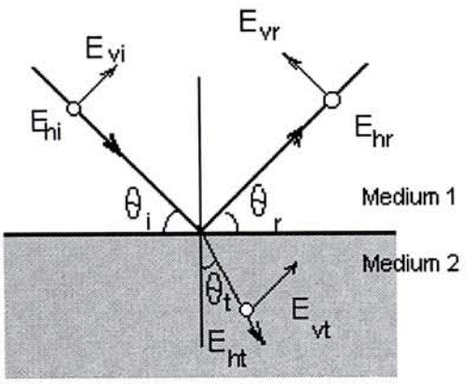
$$\begin{aligned} P_R &= \frac{P_T G_T}{4\pi d^2} \cdot A \\ &= \frac{P_T G_T}{4\pi d^2} \cdot \frac{\lambda^2 G_R}{4\pi} \\ &= G_T G_R \left[\frac{c}{4\pi f d} \right]^2 P_T \end{aligned}$$

This hypothetical free space condition never exists in its absolute form on earth. However, its analysis does give insight of the relationship of power with distance.

The concept of the above Friis formulation is important for the fact that it points out an inverse squared relation of power with distance and frequency. Furthermore, the effect of antenna patterns is also shown. These concepts are directly utilized by the ray tracing model.

2.1.1.2 Reflection and Transmission

When an electromagnetic wave propagating in one medium impinges upon another medium with a different dielectric constant, permeability or conductivity, the wave is partially reflected and partially transmitted. For the purpose of this project, only a plane reflection boundary is examined.



The angle of reflection is equal to the angle of incidence irrespective of the dielectric constants of the two media. **[19]**

$\theta_r = \theta_i$

The angle of transmission is related to the ratio of intrinsic impedances of the two media, for non-magnetic media, the relationship becomes

$$\frac{\sin\theta_t}{\sin\theta_i} = \sqrt{\frac{\varepsilon_1}{\varepsilon_2}}$$

if medium 1 is air, the relationship may be approximated as

$$\frac{\sin\theta_t}{\sin\theta_i} = \frac{1}{\sqrt{\varepsilon_{r2}}}$$

To find the reflected field strength, the incident wave must be resolved into a vertical and horizontal component because the reflection coefficients are different for

the two polarizations.

For vertical polarization

$$\Gamma_v = \frac{E_{vr}}{E_{vi}} = \frac{-\cos\theta_i + \sqrt{\varepsilon_{r2} - \sin^2\theta_i}}{\cos\theta_i + \sqrt{\varepsilon_{r2} - \sin^2\theta_i}}$$

and for horizontal polarization

$$\Gamma_h = \frac{E_{hr}}{E_{hi}} = \frac{\cos\theta_i - \sqrt{\varepsilon_{r2} - \sin^2\theta_i}}{\cos\theta_i + \sqrt{\varepsilon_{r2} - \sin^2\theta_i}}$$

The transmission coefficients are also deduced similarly. For the vertical polarization,

$$\tau_v = \frac{E_{vt}}{E_{vi}} = \frac{2\cos\theta_i}{\sqrt{1 - \frac{\sin^2\theta_i}{\varepsilon_{r2}}} + \sqrt{\varepsilon_{r2}}\cos\theta_i}$$

for the horizontal polarization,

$$\tau_h = \frac{E_{ht}}{E_{hi}} = \frac{2\cos\theta_i}{\cos\theta_i + \sqrt{\varepsilon_{r2} - \sin^2\theta_i}}$$

For a special incident angle, the Brewster angle, the reflection coefficient Γ_v is equal to zero. This angle θ_i is given by

$$\sin\theta_i = \frac{\sqrt{\varepsilon_{r2} - 1}}{\sqrt{\varepsilon_{r2}^2 - 1}}$$

2.1.2 Practical Propagation Models

Propagation models have been introduced to study propagation loss. These models may be conveniently classified either by the environmental domain being addressed or by the method being used. There are models which address either the outdoor or indoor environment. Outdoor models are mainly concerned with wave propagation in urban or suburb areas. Indoor models address propagation within a building. In terms of the methods used, there are two major approaches: statistical

and deterministic approach.

The statistical approach essentially makes use of measured data from a similar environment to make prediction. It requires extensive measurement and its accuracy is highly questionable from case to case [21]. The deterministic approach is also known as the site-specific method. It is based on applying basic phenomena (line-of-sight, reflection, diffraction, etc.) to the specific site physical layout. It does not require extensive measurement. However, a large database of the environment and long computational time are required. Its result is generally more accurate than other approaches [24], [25], [26]. Models which address the outdoor and indoor environment will be introduced briefly.

Two of the most common outdoor models will be examined in brief to provide some idea of the modeling process. The Longley-Rice Model is a deterministic model, though a very primitive one. The Okumura model and its variants are statistical or empirical models.

2.1.2.1 Longley-Rice Model

The Longley-Rice model (1968) was developed to determine propagation loss for paths with limited available information defining the terrain. This formulation includes a terrain profile of the propagation path, transmitter and receiver antenna heights, frequency, distance, polarization, and surface reflectivity. The model is applicable to point-to-point communication systems with frequency ranging from 40MHz to 100GHz[20]. It is basically a deterministic approach. It includes the effect of the direct ray, single ground reflected ray, as well as single and double diffracted rays. The diffractions considered are only knife-edge diffractions without taking into account the actual edge profile. Other geometrical approximations include neglecting the terrain height in calculating the single ground reflection point. Such inherent approximations deny Longley-Rice model good accuracy even if a more detailed terrain database is supplied. Therefore it is not suitable for complicated terrain like the urban (outdoor) or the indoor environment.

2.1.2.2 The Okumura Model

Okumura published an empirical prediction model for signal strength prediction following an extensive series of measurements in and around Tokyo. The basic formulation may be expressed as

$$L_{50} = L_F + A_{mu}(f,d) + G(h_{te}) + G(h_{re})$$

L_{50} is the median propagation loss in dB. L_F is the free space propagation loss. A_{mu} is the median attenuation relative to free space. $G(h_{te})$ is the base station antenna height gain factor. $G(h_{re})$ is the mobile antenna gain factor. His measurements led to a set of curves A_{mu} versus frequency from 100 to 3000 MHz (after interpolation). $G(h_{te})$ and $G(h_{re})$ versus antenna heights curves are also available. The primary limitation of all empirical models is their limited valid parameter ranges. It is also not applicable to transmission range less than 3 km. For this, although it is often employed as a prediction tool in the urban environment, it does not appear to be applicable to the Hong Kong environment.

Hata modified the graphical results of Okumura and provide an empirical formulation[**20**]. Akeyama proposed another modification to account for different degree of urbanization of the environment. In short, statistical models need modifications for different circumstances.

2.1.3 Indoor Propagation Models

The development of indoor models started much later than their outdoor counterparts.

2.1.3.1 Alexander Distance/Power Laws

The first major effort appeared in 1983 when Alexander[**27**] performed a series of measurements to characterize buildings at 900Mhz. It emerged with the new cordless phone at that frequency. It is a statistical model that considers only two parameters: distance and construction type. The path loss s in dB is given by

$$s(dB) = -m \log D$$

D is the distance of the receiver from the source. m is a loss gradient found by curve fitting from the measured data. For example m=2 for free space (same as the Friis' formula). Some examples are shown in Table 2-1:

Table 2-1 : Gradient of different building

Building	*Construction*	*Gradient m*
Office 1	Brick	3.9
Office 2 ground floor	brick/plasterboard	5.3
Office 2 1st floor	brick/plasterboard	4.3
Office 2 2nd floor	brick/plasterboard	4.8
Office 3	Plasterboard	2.8
Office 4	Plasterboard	3.7

It can be seen that the gradient m does not show consistent values from building to building and even within a building. This model does not try to distinguish the difference between different kinds of brick or plasterboard. Indeed not even deterministic models try to do this. Yet the major cause of the variation is the difference in geometrical layout. This model only considers distance from the transmitter. Later statistical models try to improve on this point by considering the number of structures as well.

2.1.3.2 Saleh Model

In 1987, Saleh and Valenzuela presented their results of multipath delay spread and attenuation measurements at 1.5GHz within a medium-sized two-storey office building[**28**]. In that same paper, they also presented a statistical model to study the impulse response. The central idea of the model assumes that received power decreases with time. The model starts with assuming cluster arrival of rays. The cluster arrival time is the arrival time of the first ray in the cluster. It is modeled as a double Poisson arrival process with some fixed rate. Within each cluster, subsequent rays arrive with another fixed rate. The amplitude of the rays has a Rayleigh distribution. It decreases exponentially with arrival time.

The amplitude of the first ray of the first cluster $\overline{G}(0,0)$ is first estimated using Friis' free space equation

$$G(1m) = \frac{P_T G_T}{4\pi d^2} \cdot \frac{\lambda^2 G_R}{4\pi}\Bigg|_{d=1m}$$

and $\overline{G}(0,0) = G(1m)r^{-\alpha}$

α is similar both in nature and value to the loss gradient proposed by Alexander. The value of α was found using line matching approach. For the transmitter in the hallway and receiver inside office rooms, α is found as 3.0. With both antennas in the hallway, α is 1.5.

A choice of ray arrival rate (per 5ns) was based solely on a statistics of measured power profile. Similarly, the power decay time constants of rays and clusters were found by curve fitting.

Their measured results showed that the indoor channel is quasi-static (very slowly time varying due to people's movements). Also the impulse response profile is virtually independent of the states of polarization of the antennas, providing there is no line-of-sight between them. This model is successful for office environment application but it has been shown unsuccessful in factory environments.**[29]**

2.1.3.3 Hashemi Experiments

In 1993, Hashemi performed a series of experiment**[30]**. He summarized some previous contradictions and suggested that these results (including Saleh's) need to be verified. First, some authors suggested root mean square delay spread dependence upon transmitter and receiver separation while others, like Saleh**[28]**, did not suggest the dependence. Hashemi found out an obvious dependence. Second, the distribution of the arrival times of individual multipath components was shown to be either a modified Poisson process or a double Poisson model (Saleh). Hashemi 's results agreed with the modified Poisson process. Third, both Rayleigh (Saleh) and log-normal distribution of the amplitude of individual multipath components had been

reported. Hashemi supported the log-normal fit. Hashemi did not provide an entirely novel way of impulse response modeling. His work is similar to Saleh's model. However, one important implication is that statistical modeling of impulse response may have to be varied from building to building.

2.1.3.4 Path Loss Models

In 1992, Seidel and Rappaport proposed a path loss model based on a series of measurements at 914 MHz in different environments including stores and offices.**[21]** As distinguished from the impulse response models of Saleh and Hashemi, this model is similar to Alexander's more primitive treatment of the mean power of received signal. This model tried to improve the inaccuracy by including more site-specific information.

The mean path loss is given by

$$\overline{PL}(d)[dB] = 20.0 \cdot \log_{10}\left(\frac{4\pi d}{\lambda}\right) + p \cdot \text{AF(soft partition)}[dB] + q \cdot \text{AF(concrete wall)}[dB]$$

where d is the distance between the transmitter and receiver. p is the number of soft partitions and q is the number of concrete walls in the between the transmitter and receiver. AF(soft partition) and AF(concrete wall) are the attenuation factor in dB introduced by each soft partition and concrete wall respectively. From their measurement results, AF(soft-partition)=1.39dB and AF(concrete wall)=2.38dB. This model also tried to deal with multi-floor attenuation, but not successful in reducing the results (between 12.9 to 16.2dB per floor) down to a simple floor attenuation factor.

It must be pointed out that this model was not an invention by Seidel and Rappaport. As early as 1986, similar measurements had been performed at 1290 MHz**[31]**. At that frequency, the attenuation loss of each concrete wall was found to be 8.5dB. Some similar path loss models have been developed after the Sedel/Rappaport model. Vallejo-Cabrejas and Batolome-Pascual**[32]** virtually borrowed the model and applied it to 1.8 GHz. Sheikh et al.**[33]** added the effect of antenna height for a breakpoint separation of greater than 17m. Rappaport et al. also

proceeded with another model for the factory environment. It was also available in a computer code called Simulation of Indoor Radio Channel Impulse-response Models (SIRCIM)[**20**].

2.1.3.5 Ray Optical Models

A major breakthrough in indoor propagation prediction is the introduction of the ray-tracing method in 1991 by McKown and Hamilton [**34**]. This method rigorously determines propagation paths along which the electromagnetic wave reaches the receiver from the transmitter. The model begins with the assumption that at high frequencies, where wavelength is much smaller than dimensions of environment features, electromagnetic wave propagation resembles that of optical light. Reflecting planes are modeled as mirrors. Rays reaching the reception point after a number of reflections are added to the received field strength. An impulse response prediction may also be made by tracking the path length and hence the delay of each arrival ray.

There are two major approaches of tracing the reflected rays: the image method and the brute-force approach [**29**]. The original McKown approach is now known as the image method. Rays are traced one after another following a queue of predestined reflection sequence. For each ray, the mirror image of the last reflection generates a new mirror image after reflecting from another surface. After a specific number of reflections are reached, the validity of the reflection sequence is determined by whether the point of reflection actually lies on the presumed reflection plane. The number of reflections experienced by a ray can be zero (line of sight) up to a specified number. The effects of rays with reflection number '0' up to a number "n" are added to give the final picture. Based on this approach Valenzuela developed his simulator and achieved a mean error of less than 5dB[**35**]. In 1995, an even more powerful simulator called WISE was developed at AT&T Bell Laboratories[**36**]. Besides predicting field strength at different locations, it is capable of optimizing the transmitting antenna placement to maximize the coverage area. In all of these programs, a detailed map of the environment is required. Wall and partition types and their physical layouts are crucial while furniture may be neglected.

Another approach of tracing reflected ray is known as the brute-force approach. In this approach, there will not be a reflection sequence that governs the trace. Rays are emitted from the transmitting antenna in all possible directions. Each ray is checked whether it intersects with a reflection surface. If the answer is positive, the reflection point is found and the original ray is split into a reflected ray and a transmitted ray. Each ray individually repeats the same intersection trace and splitting. A ray that finally arrives at a spherical region around the receiving antenna is considered as being successfully received. The radius of this sphere is not fixed but depends upon the total path length. The correct radius of sphere should be about $\theta d/2$ in two-dimensional space where d is the total path length (not the distance). Schaubach et. al chose the radius to be $\theta d/\sqrt{3}$ in three dimensional space**[24]**. If the radius is too large, the same path may be counted twice. If the radius is too small, the valid ray may not fall into the sphere.

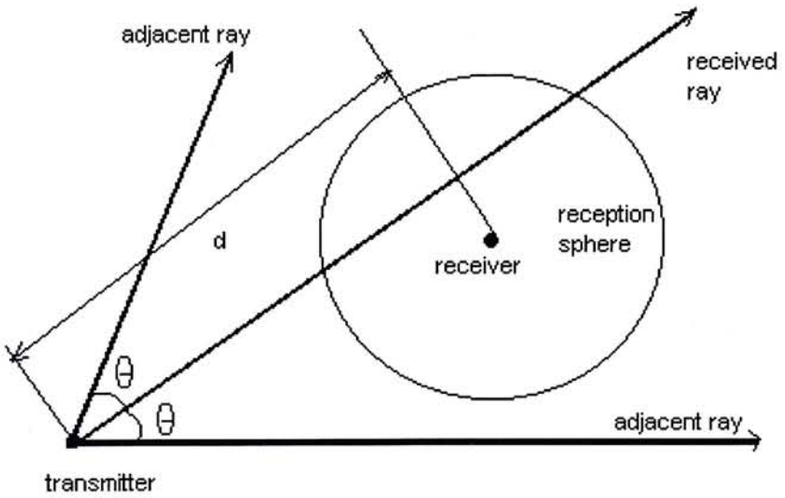


Figure 2-1 : Reception sphere illustration figure

The iterative nature of this technique poses a problem. When does the tracing of a ray stops? There are two rules which tell it to stop: (1) when the power already falls below certain threshold; (2) when it has already gone through a certain number of reflections and transmissions. One of the difficulties of this method is to maintain a constant spatial resolution. As the path length increases, the spatial resolution decreases because the angle θ is constant. That means reflecting surfaces that unfortunately fall between adjacent rays will be missed. Kreuzgruber et al. tries to ameliorate the problem by introducing a ray splitting model that splits a ray into finer rays after some additional path length**[37]**. However, this intrinsic problem cannot be

entirely solved as long as discrete rays are used to simulate a continuous plane of wave.

Although brute force approach has these theoretical problems, it is superior to image method when the simulation domain is very complicated, especially when large number of reflections is required to reach the receiving location. In image method, the propagation sequence from transmitter to receiver is exponentially increased in terms of the number of scattering surfaces in the simulation domain. This is in contrast to ray launching that is not dependent on the number of scattering surface to be considered, although the method itself may be less accurate. Also, brute force approach shows more of a modular characteristic that is very suitable for modular neural network implementation. Thus, the brute force approach is chosen in this thesis.

2.2 Ray Tracing: Brute Force approach

2.2.1 Physical Layout

The ray tracing model requires detailed description of the propagation environment. The accuracy of the prediction is highly dependent on the degree of accuracy of features like room dimensions, wall materials and furnishing that can be specified in detail. The algorithm makes no distinction between a piece of furniture or a wall or a partition. All reflecting objects are modeled as a combination of one or more reflecting surfaces. For each surface, its position and dimensions expressed in terms of relative coordinates are given. Moreover, the type of material must also be specified. Since different types of materials have very different electromagnetic properties, they have to be characterized beforehand. Notice that ideally, each of the reflecting surfaces should be characterized separately to obtain its dielectric properties such as effective dielectric constant, etc.

2.2.2 Antenna Information

Besides information of the environment, the model also requires knowledge of the transmitting and receiving antennas. The position (including height) of them must be specified. The antenna field pattern (far field) and polarization may also be specified. It is known that when no line of sight exists between the transmitting and receiving antennas, antenna polarization does not have much significance**[28]**. Nevertheless, both antenna far field pattern as well as polarization have been included in the simulator model for better accuracy. Another parameter for field strength prediction is the transmission power of the antenna.

2.2.3 Source Ray Directions

In a ray launching scheme, ray emanates from a unit sphere centered at the transmitter location. Launch points around this sphere follow a regular, computer-generated geometry. The geodesic sphere arises by tessellating on the faces of a regular polyhedron and extrapolating the intersection points to the surface of a sphere. Figure 2-2b depicts the geodesic vertices that result when the sides of an icosahedron are subdivided into smaller equilateral triangles. The geodesic vertices provide ray launching points with even angular separation around the entire sphere. Moreover, each ray will have exactly six neighboring rays that surround the original in a predictable hexagonal pattern. The reception sphere is a simple implementation for receiver-ray intersection test.

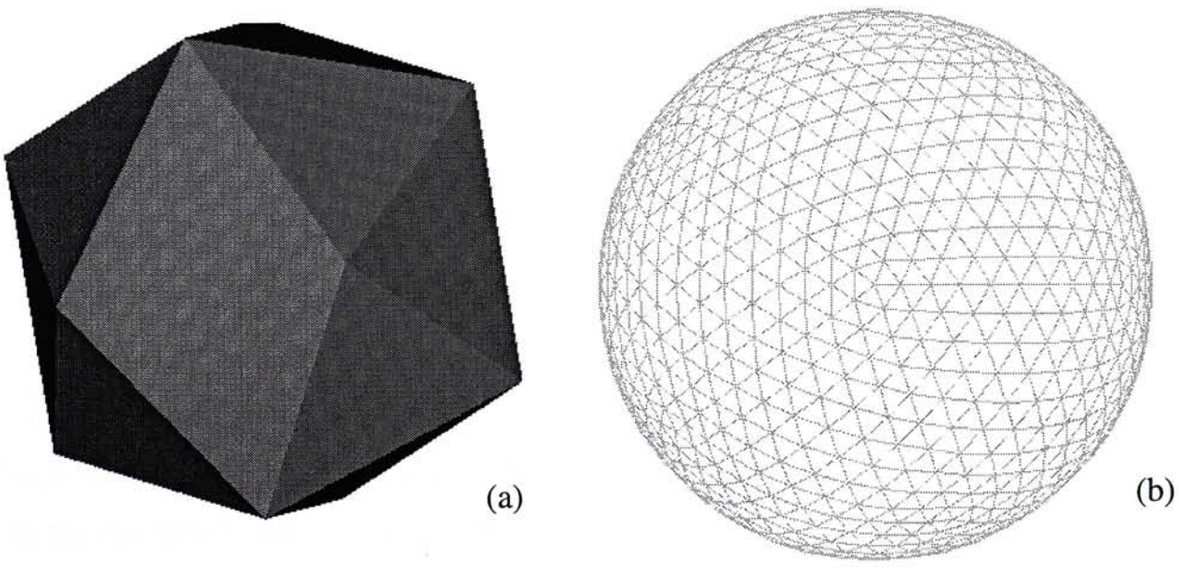


Figure 2-2(a) An icosahedron is tessellated to produce a geodesic sphere. (b) The vertices of geodesic sphere

2.2.4 Formulation

2.2.4.1 Formula of Amplitude

In ray Tracing method, all possible paths from the transmitter to each receiver will be traced and then the receiving signal can be calculated by the following equation:

$$E_i = E_o f_{ti} f_{ri} L_i(d) \prod_j \Gamma(\theta_{ji}) \prod_k T(\theta_{ki})$$

The symbols of equation are defined in Table 2-2.

Table 2-2 Summary of the variables used to describe the ray tracing propagation model

Variable	Description [units]
f_{ti}	Field amplitude radiation pattern of the transmitter antenna
f_{ri}	Field amplitude radiation pattern of the receiver antenna
$L_I(d)$	Path loss for the i^{th} multipath component
$\Gamma(\theta_{ji})$	Reflection coefficient
$T(\theta_{ki})$	Transmission coefficient
E_i	Field strength of the i^{th} multipath component [V/m]
E_o	Reference filed strength [V/m]

Ray tracing determines all ray paths along which significant levels of energy radiated from the transmitting location reaches the receiving location. Multiple receiving locations can be defined, so the procedure described here can be applied to each receiving location. Ray tracing is accomplished by an exhaustive search of a ray tree accounting for the decomposition of the ray at each planar intersection. The method traces a ray from the source in a predetermined direction and detects if an object interaction occurs. When an interaction has occurred, it divides the source ray into a transmitted and reflected ray, which are then treated in a similar fashion. This recursion continues until a maximum number of tree levels is exceeded or the energy in the ray falls below a user-specified threshold. After each interaction calculation, a segment of ray starting from the primary ray position and ending at the interaction point will be checked to see if this segment intersect with any receiving sphere. If it

does, this ray segment will be added to the receiving signal of that location.

2.2.4.2 Power Reference E_o

It is easy to understand that the received power is the transmitter power attenuated by the path loss. Finding the transmitter power by measurement at zero distance from the antenna is not practical. Instead a power reference at 1m is chosen. It is the power received by the transmitting antenna placed 1m from the transmitting antenna with an unobstructed line of sight path that coincides with the antenna patterns and polarization giving the highest power. This value corresponds to power given by one 'ray' received at a distance of 1m.

2.2.4.3 Power spreading with path length 1/d

The inverse relationship between the amplitude E_o and path length d comes from Friis' free space formula. It is stated that for free space the received power is inversely proportional to the square of path length. This may be understood as a result of the uniform spherical spreading of the transmitted power. The area of the spherical wavefront is proportional to the distance from the center that is the transmitter. The same power spreading may be inferred for a ray that is no more than a tube of wave defined by the receiver antenna area.

2.2.4.4 Antenna Patterns

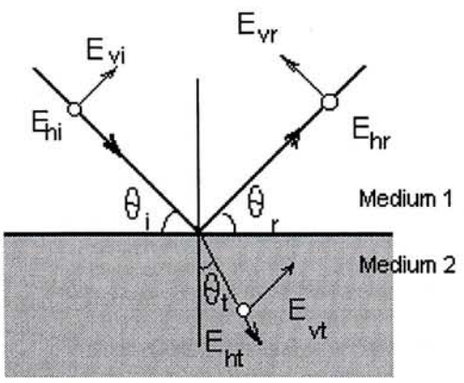
The type of antennas used for transmission greatly affects the received signal strength. This is due to the fact that different antennas have different directive gains. A three-dimensional map of the far field directive gain forms the antenna pattern or far field radiation pattern. The pattern is normalized with the peak being 'unity'. For most practical antennas, the pattern is the same for the transmission mode and the reception mode. The pattern is said to be reciprocal[24]. The far field is the region where the path length separation between the transmitting and receiving antennas exceed $2D^2/\lambda$. l is the largest dimension of the antenna. For indoor propagation considerations, the receiving antenna is usually in the far field region. For example, if

monopoles 5 cm long are used in the measurements, free space and the wavelength λ_0 at 1.8 GHz is 16.7cm, then a distance of 3 cm from the transmitting antenna is the far field region. For this reason, the far field antenna pattern is used. The transmitting and receiving antennas are modeled as two discrete points. The rays going in and out of these two 'point' antennas are then examined to determine its direction. Its amplitude is then weighted for that ray direction. The directive gain is normalized with the peak being unity. Therefore the values of G_t and G_r are less than 1.

2.2.4.5 Reflection and Transmission Coefficients

At each reflection surface, two phenomena take place: reflection and transmission. The reflection and transmission coefficients are functions of both the surface dielectric constant and the incident angle. Since reflection surfaces in reality are usually multi-layer slabs, their dielectric constants are usually given as effective values that are found experimentally. Alternatively, for each surface material, a look-up table may be created with one reflection and transmission coefficient corresponding to each incidence angle. This second method may save computational time.

The values of reflection and transmission coefficients are less than or equal to 1. For each ray, all the coefficients of reflections and transmissions it has undergone are multiplied together. The value is equal to the total path loss (attenuation factor) due to the effect of dielectric surfaces.



The angle of reflection is equal to the angle of incidence irrespective of the

dielectric constants of the two media. [4]

$$\theta_r = \theta_i$$

The angle of transmission is related to the ratio of intrinsic impedance of the two media, and non-magnetic media, the relationship becomes

$$\frac{\sin\theta_t}{\sin\theta_i} = \sqrt{\frac{\varepsilon_1}{\varepsilon_2}}$$

if medium 1 is air, the relationship may be approximated as

$$\frac{\sin\theta_t}{\sin\theta_i} = \frac{1}{\sqrt{\varepsilon_{r2}}}$$

To find the reflected field strength, the incident wave must be resolved into a parallel (vertical) and perpendicular (horizontal) component because the reflection coefficients are different for the two polarizations.

For parallel polarization

$$\Gamma_{//} = \frac{E_{//r}}{E_{//i}} = \frac{-\cos\theta_i + \sqrt{\varepsilon_{r2} - \sin^2\theta_i}}{\cos\theta_i + \sqrt{\varepsilon_{r2} - \sin^2\theta_i}}$$

and for perpendicular polarization

$$\Gamma_{\perp} = \frac{E_{\perp r}}{E_{\perp i}} = \frac{\cos\theta_i - \sqrt{\varepsilon_{r2} - \sin^2\theta_i}}{\cos\theta_i + \sqrt{\varepsilon_{r2} - \sin^2\theta_i}}$$

The transmission coefficients are also deduced similarly. For parallel polarization,

$$\tau_{//} = \frac{E_{//t}}{E_{//i}} = \frac{2\cos\theta_i}{\sqrt{1 - \frac{\sin^2\theta_i}{\varepsilon_{r2}}} + \sqrt{\varepsilon_{r2}}\cos\theta_i}$$

and for perpendicular polarization,

$$\tau_{\perp} = \frac{E_{\perp t}}{E_{\perp i}} = \frac{2\cos\theta_i}{\cos\theta_i + \sqrt{\varepsilon_{r2} - \sin^2\theta_i}}$$

For a discrete incident angle, the Brewster angle, the reflection coefficient Γ_v is

equal to zero. This angle θ_i is given by

$$\sin\theta_i = \frac{\sqrt{\varepsilon_{r2} - 1}}{\sqrt{\varepsilon_{r2}^{\,2} - 1}}$$

2.2.4.6 Polarization

The polarization describes the time-varying behavior of the electric filed intensity vector at a given point in space. Reflection and transmission coefficients have different values for different polarization. Throughout all the propagation mechanisms, the polarization vector will change as it interacts with obstacles. As the ray finally reaches the receiver, the resultant signal energy going into the receiver is the inner product of the polarization vector of incoming signal and the receiver antenna. Unmatched polarization reception may cause large power loss.

2.2.5 Mean Received Power

The primary goal of radio wave propagation model is to predict the received field strength at different reception points. Although phase does not affect the power received due to a single ray, it is important when more than one ray arrives at a single point and the difference in phase will either lead to constructive or destructive interference. The final amplitude is therefore smaller than or equal to the sum of the magnitude of all individual rays. This phenomenon is what is known as fast fading. However, in some cases, the effect of phase interference is not included for each individual point for two reasons. First, the spatial resolution of the model is not enough to accurately define variation within such a small spatial difference. Second, in practice, the receiving antenna has finite dimensions; the effect of fast fading is not as conspicuous as that for discrete points used in the simulation. Therefore the model uses a root mean square value for the mean field strength (amplitude) received to average out the effect of phase. The mean field strength $\overline{E_R}$ is the square root of the sum of power of all individual rays E_k.

$$\overline{E_R} = \sqrt{\sum_k E_k^{\,2}}$$

2.2.6 Effect of Thickness

Theoretically, when a wave travels through a wall in transmission, the direction of the transmitted wave remains the same as the incident wave except for a lateral position shift. This lateral shift is neglected here because otherwise the process of determining the reflecting paths will be unnecessarily complicated. A time consuming iterative algorithm may have to be introduced with exponential increase of computational time. This seemingly casual negligence may be justified by the simple calculation below:

For example, a 10cm thick concrete wall with a typical effective relative dielectric constant of 5 is considered.

$$\frac{\sin\theta_t}{\sin\theta_i} = \sqrt{\frac{1}{\varepsilon_{eff}}} = 0.447$$

For an incidence angle of θ_i=45°, the transmission angle inside the slab is θ_t=18.4°

the lateral shift is

$$\Delta = 10cm \cdot (\tan\theta_i - \tan\theta_t)$$

$$= 6.7\text{cm}$$

For ten such consecutive transmissions, the shift is still much less than 1m. Note that the ray that has propagated through ten such thick concrete walls is already attenuated by about 80 dB and usually does not contribute much to the total power. Another reason is that reflection tends to level out the lateral shift. Therefore it is assumed that transmission only affects the ray amplitude without disturbing its propagation path.

2.3 Neural Network

Basically, neural network is formed by simple elements called neurons connected by weights. Each neuron performs a weighted sum of all input signals and put the

value through a nonlinear transfer function. These operations including multiplication, summation and transfer function, can be performed most efficiently in digital signal processor or ASIC. Moreover, the massively parallel nature of neural network allows great parallelism in operation.

The main idea of neural network is that local connection achieves global activities. The signal passing through the network may undergo activation or suppression. Radio wave propagation has the characteristics that local propagation mechanism achieves global radio distribution. The signal in radio wave propagation is an electromagnetic wave. Therefore neural network is quite similar to radio wave propagation in this sense. However, this similarity is not enough to allow us to easily set up a neural network to model radio wave propagation.

The neuron model of most networks is a simple processor, which calculates the weighted sum of input real numbers and passes it through a differentiable activation function. Apart from real number calculation, complex number or vector number can be implemented in the neuron model to handle even higher dimensions problem. Recently, fuzzy neuron has been used as a basic element of neural network to construct a fuzzy neural network.

2.3.1 Architecture

Although neural network is constructed by simple elements called neurons, there are many different types of neural network architecture. They are introduced, for different purposes and they need specific training methods. Some typical architecture are describing here:

1. Multilayer Feedforward Network - Multilayer Perceptron, Radial Basis Network
2. Recurrent Network
3. Self Organization Map
4. Fuzzy ARTMAP
5. Modular Neural network

2.3.1.1 Multilayer feedforward network

Multilayer feedforward network is a universal function approximation that can model any non-linear function as well as linear function. Propagation itself is a linear problem that is easily implemented by Multilayer feedforward network. There are many different kind of feedforward network available for modeling such as Multilayer Feedforward Perceptron and Radial Basis Network. When the input parameters that the problem depends on are clearly defined, existing electromagnetic wave techniques or physical propagation measurement can be used to collect the training data. By training the network with a set of input-output pair, a model can be made to simulate the mapping from the input parameters to the output parameters that are desired in the propagation problem.

