Simulation and Modeling of Fading Channel and Improvement of Channel estimation using Artificial Neural Network

Hemchand Vashist, Arvind Pathak, Amarjeet Kaur, Sanjay Sharma, Ashish Sharma, Meenu Jangir

***Abstract-** We present a simulation of Rayleigh fading channel. We used MATLAB for its implementation. Further this work explored possible use of Artificial Neural network for channel estimation. This is one of the first steps in improving performance in fading environment. We present a novel algorithm for channel estimation using ANN with the help of user's usage history.*

***Keywords-** MATLAB.*

I. INTRODUCTION

Signal fading refers to the rapid change in received signal strength over a small travel distance or time interval. This occurs because in a multipath propagation environment, the signal received by the mobile at any point in space may consist of a large number of plane waves having randomly distributed amplitudes, phases, delays and angles of arrival. These multipath components combine vectorily at the receiver antenna. They may combine constructively or destructively at different points in space, causing the signal strength to vary with location. If the objects in a radio channel are stationary, and channel variations are considered to be only due to the motion of the mobile, then signal fading is a purely spatial phenomenon. A receiver moving at high speed may traverse through several fades in a short period of time. If the mobile moves at low speed, or is stationary, then the receiver may experience a deep fade for an extended period of time. Reliable communication can then be very difficult because of the very low signal-to-noise ratio (SNR) at points of deep fades.

Extensive field measurements have previously been done by many authors to characterize the small-scale spatial distribution of the received signal amplitude in multipath propagation environments. It has been found that for many environments, the Rayleigh distribution provides a good fit to the signal amplitude measurement in environments where no line-of-sight or dominant path exists. The probability density function of the Rayleigh distribution is given by

$$p(r) = \begin{cases} \frac{r}{\sigma^2}\exp(-\frac{r^2}{\sigma^2}) & r \geq 0. \\ 0 & otherwise, \end{cases} \tag{1}$$

Where σ^2 is the parameter of the distribution. A plot of the Rayleigh probability density function is shown in Figure 1.2. The Rayleigh distribution is related to the zero-mean Gaussian distribution in the following manner. Let X_1 and X_Q be two independent, identically distributed, zero-mean Gaussian random variables with variance σ^2 .The marginal probability density functions of X_1 and X_Q are given by :

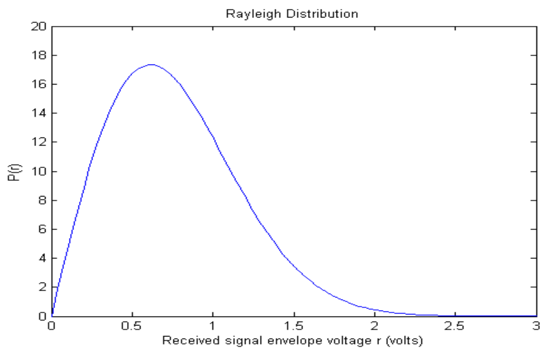
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{x^2}{2\sigma^2}) \qquad , -\infty \langle x \langle \infty . \tag{2}$$

Then the random variable R , defined as:

$$R = \sqrt{X_1^2 + X_Q^2} \tag{3}$$

It is distributed according to the Rayleigh probability density function given in Equation (1). The fact that the Rayleigh distribution provides a good fit to the measured signal amplitudes in a non-line-of-sight environment can be explained as follows. When a signal is transmitted through a multipath propagation channel, the in-phase and quadrature-phase components of the received signal are sums of many random variables. Because there is no line-of-sight or dominant path, these random variables are approximately zero-mean. Therefore, by the central limit theorem, the in-phase and quadrature-phases components can be modeled approximately as zero mean Gaussian random processes.

The amplitude, then, is approximately Rayleigh distributed.



Manuscript Received on November, 2012.

Er. Hemchand Vashist, Electronics and Communication Engineering Department, ACME, India.

Arvind Pathak, Electronics and Communication Engineering Department, Lingaya's University, India.

Er. Amarjeet Kaur, Electronics and Communication Engineering Department, ACME, India.

Sanjay Sharma, Electronics and Communication Engineering Department, ACME, India.

Er. Ashish Sharma, Electronics and Communication Engineering Department, ACME, India.