2.3.1.2 Recurrent Network

Recurrent Network is a dynamic non-linear network that can model time domain phenomenon. It is similar to the multilayer feedforward network except there is some feedback paths from the output or the hidden layers to input layer. Time domain response is another important information used in electromagnetic wave propagation. For example, It is useful if impulse response of any location can be modeled when the transmitter location and the propagation environment are known. For a static environment, the impulse response is the accumulated multi-path effect of the propagation channel. Therefore, the configuration of environment and transmitter can be input to the recurrent network and it will then provide the impulse response of the required receiver location. In most cases, the valid solution will be available when the recurrent network reaches an equilibrium state. The dynamic nature of recurrent network can then be used to provide time series result of a dynamic environment.

2.3.1.3 Fuzzy ARTMAP

Fuzzy ARTMAP is a hybrid model that combines ARTMAP network and fuzzy logic control. ARTMAP network is a clustering tool to divide the input and output domain into different regions so that each input region is mapped to a particular output region. The division process is controlled by the fuzzy logic control. It requires only one training epoch to obtain the configuration of network in terms of

connections weights. Therefore, the training of Fuzzy ARTMAP is very fast compared to the standard backpropagation network. Also, the retraining process or the process of adding additional training data into network is very convenient. In propagation modeling, if the existing simulation modeling is used to generate the training data, plenty of training data is available for network training. The question is how to choose a set of suitable training data for problem modeling. Extra training data can be easily added when the network model cannot fulfill the specification.

2.3.1.4 Self organization map

Self organization map is a different kind of neural network from the previous ones. It is an unsupervised network that can self-organize in such a way that they compete to one another for limited resources. For examples, if a fixed number of base stations available in the simulation domain is defined and the element of self organization map represent a grid in the simulation domain, they will compete to become the member of the available base stations according to a pre-defined rule. After the network reaches a stable state, the location of base station can be obtained from the self organization map. Here, the propagation result of each element in a self organization map should be calculated from the propagation model that may be existing propagation prediction techniques or the above neural network models.

2.3.1.5 Modular Neural network

Modular Neural network is a combination of neural networks. A neural network can be decomposed into different modules according to the input domain and inner function. If the neural network is decomposed by input domain, each neural network handles a subset of input data and one specific neural network module only maps the corresponding input region to the respective output region by a specific mapping function. The overall neural network will actually contain modules of different mapping functions for their respective regions.

The other way to decompose a neural network is in terms of inner function. Therefore, the connection of each member neural network module is according to functions. This method actually inserts engineering knowledge or a priori knowledge

into the development of neural network model. Although the running time of the whole neural network may be longer than a simple single neural network since the connection may contain not only parallel function but also sequential one, each module can be individually trained and debugged. The implementation of each member neural network is easy since it just handles a simple task that is a subset of the whole problem. Thus, a solution can be found even when the problem is very complicated. Another advantage is the maintenance of neural network. The network can be upgraded when the accuracy and speed of one part of algorithm is improved. Just replace the old member neural network and the whole network does not need the retaining process that cost much in the development of model.

In handling connection between member neural network, one operation that needs to be implemented is pattern rejection. Since there are some cases where the signal of one member network should not pass into the next stage. Since there is a physical connection between two member networks, even zero set is one kind of signal that can propagate through the network and the result is that the bandwidth of the whole network will be reduced. Wastebasket concept should be implemented to avoid invalid signal propagating through the network by removing it from the main path. The method is to create some terminated connection for the member neural network to throw out these invalid signals. Since the connection is physically isolated without any connection to other network, the signal can be totally discarded.

Availability of inside probing and easy development being the advantages of modular neural network, the next job is to make use of neural network concept in radio wave propagation modeling. Here, a novel modular neural network architecture has been proposed. Basically, it is a hybrid model between neural network and ray tracing model. The concept is described briefly in the following: First, brute force ray tracing algorithm is divided into different modules according to the function and physical relationship. The original problem now becomes many small modules and these modules can be individually solved. For each module, a suitable neural network model is chosen. After training and confirming that each neural network can model the sub-problem with an acceptable error, they are connected together to form the total network, which may be trained again in terms of network connection, while fixing the architecture and weights of each module. Using this methodology, the large training

effort is broken up into small pieces so that the training process of each module can be done in a concurrently fashion. After preparing different modules by individual training, the basic architecture is almost completed. Second round training is relatively simple since it just involves minor adjustment in the overall network. A large training process is now broken down into small training processes. Moreover, the network is an alternative way of performing brute force ray tracing so that the different steps of network operation can be clearly identified instead of just a chain of multiplication and summation that cannot be understood physically. In other words, having a meaningful interpretation at each part of the network implies inaccurate prediction result can be debugged and traced back to each block of the whole network. This is testability. Without debugging inside the neural network, inaccurate result can only be improved by extremely long retraining process. In the modular neural network, result can be improved by replacing incorrect block of neural network from the modular neural network.

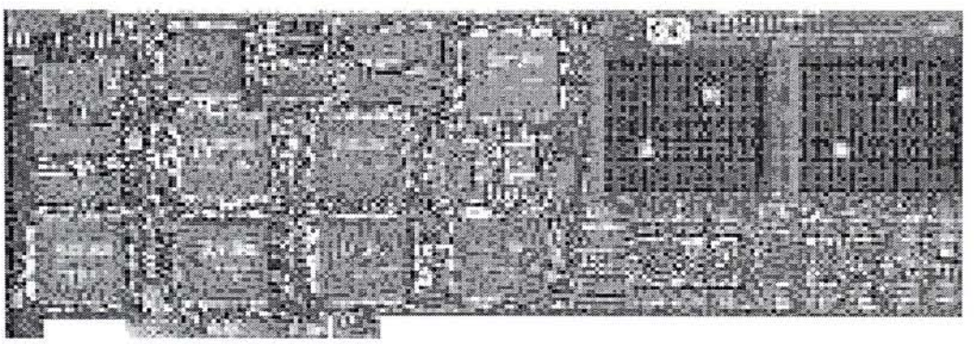


Figure 2-3 DSP dedicated neural network hardware.

2.3.2 Training Method

There is no universal training method for all kinds of neural network architecture. Even in Multilayer Perceptron, several training algorithms are available to balance between fast training and good network performance. In multilayer perceptron, backpropagation is a standard training method and there are many variations of backpropagation algorithm. Fast training algorithms are used in this thesis. There are two main categories of training algorithm. The first category uses heuristic techniques, which are developed from an analysis of the performance of the standard steepest decent algorithm. The second category uses standard numerical optimization

techniques by manipulating the Hessian matrix or approximate form of Hessian matrix. Hessian matrix is the second derivative of the performance index for the current values of weights and biases. One typical algorithm using an approximate form of Hessian matrix is Levenberg-Marquardt algorithm. The update equation is

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e$$

where x_k is the network parameters at k iteration, J is the Jacobian matrix containing first derivative of the network error with respect to the weight and biases, e is a vector of network errors, μ is a controlling scale value and I is the identity matrix. In this update equation, J^TJ is the approximate form of Hessian matrix.

There is one common problem in neural network training called overfitting. The error between desired output and network output for the training set is driven to a very small value. However, when a new data is presented to the network, this error is large. The network has memorized the training examples, but it has not learned to generalize to new situations. Therefore, besides using training method to update the weights and biases in the network, generalization improvement techniques are also used in the training process. For example, the performance, which is normally chosen to be the sum of squares of the network errors on the training set, can be modified by adding a term that consists of the mean of the sum of squares of the network weights and biases. The performance function becomes:

$$F = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 + (1-\gamma)\frac{1}{n}\sum_{i=1}^{n} w_j^2$$

where N is the number of training data, t_i is output of i^{th} training data, a_i is network output of i^{th} training data, n is number of network parameter, w_j is j^{th} network parameter and γ is the performance ratio. Using this performance function the network will have smaller weights and biases, and this will force the network response to be smoother and less likely to overfit.

2.3.3 Advantages

There are many advantages of using neural network. First, neural network is a nonlinear system. Basically, a neuron is a nonlinear device. Hence, neural network is

capable of modeling nonlinear physical phenomena. Although electromagnetic wave propagation is a linear problem, nonlinear effect exists once the microwave circuits are included in the analysis. Secondly, It is a universal approximators. Neural networks are able to approximate with arbitrary accuracy any continuous function. This makes them suitable for solving a wide variety of computational tasks. Besides the ability to model propagation, the massively parallel nature of neural network, and the simplicity of their processors make them well suited for VLSI implementation. That is why neural network is used to emulate ray launching algorithms. Finally, the uniformity of analysis and design makes all sub-net design relatively easy.

2.3.4 Definition

A neural network is a dense interconnection of computationally simple processor (i.e., neurons) that is based on the anatomy of the brain. It resembles the functionality of the brain in the following two ways:

1. Knowledge is stored in a distributed fashion as connection strengths (i.e., synaptic weights) between processors.
2. Knowledge is acquired through a learning processor that involves modification of the connection strengths between processors.

2.3.5 Software

Developing neural network model requires tools to describe and train the neural network in a logical and flexible sense. There are many programs that are available as freeware, shareware and commercial products. In fact, any computer language can be used to program neural network development. In this thesis, Matlab© neural network toolbox is used to develop all modular neural network modules. Matlab© is a common tool in engineering field to build up a mathematical prototype. With the help of neural network toolbox 3.0, individual neural network with different training methods is available in the toolbox for easy training. Also, neural network in Matlab environment is constructed as an object with flexible architecture description. Therefore, any network size and architecture can be easily constructed.

3 Hybrid Modular Neural network

Modular neural network is constructed by decomposing the brute force ray tracing algorithm into small modules. The decomposition process is possible since brute force ray tracing algorithm involves a series of sequential process for every individual ray. These sequential processes include surface-ray hit checking, surface selection with minimum distance criteria, reflection-transmission calculation and update of the received signal in each receiver. The mentioned process is just the first step for the decomposition process. Repeating the task in each process can further reduce the system into small modules. Besides the construction of each module, the limitation needs to be defined so that all the network designs can follow the same rule and fewer problems will be introduced when they are finally combined. In other words, the ranges of parameters used in the simulation need to be designed.

3.1 Input and Output Parameters

Suppose a scattering surfaces exists completely within a simulation domain of 50 m × 50 m × 50 m in dimension with six absorbing boundary surfaces at the far end to stop rays coming from any position in any direction. The boundary surfaces are chosen because there is no master control unit that monitors outgoing ray in the system. In this distributed system, a boundary surface can be used to absorb any ray that leaves the simulation region. The maximum dimensions of the scattering surface are limited by the environmental dimensions and the range of surface thickness is between 0.02 m and 0.1 m. Also, the dielectric constant of surface material ranges from 2 to 6. The allowable range should be defined before building the neural network model since extrapolation is a well-known weakness of multi-layer perceptron. The ranges of all parameters are listed in Table 3-1. The ranges of these parameters define the physical limitation of the modular neural network model. However, it is not a complete list. When going into each sub-network design, other parameters will be included to describe the problem in more detail. For example, a logical parameter should be used to describe whether an incident ray hit the scattering surface or not. In this case, 0 and 1 are used to represent hit or not hit status. In later

chapters that talks about the detail of each sub-network design, a complete description of all the network architectures as well as input and output parameters will be given.

Table 3-1 Modeling Parameter and Their Range

Aspect	Variable	Range
Domain Size	X	0 m-50 m
	Y	0 m-50 m
	Z	0 m-50 m
Tx Location	X	0 m-50 m
	Y	0 m-50 m
	Z	0 m-50 m
Rx Position	X	0 m-50m
	Y	0 m-50 m
	Z	0 m-50 m
Propagation vector	θ_d	0° - 180°
	ϕ_d	0° - 360°
Polarization vector	θ_p	0° - 180°
	ϕ_p	0° - 360°
Amplitude	A	-120 dB – 0 dB
Propagation Distance	D	4500 m
Surface Center	X	0 m-50 m
	Y	0 m-50 m
	Z	0 m-50 m
Surface Dimension (Two of Sx, Sy and Sz are nonzero at a time)	Sx	0.2 m – 50 m
	Sy	0.2 m – 50 m
	Sz	0.2 m – 50 m
	Thickness	0.02m–0.1 m
Surface Characteristics	εr	2 -6

Although the simulation domain is fixed after the ranges of parameters are set in the modular neural network, not all networks need to be redesigned if another large environment is simulated. Later some neural network design that is invariant to the dimensions of simulation domain will be pointed out. Therefore, a new set of neural networks for a larger simulation domain can be constructed with some modules trained previously reused.

3.2 Architecture

Suppose the transmitter emits N rays in all directions with vertical polarization. First, the separation distance matrix between all the scattering surfaces and N rays is calculated as shown in Figure 3-1. If a ray hits a surface, dij represents the separation distance between hitting point on scattering surface i and source position of ray j. Otherwise, dij is zero.

	R_1	R_2	R_3	R_4	R_5	R_6	:
S_1	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}	
S_2	d_{21}	d_{22}	d_{23}	d_{24}	d_{25}	d_{26}	
S_3	d_{31}	d_{32}	d_{33}	d_{34}	d_{35}	d_{36}	
S_4	d_{41}	d_{42}	d_{43}	d_{44}	d_{45}	d_{46}	
S_5	d_{51}	d_{52}	d_{53}	d_{54}	d_{55}	d_{56}	
S_6	d_{61}	d_{62}	d_{63}	d_{64}	d_{65}	d_{66}	
:							.

Figure 3-1 Separation distance matrix between surfaces and rays

The calculation procedure is:

1. Center transformation – translates the surface center to origin and carries out the same translation to any incident ray
2. Orientation transformation – rotates any surface normal to positive y-axis and carries out the same rotation to any incident ray
3. Checks if the ray hits the surface and calculates the separation distance between source position and hitting point. (the separation distance may be wrong but the calculation will be rejected by the result of hit checking network)
4. Multiplies the hit check status and separation distance in step 3. Since hit status is represented by binary value, the resultant value represents the separation distance if it is greater than or equal to the resolution threshold. Otherwise, the resultant value represents not hit if it is smaller than the resolution threshold.

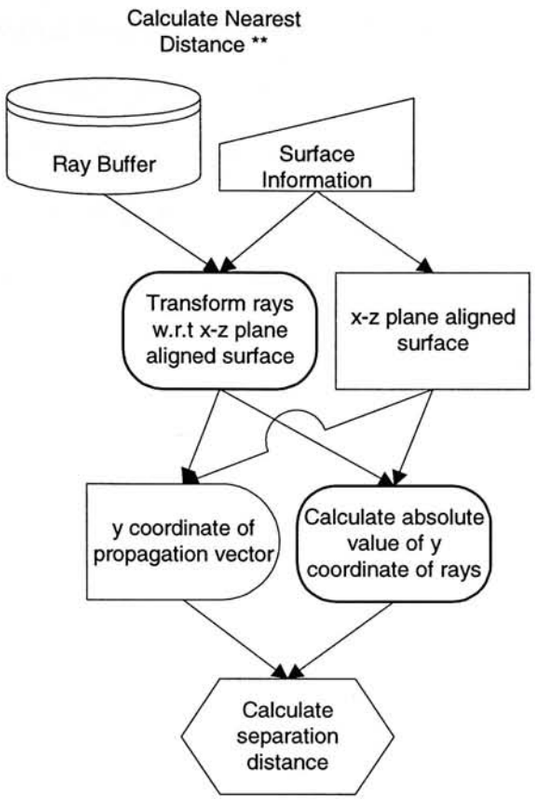


Figure 3-2 Sub-net layout in calculating nearest distance

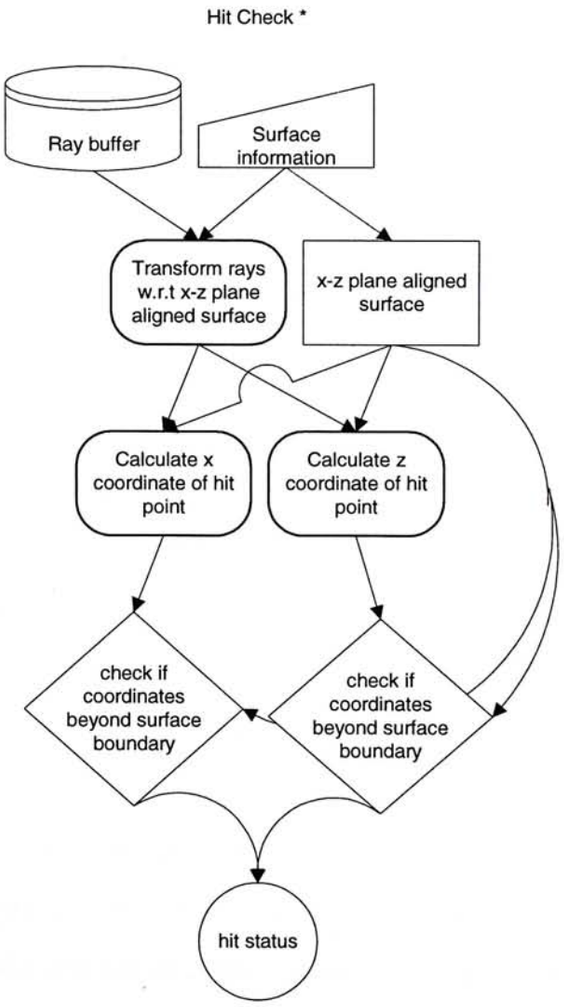


Figure 3-3 Sub-net Layout in hit checking

Then, reset any value that is smaller than the resolution threshold to maximum separation distance in that simulation domain. Select the surface for each ray if the separation distance between that surface and that ray is minimum and collect N surfaces (they may be repeated) for N rays for the next calculation.

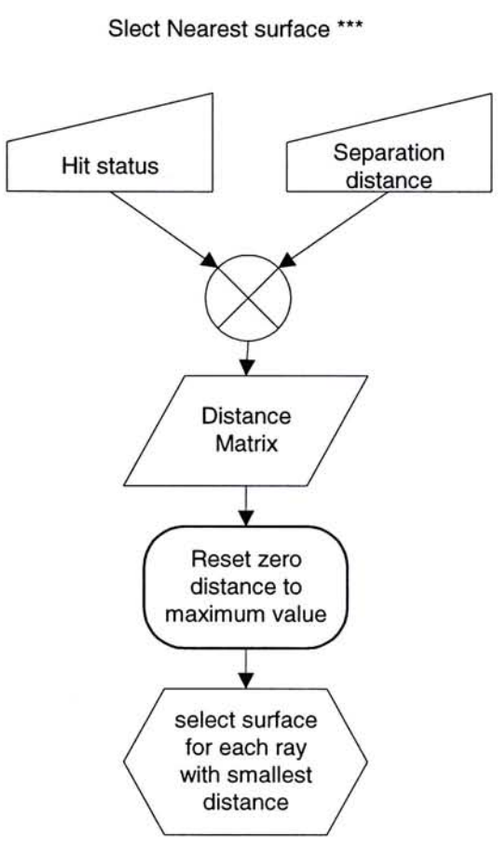


Figure 3-4 Sub-net layout in selecting nearest surface

These N rays and N corresponding scattering surfaces will be presented to the reflection and transmission ray calculation network. The following procedure is used to find out the two secondary rays (Figure 3-6):

1. Center transformation (same as previous description)
2. Orientation transformation (same as previous description)
3. Calculates the position and direction of the reflection ray and transmission ray
4. Calculates the polarization vector direction, additional amplitude loss and additional phase. The vertical and horizontal polarization coefficient will be considered in terms of the additional amplitude loss and additional phase. The resultant amplitude and phase is that of incident ray and the additional one due to reflection or transmission.

5. Calculates the propagation distance between source position and hitting position and add the propagation distance to that total distance of the reflection ray and transmission ray
6. There are 2N secondary rays to be generated

When 2N rays are obtained from the primary N rays, all receivers will be updated if the receiver interacts with the primary rays. Also, the secondary rays are filtered out if their amplitudes are smaller than the antenna-receiving threshold. The left ray will be placed in the ray buffer for next iteration. Four basic procedures will be repeated until the ray buffer is exhausted. The modular neural network architecture is shown in Figure 3-5.

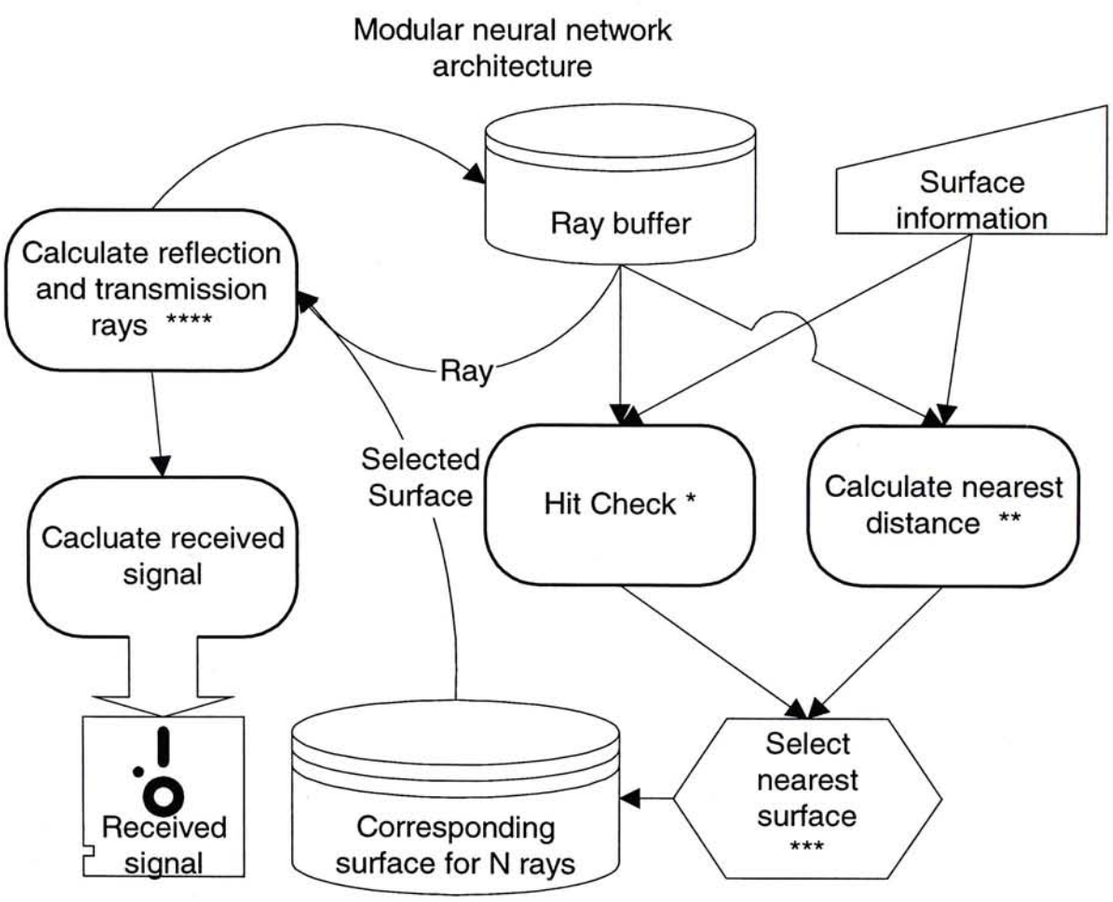


Figure 3-5 : Modular neural network architecture

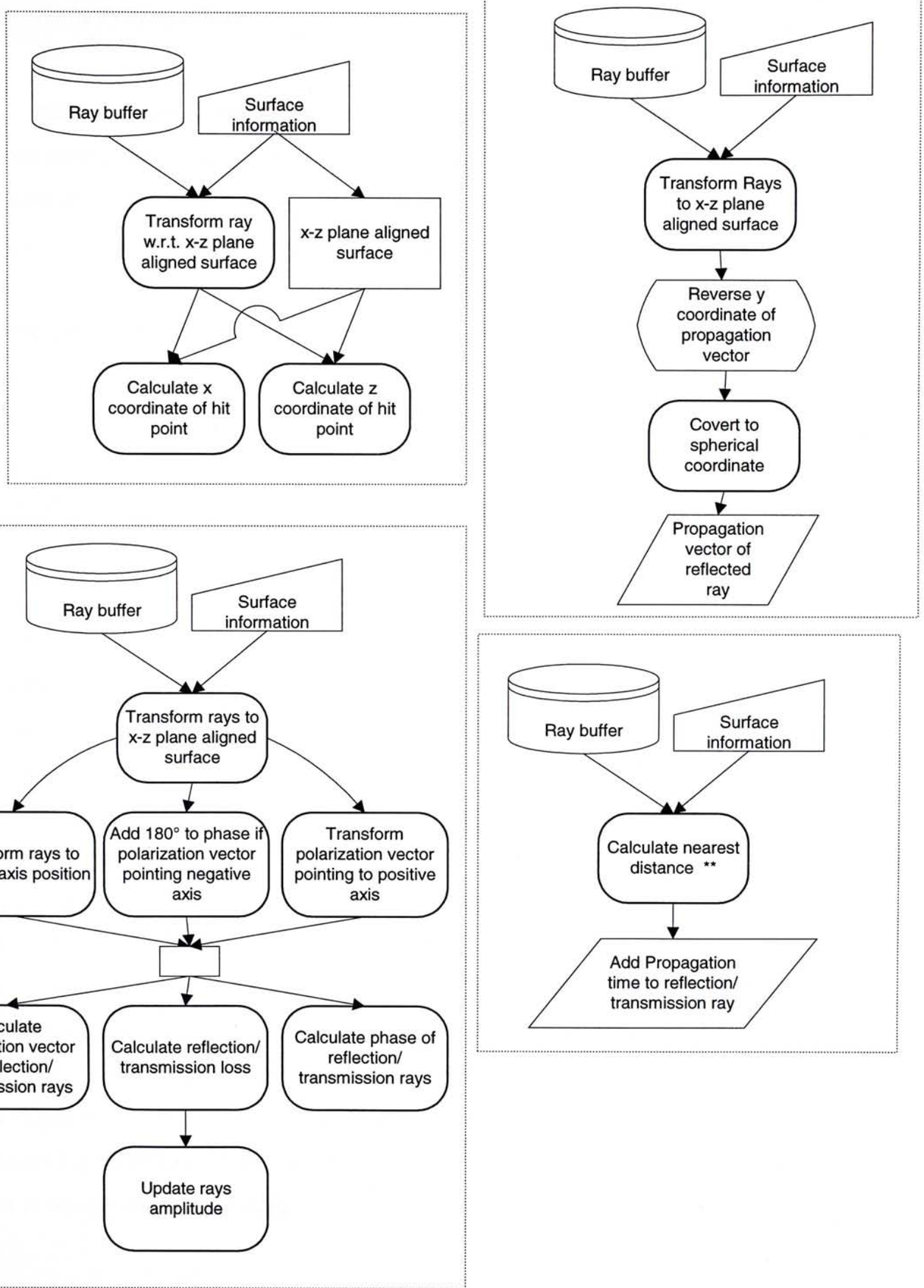


Figure 3-6 Sub-net Layout in calculating secondary ray

3.3 Data Preparation

All the training data for each module should be prepared in terms of input and output parameters. Since the model is an alternative form of brute force ray tracing method, the data preparation can be extracted from the traditional ray tracing method. Matrix generation and mapping function table look-up is written to generate the required training data. The number of input parameters and data range is specified in matrix generation configuration file. The program will first generate a set of input parameters based on n-dimensions matrix generation where n is the number of input parameters. Then, each input vector will pass through the specific function and the output vector will be calculated and written to the output file. Once the training data is prepared, the training process is ready to proceed.

After obtaining the input and output pairs for each neural network, the data needs to be normalized so that the nonlinear transfer function can operate without saturation. In one training data set, the maximum and minimum value of all input and output parameters are sorted out. Each parameter has its own boundary values. Then, linear normalization to transform the data into 0.1 to 0.9 range is carried out. Since the input range of all transfer functions is limited from 0 to1, the neural network will not go into saturation where the output of the transfer functions is constant. Training process will becomes very slow if neural network enters that region. However, such normalization generates some defects in network performance. Since the desired output range is exactly equal to the range of transfer functions output, the actual network output may be within the uncovered region, i.e., smaller than 0.1 and larger than 0.9. After de-normalization, this output will be out of the physical range and propagate to another stage in the modular neural network. Therefore, in some cases, another layer needs to be added after the output layer to remove these network output that is beyond the desired range.

3.4 Advantages

As each network has its position and meaning in the modular neural network architecture, inaccurate prediction can be back traced module by module in order to find out which one contributes to the incorrect prediction. This is testability. Without debugging inside the neural network, inaccurate result can only be improved by extremely long retraining process. In the modular neural network, result can be improved by replacing incorrect block of neural network from the modular neural network.

Immediate solution can be obtained for debugging and development during the simulation process. Each module can be trained individually so that handling large network structure and long training process can be avoided. Each module can be optimized according to the characteristics of the modeling task. For example, functional mapping and classification requires different type of network architecture.

3.5 Limitation

As the training data for neural network training is prepared by ray tracing, the performance of modular neural network is limited by the accuracy of the ray tracing algorithms that is being used for data preparation.

The range of input parameters or the applicable environment is fixed and limited after the modular neural network architecture design. The reason to limit the range of input parameters is that neural network can only operate inside the designed range. It is rather weak in extrapolation. Therefore, the training environment dimensions limit the size of environment that the neural network can handle. However, multi-session simulation is possible when the model is cascaded in a grid-based manner.

3.6 Applicable Environment

The training environment of the modular neural network in this thesis is large enough to handle most indoor environment. For example, a single floor of

engineering building in CUHK is 22 m × 25 m × 4 m. In this thesis, 8 to 10 floors indoor coverage can be simulated. However, modular neural network is not limited to this kind of simulation. There are two possible approaches to simulate larger environment. First, some neural network modules can be re-constructed that are domain dimension dependent. In the modular neural network design, two modules are domain dimensions dependent: the hit point calculation and the propagation distance calculation. The input and output parameters of these modules should be adjusted according to the maximum simulation dimensions. One point should be noted here. When the range of input and output parameters is enlarged, higher accuracy in network performance is needed to maintain the modeling resolution. Since all the data in the modular neural network is normalized from 0 to 1 no matter how large the parameter represent, the effective resolution will be lower down for the same prediction performance in terms of mean testing error. The second approach is Multi-session modular neural network. Since there are six absorbing boundary surfaces to define the simulation region in one modular neural network, larger environment can be represented by multi-session of modular neural network and their connection is the common absorbing boundary surfaces. By using ray routing mechanism between two successive modular neural networks, the large environment can be divided into several blocks. Thus, modular neural network can be extended to larger environment simulation.

4 Individual Modules in Hybrid Modular Neural Network

There are all together 18 neural network sub-nets in this modeling. All of them are Multilayer Perceptron. Apart from input and output layers, there are at most two hidden layers in all neural network sub-nets. Their sizes are not fixed but depend on their respective functions. In some cases such as absolute function modeling, the network parameters are calculated so their sizes are minimal. In another case, training process is needed to prepare the neural network. When the networks are trained and tested, these networks can be minimized by applying some advanced techniques to eliminate the redundant weights and hidden neurons. However, in this thesis, the sizes of networks have not been optimized. When the networks require training, Levenberg-Marquardt algorithm with bayesian regularization is mainly used since the trained network has better generalization capacity than other algorithms[**38**]. A detailed discussion of the use of Bayesian regularization, in combination with Levenberg-Marquardt training can be found in [**39**].

Problem transformation technology has been widely applied in modular neural network design. There are two reasons to apply problem transformation:

- simplify the structure of the required model
- avoid discontinuity in the underlying function since discontinuity may cause defects in neural network approximation of the function

By using suitable problem transformation, the required neural network can be simplified and the training process can be minimized.

There are three techniques used in neural network design. First, position parameters are in Cartesian coordinate system while direction parameters are in spherical coordinate system with unit radius. Spherical coordinate system with unit radius is used to reduce the necessary domain in vector direction modeling. Secondly, modeling domain should be reduced as small as possible by transformation and reducing redundant input parameters so that the performance of neural network can be improved. For example, in modeling the phi parameter in spherical coordinate from x and y in Cartesian coordinate, considering the first quarter is easier than considering

all quarters at the same time. Finally, additional constraints can be applied in individual neural network to reduce the domain size and complexity. In the hit point calculation, the possible valid positions are found on the maximum dimension surface. Any hit points outside this range are classified as invalid hit point location. When this restriction is applied, hit point can be limited within a certain value even if the ray does not hit the surface or hit at a point that is far away from the surface. Thus, the dynamic range of neural network output will not be too large.