Er. Meenu Jangir, Electronics and Communication Engineering Department, ACME, India.

Figure 1: Rayleigh probability density function (pdf).

II. USING ARTIFICIAL NEURAL NETWORK FOR IMPROVING CHANNEL ESTIMATION

A. Biological Neuron:

Artificial neural networks could surpass the capabilities of conventional computer-based pattern recognition systems. An artificial neural network seeks to emulate the function of the biological neural network that makes up the brains found in nearly all higher life forms found on Earth. Neural networks are made up of neurons. A diagram of an actual neuron is shown in Figure 2

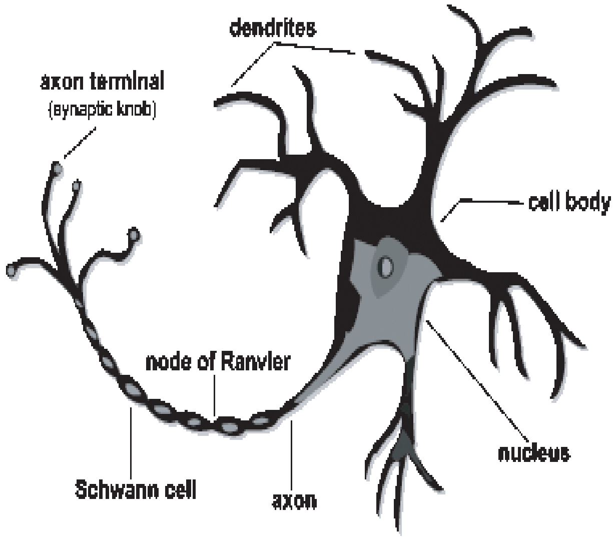


Figure2: A biological neuron

From Figure 2, the neuron is made up of a core cell and several long connectors, which are called synapses. These synapses are how the neurons are connected amongst themselves. Neural networks, both biological and artificial, work by transferring signals from neuron to neuron across the synapses.

Neurons are the most important units in the nervous system. There are approximately 100 billion neurons in the brain, each of which is amazingly complex in itself. From a simplistic viewpoint, a neuron is a basic processing unit. A neuron receives input from other neurons, processes and integrates it, then bases its output (or lack thereof) on this integration.

Neurons are made up of several distinct parts. The soma (cell body) surrounds the nucleus of the cell, where the genetic information is stored. Floating around inside the soma (in a fluid called the cytoplasm) are molecular structures which perform various tasks.

Connected with the soma are **dendrites**, branch-like extensions which receive input from other neurons. On the other end of the cell is the axon, or the output cable of the cell. The **axon** branches out in a dense network; each branch stretches out to other neurons. At the end of these branches are small bumps, called **terminal buttons**, which contain the chemicals necessary for communication with other cells. The terminal buttons link to the dendrites of other neurons. The two cells do not actually touch; there is a very small gap between the terminal button and the dendrite across which chemical communication takes place. This junction of terminal button of one cell to the dendrite of another is called a **synapse** (synapse can also be used as a verb, meaning "to form a synapse with").

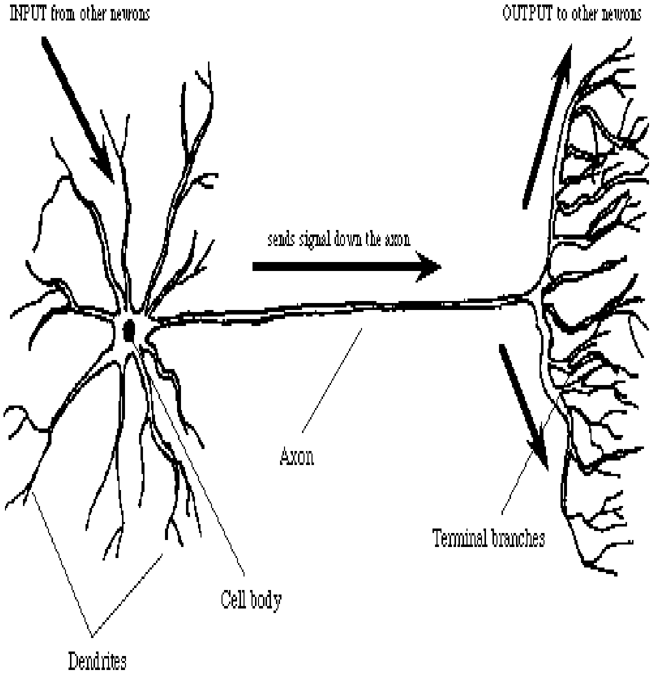


Figure 3: Schematic View of A Neuron

B. Artificial Neural Network

Artificial Neural Network is to mimic the human ability to adapt to changing circumstances and the current environment. **Artificial Neural Network** consists of many nodes i.e.:- **Processing Units.**

The other part of the "art" of using neural networks revolves around the myriad of ways these individual neurons can be clustered together. This clustering occurs in the human mind in such a way that information can be processed in a dynamic, interactive, and self-organizing way. Biologically, neural networks are constructed in a three-dimensional world from microscopic components. These neurons seem capable of nearly unrestricted interconnections. That is not true of any proposed, or existing, man-made network. Integrated circuits, using current technology, are two-dimensional devices with a limited number of layers for interconnection. This physical reality restrains the types, and scope, of artificial neural networks that can be implemented in silicon.

Finally, the processing element is ready to output the result of its transfer function. This output is then input into other processing elements, or to an outside connection, as dictated by the structure of the network. All artificial neural networks are constructed from this basic building block - the processing element or the artificial neuron. It is variety and the fundamental differences in these building blocks which partially cause the implementing of neural networks to be an "art."

C. Comparison of BNN and ANN

Table 1: Comparison Between BNN and ANN

Parameter	ANN	BNN
Speed	ANN are faster in processing information	BNN are slow in b/n processing information
Processing	Many programs have large no of instruction and they operate in a sequential Mode.	It can perform Massively Parallel operation
Size and Complexity	These do not involve as much computational neuron	NN have large no of computing elements
Storage	In a computer ,the information is stored in the memory,	NN store information in the strengths of the interconnection
Fault Tolerance	Not Fault Tolerant	Fault Tolerant

Table 2: Comparison Between Computer and BNN

S. No.	COMPUTER	NN
Processor	-Complex -high Speed -One or a few	-Simple -Low Speed -A large No.
Memory	-separate from a processor -localized -non content addressable	-Integrated into Processor -Distributed -Content Addressable.
Computer Net	-Centralized -Sequential -Stored Program	-Distributed -Parallel -Self –Learning.
Operating Environment	-Well Defined -Well constrained	-Poorly Defined -unconstrained.

III. NETWORK ARCHITECTURE

A. Network Architecture:-

Following are the types of Network Architectures:-

I. Feed Forward
II. Feed Back
III. Fully Interconnected
IV. Competitive.

Feed Forward Net: - It may have a single layer of wts where the inputs are directly connected to the output or multilayer with intervening set of hidden units. The ANN use hidden units to create internal representation of the input Pattern.

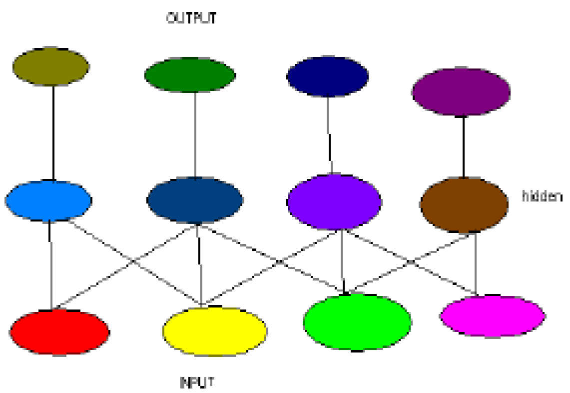


Figure 4:Multi-layer

1. **Single Layer network: -** It is a feed forward network. It has only one layer of weighted interconnection. The input may be connected fully to the output units.
2. **Multi layer network:-** It is a feed forward network ,the network where the signals flow from the input units to the output units in a forward direction.
3. **Recurrent Network: -** All units are connected to all other units and every units are both an input and an output .It is allow to process sequential information.Setting the weights:-

The method of setting the value for the weights enables the process of learning or training. The Process of modifying the weights in the connection b/n n/w layers with the objective of achieving the expected output is called training a network.