These 18 neural network sub-nets are then used to construct nine different modules that are the basis of hybrid modular neural network model. Some neural network sub-net may be used in several modules while some of them are unique components in particular module. In the following sections, these nine modules will be described in detail. All neural network sub-nets in each module will be described as a whole module so that a clear picture will be given. The input and output design of each individual sub-nets will be stated for reference.

4.1 Conversion between spherical coordinate and Cartesian coordinate

Spherical coordinate is chosen to represent all direction parameters in order to reduce the modeling domain. However, cartesian coordinate is still used in some neural network design because they perform better in modeling. Neural network "tools" to convert between spherical and cartesian coordinate is necessary.

4.1.1 Architecture

There are five neural networks involved in coordinate transformation. Three of them are used to calculate the phi value in spherical coordinate from cartesian coordinate. Two of them are used to calculate the cartesian coordinate (x, y, z) from spherical coordinate. The last one is used to calculate the theta value in spherical from cartesian coordinate. In phi calculation, the phi angle in the first quarter is first calculated by using the absolute x and y coordinates. At the same time, another neural network is used to determine the actual quarter of the x and y value. This

network contains four-output signals that represent the quarters that the phi coordinate belonging to. One of the outputs will be 1 and the others will be 0. Finally, the last network will use the quarter information and the equivalent phi angle in first quarter to construct the actual phi angle. There are two reasons to implement such complicated structure. First, phi angle is calculated from x and y value which are the coordinates of any point in the unit circle. One phi angle can be equivalent to multi-pair of x any y value (they are different in radius). This input domain is so large that a single network cannot perform well. Second, phi angle varies from 0° to 360° but the region near to 360° is neighborhood to the region near to 0°. Therefore, similar x and y value may map to totally different phi angle and single network is difficult to model such large nonlinear effect. As a result, the region is divided into quarters and a common neural network is used to find out the equivalent phi angle in the first quarter. Then, the corresponding phi can be easily found with the quarter information.

4.1.2 Input and Output Parameters

In Table 4-1, The input and output parameters as well as the ranges of all neural network modules in coordinate conversion are shown.

Table 4-1 Input and output parameters of sub-nets in cartesian to spherical coordinate conversion

Network in converting cartesian coordinate to theta in spherical coordinate			
Input Parameters	Range	Output Parameters	Range
Z	-1 ~ 1	Theta	0° ~ 180°
Network in calculating phi value in first quarter from x and y component			
Input Parameters	Range	Output Parameters	Range
X	-1 ~ 1	Phi value in first quarter	0° ~ 90°
Y	-1 ~ 1	N/A	N/A

Network in classifying which quarter does the phi belongs to			
Input Parameters	Range	Output Parameters	Range
X	-1 ~ 1	In first quarter	0 / 1
Y	-1 ~ 1	In second quarter	0 / 1
N/A	N/A	In third quarter	0 / 1
N/A	N/A	In forth quarter	0 / 1
Network in calculating the actual phi value			
Input Parameters	Range	Output Parameters	Range
X	-1 ~ 1	Actual phi	0° ~ 360°
Y	-1 ~ 1	In second quarter	0 / 1
In first quarter	0 / 1	In third quarter	0 / 1
In second quarter	0 / 1	In forth quarter	0 / 1
In third quarter	0 / 1	N/A	N/A
In forth quarter	0 / 1	N/A	N/A

Table 4-2 : Input and output parameter of cartesian to spherical coordinate conversion module

Input Parameters	Range	Output Parameters	Range
X	-1 ~ 1	Theta	0° ~ 180°
Y	-1 ~ 1	Phi	0° ~ 360°
Z	-1 ~ 1	N/A	N/A

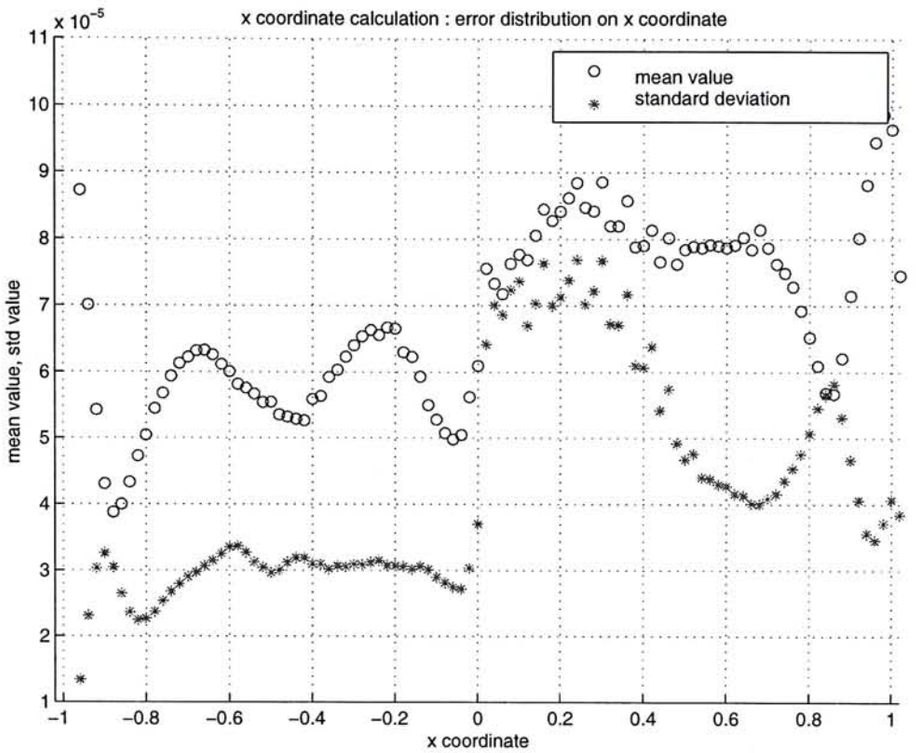
Table 4-3 : Input and output parameter of spherical to cartesian coordinate conversion module

Input Parameters	Range	Output Parameters	Range
Theta	0° ~ 180°	X	-1 ~ 1
Phi	0° ~ 360°	Y	-1 ~ 1
N/A	N/A	Z	-1 ~ 1

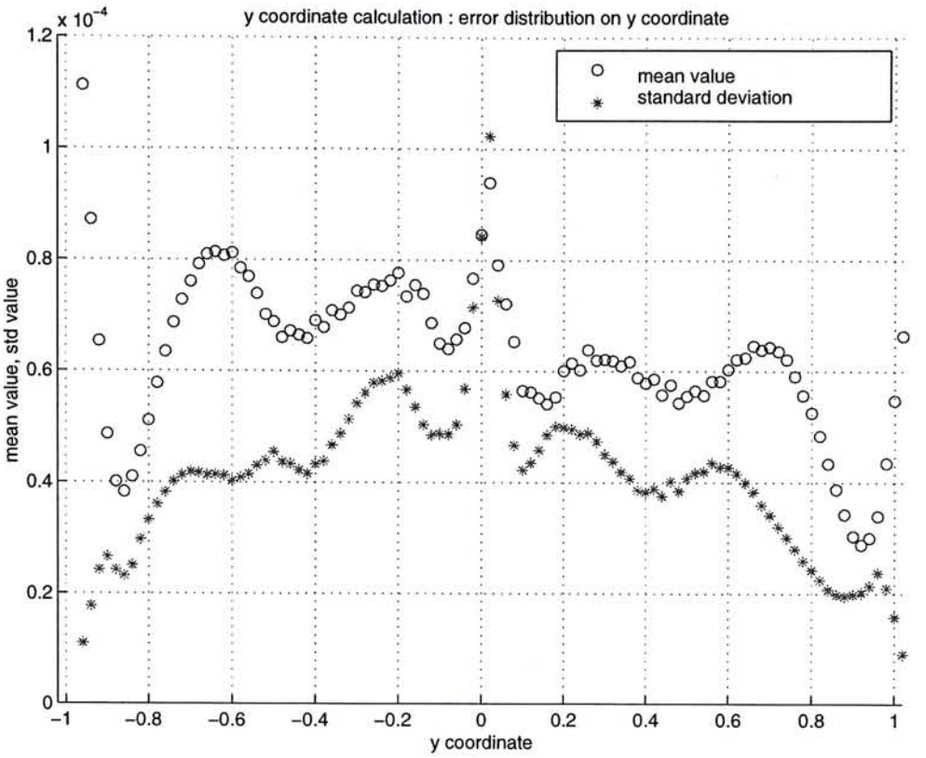
4.1.3 Testing result

80601 testing data is generated to test the performance of these neural networks. Since four neural networks are used in converting cartesian coordinate to spherical coordinate, they are combined for evaluation. In spherical to cartesian coordinate, Figure 4-1 shows the modeling error of x, y and z coordinate distributed on their desired output. The overall maximum error is 1e-3 for all cartesian

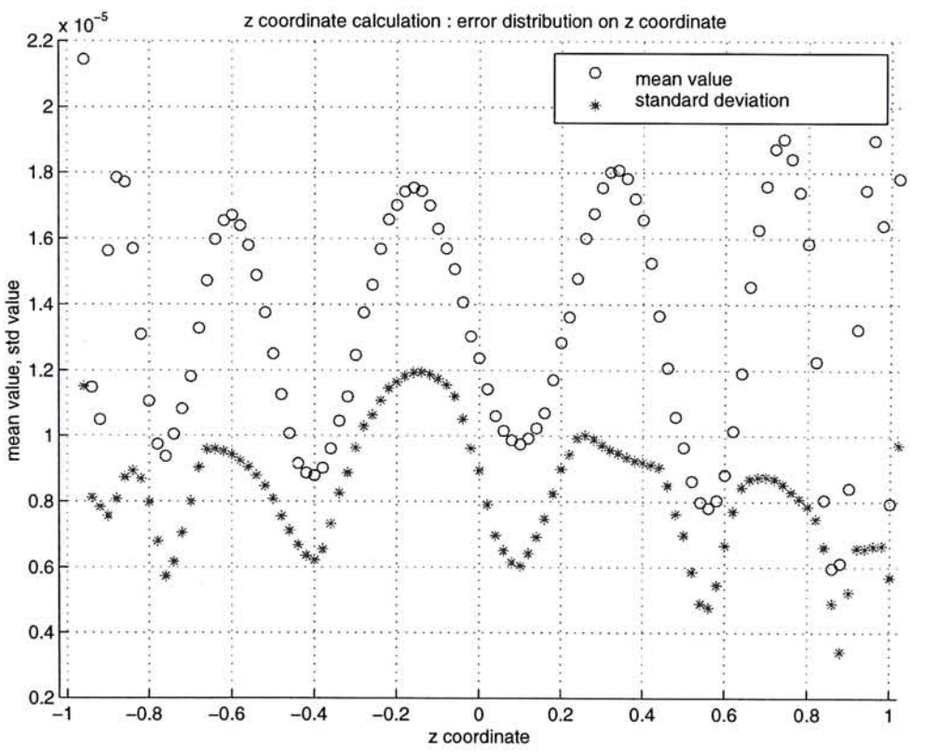
coordinate. Basically, good approximation can be achieved in this conversion except those testing data that has 0 output for x and y coordinate. In that case, testing data is located along the axis and the other coordinate has maximum absolute value (for example, when x = 0, absolute y and/or z is 1). Thus, this exceptional case affects the conversion a little bit.



(a)

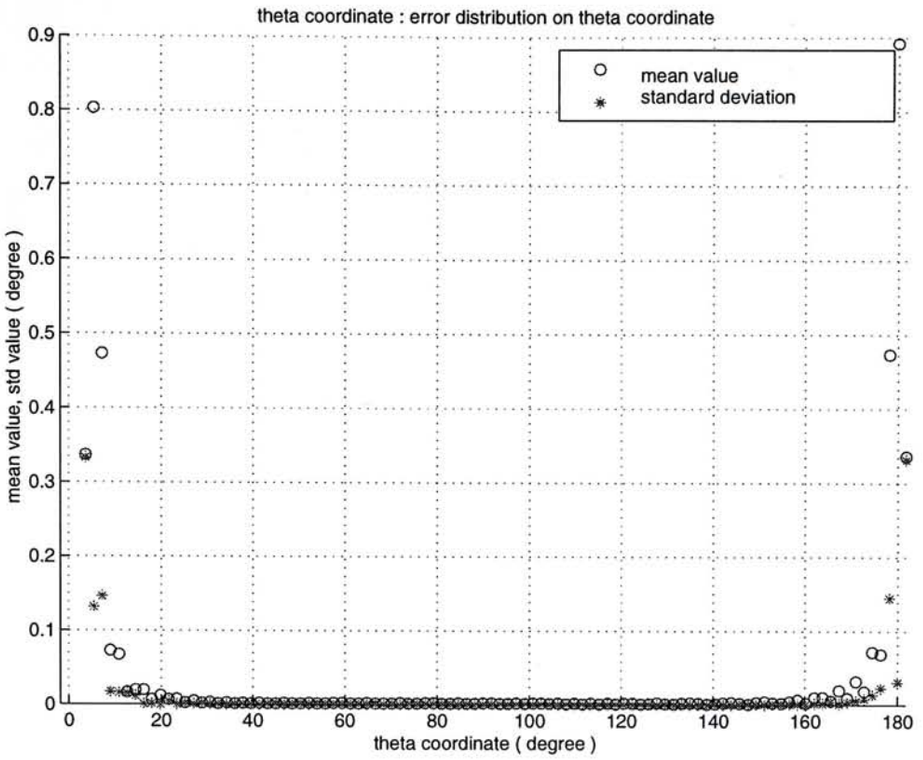


(b)

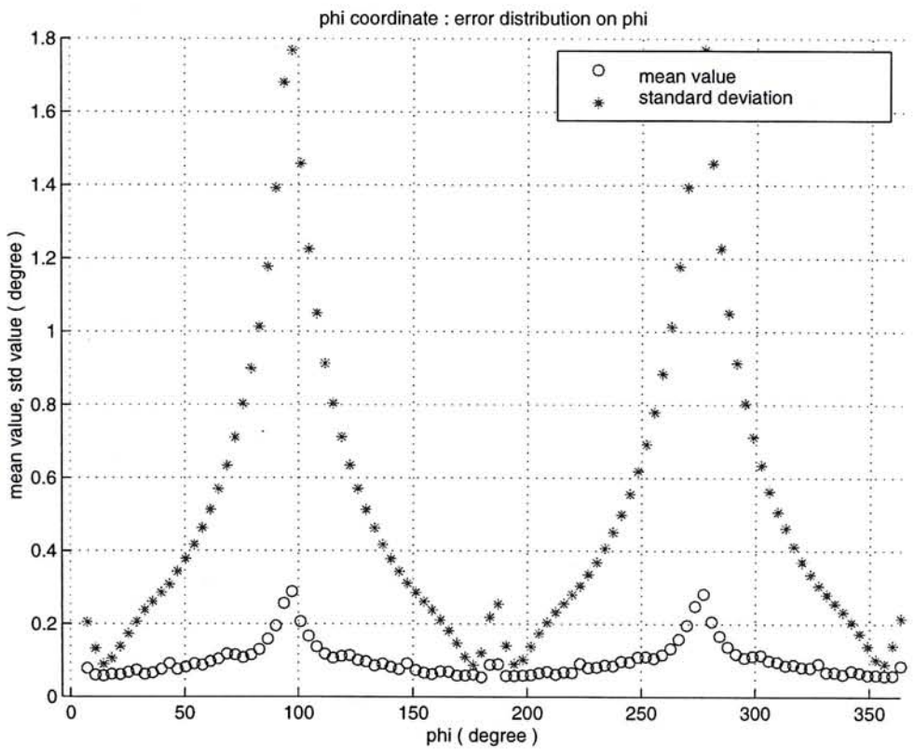


(c)

Figure 4-1 Network performance in converting spherical coordinate to cartesian coordinate for testing data. a) x coordinate b) y coordinate c) z coordinate

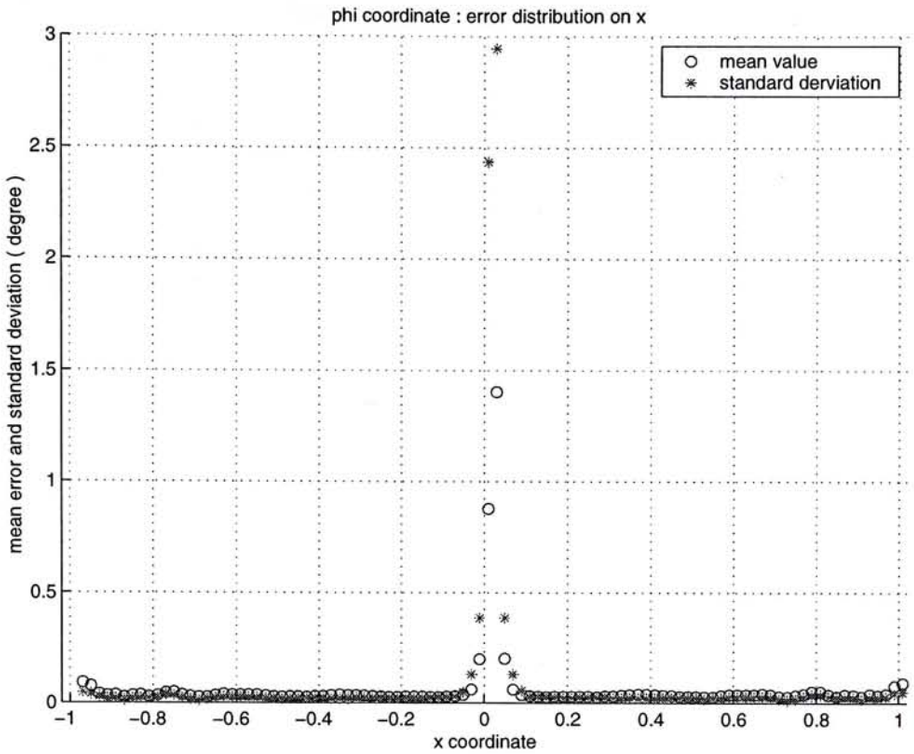


(a)

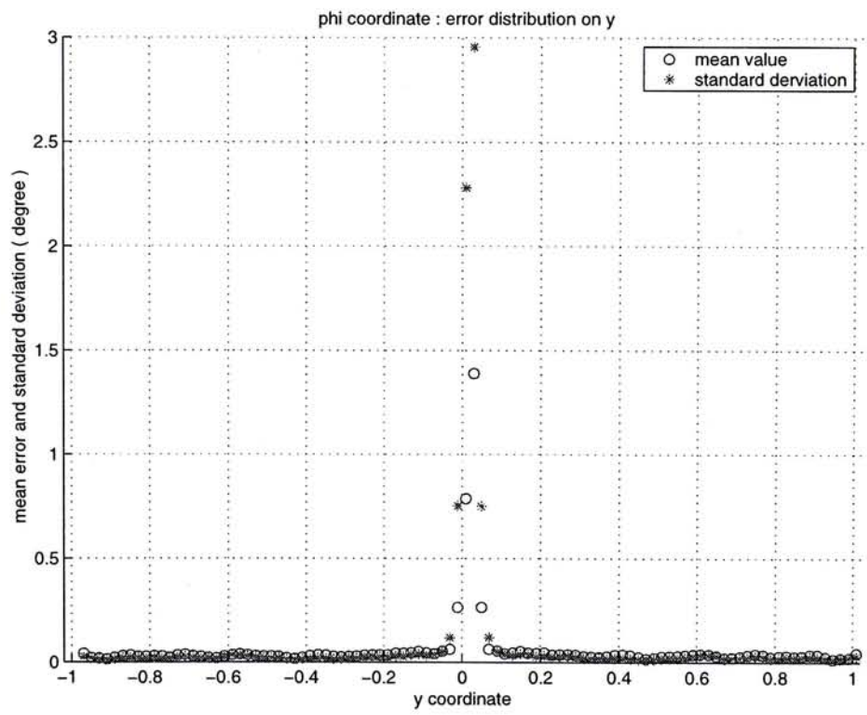


(b)

Figure 4-2 Network performance in converting cartesian coordinate to spherical coordinate for testing data. a) theta coordinate b) phi coordinate



(a)



(b)

Figure 4-3 The error in calculating phi coordinate from cartesian coordinate is plotted against (a) x coordinate of cartesian (b) y coordinate of cartesian

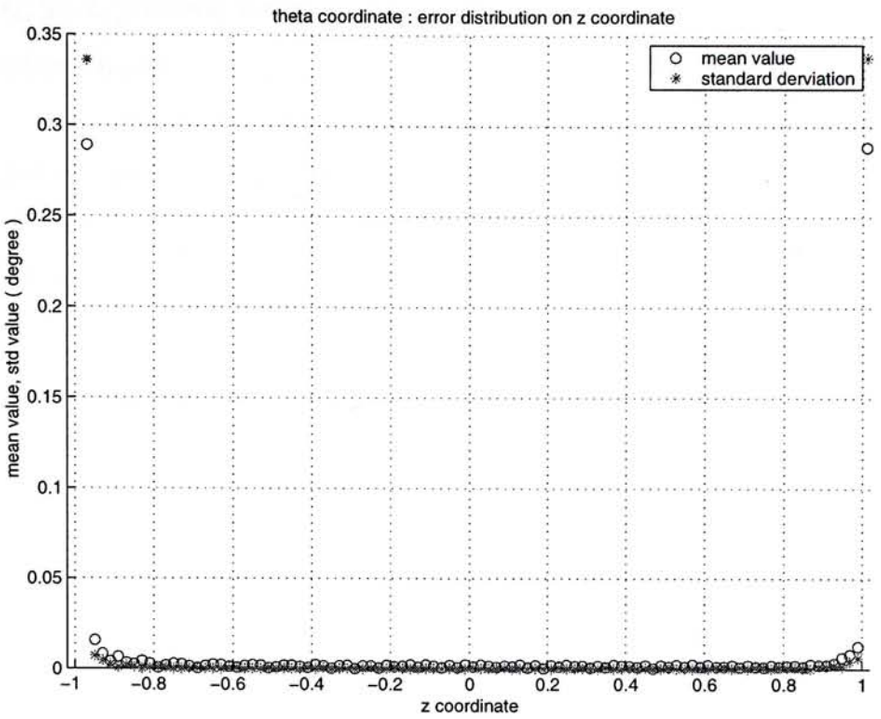


Figure 4-4 The error in calculating theta coordinate from cartesian coordinate are plotted against z coordinate

In reverse conversion, neural network cannot perfectly model the calculation, although the error is not too high. The error in theta coordinate calculation is acceptable, as the average mean error is 0.39 degree. Typically, in ray launching method, 1-degree angular separation distance is used and 40000 rays will be performed in ray tracing simulation. In phi calculation, major error occurs when the absolute value of x and y coordinates are smaller. In those situations, direction vector is pointing to positive or negative z-axis and the error in phi calculation causes little variation in the resultant direction. In fact, the phi value is a redundant parameter in the extreme case when the theta coordinate is equal to 0° or 180°.

4.2 Performing Rotation and translation transformation

Given a surface centering at x_0 , y_0, z_0 and surface normal pointing to $\hat{N}$ direction, the scattering surface and all incident ray pointing to this surface need to be transformed in such a way that the transformed surface is aligned on the X-Z plan with origin at the center. The center and orientation transformation for an incident ray

starting at x_r, y_r, z_r position with ray direction $\hat{R}$ are:

Transformation Matrix Γ_{rot}

$$= \begin{bmatrix} \sin\beta_0\cos\theta_0\cos\phi_0 + \cos\beta_0\sin\phi_0 & \sin\beta_0\cos\theta_0\sin\phi_0 - \cos\beta_0\cos\phi_0 & -\sin\beta_0\sin\theta_0 \\ \sin\theta_0\cos\phi_0 & \sin\theta_0\sin\phi_0 & \cos\theta_0 \\ -\cos\beta_0\cos\theta_0\cos\phi_0 + \sin\beta_0\sin\phi_0 & -\cos\beta_0\cos\theta_0\sin\phi_0 - \sin\beta_0\cos\phi_0 & \cos\beta_0\sin\theta_0 \end{bmatrix}$$

Center Transformation

$$[x_n \quad y_n \quad z_n] = ([x_0 \quad y_0 \quad z_0] - [x_r \quad y_r \quad z_r])\Gamma_{rot}$$

Orientation Transformation

$$[p_x \quad p_y \quad p_z] = [\sin\theta_r\cos\phi_r \quad \sin\theta_r\sin\phi_r \quad \cos\theta_r]$$

$$[p_x' \quad p_y' \quad p_z'] = [p_x \quad p_y \quad p_z]\Gamma_{rot}$$

$$[\theta_n \quad \phi_n] = \left[\cos\left(\frac{p_y'}{p_x'}\right) \quad \sin\left(\frac{p_y'}{p_x'}\right) \quad \tan^{-1}\left(\frac{p_z'}{\sqrt{(p_x'^2 + p_y'^2)}}\right)\right]$$

Since direction information is stored in spherical coordinate and they need to be converted into cartesian coordinate first and multiply it with the rotation matrix. The rotation matrix is calculated before the start of simulation since it is fixed once the location and orientation of scattering surface is specified. Since the rotation matrix is not calculated at runtime, special neural network module is not required. In the rotation matrix, transformation involves addition and multiplication that is similar to neuron-operation.

4.3 Calculating a hit point

Suppose there is a pair of scattering surfaces and an incident ray. The scattering surfaces in this section are assumed to be of maximum size. First, they are transformed into a new coordinate in such a way that the surface is aligned with X-Z plane and the center of surface is at the original. At the same, the incident ray

undergoes the same translation and rotation accordingly. Although the hit point on the transformed surface does not change with respect to the ray-surface configuration, the allowable coordinate range is changed. Therefore, the range of parameters in this module is different from the simulation environment definition. For example, the coordinate of the source position of an incident ray should be limited from 0 m to 50 m. After transformation, the allowable coordinate of the source position will be expended to –90 m and 90 m that the absolute value is approximately equal to $\sqrt{3}D$ where D is the maximum coordinate. These change affect surface dimensions too and more detail can be found in section 4.1.

4.3.1 Architecture

The calculation of x and z coordinate of hit point is separate. Since they are more or less equal to each other, there are two identical neural networks to perform coordinate calculation. First of all, the y-axis projection of the source point is obtained. Then, the hit point is calculated from the perpendicular separation distance, y coordinate and x/z coordinates of propagation vector. X and z coordinates of propagation vector are chosen as input in calculating x and z coordinates separately. A neural network module is trained to perform this operation. After the hit point coordinate is calculated for the projection case, the actual hit point coordinate is the summation of the network result and coordinate of source position.

One point should be mentioned here. The designed deflected length is extended up to 120 m instead of 90 m and any calculated deflected length beyond 90 m will be squeezed into 90 m and 120 m by an exponential function. This extension will not affect the calculation of hit point but it can be used in ray-surface hit checking. More details can be found in section 4.4.

4.3.2 Input and Output Parameters

The input and output configuration of two neural networks are listed in Table 4-4.

Table 4-4 The input and output parameters of sub-net in hit check module

Network in calculating shift distance when a incident ray coming with an incident angle			
Input Parameters	Range	Output Parameters	Range
X/Z component in cartesian of ray direction	-1 ~ 1	Shift distance from original	0 m ~ 50 m
Y component in cartesian of ray direction	-1 ~ 1	N/A	N/A
Perpendicular separation distance	0.05 m ~ 90 m	N/A	N/A

Table 4-5 The input and output parameters of hit check module

Input Parameters	Range	Output Parameters	Range
X/Z component in cartesian of ray direction	-1 ~ 1	X coordinate	-90 m ~ 90 m
Y component in cartesian of ray direction	-1 ~ 1	Z coordintae	-90 m ~ 90 m
Perpendicular separation distance	0.05 m ~ 90 m	N/A	N/A

4.3.3 Testing result

In verification, 1318761 rays will be generated from the designed range and they will propagate to a X-Z plane aligned surface with maximum dimensions (50 m × 50 m). By using the above mentioned neural networks and transformation techniques, the hit point can be calculated. The calculation of deflected distance is shown in Figure 4-5 and the error distributions on input parameters are shown in Figure 4-6. The error distributions show that deflected distance calculation has 1 m mean error when the perpendicular separation distance between source position and surface is smaller than 5 m as well as the y coordinate of propagation direction is smaller than 0.1. It is acceptable since the neural network cannot model such large variation of deflected distance in that region.