Three types of training:-

I. Supervised Training
II. Unsupervised Training
III. Reinforcement Training

Supervised Training: - It is the process of providing the n/w with a series of sample inputs and comparing the output with the expected responses.

I. Binary Adaptive Resonance Theory (ART1)
II. Analog Adaptive Resonance Theory (ART2, ART2a)
III. Discrete Hopfield (DH)
IV. Continuous Hopfield (CH)
V. Discrete Bidirectional Associative Memory (BAM)
VI. Kohonen Self-organizing Map/Topology-preserving map

ARTMAP is based on ART:-

I. ARTMAP's internal control mechanisms create stable recognition categories of optimal size by maximizing code compression while minimizing predictive error during on-line learning.
II. ARTMAP incorporates fuzzy logic in its ART modules
III. ARTMAP has fuzzy set-theoretic operations instead of binary set-theoretic operations.
IV. It learns to classify inputs by a fuzzy set of features (or a pattern of fuzzy membership values between 0 and 1)

Unsupervised Training: - In a N Net, if for the training input vector, the target o/p is not known, the training method adopted is called as unsupervised training. It is called self –learning n/w or SOM., because of their ability to carry out self learning.

I. Learning Matrix (LM)
II. Sparse Distributed Associative Memory (SDM)
III. Fuzzy Associative Memory (FAM)
IV. Counter propagation (CPN)

Reinforcement: - In this method, a teacher is also assumed to be present, but the right answer is not presented to the network. In this process, the output may not be indicated as the desired output but condition may be indicated.

B. Learning Rules:-

Learning is the process by which the free parameter of neural N/W gets adapted through a process of simulation by the environment in which the N/W is embedded. The type of learning is determined by the manner in which the parameter changes take place. The set of well defined rules for the solution of a learning problem is called a learning algorithm. Each learning algorithm differs from the other in the way in which the adjustment to a synaptic wts of a neuron is formulated.

There are various learning Rules:-

I. Hebbian Learning Rules
II. Perception Learning Rules
III. Delta Learning Rules

IV. Competitive Learning Rules
V. Out Star Learning Rules
VI. Boltzmann Learning Rules
VII. Memory Based Learning Rules

Delta Learning Rules:-

It is valid only for continuous activation functions and in the supervised training Mode. The learning signal for this rule is called DELTA.

Delta rules stated as:-

"The adjustment made to synaptic weights of a neuron is proportional to the product of the error signal and the input signal of the synaptic".

The delta rules assume that the error signal is directly measurable. The aim of the delta rule is to minimize the error over all training Patterns .Here enhance the delta bar delta by applying an exponential decay to the learning rate increase, add the momentum component back in, and put a cap on the learning rate and momentum coefficient. As discussed in the section on back-propagation, momentum is a factor used to smooth the learning rate. It is a term added to the standard weight change which is proportional to the previous weight change. In this way, good general trends are reinforced, and oscillations are dampened.

The learning rate and the momentum rate for each weight have separate constants controlling their increase and decrease. Once again, the sign of the current error is used to indicate whether an increase or decrease is appropriate. The adjustment for decrease is identical in form to that of Delta Bar Delta. However, the learning rate and momentum rate increases are modified to be exponentially decreasing functions of the magnitude of the weighted gradient components. Thus, greater increases will be applied in areas of small slope or curvature than in areas of high curvature. This is a partial solution to the jump problem of delta bar delta.

To take a step further to prevent wild jumps and oscillations in the weights, ceilings are placed on the individual connection learning rates and momentum rates. And finally, a memory with a recovery feature is built into the algorithm. When in use, after each epoch presentation of the training data, the accumulated error is evaluated. If the error is less than the previous minimum error, the weights are saved in memory as the current best. A tolerance parameter controls the recovery phase. Specifically, if the current error exceeds the minimum previous error, modified by the tolerance parameter, than all connection weight values revert stochastically to the stored best set of weights in memory. Furthermore, the learning and momentum rates are decreased to begin the recovery process.

C. Feed Forward Network :-

Two basic types of this network are :-

Network with Feedback

Network without Feedback

In the networks with Feedback, The o/p values can be traced back to the input values. However there are N/W wherein for every input vector laid on the N/W,an output vector is calculated and this can be read from the output neuron . There in no Feedback Hence only a forward flow of information is present .N/W having this structure are called as feed forward N/W. One of the most important types of feed forward N/W is the Back- Propagation N/W.

Back Propagation Network: - It is a systematic method for training MNN . It has a Mathematical foundation. It is multi-layer forward N/W using Delta Learning Rule. I.e. also Known as back-propagation rule.

The aim of this N/W is to train the net to achieve a balance b/n the ability to respond correctly to the input Patterns that are used for training and the ability to provide good response to the input that are similar.

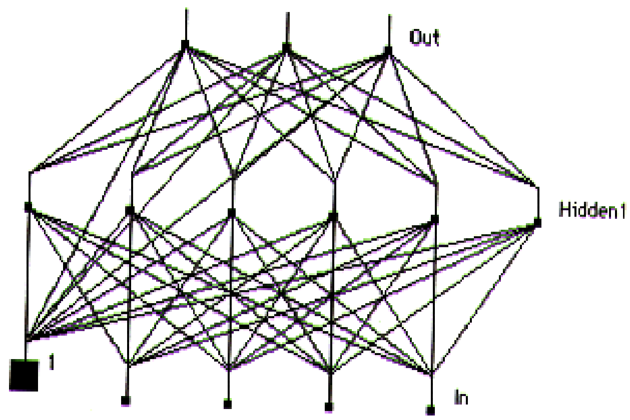


Figure 5: An Example Feed forward Back-propagation Network.

The number of layers and the number of processing element per layer are important decisions. These parameters to a feed forward, back-propagation topology are also the most ethereal. They are the art of the network designer. There is no quantifiable, best answer to the layout of the network for any particular application. There are only general rules picked up over time and followed by most researchers and engineers applying this architecture of their problems.

Rule One: As the complexity in the relationship between the input data and the desired output increases, then the number of the processing elements in the hidden layer should also increase.

Rule Two: If the process being modeled is separable into multiple stages, then additional hidden layer(s) may be required. If the process is not separable into stages, then additional layers may simply enable memorization and not a true general solution.

Rule Three: The amount of training data available sets an upper bound for the number of processing elements in the hidden layers.

To calculate this upper bound, use the number of input output pair examples in the training set and divide that number by the total number of input and output processing elements in the network. Then divide that result again by a scaling factor between five and ten. Larger scaling factors are used for relatively noisy data. Extremely noisy data may require a factor of twenty or even fifty, while very clean input data with an exact relationship to the output might drop the factor to around two. It is important that the hidden layers have few processing elements. Too many artificial neurons and the training set will be memorized. If that happens then no generalization of the data trends will occur, making the network useless on new data sets.

International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075, Volume-1 Issue-6, November 2012

Once the above rules have been used to create a network, the process of teaching begins. This teaching process for a feed forward network normally uses some variant of the Delta Rule, which starts with the calculated difference between the actual outputs and the desired outputs. Using this error, connection weights are increased in proportion to the error times a scaling factor for global accuracy. Doing this for an individual node means that the inputs, the output, and the desired output all have to be present at the same processing element. The complex part of this learning mechanism is for the system to determine which input contributed the most to an incorrect output and how does that element get changed to correct the error. An inactive node would not contribute to the error and would have no need to change its weights.

To solve this problem, training inputs are applied to the input layer of the network, and desired outputs are compared at the output layer. During the learning process, a forward sweep is made through the network, and the output of each element is computed layer by layer. The difference between the output of the final layer and the desired output is back-propagated to the previous layer(s), usually modified by the derivative of the transfer function, and the connection weights are normally adjusted using the Delta Rule. This process proceeds for the previous layer(s) until the input layer is reached. There are many variations to the learning rules for back-propagation network. Different error functions, transfer functions, and even the modifying method of the derivative of the transfer function can be used. The concept of ³momentum error² was introduced to allow for more prompt learning X while minimizing unstable behavior. Here, the error function, or delta weight equation, is modified so that a portion of the previous delta weight is fed through to the current delta weight. This acts, in engineering terms, as a low-pass filter on the delta weight terms since general trends are reinforced whereas oscillatory behavior is canceled out. This allows a low, normally slower, learning coefficient to be used, but creates faster learning.