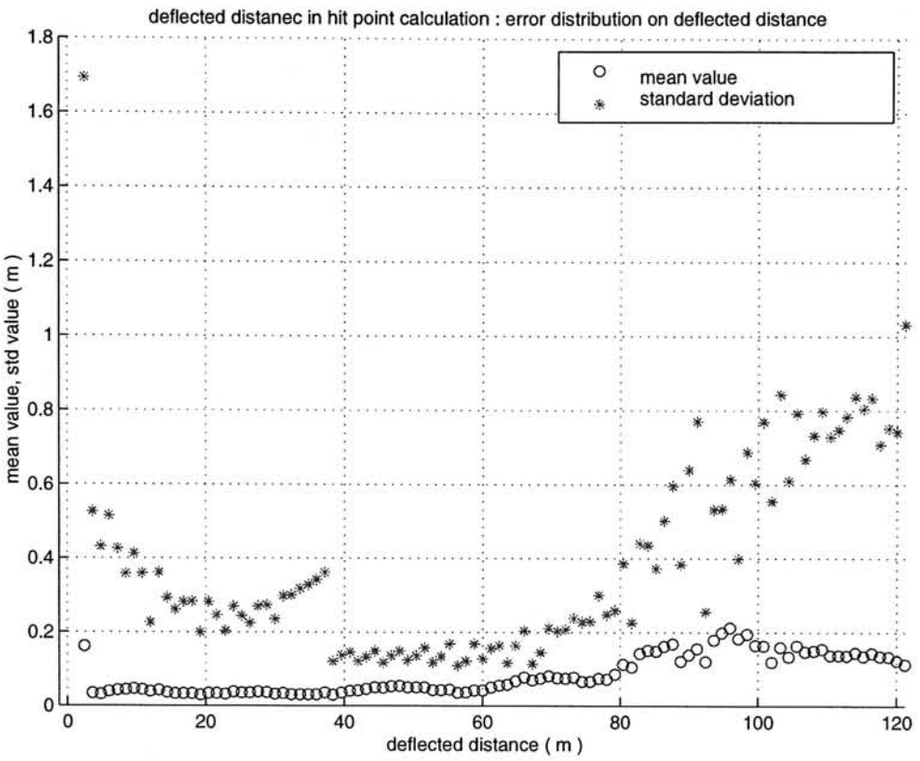
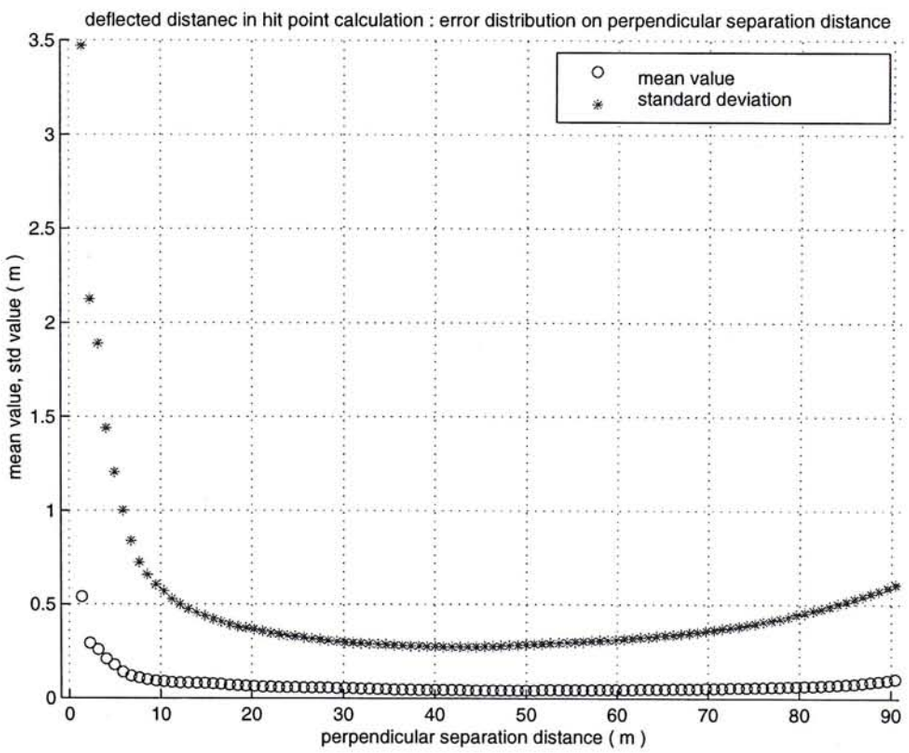
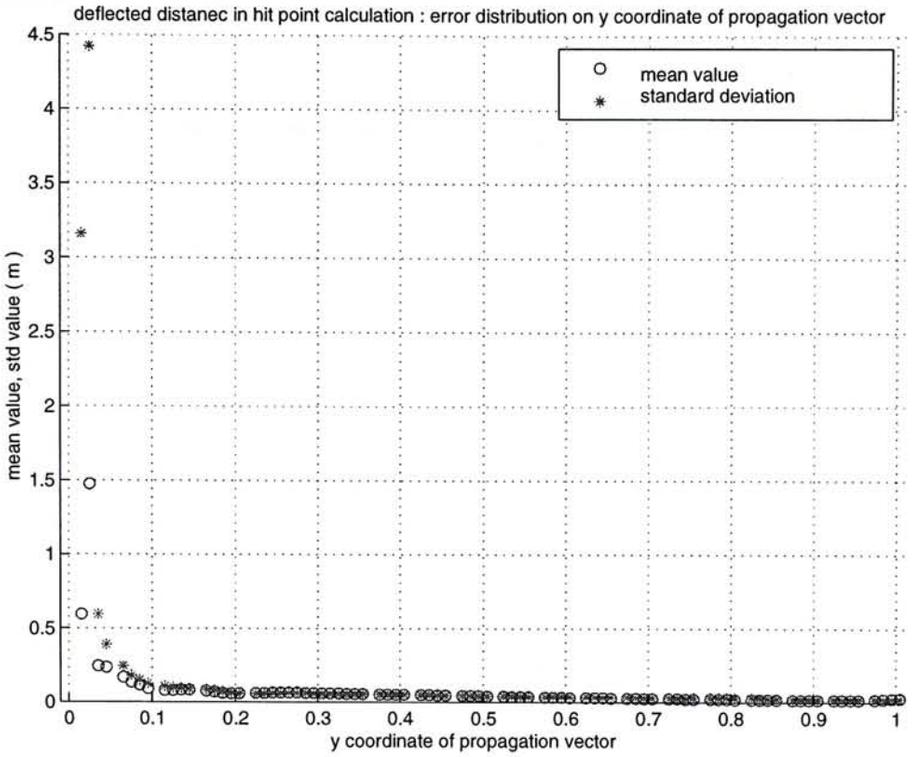


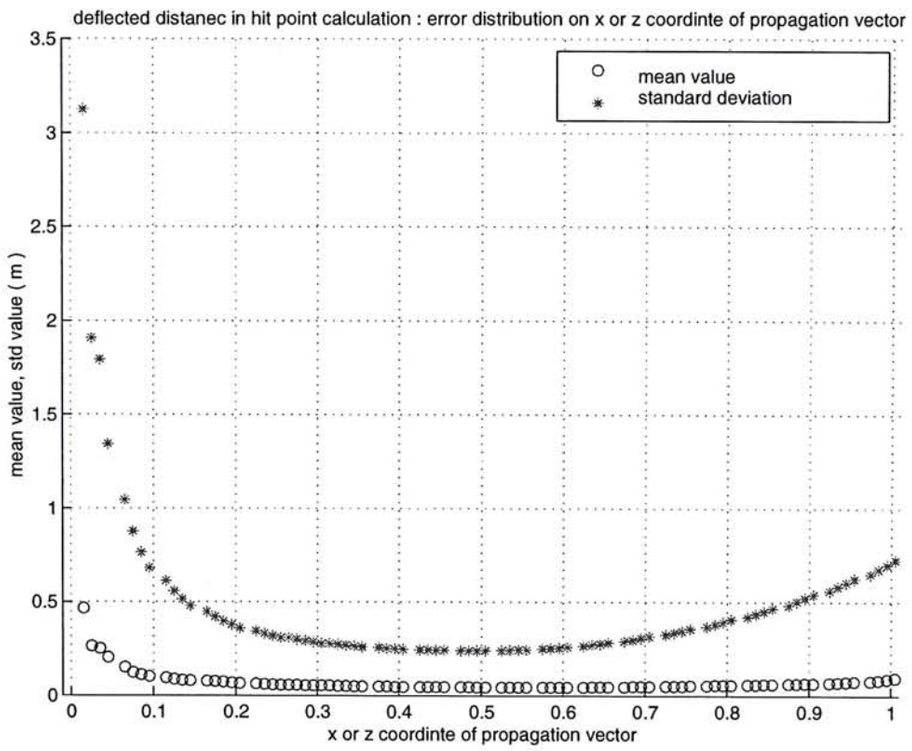
Figure 4-5 The error of calculating deflected distance are plotted against desired output



(a)



(b)



(c)

Figure 4-6 The error of calculating deflected distance are plotted against (a) the perpendicular separation distance between source position and surface (b) y coordinate of propagation vector (c) x or z coordinate of proapgation vector

4.4 Checking if an incident ray hits a Scattering Surface

There is no special module check if an incident ray hits a scattering surface but the modified hit point calculation module is utilized to do this task. When an X-Z aligned surface is considered, a ray will hit on the surface under two conditions. The first condition is that an incident ray should point toward the scattering surface. In hit point calculation, the y coordinate will not be calculated and set to be zero for all cases. The effect is that a virtual hit point on X-Z plan will be generated even if the incident ray is pointing away from the scattering surface but the x and z value of virtual point may still be inside the surface boundary. The simple method is to check if the transformed y coordinate of ray position and the y coordinate in cartesian form of ray direction vector are different in sign. In the second criteria, the calculated hit point should be within the surface boundary if an incident ray hit the scattering surface. However, hit point calculation may need to handle some coordinate output that is far away from the original. Therefore, a trick is used to avoid this difficulty by limiting the calculated coordinate that is beyond ±90 m to be within ±120 m even if the actual coordinate is much larger. Since 90 m in deflected length is the maximum case for all possible rays and surface configurations, larger deflected length does not contribute much in hit point calculation but hit check is still valid if the generated hit point coordinate is beyond the possible maximum surface dimensions. After the actual hit point coordinate is generated in transformed domain, hit status can be obtained by testing if the absolute value of hit point coordinate is smaller than the surface boundary dimensions. Based on these two conditions, an incident ray with arbitrary direction at any position inside the simulation can be checked whether it hits on a scattering surface with arbitrary orientation and surface dimensions.

4.5 Calculating separation distance between source point and hitting point

Separation distance calculation is similar to the hit point calculation. If a triangle with three vertexes is considered with the vertexes of the y-axis projection of ray position, original point of scattering surface and hit point, then two sides of that triangle are the required result in previous section, hit point calculation, and

separation distance. Therefore, same approach is applied in calculating separation distance between ray position and hit point. However, one restriction can be released in this network design. all hit points coming from maximum surface in the designed domain will be considered. Any surface-ray configuration that generates larger separation distance will be ignored in this network. Therefore, the separation distance may be wrong if the data is out of the region that is considered in the training. The correctness of this separation distance is validated by hit checking result. In the real operation, surface is first aligned with X-Z plan after the standard transformation. Then, the separation distance can be calculated from the y coordinate of propagation vector and perpendicular separation distance between ray position and surface, i.e. absolute value of y coordinate of ray position.

4.5.1 Input and Output Parameters

The input and output configurations of two neural networks are listed in Table 4-6.

Table 4-6 : The input and output parameters of sub-net in separation distance calculation module

Network in calculating separation distance between ray position and hit point			
Input Parameters	Range	Output Parameters	Range
Y component in cartesian of ray direction	-1 ~ 1	Separation distance	0 m ~ 100 m
Perpendicular separation distance	0.05 m ~ 90 m	N/A	N/A

* The input and output parameters of this module is the same as that in sub-net

4.5.2 Data Preparation

In separation distance calculation, all incident rays will be considered to hit the scattering surface with maximum dimensions. In data preparation, all possible combinations between perpendicular separation distance and y coordinate of propagation vector are generated. The corresponding separation distance is also

calculated. Then, all separation distance results that are larger than 90 m will be filtered out. Apart from this normal sampling, the sample data should be increased when the perpendicular separation distance is smaller than 10 m because the required result is changing rapidly in term of the y coordinate of propagation vector. Consequently, not all propagation direction will be considered in training and even testing data. In real simulation, input data that is out of these ranges will possibly be input to the network and thus unpredictable result will be generated. However, this unstable result is expected and hit check sub-net is used to validate the result of separation distance calculation. This simplification reduces the dynamic range of output parameter and non-linear effect (if the squeezing techniques described in section 4.3 is used) without loss in generalization capability.

4.5.3 Testing result

37901 pairs of testing data are generated to verify the trained network. The target surface is also a X-Z plane aligned surface with maximum dimensions (50 m $\times$ 50 m). Similar to hit point calculation, large error is expected when the perpendicular separation distance is smaller than 5 m while the y coordinate of propagation direction is smaller than 0.1.

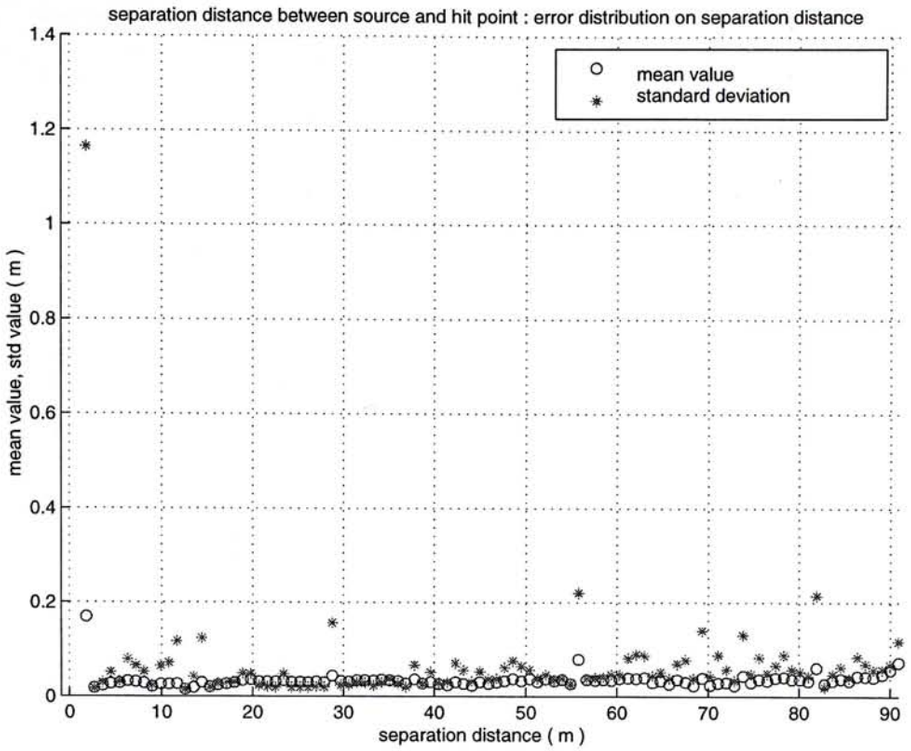
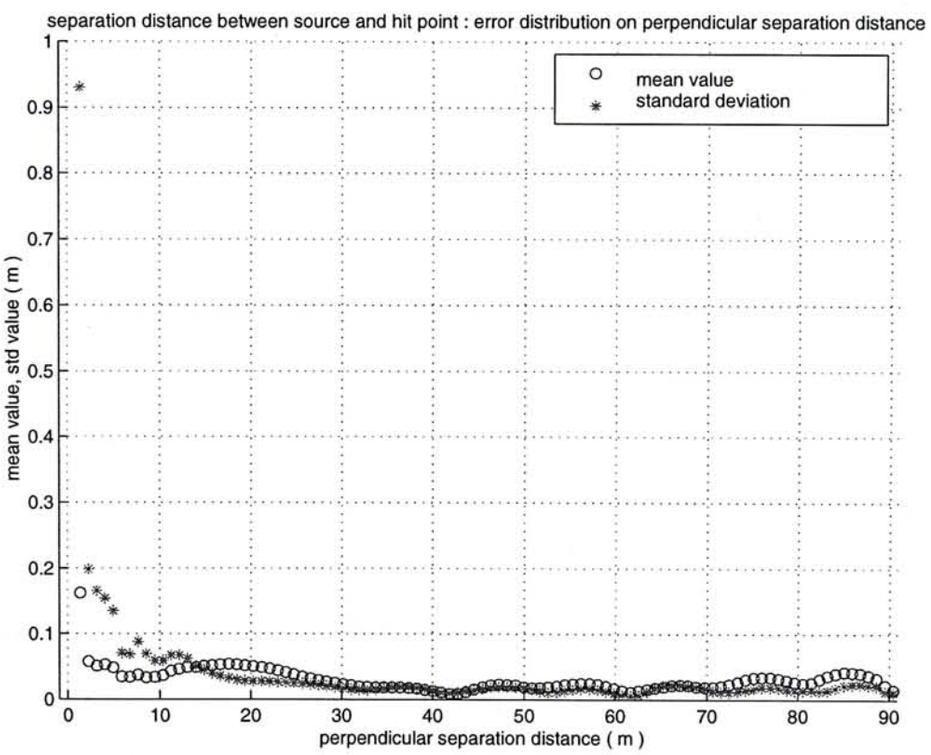
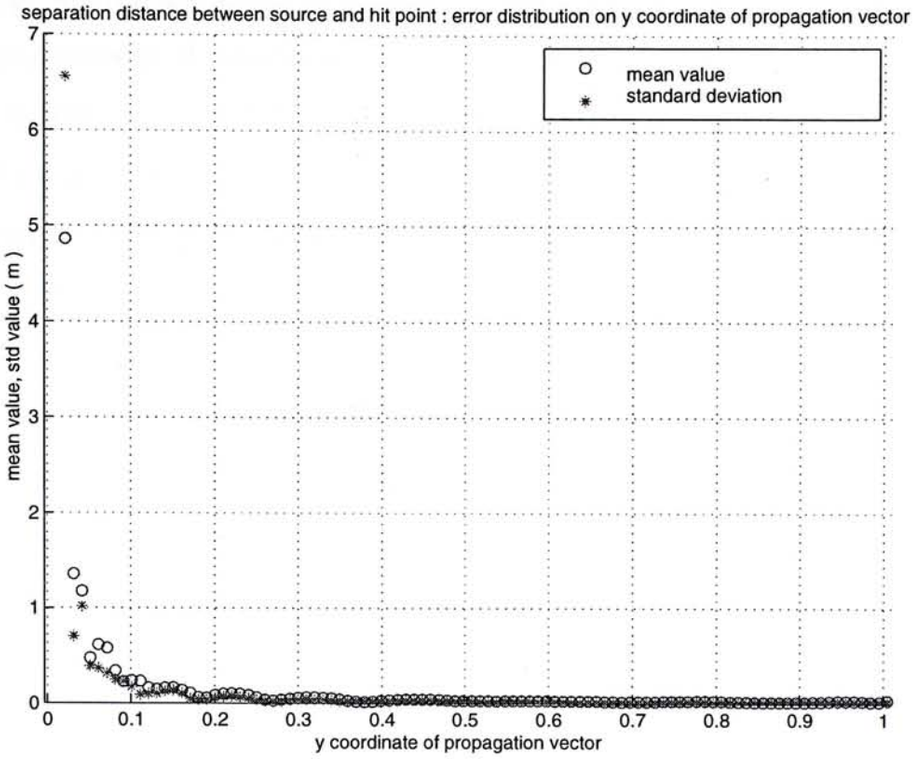


Figure 4-7 : The error of calculating separation distance is plotted against the desired output



(a)



(b)

Figure 4-8 : The error of calculating separation distance is plotted against a) the perpendicular separation distance between source position and surface b) y coordinate of propagation vector

4.6 Calculating propagation vector of secondary ray

There is no special neural network module to calculate propagation vector of secondary ray. For transmission ray, propagation vector is the same as that of incident ray. For reflection ray, propagation vector is simply calculated by reversing the sign of y coordinate of the incident ray.

4.7 Calculating polarization vector of secondary ray

When an incident ray does hit on the scattering surface, the calculation of the polarization vector of secondary ray is independent of ray position and surface dimensions. In this sub-net architecture, translation and rotation transformations are still applied to simplify the problem complexity for neural network. However, the calculated polarization direction is the result in transformed domain and de-transformation is needed to obtain the final result. A question in the efficiency of

calculating polarization vector and neural network training should be considered now to justify the necessity of transformation. The amount of training data is compared in two cases: training data with and without transformation. Then, the training time is estimated for both cases so that it is a successful training. As a result, it is 20 times larger in training data if the neural network is constructed without transformation. This suggests that 20 times more training is needed if the relationship is linear. In conclusion, although transformation creates extra workload in simulation, it can simplify the training process and also makes training more easily successful. This argument can be applied to all network construction for secondary ray calculation. Apart from standard transformation, another transformation is applied so that modeling can be further reduced. The incident ray will be transformed so that all incident ray is coming from positive Y-axis. At the same time, the polarization vector is modified by manipulating the phase of incident ray so that the phi coordinate of polarization vector will be less than 180°. The former transformation affects the propagation direction. The later modification is done as follows: if the phi coordinate of polarization vector is larger than 180°, 180° is added into the phase and the polarization vector is flipped. This transformation will not affect the representation since the original one and transformed one is equivalent.

4.7.1 Architecture

The formulation of polarization calculation involves several procedures. First of all, the polarization vector is decomposed into parallel polarization and vertical polarization according to the surface normal and propagation vector of the incident ray. Then, the reflection or transmission coefficients will be multiplied to the polarization vector. Finally, the resultant polarization of secondary ray is the summation of parallel and vertical polarization. However, in modular neural network architecture, it is difficult to decompose the polarization calculation into these three steps. The data conversion and auxiliary modules will overload the system. Therefore, a "single" neural network is developed for this task but the least number of input parameters is used to reduce the network complexity. Here, the "single" neural network means single network is used to map from incident ray information directly to the polarization vector of secondary ray without decomposition. Each output

utilizes one single network in order to reduce the network complexity. Single network for single output doesn't break down the parallelism but increases the network size. Further reduction in network size is possible afterward. There are nine network outputs in section. Six of them are polarization vector direction. These six neural networks can be divided into three categories: dielectric surface reflection, dielectric surface transmission and metal surface reflection. Three of them are the loss in amplitude and additional phase due to reflection or transmission. Since metal surface will reflect all signals back, it is not necessary to specifically calculate the reflection loss and additional phase. The amplitude will convert in dB. Finally, the amplitude and phase of secondary rays are the summation of that in the incident rays and the calculated amplitude loss and additional phase respectively.

4.7.2 Input and Output Parameters

The related parameters are propagation direction of incident ray, scattering surface thickness, dielectric constant of surface material and, of course, polarization vector of incident ray. Besides these parameters, the amplitude and phase of incident ray are considered as well. However, the loss and additional phase are calculated instead of actual amplitude and phase of secondary ray. By using this approach, two input parameters can be saved. Also, scattering surface dimensions are not included since it has already been used in the previous section in testing whether the incident ray hit the scattering surface or not.

Table 4-7 The input and output parameters of polarization vector calculation sub-net

Network in calculating polarization theta and phi value in transmission and reflection for dielectric surface			
Input Parameters	Range	Output Parameters	Range
Propagation direction theta value	0° ~ 180°	Polarization theta value	0° ~ 180°
Propagation direction phi value	0° ~ 360°	Polarization phi value	0° ~ 180°
Polarization theta value	0° ~ 180°	N/A	N/A

Polarization phi value	0° ~ 360°	N/A	N/A
Dielectric constant of surface material	2 ~ 6	N/A	N/A
Network in calculating polarization theta and phi value in reflection for metal surface			
Input Parameters	Range	Output Parameters	Range
Propagation direction theta value	0° ~ 180°	Polarization theta value	0° ~ 180°
Propagation direction phi value	0° ~ 360°	Polarization phi value	0° ~ 180°
Polarization theta value	0° ~ 180°	N/A	N/A
Polarization phi value	0° ~ 360°	N/A	N/A
Network in calculating additaional loss in transmission and reflection amplitude			
Input Parameters	Range	Output Parameters	Range
Propagation direction theta value	0° ~ 180°	Additional loss in amplitude	0 ~ 1
Propagation direction phi value	0° ~ 360°	N/A	N/A
Polarization theta value	0° ~ 180°	N/A	N/A
Polarization phi value	0° ~ 360°	N/A	N/A
Dielectric constant of surface material	2 ~ 6	N/A	N/A
Network in calculating additional phase in reflection			
Input Parameters	Range	Output Parameters	Range
Propagation direction theta value	0° ~ 180°	Additional phase	0° or 180°
Propagation direction phi value	0° ~ 360°	N/A	N/A
Dielectric constant of surface material	2 ~ 6	N/A	N/A

Table 4-8 The input and output parameters of polarization vector in dielectric reflection

Input Parameters	Range	Output Parameters	Range
Propagation direction theta value	0° ~ 180°	Polarization theta value	0° ~ 180°
Propagation direction phi value	0° ~ 360°	Polarization phi value	0° ~ 360°
Polarization theta value	0° ~ 180°	Amplitude	-80 dB ~ 0 dB
Polarization phi value	0° ~ 360°	Phase	0° ~ 360°
Dielectric constant of surface material	2 ~ 6	N/A	N/A

Table 4-9 The input and output parameters of polarization vector in dielectric transmission

Input Parameters	Range	Output Parameters	Range
Propagation direction theta value	0° ~ 180°	Polarization theta value	0° ~ 180°
Propagation direction phi value	0° ~ 360°	Polarization phi value	0° ~ 360°
Polarization theta value	0° ~ 180°	Amplitude	-80 dB ~ 0 dB
Polarization phi value	0° ~ 360°	Phase	0° ~ 360°

Table 4-10 The input and output parameters of polarization vector in metal reflection

Input Parameters	Range	Output Parameters	Range
Propagation direction theta value	0° ~ 180°	Polarization theta value	0° ~ 180°
Propagation direction phi value	0° ~ 360°	Polarization phi value	0° ~ 360°
Polarization theta value	0° ~ 180°	Amplitude	-80 dB ~ 0 dB
Polarization phi value	0° ~ 360°	Phase	0° ~ 360°

4.7.3 Testing result

570834 testing data is generated to validate the trained neural network.

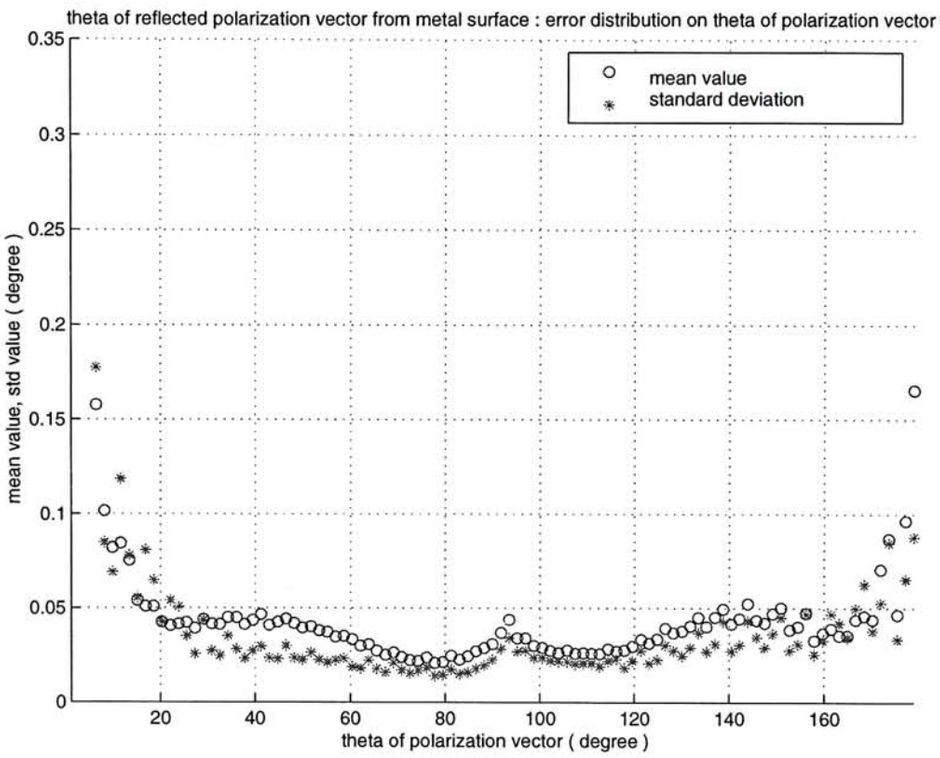


Figure 4-9 : Network performance of calculating theta coordinate in reflection polarization vector from metal surface

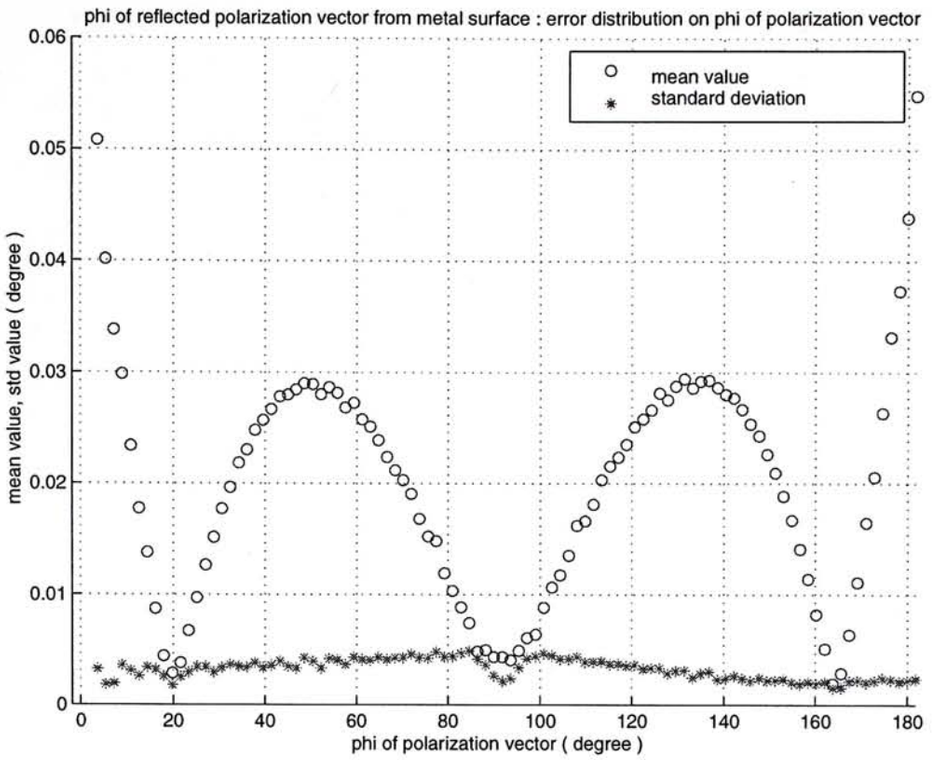


Figure 4-10 : Network performance of calculating phi coordinate of reflection polarization vector from metal surface

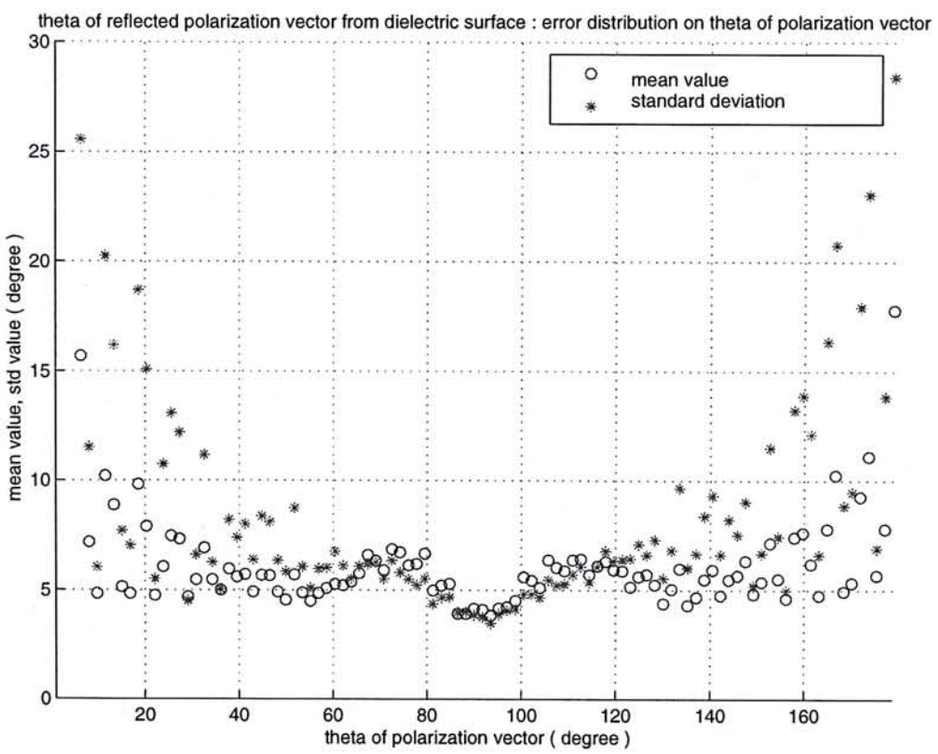


Figure 4-11 : Network performance of calculating theta coordinate of reflection polarization vector from dielectric surface

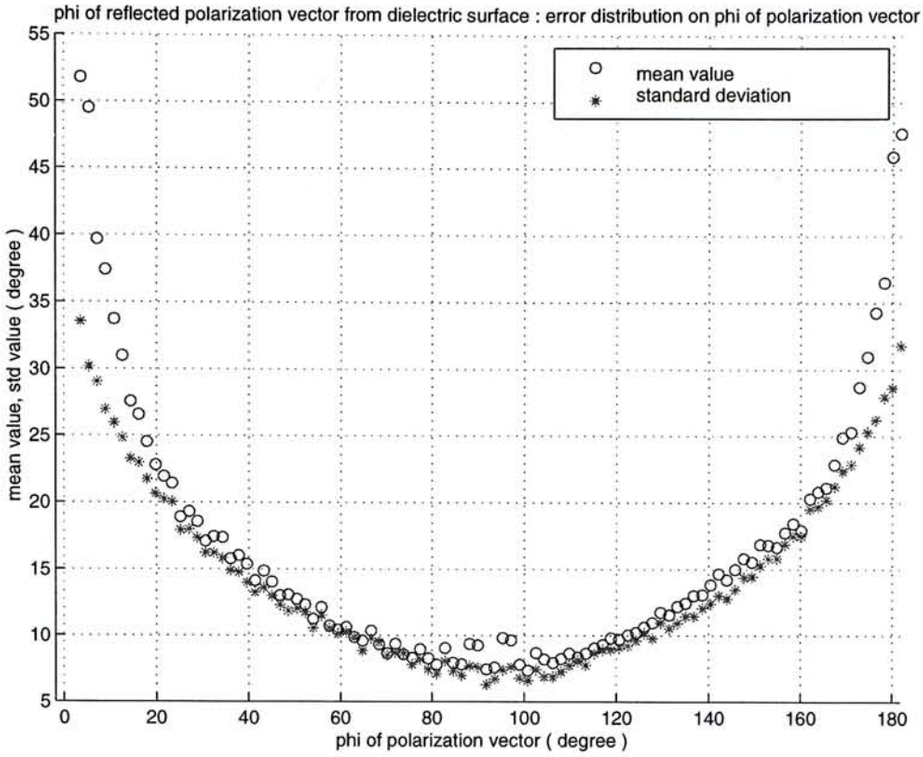


Figure 4-12 : Network performance of calculating phi coordinate of reflection polarization vector from dielectric surface