IV. IMPLEMENTATION

A. Applying ANN for fading channel estimation

One of the most important steps in fading wireless communication systems is to do channel estimation. This is shown in the following figure.

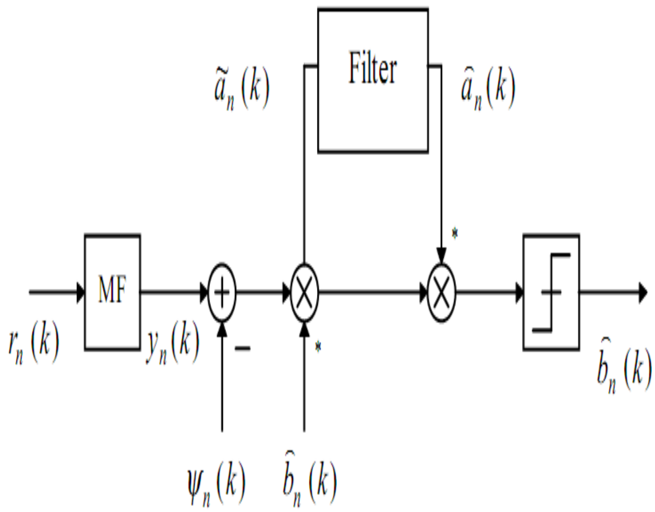
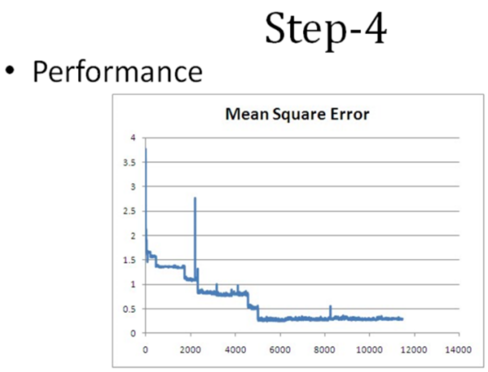


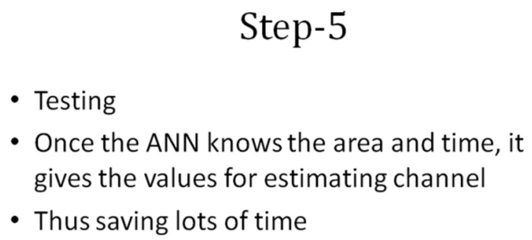
Figure 6: Feed Forward N/W

Clearly it shows that it involves lots of complex calculations which will involve considerable computational time. In this work we propose a novel method for channel estimation using ANN.

The core idea is our system will take into account the users history of talking habits and its location of roaming. For example, a user might be at fixed location daily at 9:30 AM. So this information can be used by the ANN and will help in estimating the channel dynamically. This will thus gives the calculations at much higher rate.

We describe these ideas first using a block diagrams as shown below:





V. RESULT

Matlab program

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%
close all
clear all
N = 10^6;
x = randn(1,N); % gaussian random variable, mean 0, variance1
y = randn(1,N); % gaussian random variable, mean 0, variance1
z = (x + j*y); % complex random variable
% probability density function of abs(z)
zBin = [0:0.01:7];
sigma2 = 1;
pzTheory = (zBin/sigma2).*exp(-(zBin.^2)/(2*sigma2)); % theory
[nzSim zBinSim] = hist(abs(z),zBin); % simulation
% probability density of theta
thetaBin = [-pi:0.01:pi];
pThetaTheory = 1/(2*pi)*ones(size(thetaBin));
[nThetaSim thetaBinSim] = hist(angle(z),thetaBin); % simulation
figure
plot(zBinSim,nzSim/(N*0.01),'m','LineWidth',2);
hold on
plot(zBin,pzTheory,'b.-')
xlabel('z');
ylabel('probability density, p(z)');
legend('simulation','theory');
title('Probability density function of abs(z)' )
axis([0 7 0 0.7]);
grid on