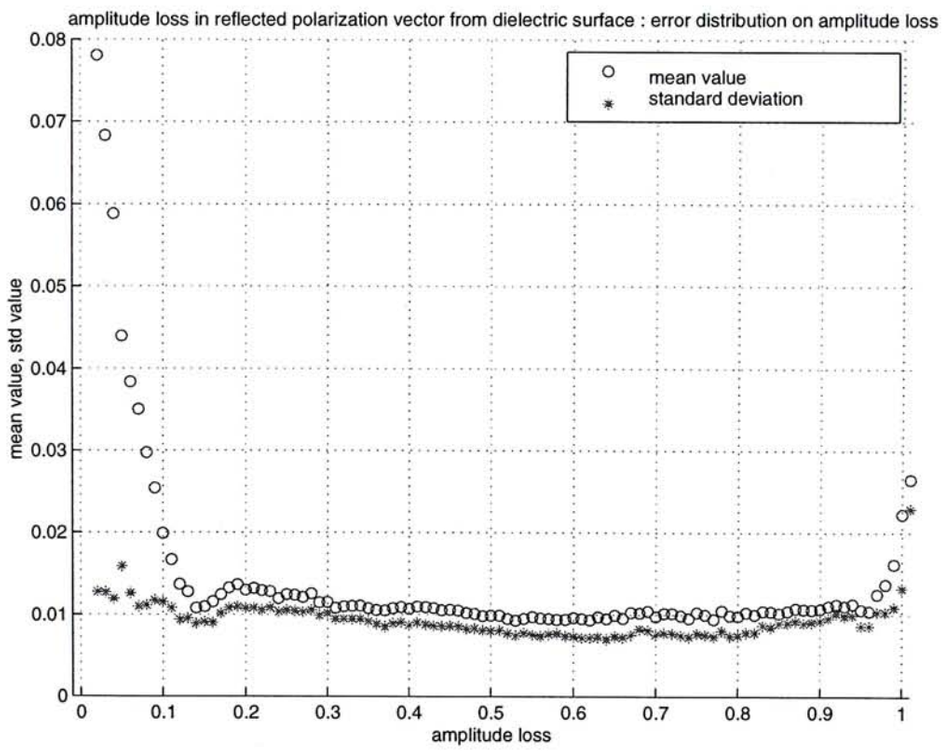


Figure 4-13 : Network performance of calculating amplitude loss of reflection polarization vector from dielectric surface

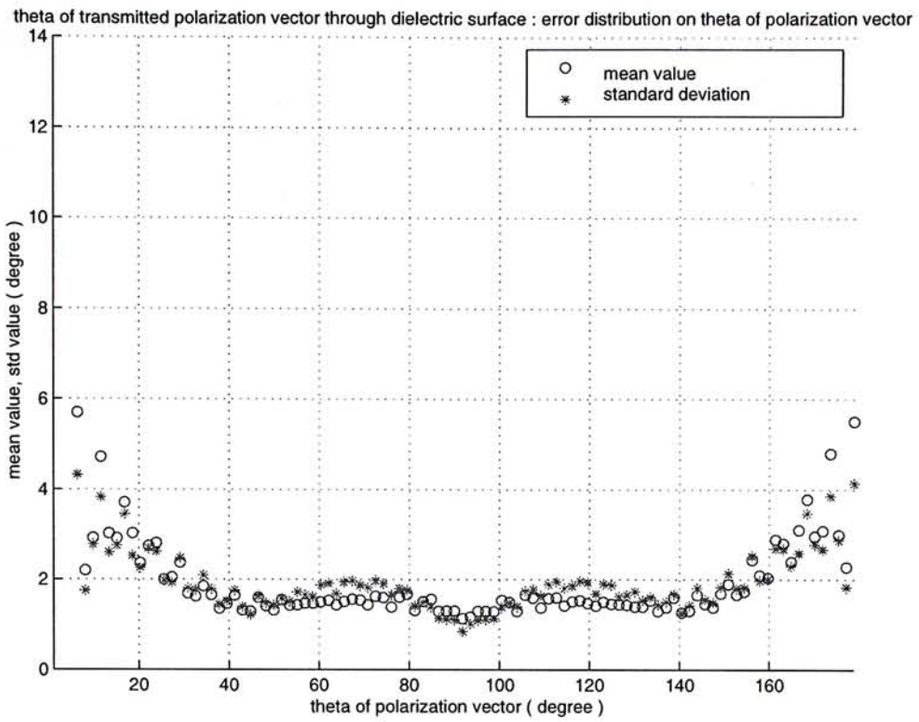


Figure 4-14 : Network performance of calculating theta coordinate of transmission polarization vector through dielectric surface

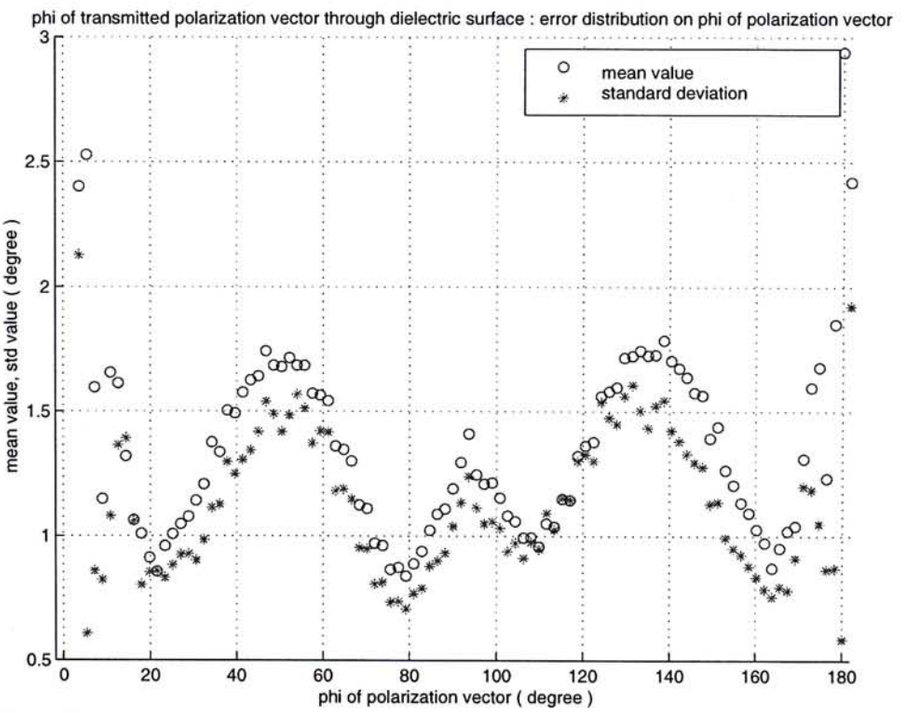


Figure 4-15 : Network performance of calculating phi coordinate of transmission polarization vector through dielectric surface

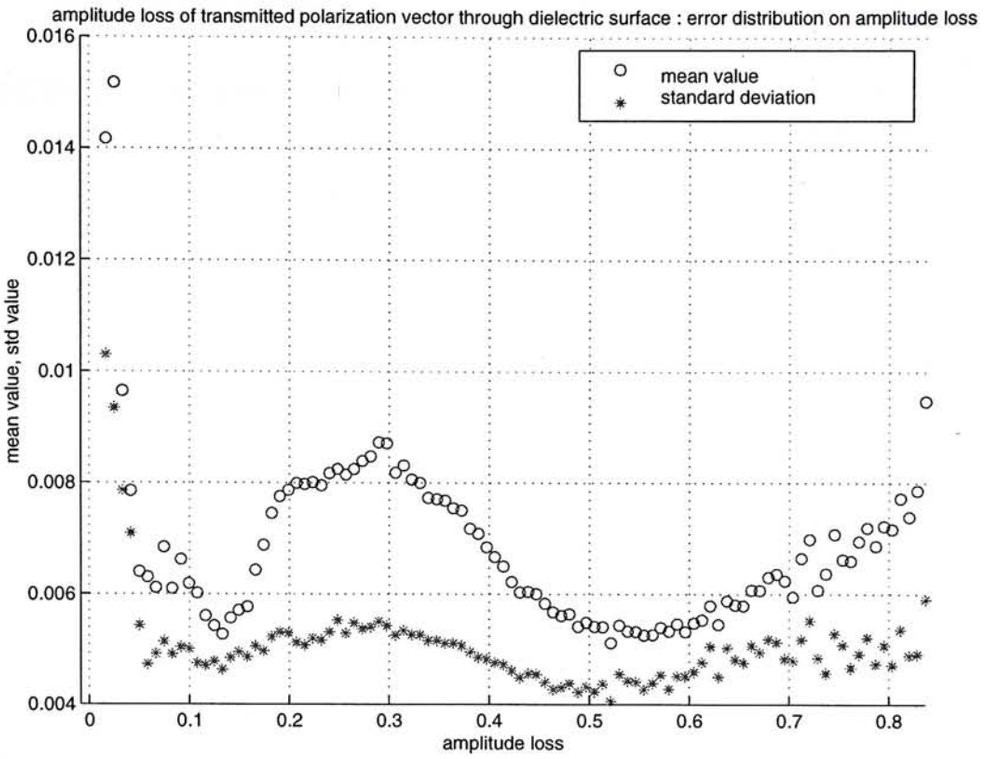


Figure 4-16 : Network performance of calculating amplitude loss of transmission polarization vector through dielectric surface

4.8 Rejecting ray from simulation

There are two criteria to reject a ray from simulation. The first is the absorbing boundary rejection. If a ray reaches an absorbing boundary surface, it means that this ray is going outside the simulation environment and it should be rejected from the ray tracing iteration. A simple method to detect which ray belongs to this criterion is to check whether the coordinate value of the ray position is near to the environment limits no matter it is pointing outwards and inwards. When a ray is positioned on the absorbing surface and pointing inwards, it should be the reflected ray from the absorbing surface and rejected from ray tracing iteration. In other words, if the ray is positioned within (0 m + THRESHOLD) and (50 m - THRESHOLD), this ray will be retrieved in the ray buffer for the next iteration. Another criterion is low amplitude rejection. If the energy in the ray falls below a user-specified threshold, this ray is also rejected. These two criteria are easy to implement through hard-limited function and suitable threshold. Therefore, these neural network modules can be constructed without training.

4.9 Calculating receiver signal

In most ray tracing program, received signal calculation is classified as a post-processing step. In the ray tracing simulation, all multi-path information at all receiver locations will be printed to an output file. After the simulation is completed, the multi-path information is processed to generate particular analysis indicator for evaluation such as received power, delay-spread profile, and angle of arrival. Traditional method can be used to calculate these analysis parameters.

In post-processing, brute force checking method is used to eliminate double count problem that is commonly found in ray launching method. Double count error is caused by the reception sphere model. In two dimensions, reception spheres work perfectly but it generates double count error in three dimensions. The angular separation of geodesic sphere (**Figure 2-2**) is not absolutely constant over the geodesic surface. A constant angular separation is used in the calculation of the radius of reception sphere. The minimum radius for a reception sphere to guarantee the collection of at least one ray from a wavefront is $1/\sqrt{3}$ the distance between rays. Sometimes, this reception method causes two rays to be received. However, since modular neural network is just a approximation of ray tracing method, opposites problem may happen that ray miss is possible due to the calculation error in ray direction. The equivalent error in simulation result due to ray missing is much larger than that for double count error. Therefore, the biggest angular separation is chosen to ensure less ray miss in modular neural network.

Brute force checking is used to eliminate such double count error by removing a ray if it is similar to another ray in the same reception sphere. Since the considered ray is compared to all other rays in the same reception sphere, the process should be worked out after the simulation is completed. Several criteria are used to define the similarity of two rays. These criteria include propagation profile, unfold separation distance, arrival direction and last interaction position. Propagation profile is the number of reflection and transmission when the ray propagates from transmitter to receiver. Unfold separation distance is the total propagation distance from transmitter

to receiver. Arrival direction is the direction where the signal arrives at the receiver. Finally, last interaction position is the last hit point if the ray propagates with reflection or transmission, or the transmitter position if it is a line-of-sight signal. If a weighted sum of the difference of these parameters in two rays is bounded within a defined threshold, these two rays are classified as the same and one of them can be ignored in the received signal calculation.

4.10 Further comment on preparing neural network

4.10.1 Data preparation

In the preparation of training data in most modules, an evenly distributed sampling method is used to generate the required training data. The method has the advantage that the generation is simple to implement. When there are N input parameters available in a certain module, the interval of each parameter is first determined. Then, sample is taken from that N dimensions space and compiled to form the training data set. Since the range of each parameter is known before the generation of training data, evenly distributed sampling can cover all possible values if the interval of each parameter is fine enough. A problem arises: How to determine the suitable size of interval for different parameters? If the interval size is too small, the number of training data will be exponentially large with reference to the sampling frequency of each parameter. If the interval size is too large, the number of samples may not be enough to represent the variation of multi-dimensional function that is described by the relationship between input and output parameters. In this research, the interval size is decided by exploring the variation of that multi-dimensions function. Then, a slightly smaller interval size is used so that the underlying function can be described by the sample data itself.

In some cases, even distributed sampling is not applicable since the corresponding output space is dominated by certain range of values. In the evenly distributed sampling method, the corresponding output space is supposed to be similarly distributed. Therefore, the output data in training data set can cover the

entire possible output domain. The assumption will be violated when the output always has a similar value for most input combination but a great variation occur in a small region of the input space. In such cases, sampling data seems to be not enough for that particular region where the output has great variation. As a result, neural network will be often trained to be biased to the dominant value and the neural network cannot model the variation. In some extreme case, the network output is constant for all possible input combination. There is one possible solution to ease the problem. In the first stage, using even distribution method generates sufficient sample data. Then, the training data is filtered by considering the distribution of output parameters so that the sampling data will be reduced or determinated in over-sampling region while maintain the sampling data where the output variation is large.

4.10.2 Batch training

Almost all the training data sets in this research are very large in order to adequately describe the underlying functions. For example, if a training set contains 10000 sample, it is impossible to consider all the training data in one epoch especially when second order training method is used to speed up the process. These methods mostly manipulate Hessian matrix and find out the optimal searching direction so that the minimum point can be reached speedily. However, the calculation of Hessian matrix is a third-order computation in terms of the number of training data. Therefore, large training data in one epoch results in large memory requirement in the training process. An alternative method is to use batch training. In batch training, part of the training set will be used to train the network in each epoch. The selection of partial data in each epoch is randomly chosen so that all the training data has equal chance for presentation to the neural network.

Batch training has disadvantages. One problem is oscillation in the network performance. Batch size cannot fully represent the whole training set. Smaller training size can be used without reducing the performance. Little variation is expected between one batch and another. This difference in batch set will disturb the training and artificial oscillation will be added. The consequence is that the network cannot reach the truly minimal that can be reached by using total training data in one

epoch. The second problem is that additional training time is required. As batch data can only represent part of underlying function characteristics, there will be some error in the training process due to the wrong search direction. Oscillation may happen and long search path is expected. Therefore, more training cycle is needed to reach the goal.

In Levenberg-Marquardt training method, one useful parameter can be used to evaluate how well the training is progressing. This is the number of parameters out of the total network parameters that are effective in the training cycle. Although this indicator will be reset to maximum network parameters when a new batch of training data is presented in the next batch training epochs, it will come to a stable value after the end of one batch training cycle. Therefore, if the number of effective parameters versus the training process is plotted, an interesting figure can be obtained to illustrate the training process. A good or normal training process will lead to an increase in the number of effective parameters. The "good" conditions somehow relate to the training data preparation, batch size, randomization of batch training data. Three different kinds of training is plotted in terms of effective parameters. In Figure 4-17, the number of effective parameters at the end of the batch training is steadily increasing when more and more training data is presented. This shows that the network can capture the underlying function and effectively transform it into its network weight parameters. In Figure 4-18, the number of effective parameters at the end of each batch cycle is constant throughout the training. This shows that the network is capable of modeling the problem and it should be reflected from the sum squared error of training and testing data. However, the network cannot further improve the modeling performance even when more new training data is presented. This may be caused by the training data preparation and the number of training data in each batch training. In the last case, Figure 4-19, the number of effective parameters is equal to an unreasonable level: one. As a result, the network is expectedly unable to model the relationship between input and output parameters. Experience indicates that this situation happens, the network output is always equal to a constant value no matter what input data is presented to the network.

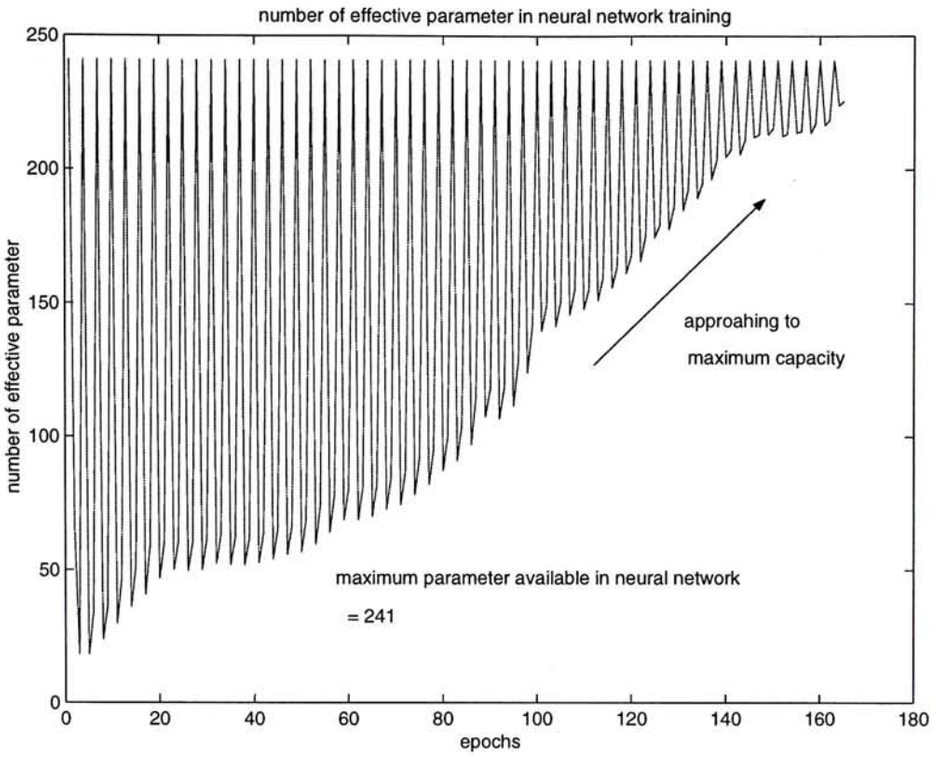


Figure 4-17: Variation of effective parameter throughout the training when network is capable of modeling the problem

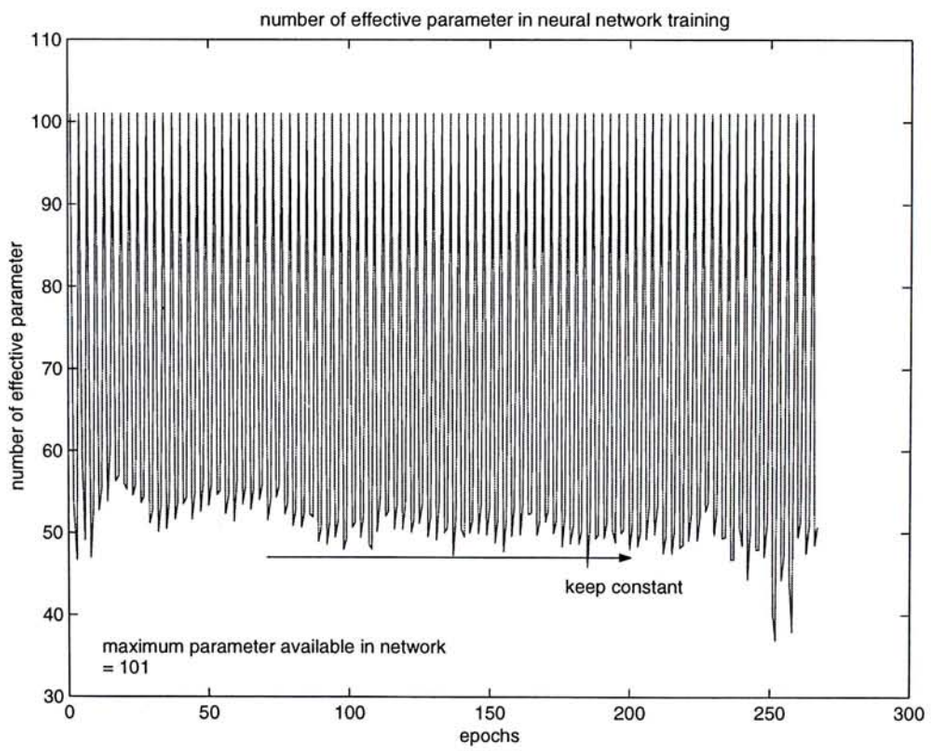


Figure 4-18: Variation of effective parameters throughout the training when the network cannot further improve the overall performance

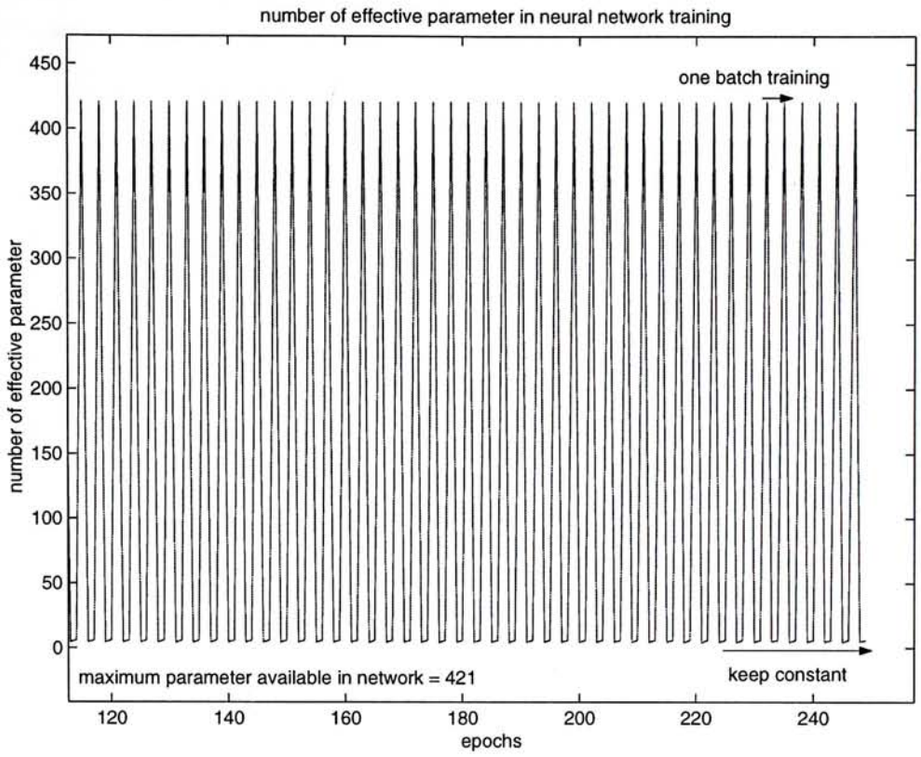


Figure 4-19: Variation of effective parameters when the network cannot model the problem

4.10.3 Batch size

When batch training is used, the amount of training data presented to the network at each epoch has to be decided. Result shows that this factor is critical. For example, in the same problem, if two different batch sizes are used in its training, the number of effective parameters is only slightly different in value. According to the previous discussion, these effective parameters can illustrate how well the network training is processing. The smaller batch size will reduce the speed of training. This is reasonable since the data in smaller batch size may represent different information from that of the total original training data set. Therefore, batch size should be carefully chosen.

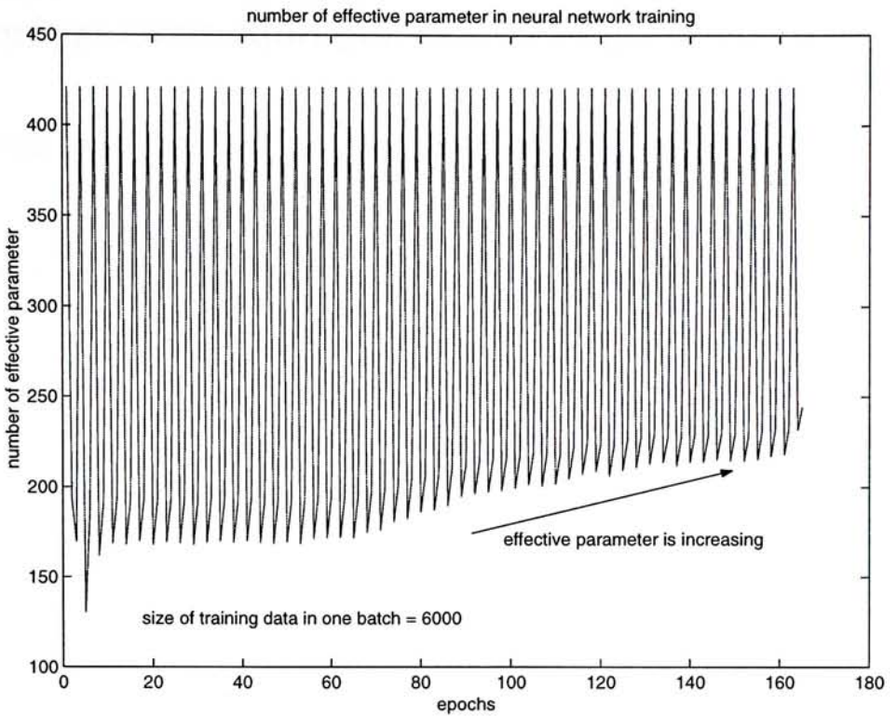


Figure 4-20 : Variation of effective parameters when a sufficient large batch size of training data is used in training process

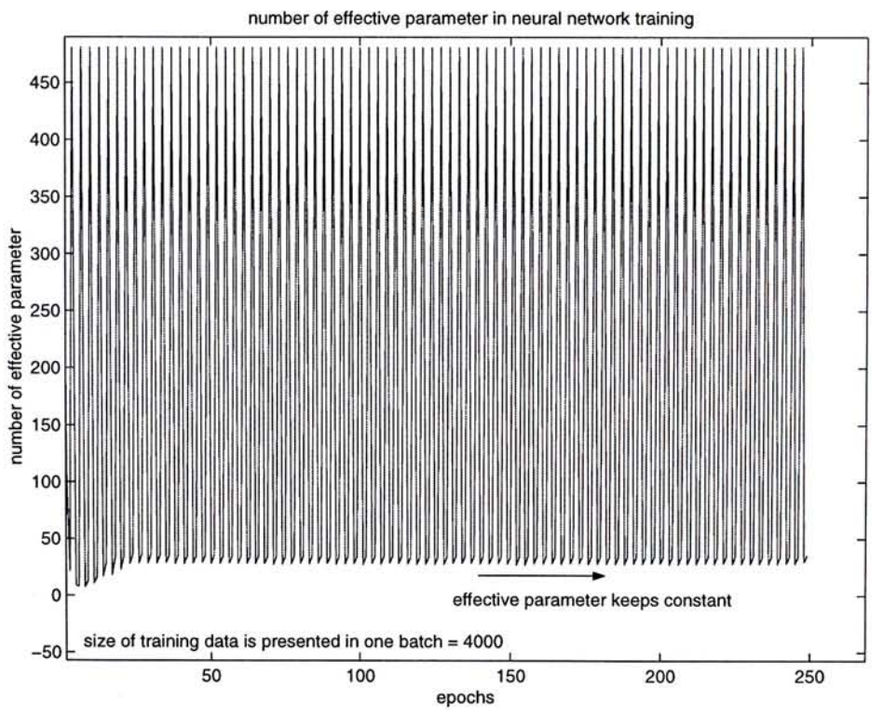


Figure 4-21 : Variation of effective parameters when a small batch size of training data is used in training process.

5 Canonical Evaluation of modular neural network

5.1 Typical environment simulation compared with ray launching

In this section, the modular neural network model and the brute force ray tracing method are used to simulate some canonical environment. Based on the comparison between the two simulated results as well as deterministic calculations, the modular neural network approach is verified in accuracy. These typical environment simulations are chosen because they can show different simulation characteristics and thus a complete verification can be achieved.

5.1.1 Free space

In this free space example, the path loss along path A is simulated while an isotopic vertical polarization source is placed at (5 m , 25 m , 18 m) location. The path A starts from (6 m , 25 m ,12 m) and ends at (40 m ,25 m ,12 m) with 0.1 m sampling distance. The simulation configuration is shown in Figure 5-1.

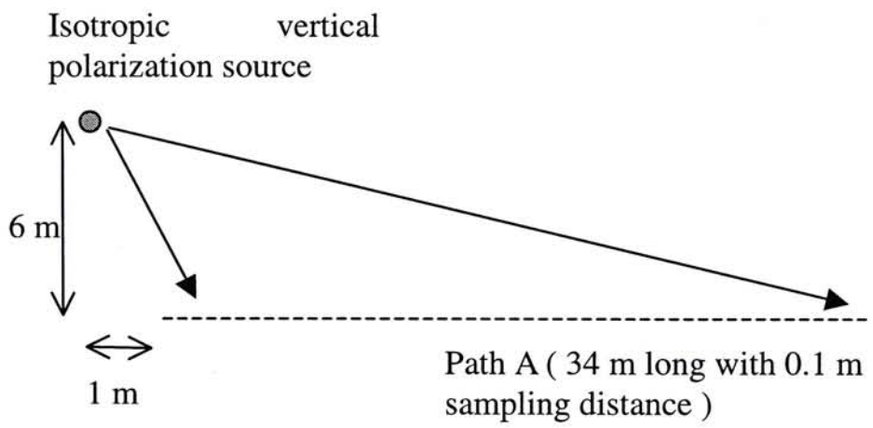


Figure 5-1 Free space simulation configuration

In reception, dipole antenna is used as receiver so polarization effect will be considered. In Figure 5-2, the result of calculation and simulations are compared in terms of path loss. The dash-dotted line with star marker is deterministic Frii's formulation result. The dash line with circle marker is simulation result of brute force ray tracing method. The solid line with cross marker is simulation result of modular

neural network. The result is expectedly good with all three lines in the graph overlapping. Looking at the result, the higher path loss in the beginning of path A is caused by the far field pattern of the dipole. When the separation distance increases, the effect of far field pattern of the receiver becomes minor while the distance dependent factor dominant the path loss result. Propagation distance, polarization vector, amplitude and phase are used in this simulation.