figure
plot(thetaBinSim,nThetaSim/(N*0.01),'m','LineWidth', 2);
hold on
plot(thetaBin,pThetaTheory,'b.-')
xlabel('theta');
ylabel('probability density, p(theta)');
legend('simulation','theory');
title('Probability density function of theta')
axis([-pi pi 0 0.2])
grid on
```

Lab shots for the required program is as given below

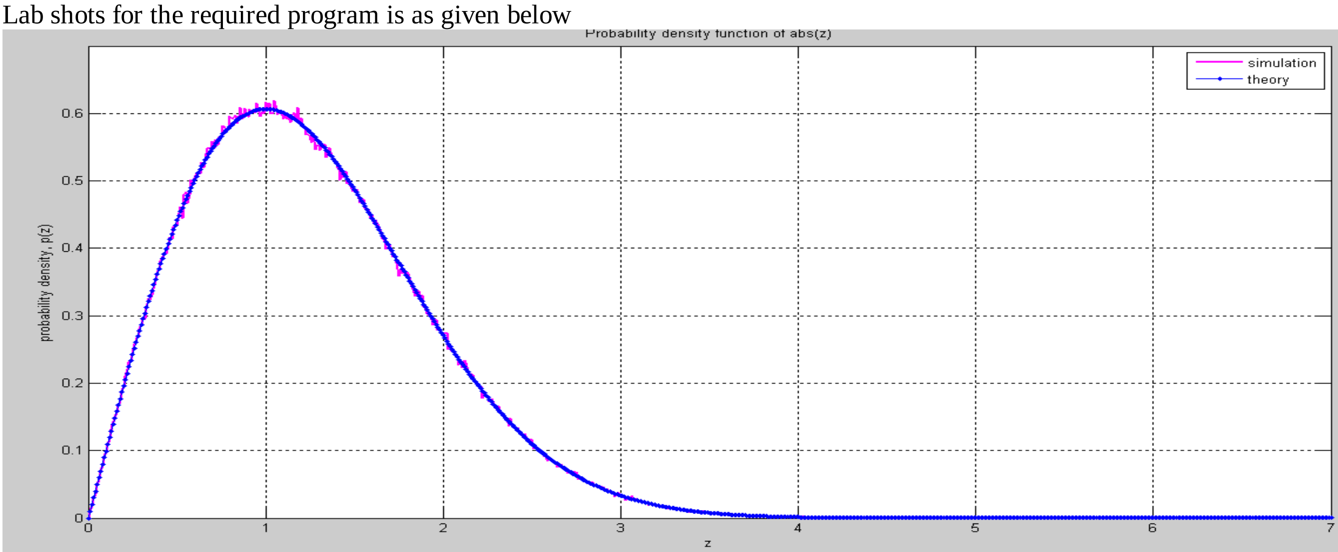


Figure 7: PDF of abs(z)

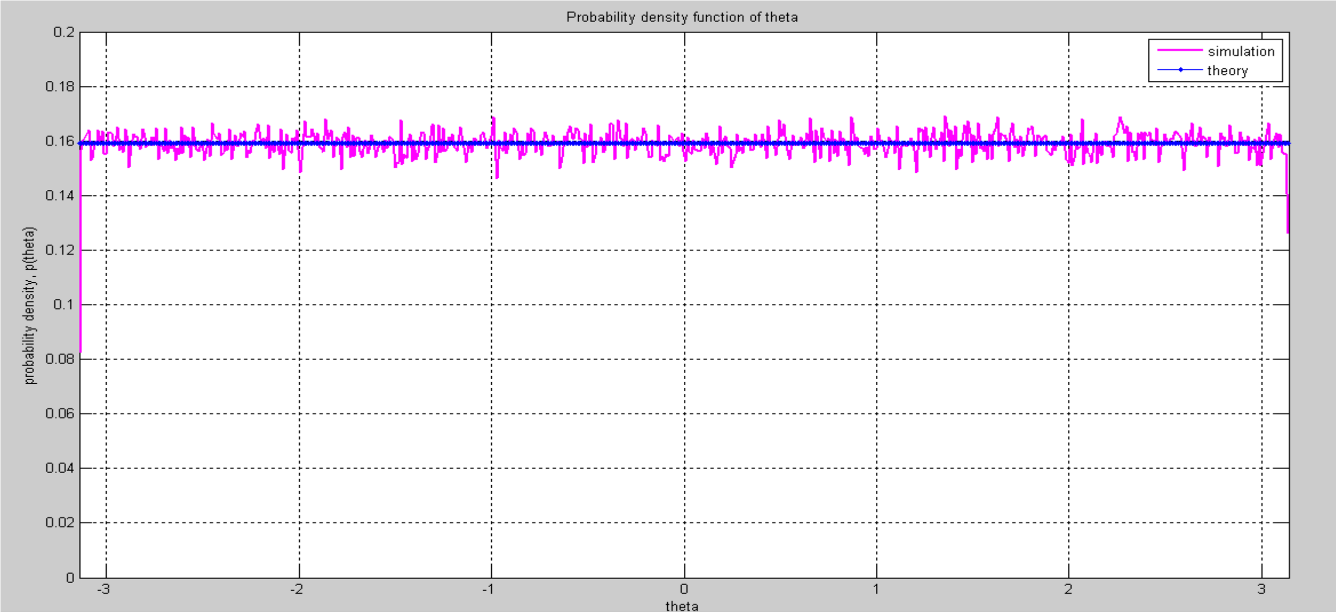


Figure 8: PDF of Theta

VI. CONCLUSION

Study of fading channel often requires simulation on PCs and is a necessary step in actual implementation. Our simulation in MATALAB presented a high level view of how Fading can affect systems performance. MATLAB's flexibility helped us in simulating the random behavior of the Reyliegh channel.

A Rayleigh distribution is often observed when the overall magnitude of a vector is related to its directional components. One example where the Rayleigh distribution naturally arises is when wind speed is analyzed into its orthogonal 2-dimensional vector components. We compared our simulated channel performance with the practical one and as shown in the figure generated by MATLAB we can see that we have got very good accuracy.

Further we proposed an Idea for improving channel estimation by applying Artificial Neural Network. The basic concept is to use the mobile user's habit of talking and using this information for training the ANN. The ANN can then be helpful in estimating channel parameters far quickly without sending the training sequence. This has several benefits including faster response and lesser bandwidth conjunctions.