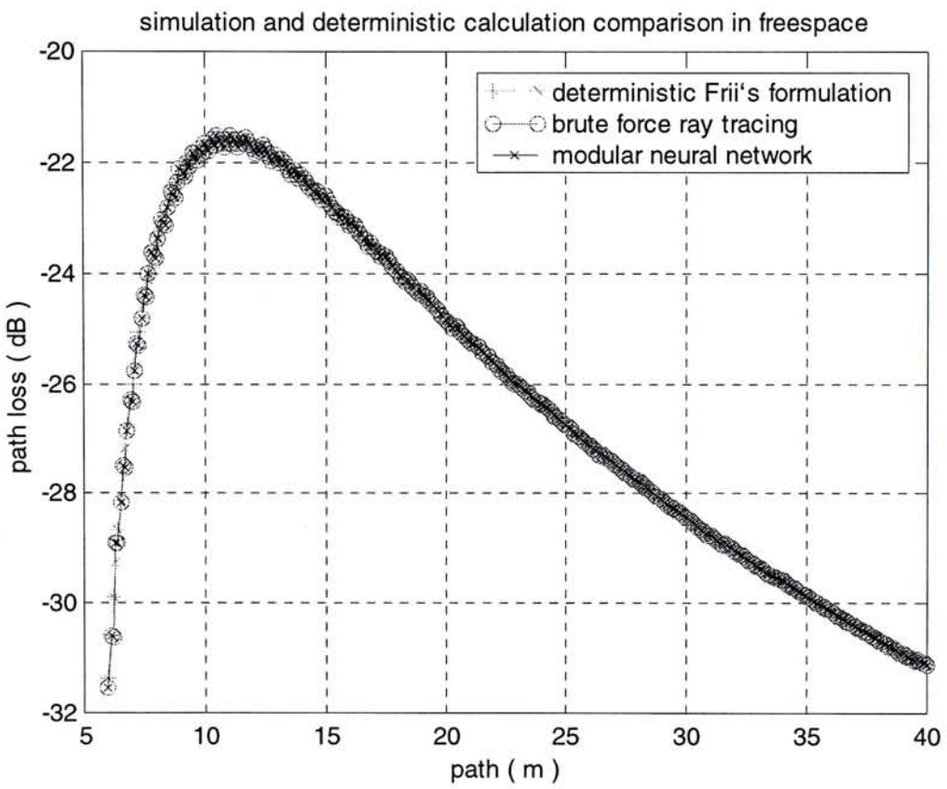


Figure 5-2 Comparison between simulations and calculation in free space

5.1.2 Metal ground reflection

After the line of sight calculation is verified, ground plane reflection is simulated by adding a metal surface 8 meters below the transmitter. In this simulation, single reflection is expected from the metal surface and all locations will receive two signals: line of sight signal and ground reflection signal. The simulation configuration is the same as that in free space model but a metal surface is added. The height between transmitter and ground is 8 m while the height between all receivers and ground is 2 m. Deterministic two ray model is also provided to verify

the simulation. In Figure 5-3, the dash-dotted line with star marker is calculation result of deterministic two ray model. The dash line with circle marker is simulation result of brute force ray tracing method. The solid line with cross marker is simulation result of modular neural network. The figure shows that ray tracing simulation result follows the calculation very well and modular neural network can model the problem except there are some error when the receiver locations is far away from the transmitter. The fading depth and fading width of both brute force ray tracing method and modular neural network are similar to deterministic two ray model result along the path especially the increasing fading depth can be modeled in two simulations. At the receiver locations where are far away from the transmitter, three aspects in modular neural network simulation cause the prediction error. The first aspect is ray missing due to the error in hit point calculation along the path. When the propagation vector of the incident ray is parallel to either one of axis, the deflected distance is very small. Referring to section 4.3.3, the modeling error in hit point modules is relatively large when the deflected distance is smaller. Therefore, some rays that are supposed to be reflected at the path locus now are reflected with position shift and they will not enter the reception sphere. The second aspect is the calculation error in separation distance. Since the total propagation distance will affect the exact phase when the ray reach the receiver. Also, the calculation in polarization is not prefect and the wrong polarization vector direction will directly affect the partial of energy to be received when the polarization of receiver is considered. In two ray model, the two possible rays are line-of-sight and reflected ray from the ground. They have different phase and different polarization vector direction and therefore, fading occurs in the simulations and calculation. The last two aspects mainly affect the vector summation of two signal in two ray model and thus an incorrect prediction result is generated. In Figure 5-4 (Figure 5-5 is the close up of the hit point near to transmitter on the surface), the hit points calculation of ray tracing and modular neural network are plotted for comparison. The calculation of hit point in far away region and near region with respect to the transmitter shows good accuracy.

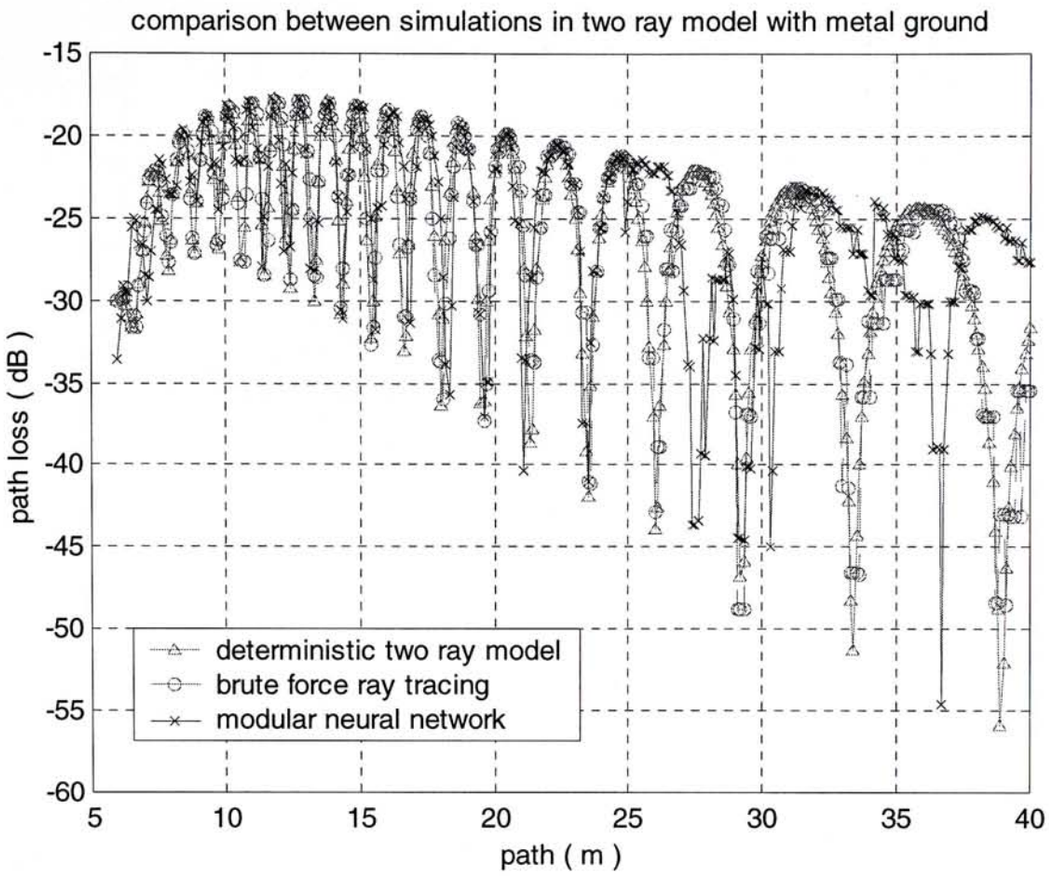


Figure 5-3 : Comparison between the result calculation and simulations in two ray model with metal ground reflection

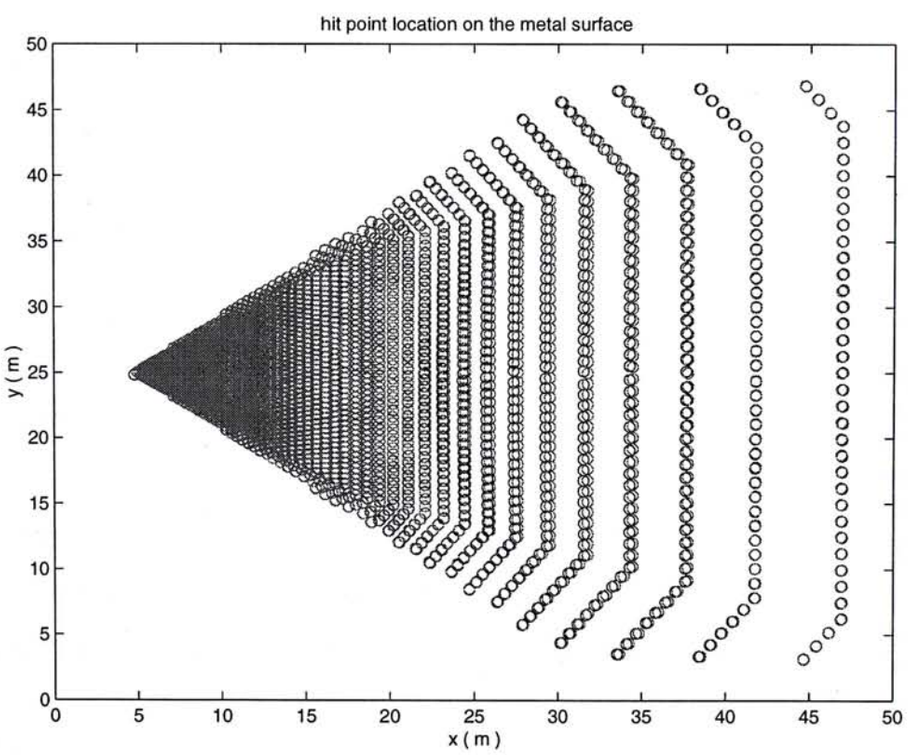


Figure 5-4 : The interaction points of ground reflection in modular neural network and ray tracing: brute force approach

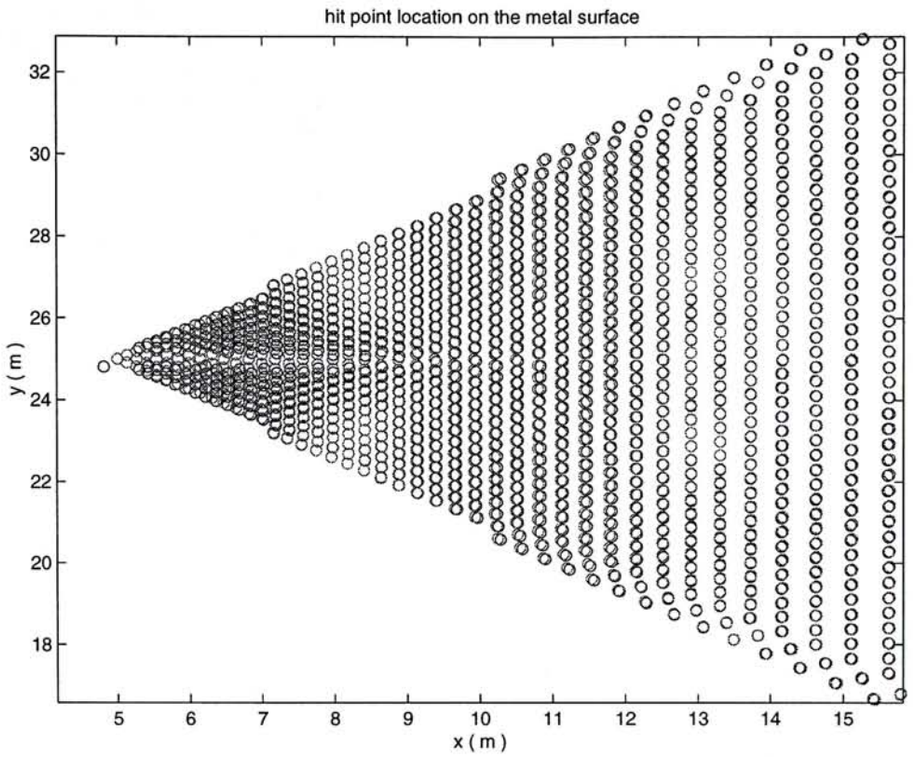


Figure 5-5 : The close up of the interaction points of ground reflection in modular neural network and ray tracing: brute force approach

Table 5-1 : Statistics of prediction error for modular neural network in two ray model with metal reflection

Mean error (dB)	4.48
Standard deviation (dB)	5.15

5.1.3 Dielectric ground reflection

This simulation configuration is the same as the previous environment except the ground surface is replaced by a dielectric slab. While the calculation of ray geometry is proved in the metal ground reflection example, the reflection signal from dielectric surface is tested in this example. Underground reception is also considered as a ray can propagate through the dielectric ground reach any location under the ground layer. This calculation will then test the transmission accuracy.

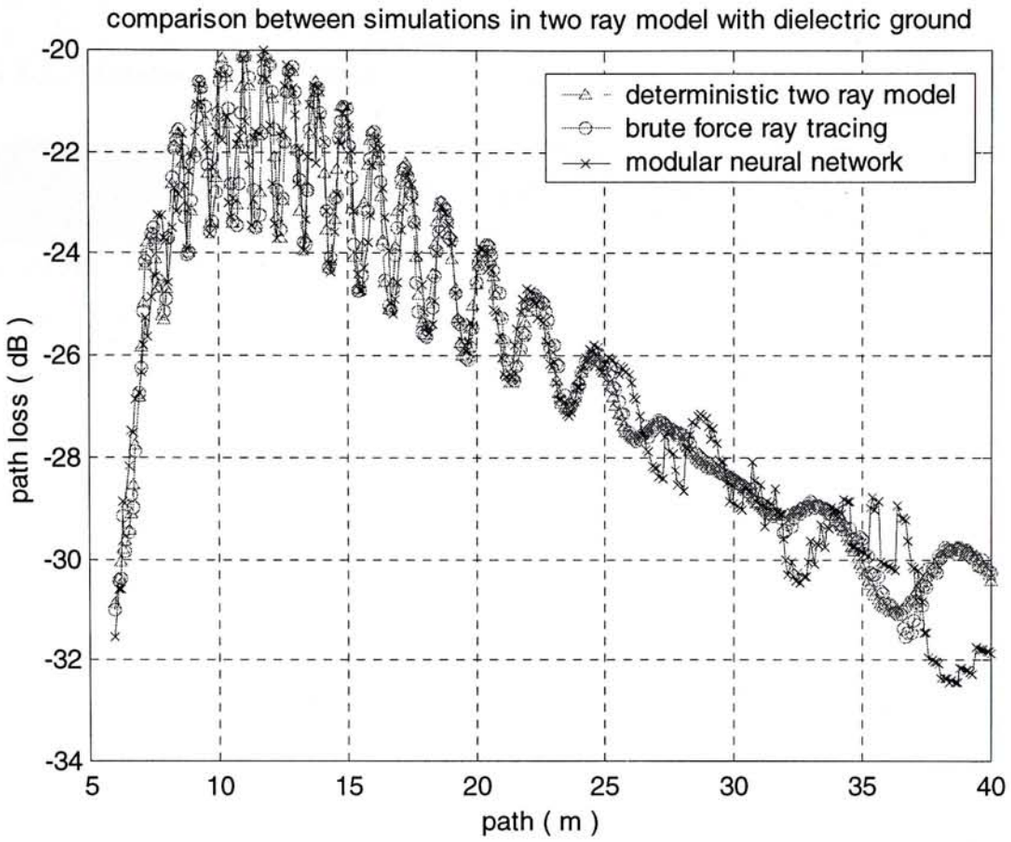


Figure 5-6 Comparison between ray tracing: brute force approach and modular neural network simulation result in two ray model over a dielectric surface ground. (dielectric constant is equal to 4)

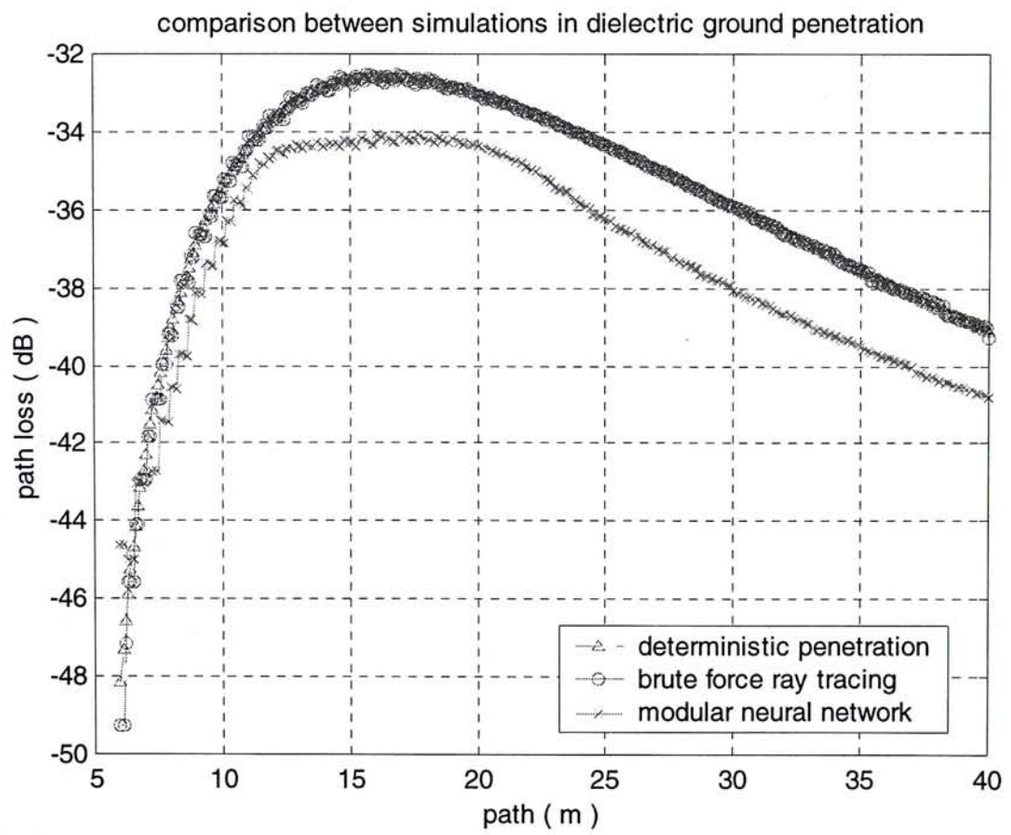


Figure 5-7 Comparison between ray tracing: brute force approach and modular neural network simulation result in underground reception. (dielectric constant is equal to 4)

Table 5-2 : Modular neural network prediction error in two ray model and ground penetration simulation

	Two ray model with dielectric ground	*Penetration through dielectric ground*
Mean error (dB)	0.98	1.67
Standard deviation (dB)	0.78	0.47

5.1.4 Empty Hall

In this section, a simple indoor environment is considered. It is a rectangular hall. The four vertical walls and the ceiling of this rectangular hall are dielectric slab while the ground is made of metal. On two opposite sides, one door each is created to allow energy to go outside so that the number of reflection can be reduced. Thus, there are a total of ten surfaces in this model. The dimensions of the room are 22 m × 26 m × 8 m. A single transmitter is mounted on the ceiling (the top dielectric slab) and the signal at 1.5 m above the floor (the bottom dielectric slab) is considered. This is the average level for a person holding a mobile device. The transmitter is a directional antenna that radiates downward. The geometry model of this empty hall is shown in Figure 5-8. In this figure, ceiling surface and floor surface are removed to illustrate the configuration of antenna location as well as receiver locations. Red dot represents the transmitter position while the blue dots represent the receiver location.

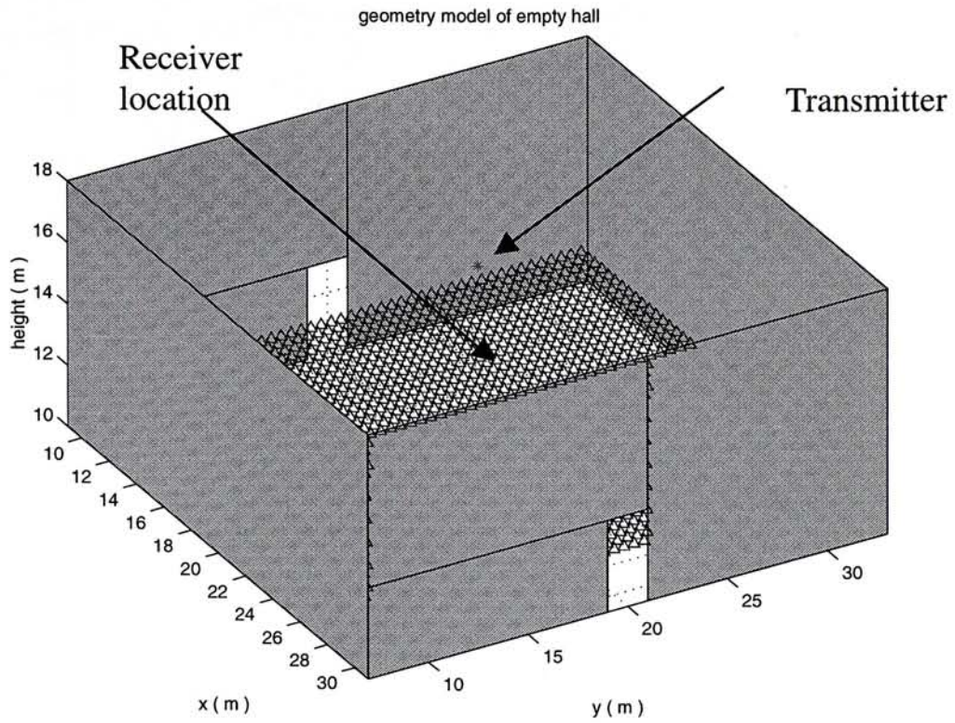


Figure 5-8 : Geometry model of empty hall.

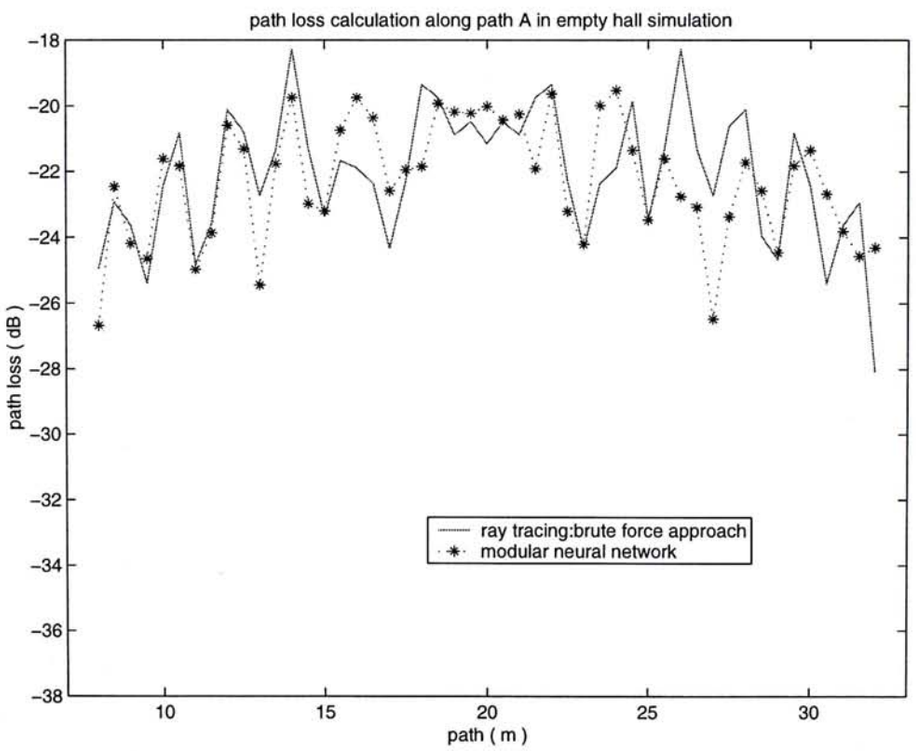


Figure 5-9 : Path loss along Path A in ray tracing: brute force approach and modular neural network simulation result.

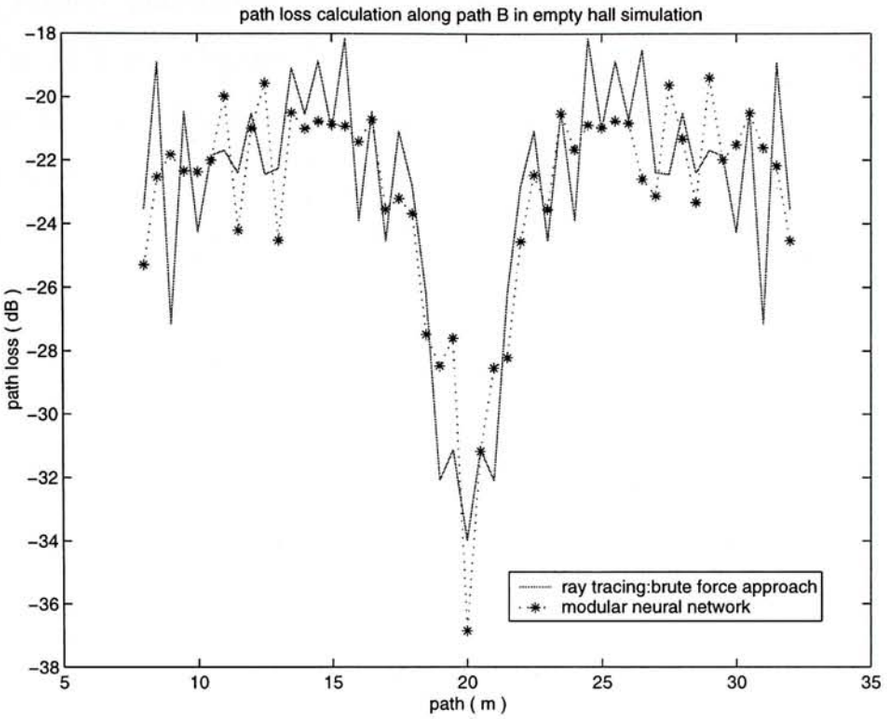


Figure 5-10 : Path loss along Path B in ray tracing: brute force approach and modular neural network simulation result.

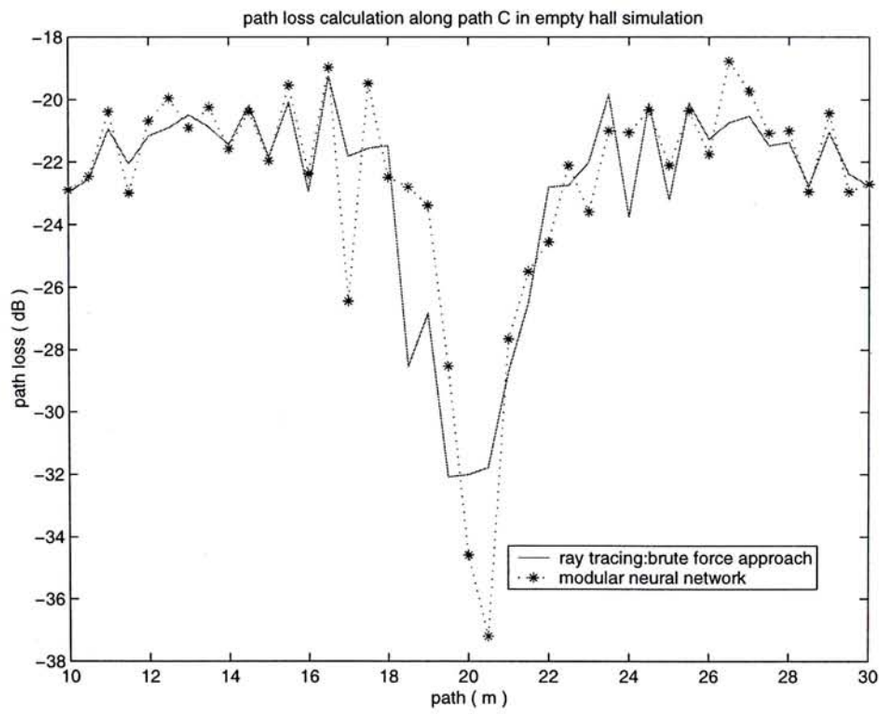


Figure 5-11 : Path loss along Path C in ray tracing: brute force approach and modular neural network simulation result.

In the simulations, 2009 samples are considered. They are evenly distributed with 0.5 m separation distance. Three simulation paths are chosen to compare the

simulation results graphically. Path A is the collection of locations where y coordinate is 11.7 m. (Figure 5-9) Path B is the collection of locations where y coordinate is 17.5 m. (Figure 5-10) Similarly, Path C is that where x coordinate is 23 m. (Figure 5-11) In these three figures, the solid line is the simulation result of brute force ray tracing method while the dotted line with star mark represents the simulation result of modular neural network. In path B and path C, the large path loss at the center of the path is caused by the far field pattern of the dipolar receiver antenna because that region is the bottom of the transmitter. This large variation in path loss can be modeled by both simulation methods. Along path A, the signal characteristic is very different from the previous two paths. The incoming signal has similar signal level, so the path loss at all location is around –22 dB. Both simulations has quite equal result especially, there are some locations where the simulation results of both simulation overlaps each other. The overall error in different between two simulations are shown in Table 5-3.

In the next chapter, the two simulation models will be compared, with indoor measurement results to further verify the models.

Table 5-3 : Statistics of prediction error of modular neural network in empty hall with reference to the simulation result of brute force ray tracing model

Mean error	**1.57 dB**
Standard Deviation	**1.39 dB**
Maximum error	**10.37 dB**

6 Indoor Propagation Environment Application

6.1 Introduction

In the previous chapter, four canonical geometries were simulated using brute force ray tracing method as well as hybrid modular neural network. Good agreement between the two methods was observed. In this chapter, the techniques were applied to a real environment, namely, third floor of the Ho Sin Hang Engineering Building of the Chinese University of Hong Kong. In this chapter, the simulation is shown to agree reasonably well with measurement and the accuracy of the hybrid neural network is confirmed.

In both simulation methods, there are 29 surfaces in the geometry model and 20000 rays are launched from the transmitter. The maximum angular separation between rays is 1.68° at the source. Both simulations calculate the path loss with reference to the electric field strength at a 1-meter reference point from the transmitter.

6.2 Indoor measurement on the Third Floor of Engineering Building

The third floor of the Engineering Building at the Chinese University of Hong Kong was modeled and shown in Figure 6-1. Each of the four corridors is about 20 meters in length. They form a rectangle embracing some laboratories and soft-partitioned offices. For each measurement point, 200 samples were taken randomly over a circle of 1-meter diameter around the measurement point. There were altogether 89 measurement points along four paths, one along each corridor. A dipole antenna connected to an HP-8594EM spectrum analyzer was used to pick up the signal. In Figure 6-2, the path loss of all the measured locations along the four paths are plotted. More details can be found in **[40]**

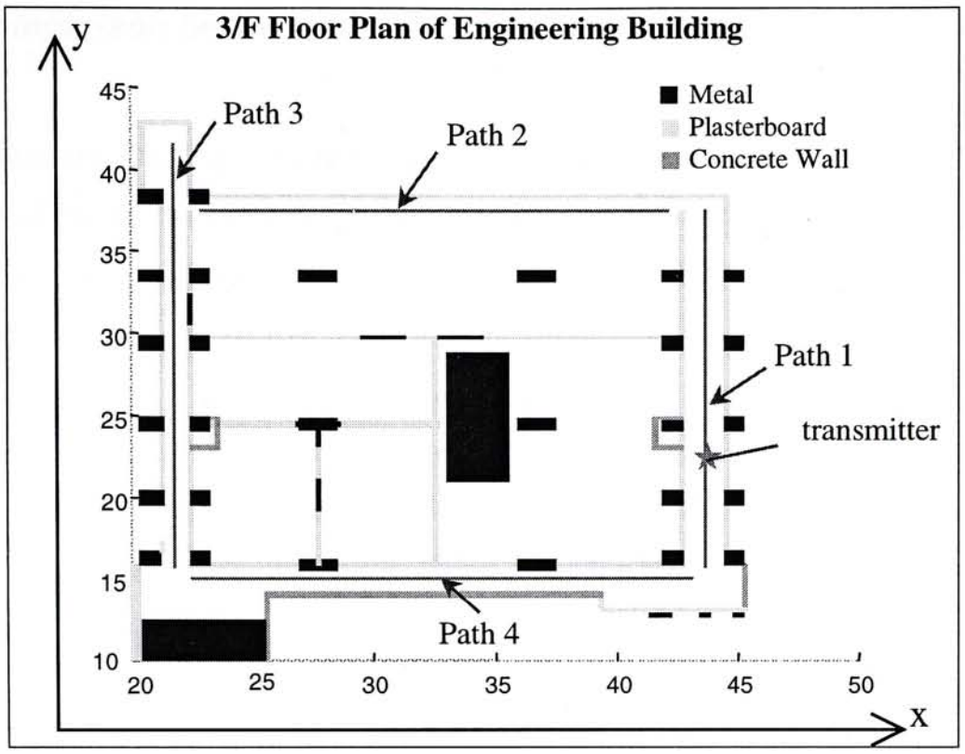


Figure 6-1 : Third floor plan of Engineering Building

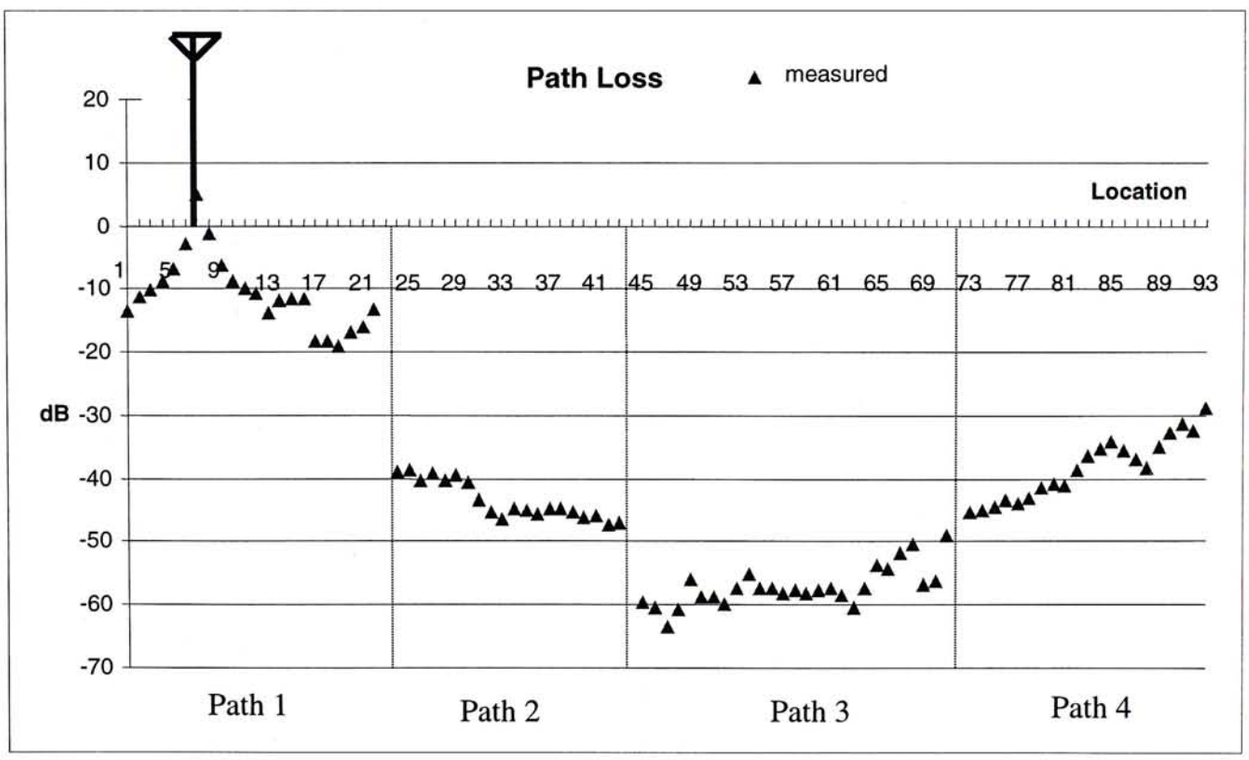


Figure 6-2 : Measurement result of four paths in third floor of Engineering Building

6.3 Comparison between simulation and measurement result

In this section, the simulation results of brute force ray tracing method and hybrid modular neural network method are compared with measurement for the third floor of engineering building. The measurement and simulation results of four measurement paths are plotted separately in Figure 6-4, Figure 6-6, Figure 6-8 and Figure 6-10. The solid line is the measurement result. The solid line with triangle marker is brute force ray tracing simulation result and the dotted line with plus marker is the modular neural network simulation result. In these four figures, the x-axis is the position coordinate of the measurement locations along the path. The statistical information of prediction error for both simulations with reference to measurement result is summarized in Table 6-1.

Table 6-1 : The statistics of the prediction result of both modular neural network and brute force ray tracing simulation with reference to the measurement result

Location	Path 1		Path 2		Path 3		Path 4		Overall	
Method	Brute force ray tracing	Modular neural network	Brute force ray tracing	Modular neural network	Brute force ray tracing	Modular neural network	Brute force ray tracing	Modular neural network	Brute force ray tracing	Modular neural network
mean error (dB)	3.25	5.27	7.19	5.21	16.02	9.4	13.27	7.45	10.37	6.93
standard deviation (dB)	2.6	4.63	4.37	4.02	8.04	8.48	3.49	4.3	7.33	6.01

6.3.1 Path 1

The transmitter was placed in Path 1. The location in the map marked with star or the antenna symbol on the path loss diagram represents the transmitter location. All the measurement or simulation points along this path were in the line-of-sight region of the transmitter. In this set of measurement or simulation, both the transmitting and receiving antennas were kept at the same height of 1.15m above the floor. The transmitting antenna is a monopole. The receiving antenna is a dipole. Their corresponding theoretical antenna patterns were modeled in the simulation. All together 21 positions were taken within a length of 21 meter.

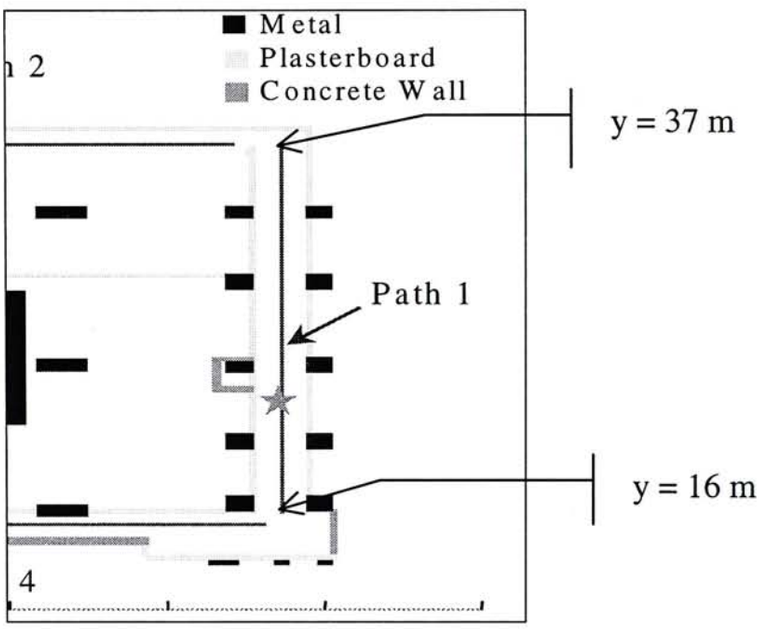


Figure 6-3 : The coordinate of the two ends of path 1

The path loss measurement result ranges from –1 dB to –20 dB. The simulation result of brute force ray tracing ranges from 0 dB to –29 dB while that of modular neural network is between 0 dB and –33 dB. The dynamic range of measured result and simulated result is similar. Since there is a strong line-of-sight signal existing in path 1, the received signal decreases steadily as a function of the separation distance between transmitter and receiver. On the side of the transmitter where the coordinate values of locations are larger than that of transmitter, both brute force ray tracing and modular neural network can predict the trend in path loss. When the coordinate value is larger than 27 meters, there are two slight increase in the received signal found in measurement. It is reasonable since the corridor is so narrow that the channeling

effect is significant. However, both simulations cannot capture such variations. It is caused by the over simplification in the geometry model. Modular neural network has two locations where the signal drop suddenly while brute force ray tracing has one locations where the signal drop suddenly. Modular neural network has large fluctuation in path loss compared with brute force ray tracing. On the left side of the transmitter, since this part of path 1 is relatively short, it is difficult to compare both simulations. One contrast can be found is the sudden drop in simulated received signal of modular neural network when the coordinate value is smaller than 17 meters. In the overall results, modular neural network can still provide acceptably accurate prediction of the path loss in path 1 compared with brute force ray tracing. As modular neural network is the approximation of brute force ray tracing, this prediction performance is good.

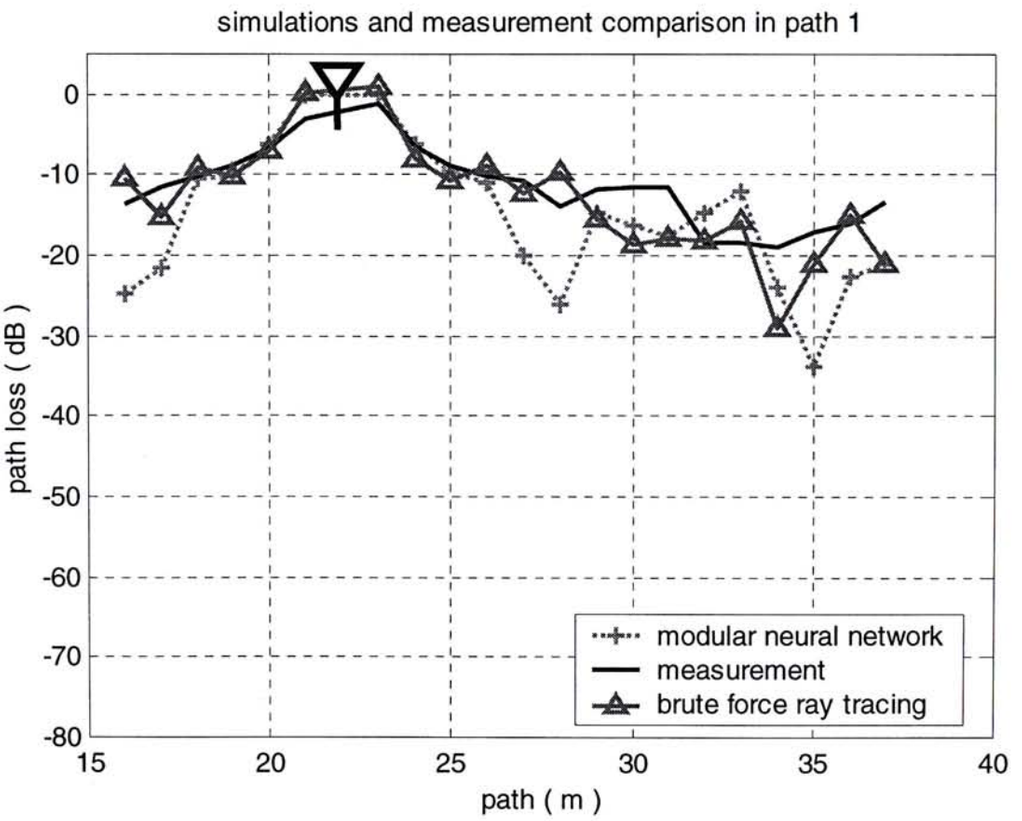


Figure 6-4 : Measured and simulated result of path loss in path 1

6.3.2 Path 2

The measurement locus of path 2 is shown in Figure 6-5. It is a shadow region with respect to the source. The locations of large coordinate value (value in x-axis) are near to the path where the transmitter is located.

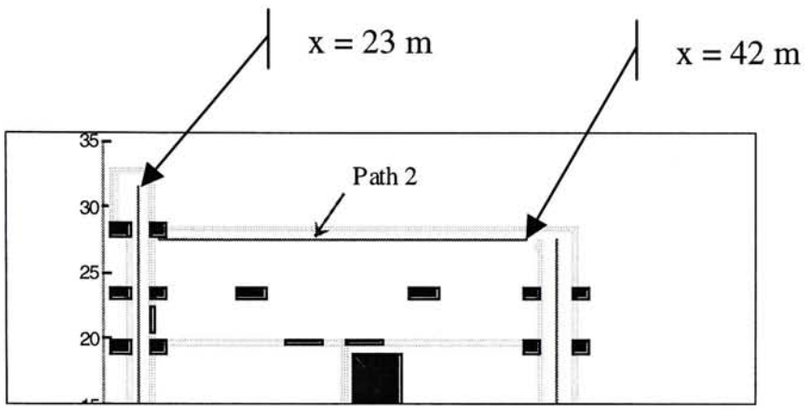


Figure 6-5 : The coordinate of the two ends of path 2

The ranges of measurement result is between –39 dB and –48 dB. The simulation result of brute force ray tracing ranges from –28 dB to –58 dB while that of modular neural network ranges from –32 dB to –61 dB. The dynamic range of modular neural network is similar to that of measurement result, especially the right part where the coordinate value is large. When the locations are moving far away from the transmitter, the measured path loss is gradually increasing without large variation. It shows that a relatively strong signal still exists in this path so that multi-path effect is not dominated. Both simulations can predict the decay trend of the path loss along the path when the locations move from large coordinate to smaller coordinate. There are four fading locations in brute force ray tracing when the coordinate is 43 meters, 34 meters, 29 meters and 24 meters. These fading locations have similar separation distance between them. Brute force ray tracing predicts a periodical fading along path 2 and two to three significant signals exist in this region. However, modular neural network does not show this characteristics. Although there are two minor fading locations, simulated path loss is decreasing gradually in contrast to brute force ray tracing that predicts a similar signal level but with fading. When the coordinate of the location is smaller than 35 meters, the metal-liked pillar near to the transmitter and part of metal chamber in the laboratory block the signal. It can be

shown in the measurement result where the path loss is 5 dB larger than that of the locations near to transmitter. Modular neural network over-estimates the effect of metal chamber blockage while brute force ray tracing has large variation over the measurement result. In this path, modular neural network has slightly lower prediction error than that of brute force ray tracing method.

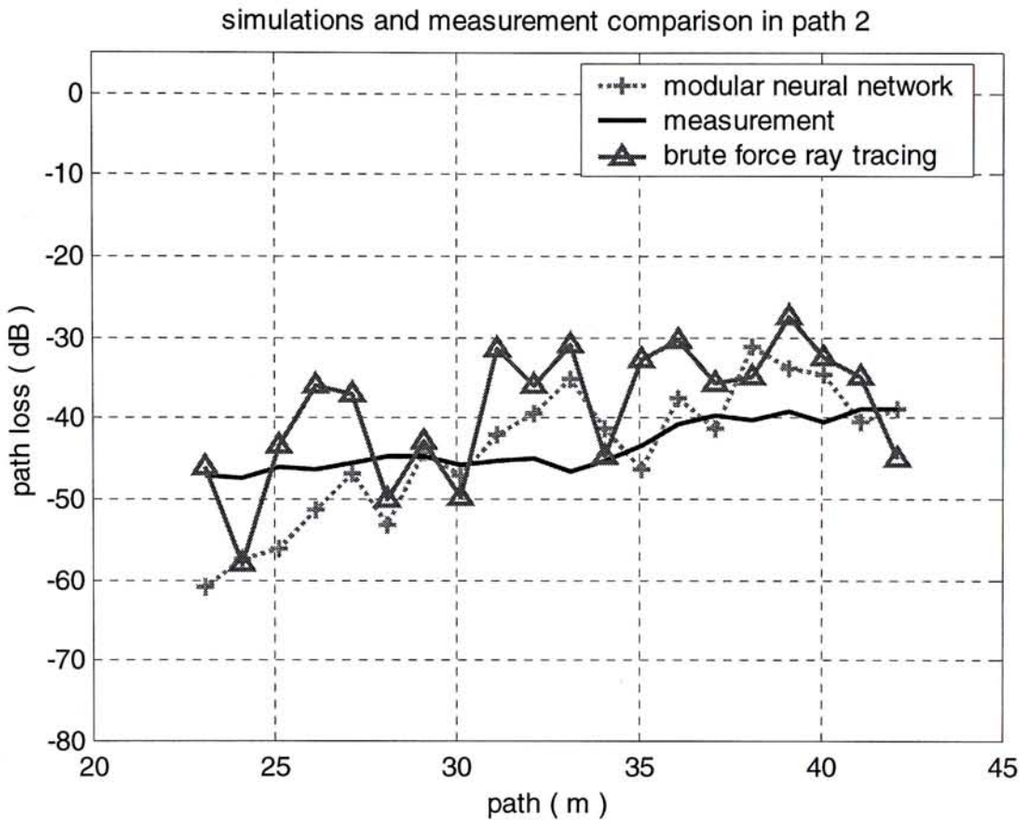


Figure 6-6 : Measured and simulated result of path loss in path 2

6.3.3 Path 3

The third path is the longest of all. It measures 27 meters and 27 locations were measured. In path 3, the environment is a deep shadow region and multi-reflection signal is expected in most locations. Also, in our geometry model, all rooms between the transmitter and path 3 are empty except the metal chamber in the room beside the transmitter. Therefore, the simulated path loss is lower than measurement.

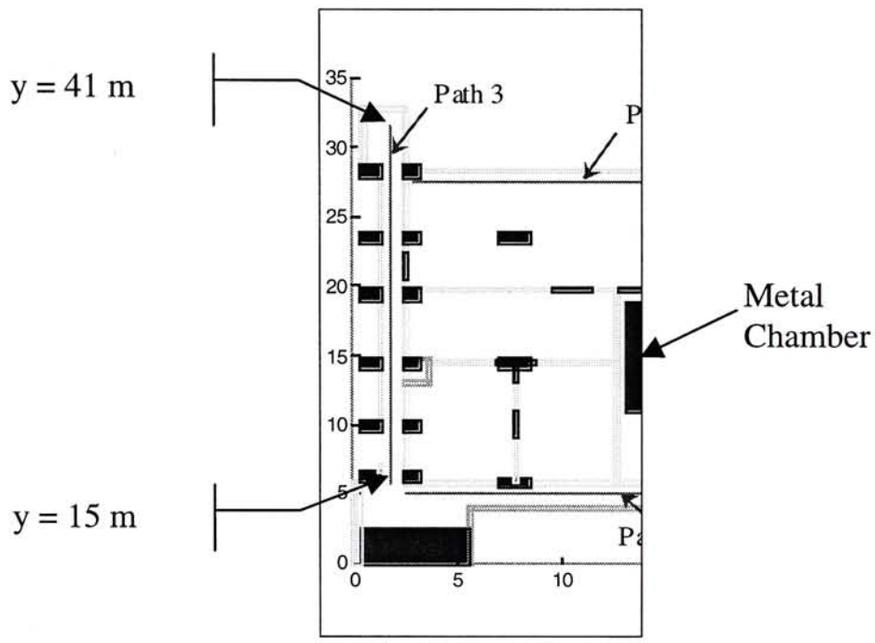


Figure 6-7 : The coordinate of the two ends of path 3

For the locations in the middle of path 3, the metal chamber exactly blocks the LOS path. Therefore, those locations will receive signals lower in amplitude. The range of measurement result is between –49 dB and –63 dB. The dynamic range of measurement result is 14 dB, that is the largest dynamic ranges among four paths. The simulation result of brute force ray tracing ranges from –24 dB to –72 dB while that of modular neural network ranges from –24 dB to –72 dB. Apart from the left part where the coordinate value is 20 meters, the dynamic range of modular neural network simulation is closer to that of measurement compared with brute force ray tracing. On average, brute force ray tracing under-estimate the path loss by 15 dB. In the part where the coordinate value is smaller than 20 meters, both simulations cannot predict the measurement and the prediction error is 20 dB. Again, the over simplification of geometry model causes large prediction error in simulations since the modeling does not include any furniture of these rooms in the middle of the third

floor.

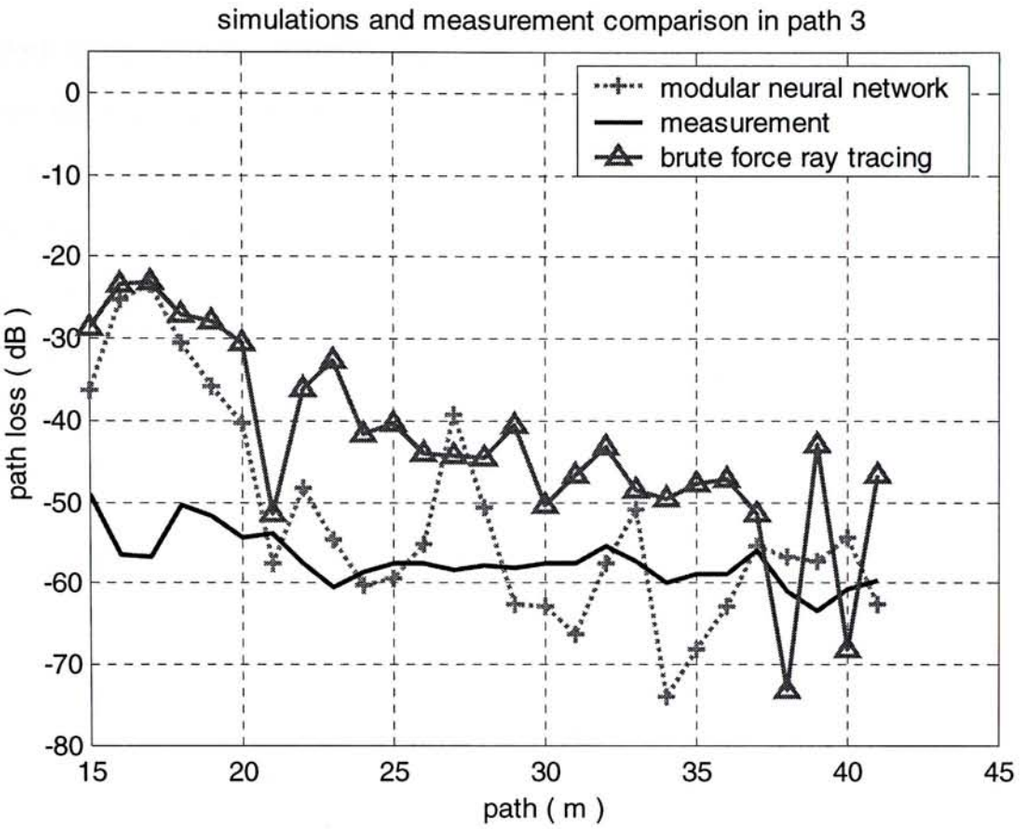


Figure 6-8 : Measured and simulated result of path loss in path 3

6.3.4 Path 4

Path 4 has no line-of-sight signal from the transmitter but the separation distance between the turning corner and transmitter is the shortest among the four paths. As a result, the signal can reach receivers with less reflection processes. Also, path 4 is near to the concrete outer wall of the building. The reflected signal from the outer wall causes higher amplitude. There is no major metal object between transmitter and path 4. The transmitted signal that passes through the room containing metal chamber and the reflected signal that comes from the outer wall have similar amplitude. Therefore, the received signal in path 4 has more fluctuation than path 2 where is also a shadow region. There is a decrease in received power in measurement at the locations where the coordinate values are 31 meters, 37 meters and 42 meters

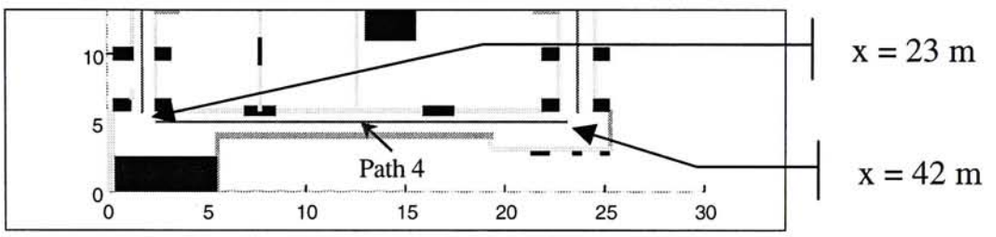


Figure 6-9 : The coordinate of the two ends of path 4

The range of measurement result is between –29 dB and –35 dB. The simulation result of brute force ray tracing ranges from –20 dB to –33 dB while that of modular neural network ranges from –24 dB to –44 dB. The simulation result of brute force ray tracing is higher on average compared with measurement but the trend is very similar. There is a 10 dB difference between measurement and brute force ray tracing simulation on average. Modular neural network provides better prediction than brute force ray tracing especially when the coordinates of locations are larger than 37 meters.

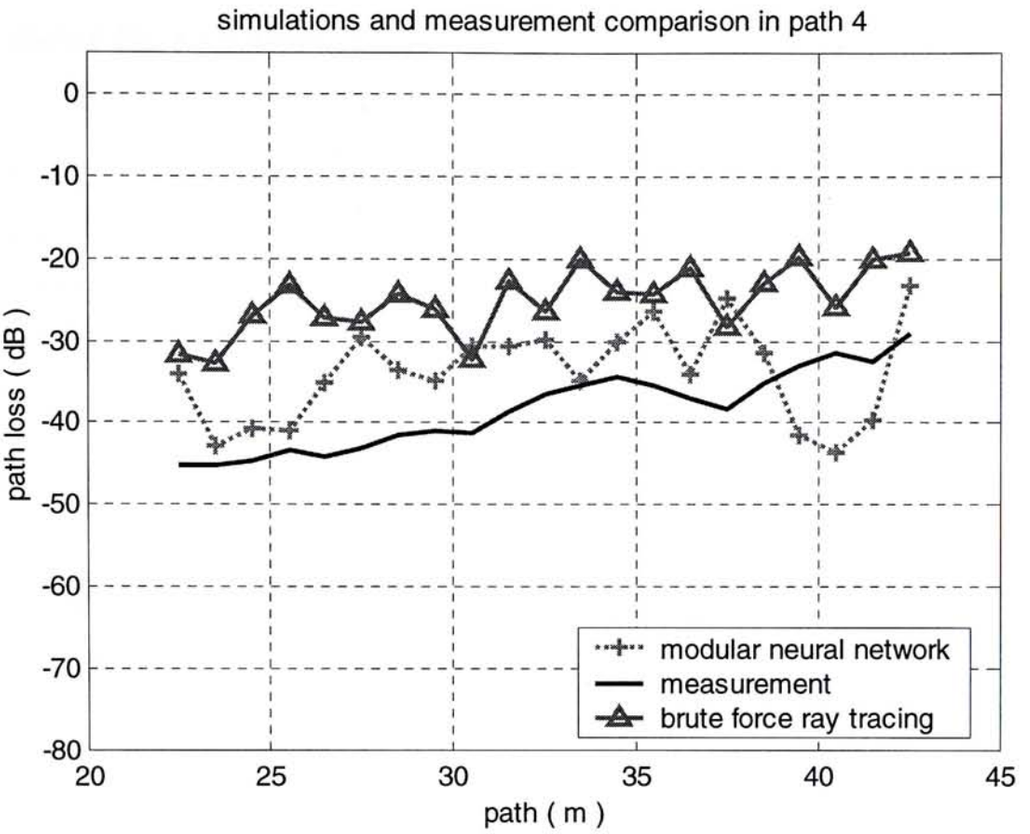


Figure 6-10 : Measured and simulated result of path loss in path 4

6.3.5 Overall Performance

From the above comparison of four different paths, brute force ray tracing under-estimates the path loss in general. This is caused by an over simplified geometry model that has not included the furniture inside in the laboratories in the middle of the floor and the some metal-liked pillar around all the rooms in the middle of the third floor. Modular neural network generates better prediction than brute force ray tracing method. Table 6-1 summarizes the prediction error of both simulations with reference to measurement result. When line-of-sight conditions exists, the prediction is much better. In the deep shadow region especially in path 3, the prediction error is relatively large. Modular neural network has higher prediction accuracy than brute force ray tracing method in path 3 and the overall average prediction error of modular neural network is 4 dB better than that of brute force ray tracing.

6.4 Delay Spread Analysis

In the previous section, received power is analyzed. In this section, the simulation result of brute force ray tracing and hybrid modular neural network are used to analyze the power delay spread of the third floor of engineering building. The purpose of this section is to study the effect of multi-path. In delay spread analysis, the multi-path effect can be clearly analyzed using the power delay profile to calculate mean excess delay, root-mean squared excess delay and maximum excess delay. These parameters are then used to quality and quantify the data rate permissible in this environment.

Three basic time dispersion parameters were calculated to describe the power delay profile statistically. They were the mean excess delay, root-mean-squared delay spread and excess delay spread with threshold –10 dB. Five locations were chosen to study the power delay profile. The five locations are marked with square in the Figure 6-11.

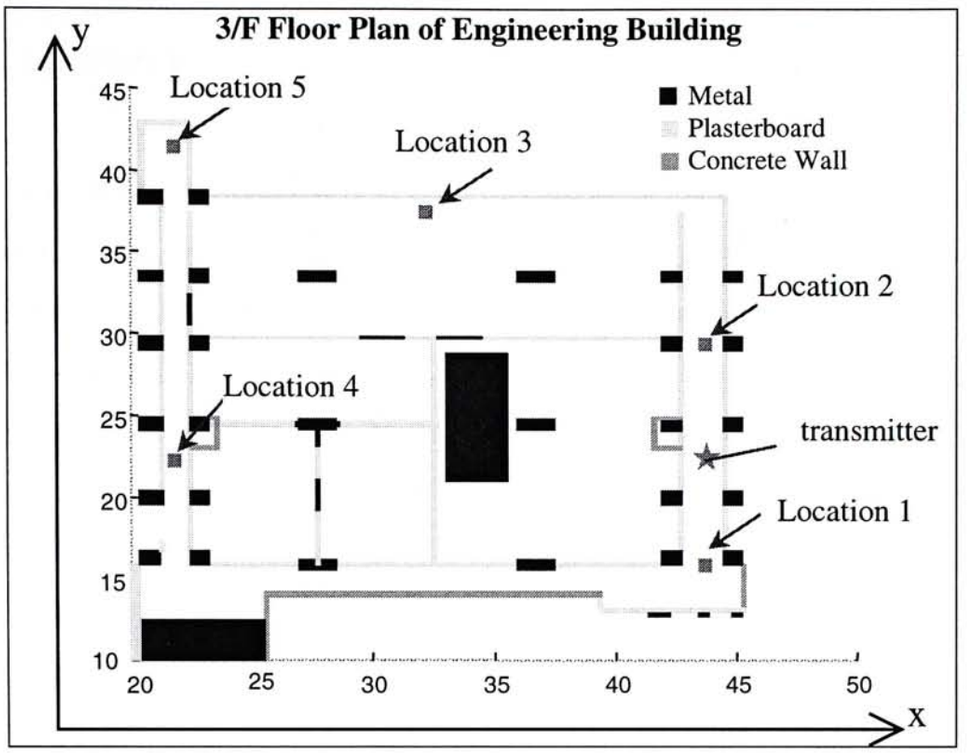


Figure 6-11 : Third floor plan of Engineering Building

6.4.1 Location 1

The coordinate of location 1 is (43.6 meters, 16 meters, 21.15 meters). The separation distance from transmitter is 6 meters. Location 1 is in the line-of-sight region of the source so the first approaching signal should be line-of-sight signal. It can be shown in the power delay profile that the first approaching signal with 20 ns delay corresponds to the line of sight path between transmitter and receiver. Since the receiver is located at the corner region, the three surrounding walls provide sufficient reflection surface reflecting the signal back to the receiver. Multi-path signal with substantial amplitude lasts for longer period than other locations that are also in the line-of-sight region. The power delay profile is plotted in Figure 6-1. The simulated delay spread in modular neural network and brute force ray tracing are 11.4 ns 9.79 ns separately. Significant signals still reach the receiver long after the first approaching signal. That is why the rms excess delay is half of the value of mean excess delay. The simulation result of modular neural network matches that of brute force ray tracing very well. The first approaching signal and the time dispersion statistics are similar in both simulations. The coherence bandwidth is defined as the bandwidth over which the frequency correlation function is above 0.5, then the coherence bandwidth is approximately equal to $\frac{1}{5\sigma_\tau}$ where σ_τ is the rms excess delay. In location 1, if the data in modular neural network is used, the coherence bandwidth is 17.54 MHz. So, in the line-of-sight region, equalizer is not necessary if a data rate of 2 Mbps is required. The supported data rate is 8.77 Mbps even if ten times of rms excess delay time is used as the symbol period.

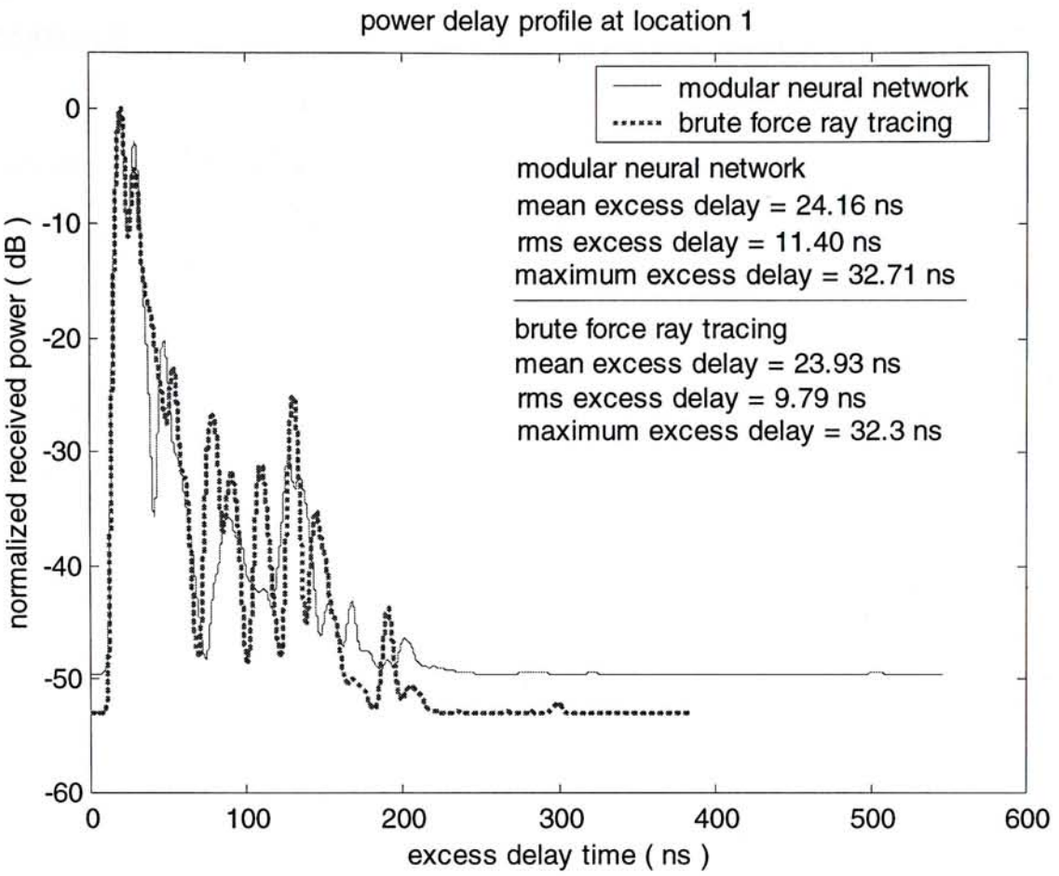


Figure 6-12 : Power delay profile at location 1

6.4.2 Location 2

The coordinate of location 2 is (43.6 meters, 28 meters, 21.15 meters). The separation distance from transmitter is 6 meters that is the same as that in location 1. Location 2 is in the line-of-sight region of the source but not at the corner region. Apart from line-of-sight signal, the secondary strongest signal is one reflection from walls besides the receiver. In contrast to location 1, the signal propagating to the end of the corridor and backing to receiver is relatively lower in amplitude. Therefore, the rms excess delay in location 2 is slightly smaller than that at location 1. The mean excess delay in location 1 and location 2 is the same, so the simulator can correctly capture the line-of-sight signal. The delay spread in modular neural network and brute force ray tracing are 8.51 ns and 12.49 ns separately. Again, both simulations result are similar although there is one missing signal that has lower in amplitude in modular neural network simulation compared to that of brute force ray tracing. The coherence bandwidth is equal to 23.5 MHz that is slightly larger than that in location 1. Also, the supported data rate is 11.75 Mbps. From these two locations illustration, higher data rate can be supported by the system when the receiver exists in the line-of-sight region. These are still substantially higher than the 2 Mbps requirement.