REFERENCES

1. S. Theon, "Modeling the Channel Time-Variance for Fixed Wireless Communications", IEEE Communication Mag., vol.6, NO.8, August 2002, pp 331-333.
2. B. Skler, "Rayleigh Channels in Mobile Digital Communication Systems,"IEEE Communication Mag.,vol.29, no 4, July 1997, pp 90-100.
3. T. S. Rappaport, Wireless Communication, Chapters. 3 and 4,Upper Saddle River, NJ: Prentice Hall, 1996.
4. W.C. Jakes, microwave Mobile Communications, IEEE Press, 1994.
5. D. C. Cox, R. R. Murray,and A W. Norris, "Measurements of 800 MHz Radio Transmission into Buildings with Metallic Walls," Bell System Technical Journal,Vol. 62, pp. 2695~2717,November 1983.
6. D. C. Cox, R. R. Murray,and A W. Norris, "Measurements of 800 MHz Attenuation Measured In and Around Suburban Houses,"Bell System Technical Journal,Vol. 63, pp. 921~954,November 1984.
7. R. J. C. Bultitude and G. K. Bedal, "Propagation Characteristics on Microcellula Urban Mobile Radio Channels at 910 MHz," IEEE Journal on Selected Areas in Communications, Vol. 7, No. 1, pp. 31_39, January 1989.
8. T. S. Rappaport, "Characterization of UHF Multipath Radio Channels in Factory Buildings," IEEE Transactions on Antennas and Propagation, Vol. 37, No. 8, pp. 1058~1069, August 1989.
9. D. C. Cox, "Universal Digital Portable Radio Communications," IEEE Proceedings,Vol 75, No. 4, pp. 436~ 477, April 1987.
10. H. Hashemi, "Impulse Response Modeling of Indoor Radio Propagation Channels,"IEEE Journal on Selected Areas in Communications, Vol 11, No. 7, pp. 967_978, September 1993.