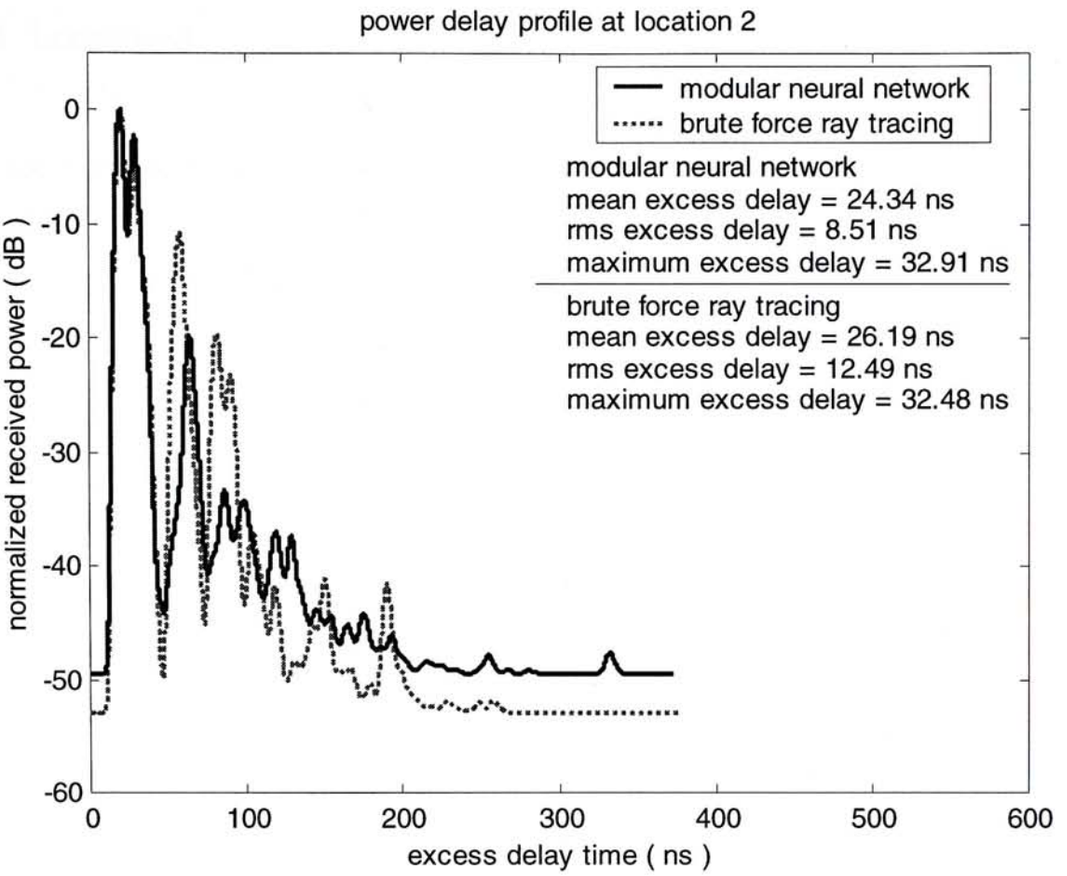


Figure 6-13 : Power delay profile at location 2

6.4.3 Location 3

The coordinate of location 3 is (33.1 meters, 37 meters, 21.15 meters). The separation distance from transmitter is 18.31 meters. Location 3 is positioned at the shadow region with reference to the source. The first approaching signal is expected to be low in amplitude since there is a metal-liked pillar between the transmitter and location 3. The first approaching signal is a direct path since the delay time is 60 ns, exactly the time for a ray propagating from transmitter to receiver directly. In the power delay profile, the first approaching signal is the strongest signal. It shows that the metal-liked pillar cannot block the signal significantly in the simulation. Multi-path effect is more severe compared with the previous locations since it is located at the shadow region. Also, the multi-path signal after the first approaching signal is clearly separated because location 3 is at the middle of path 3 and it is located within a large empty soft-partitioned office. All four significant signals can be accurately predicted by both modular neural network and brute force ray tracing. The coherence bandwidth is 3.91 MHz and the supported data rate is 1.96 Mbps. This is significantly lower than the previous two locations.

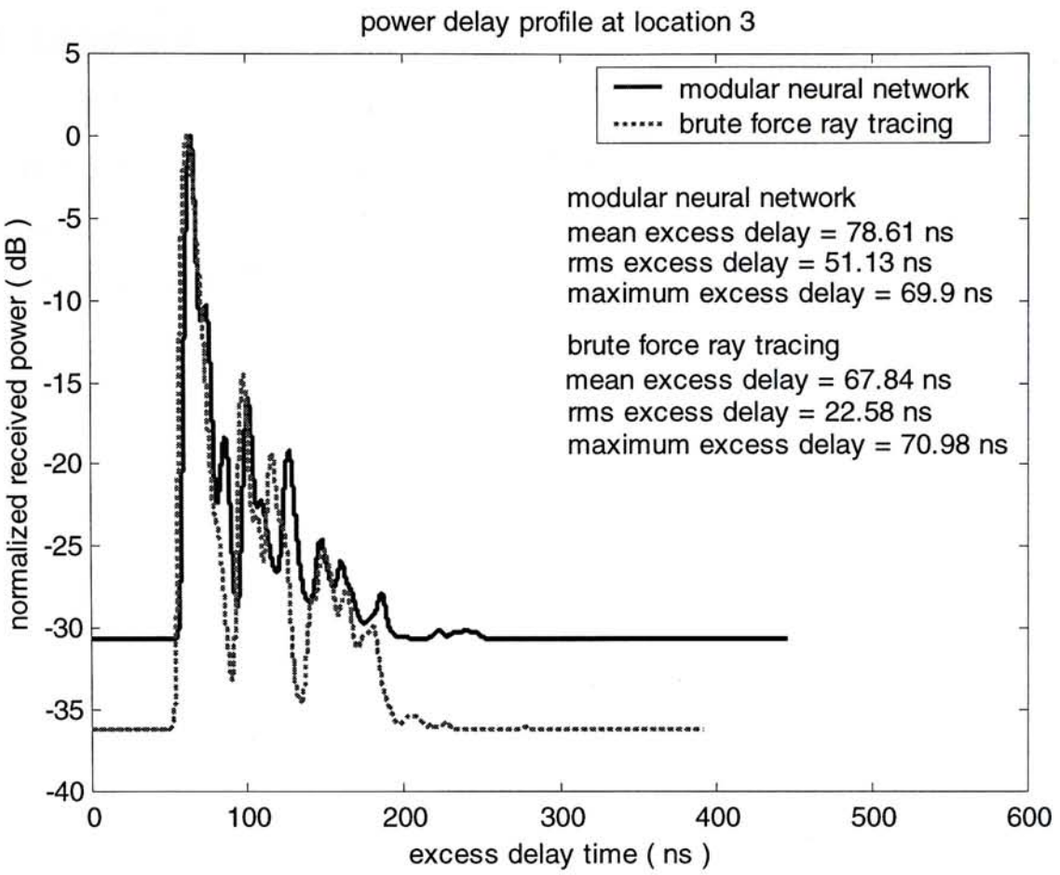


Figure 6-14 : Power delay profile at location 3

6.4.4 Location 4

The coordinate of location 4 is (21.5 meters, 22 meters, 21.15 meters). The separation distance from the transmitter is 22.1 meters. The metal chamber exactly blocks the signal from the transmitter. It is expected that the mean excess delay is relatively large since the first approaching signal will pass through substantial reflection processes. This is confirmed in the power delay profile, since the first approaching signal has a 110 ns delay. If signal came from the transmitter directly, the delay should be 73.67 ns. The situation of location 4 is different from that of location 3 since it is located in the deep shadow region. Apart from the blockage of metal chamber, signal has to undergo multiple reflections and transmissions before it can reach the receiver at location 4. The dynamic range of the predicted data from the modular neural network is only 10 dB while that of brute force ray tracing is 20 dB. All signals are low in amplitude. The primary and secondary signal comes with large difference in delay. The difference in delay time is 50 ns, three to five times of the rms excess delay in location1 and location 2 where line-of-sight exists. The coherence bandwidth is 1.628 MHz and the supported data rate is 814 kbps. The supported data rate is much smaller than that of other locations.

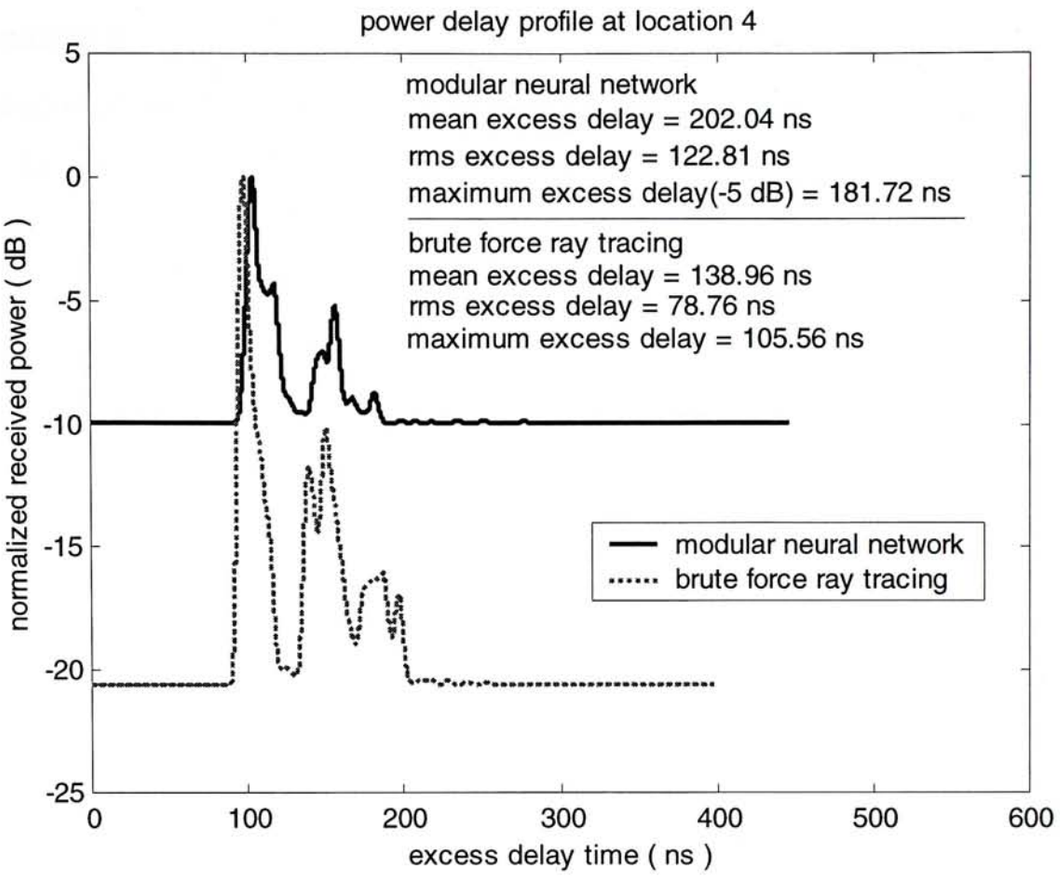


Figure 6-15 : Power delay profile at location 4

6.4.5 Location 5

The coordinate of location 5 is (21.5 meters, 41 meters, 21.15 meters). The separation distance from the transmitter is 29.14 meters. It is the longest distance from the transmitter in all measurement locations. Since there is no metal object between location 5 and transmitter, the first approaching path is the transmitted signal coming from the transmitter directly. The delay time for a ray from transmitter to this location is 97.13 ns. In the modular neural network results, there are two dominant signals in the first detectable region but only one dominant signal exists in brute force ray tracing. Moreover, the power of the first approaching signal in the modular neural network result is not the largest of all and there is another major signal arriving 10 ns later. As the separation distance between location 5 and the transmitter is very long, many possible ray paths are available. Also, the rms excess delay in modular neural network is much larger than that of brute force ray tracing. From the simulations, multi-path effect of location 5 is the worst among all five locations. The rms excess delay is the largest value if the simulation of modular neural network is used. The coherence bandwidth is 1.30 MHz and the supported data rate is 648 kbps.

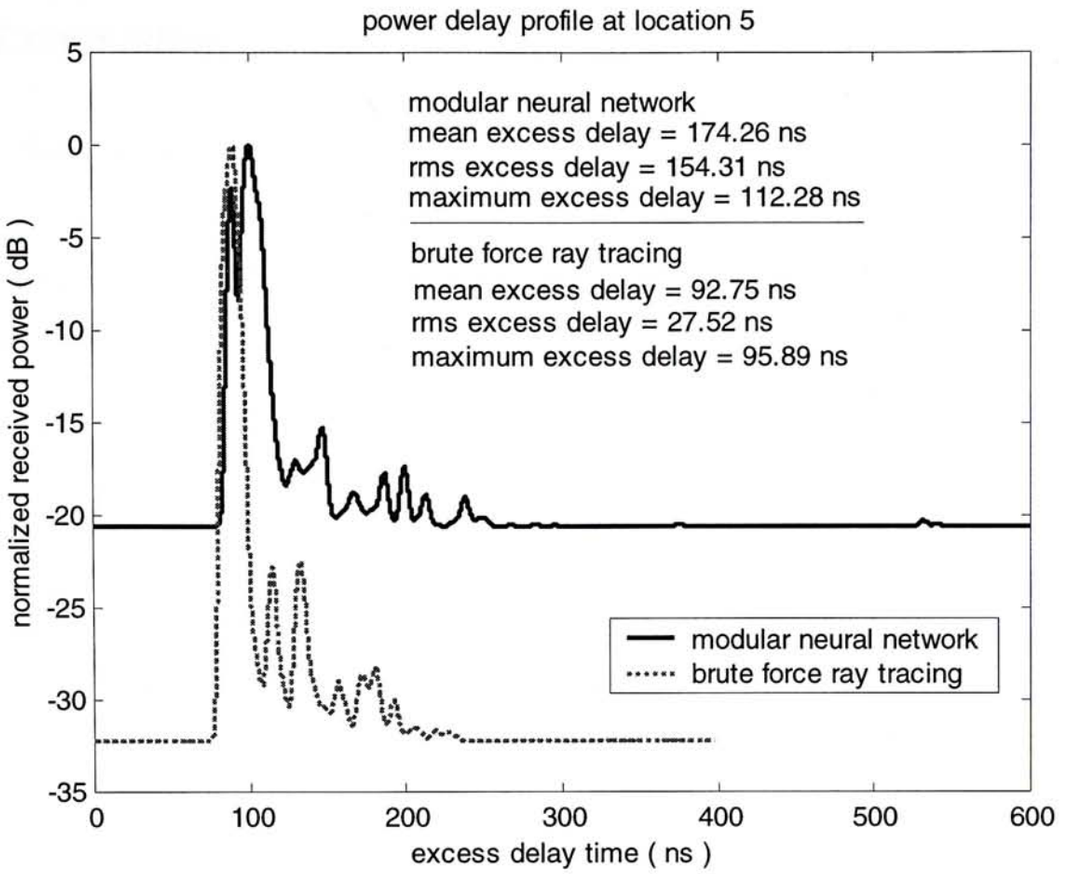


Figure 6-16 : Power delay profile at location 5

6.5 Summary

In this chapter, modular neural network results are compared with brute force ray tracing and measurement, confirming its usefulness and accuracy. It is then used to calculate the multi-path effects and the resulting acceptable data rate. It shows that a large variation of acceptable data rate exists in the environment tested.

7 Conclusion

7.1 Summary

Advanced mobile communications system requires site-specific radio wave propagation prediction tool to optimize the network in terms of base station locations and the frequency allocation scheme. Applying deterministic modeling in network optimization in urban environment is not practical because of their large computation resource requirement. Recently, some researchers used neural network architecture to model radio propagation. However, their models are not truly general for all kinds of environment due to the limitation of their defined model and training data collection methods. In this thesis, function-based and physics-based hybrid modular neural network architecture with true generalization capability was developed for radio wave propagation prediction.

Modular neural network architecture for radio wave propagation modeling is developed in the following manner. The architecture is shown in Figure 7-1. First, the brute force ray tracing algorithm is divided into different sections according to their functions and physical relations. The original problem now becomes many small problems and these problems can be individually solved. A suitable neural network model is applied to each module. Each neural network is then trained to solve its corresponding problem. Finally, these neural networks are combined to form a large network. The whole network may need to be trained again in terms of network connection, while the architecture of each module remains fixed. This methodology distributes the large training effect into small pieces. The training process of each module can be done concurrently. After training of each small module is done, the overall architecture is almost ready. Second round training is relatively simple since it consists of minor adjustments in the whole network. The network is highly testable because it is carrying out ray tracing: the physics of that is clearly presented at the output of each module and not lost in the myriad of a massive neural connections.

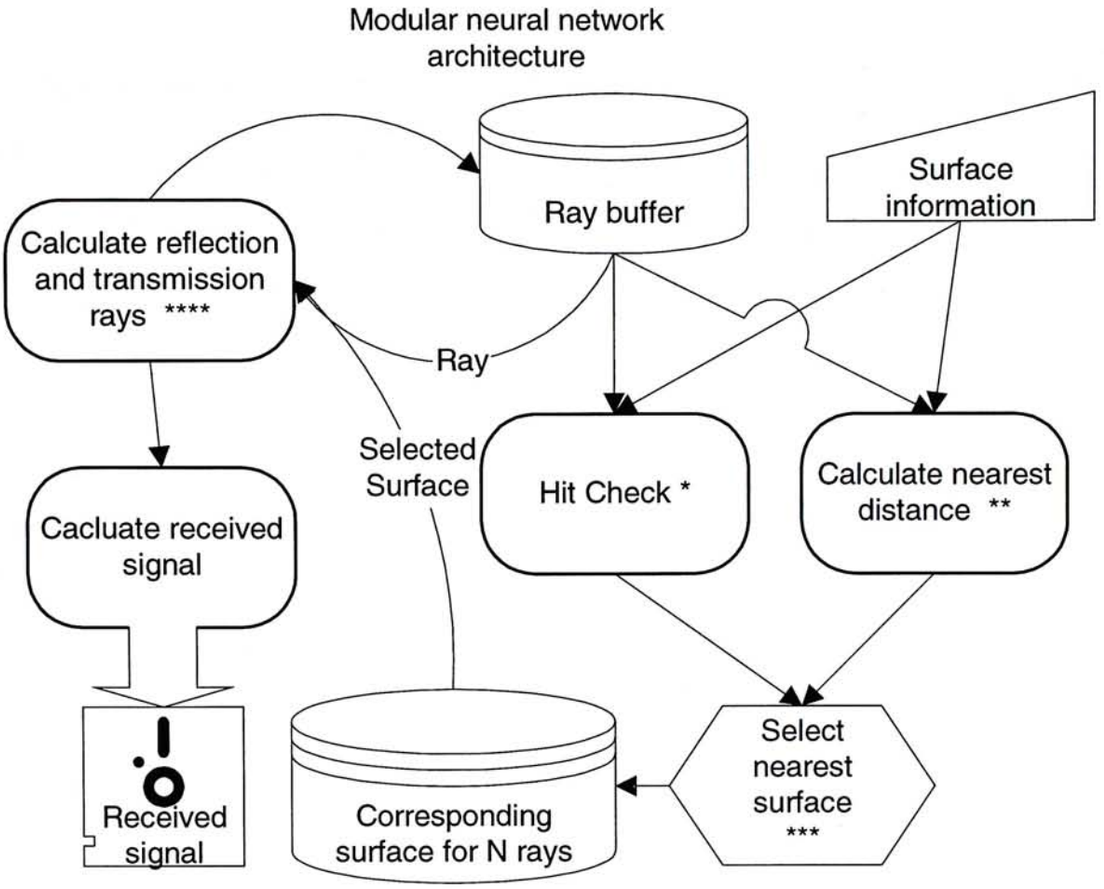


Figure 7-1 : Modular neural network architecture

In this architecture, there are 18 neural network modules that are prepared individually. These 18 neural network modules include coordinate conversion, hit point calculation, hit check, separation distance calculation, polarization vector of secondary ray calculation, lower amplitude ray rejection. Each module may contain more than one neural network sub-net. They are connected together to form the final model. Multilayer Perceptron was used in all the neural network models and deterministic calculation result was used to train the neural network. All individual network was evaluated by using testing data generated from the designed ranges.

As each network has its position and meaning in the modular neural network architecture, inaccurate prediction can be back traced module by module in order to find out which one contributes to the incorrect prediction. This is testability. Also, immediate solution can be obtained for debugging and development during the simulation process. Each module can be trained individually so that handling large network structure and long training process can be avoided. Each module can be optimized according to the characteristics of the modeling task.

The first version of modular neural network can simulate 50 m × 50 m × 50 m environment and the allowable surface dimension is 50 m × 50 m with 0.1 m in thickness. Also, the dielectric constant of surface material ranges from 2 to 6. The training environment of the modular neural network in this thesis is large enough to handle most indoor environment.

Four canonical geometries were studied using brute force ray tracing method, modular neural network and deterministic method. These four geometries were: free space, ground earth, ground penetration and empty hall. The result shows that modular neural network can generate accurate prediction results. In empty hall simulation, modular neural network has 1.57 dB mean error and 1.39 dB standard deviation with reference to the simulation result of brute force ray tracing model.

As an actual test, modular neural network is used to predict propagation inside the third floor of the engineering building of the Chinese University of Hong Kong. The average prediction error of modular neural network is 6.93 dB and 6.01 dB standard deviation for shadow region, and 5.27 dB with 4.63 dB standard deviation for line-of-sight region.

7.2 Recommendations for Future Work

For future work, the network needs to be trained and tested in more environments. In the process of further training and perfecting of the proposed model, various important properties of the neural network predictor should be addressed, including speed of solution, uniqueness and repeatability of solution, generalization of solution, convergence and stability. The size and number of modules should be reduced in modular neural network in order to optimize the performance in hardware acceleration. A DSP based neural network accelerator should be developed to realize the model described in this thesis so that the true speed up advantage can be realized.

Another interesting topic is to compensate the lack of geometry details by actual

measurement. In the calculation of ray tracing, accuracy of the geometry model will affect the accuracy of prediction. If a good geometry database is not available for simulation, large prediction error is inevitable. Although digital map is available from Hong Kong Government as well as other countries, some important information is still missing, such as furniture and office layout inside the buildings. Therefore, a geometry inverse process is needed to determine effects of these hidden objects and include them in the simulation model to improve the prediction accuracy. For example, an unknown object neural network module can be developed to represent any undetermined object. However, its reflection and transmission characteristics are learned from measurement result that is obtained in the testing environment

.

Publication List

1. J.W.H. Lee and A. K. Y. Lai, "Hybrid neural network/ray tracing model for radiowave penetration into buildings," Electronics Letters, vol. 33 19, September 1997, pp. 1609-610

2. J.W.H. Lee and A. K. Y. Lai, "FDTD analysis of indoor radio propagation," IEEE Antennas and Propagation Society International Symposium, 1998, vol. 3, pp. 1664-1667

Bibliography

[1] W. T. Smith, R. J. Basian, S. Y. Cheah, "An Efficient Neural Network Algorithm for Reflector Surface Error ompensation," IEEE Trans. AP, Feb. 1996, vol. 44, no. 2, pp. 137.142

[2] S. J. Blank and W. A. Impriale, "Arrayfeed Synthesis for Correction of reflector Distortion and Vernier Beam Steering," IEEE Trans. AP, Oct. 1988, vol. 36, pp. 1351-58

[3] S. Sagoroglu and K. Guney, "Calculation of Resonant Frequency for an Equilateral Triangular Microstrip Antenna with the Use of Artificial Neural Networks," Microwave and Optical Technology Letters, Feb. 1997, vol. 14, no. 2

[4] G. Washington, "Aperture Antenna Shape Prediction by Feedforward Neural Network," IEEE Trans. AP, Arpil 1997, vol. 45, no. 4, pp. 683-688

[5] K. E. Stocker, F. M. Landstorfer, "Empirical Prediction of Radiowave Propagation by Neural Network Simulator," Electronics Letters, vol. 28 no. 8, pp. 724-726, April, 1992

[6] B. E. Gschwendtner, F.M. Landstorfer, "An Application of Neural Networks to the Prediction of Terrestial Wave Propagation," COST 231 TD (93) 21. Barcelona (Spain), January, 1993

[7] T. Balandier, A. Caminada, V. Lemoine, F. Alexandre, "170 MHz Field Strength Prediction in Urban Environments Using Neural Networks," IEEE International Symposium on Personal Indoor and Mobile Communications, vol. 1, pp. 120-124, September, 1995

[8] O. Perrault, J. P. Rossi, T. Balandier, A. Levy, "Prediction Field Strength with a Neural Ray-Tracing Model," COST 231 TD (96) 009 Belfort (France), January, 1996

[9] B. E. Gschwendtner, F. M. Landstorfer, "Adaptive Propagation Modeling Using a Hybrid Neural Network Technique," Electronics Letters, vol. 32 No. 3, pp. 162-164, February, 1996

[10] R. Fraile, N. Cardona, "Macrocellular Coverage Prediction for all Ranges of Antenna Height using Neural Networks," Universal Personal Communications, 1998. ICUPC '98. IEEE 1998 International Conference,

vol. 1, pp. 21-25

[11] G. Wolfe, F. M. Landstorfer, "A Recursive Model for the Field Strength Prediction with Neural Networks," 10th International Conference on Antennas and Propagation, April, 1997, pp. 174-177

[12] G. Wolfe, F. M. Landstorfer, "Field Strength Prediction in Indoor Environments with Neural Networks," Proceedings of the 47th Vehicular Technology Conference. Denver (U.S.A.). July, 1997, pp. 82-86

[13] G. Wolfle, F. M.. Landstorfer, "Dominant paths for the field strength prediction," Vehicular Technology Conference, 1998. VTC 98. 48th IEEE vol. 1 , pp. 552 -556

[14] J. Stephen. Judd, Neural network design and the complexity of learning, 1990

[15] Young-Joo Moon, Se-Young Oh, "On an efficient design algorithm for modular neural networks," IEEE International Conference on Neural Networks, 1995, vol. 3, pp. 1310 –1315

[16] Cheng-Chin Chiang, Hsin-Chia Fu, "A divide-and-conquer methodology for modular supervised neural network design," IEEE International Conference on Neural Networks, 1994, vol. 1 , pp. 119 –124

[17] Yong Liu, Xin Yao, "Evolving modular neural networks which generalise well," IEEE International Conference on Evolutionary Computation, 1997 , pp. 605 –610

[18] W. J. Jansen, M. Diepenhorst, J. A. G. Nijhuis, L. Spaanenburg, "Assembling engineering knowledge in a modular multi-layer perceptron neural network," International Conference on Neural Networks,1997, vol. 1 , pp. 232 –237

[19] D. Cheng, Field and Wave Electromagnetics, 1989

[20] T. S. Rappaport, R. Muhamed,and V. Kapoor, Propagation Models-The Mobile Communications Handbook, ed. J.D. Gibson, IEEE Press, 1996, pp.355-369

[21] S.Y. Seidel and T.S. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings, " IEEE Trans. Antennas Propagat., February, 1992, Vol.40, No.2, pp.207-217

[22] B. Gschwendtner and F. Landstorfer, "Adaptive propagation modelling

based on neural network techniques, "IEEE Vehicular Technology Conference, 1996, pp.623-626

[23] Q. Zhou and K. Y. Lai, Neural network predictor of relative received mean signal strengths in in-room radio channel, Internal Report, 1996

[24] K. R. Schaubach, N. J. Davis, and T. S. Rappaport, "A ray tracing method for predicting path loss and delay spread in microcellular environments," IEEE Proc. of Vehicular Technology Conference, 1992, pp. 932-935

[25] W. C. Lawton and J.P. McGeehan, "The application of GTD and ray launching techniques to channel modelling for cordless radio systems," IEEE Proc. of Vehicular Technology Conference, 1992, pp. 125-130

[26] P. F. Driessen, M. Gimersky, T. Rhodes, "Ray model of Indoor Propagation," Wireliess Personal Communications, ed. M.J. Feuerstein, T.S.Rappaport. Kluwer Academic Publishers, 1993, pp.225-249

[27] S. E. Alexander, "Characterising buildings for propagation at 900 MHz," Electronics Letters, 29th Sept., 1983, Vol.19 No.20, pp.860

[28] A. A. M. Saleh and R. A. Valenzuela, "A statistical model for indoor multipath propagation," IEEE Journal on Selected Areas in Communications, February, 1987, Vol.SAC-5, No.2, pp.128-137

[29] W. K. Tam and V.N. Tran, "Propagation modelling for indoor wireless communication," Electronics & Communication Engineering Journal, October, 1995, pp.221-228

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

[31] J. Horikoshi, K. Tanaka, and T. Morinaga, "1.2 GHz band wave propagation measurements in concrete building for indoor radio communications," IEEE Trans. Veh. Techno., November, 1986, Vol.VT-35, No.4, pp.146-152

[32] G. Vallejo-Cabrejas and P. Batolome-Pascual, "Application of radio channel modelling to a planning tool in a mobile radio indoor communication system," IEEE Proc. of Vehicular Technology Conference, 1993, pp. 219-222

[33] A.U. Sheikh, M.Handforth and M. Abdi, "Indoor mobile radio channel at 946 MHz: Measurements and modeling," IEEE Proc. of Vehicular

Technology Conference, 1993, pp. 73-76

[34] J. W. McKown and R.L. Hamilton, Jr., "Ray tracing as a design tool for radio networks," IEEE Network Magazine, 1991, pp.27-30

[35] R. A. Valenzuela, "A ray tracing approach to predicting indoor wireless transmission," IEEE Proc. of Vehicular Technology Conference, 1993, pp.214-218

[36] S. J. Fortune, D. M. Gay, B. W. Kernighan, O. Landron, R. A. Valenzuela, and M. H. Wright, "WISE design of indoor wireless systems: Practical computation and optimization," IEEE Computational Sci. & Engineering, Spring, 1995, pp.58-68

[37] P. Kreugruber, P. Unterberger, and Rainer Gahleitner, "A ray splitting model for indoor radio propagation associated with complex geometries," IEEE Proc. of Vehicular Technology Conference, 1993, pp. 227-230

[38] D. J. C. MacKay, "Bayesian interpolation," Neural Computation, vol. 4, no. 3, pp. 415-447, 1992.

[39] F. D. Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian regularization," Proceedings of the 1997 International Joint Conference on Neural Networks, 1997, pp. 1930-1935

[40] K. T. Lam, "Indoor Wireless Propagation," Internal report, 1997