Sardar Patel College of Engineering, Bakrol

WIRELESS COMMUNICATION (3171608)

SARDAR PATEL COLLEGE OF ENGINEERING	
BAKROL, ANAND	
Academic Year: 2020-21	**Sem: VII**
Name of Teacher: Garima Pandey	**Name of Department: IT**
Subject : Wireless Communication (3171608)	**Hrs./Week: 2**
Theory/Practical: Practical	

LIST OF EXPERIMENTS

SR.NO	DATE	EXPIREMENT	SIGNATURE
1		To Briefly introduction about MATLAB.	
2		To determine the freespace loss and the power received using MATLAB program.	
3		To write a MATLAB program to calculate the link budget for satellite communication.	
4		To write a MATLAB program to calculate the median path loss for Hata model for outdoor propagation.	
5		1. To study the BER performance of DS-CDMA using mixed codes in multipath channel using RAKE receiver for single user case. 2. To study the BER performance of MRC combining and equal combining varying with SNR.	
6		To stimulate wireless channel including Rayleigh Fading using MATLAB	
7		To stimulate Orthogonal Frequency Division Multiplexing (OFDM) using MATLAB	
8		To study Gaussian Minimum Shift Keying (GMSK) modulation technique To design a receiver using Viterbi algorithm To study the BER using Viterbi	

EXPERIMENT: 1

INTRODUCTION TO MATLAB

Aim:

To Briefly introduction about MATLAB.

Theory:

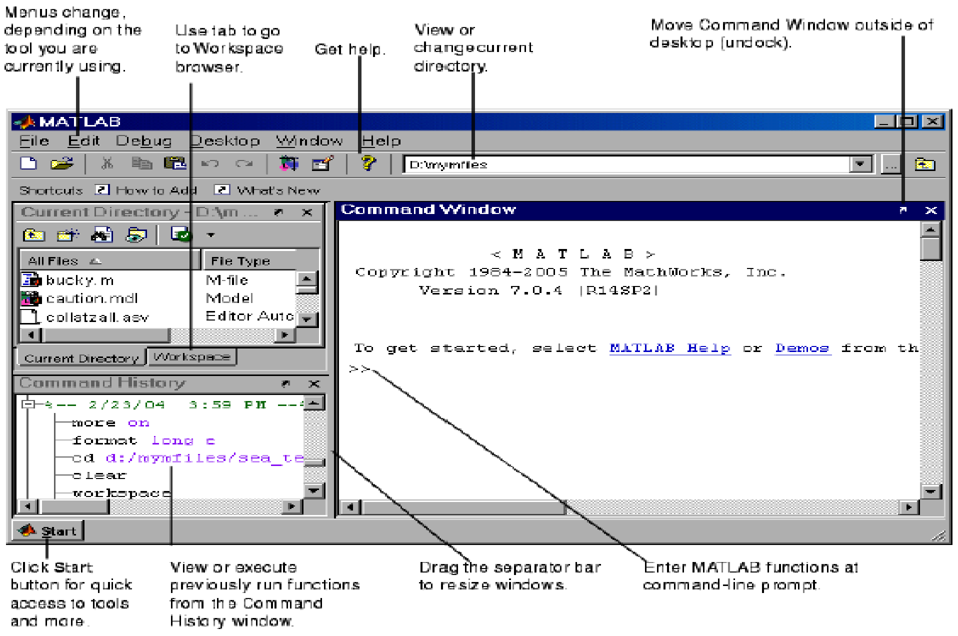
INTRODUCTION TO MATLAB:

- The name MATLAB stands for MATrix LAboratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects.
- MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research.
- MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide.
- It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific

  applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering.

- Starting MATLAB: you can enter MATLAB by double-clicking on the MATLAB shortcut icon on your Windows desktop. When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains other windows. The major tools within or accessible from the desktop are:
  - The Command Window
  - The Command History

- The Workspace
- The Current Directory
- The Help Browser
- The Start button



Menus change, depending on the tool you are currently using.

Use tab to go to Workspace browser.

Get help.

View or change current directory.

Move Command Window outside of desktop (undock).

Click Start button for quick access to tools and more.

View or execute previously run functions from the Command History window.

Drag the separator bar to resize windows.

Enter MATLAB functions at command-line prompt.

```
-- 2/23/04  3:59 PM --
more on
format long e
cd d:/mymfiles/sea_te
clear
workspace
```

```
                < M A T L A B >
    Copyright 1984-2005 The MathWorks, Inc.
           Version 7.0.4 (R14SP2)

 To get started, select MATLAB Help or Demos from th
>>
```

Fig:1 The graphical interface to the MATLAB workspace

When MATLAB is started for the first time, the screen looks like the one that shownin the Figure 1.1. This illustration also shows the default configuration of the MATLABdesktop. You can customize the arrangement of tools and documents to suit your needs.Now, we are interested in doing some simple calculations. We will assume that you have sufficient understanding of your computer under which MATLAB is being run.You are now faced with the MATLAB desktop on your computer, which contains the prompt (>>) in the Command Window. Usually, there are 2 types of prompt: >> for full version EDU> for educational version

- Quitting MATLAB: To end your MATLAB session, type quit in the Command Window, or select File Exit MATLAB in the desktop main menu.

- Creating MATLAB variables: MATLAB variables are created with an assignment statement. The syntax of variable assignment is

variable name = a value (or an expression) For example,

```
>> x = expression
```

Where expression is a combination of numerical values, mathematical operators, variables, and function calls. On other words, expression can involve:

1. manual entry

2. built-in functions

3. user-defined functions

 Overwriting variable: Once a variable has been created, it can be reassigned. In addition, if you do not wish tosee the intermediate results, you can suppress the numerical output by putting a semicolon(;) at the end of the line. Then the sequence of commands looks like this:

```
>> t = 5;

>> t = t+1

t =       6
```

- Error messages: If we enter an expression incorrectly, MATLAB will return an error message. For example, in the following, we left out the multiplication sign, *, in the following expression

```
>> x = 10;

>> 5x

??? 5x

|

Error: Unexpected MATLAB expression.
```

- Making corrections: To make corrections, we can, of course retype the expressions. But if the expression is lengthy, we make more mistakes by typing a second time. A previously typed command can be recalled with the up-arrow key". When the command is displayed at the command prompt, it can be modified if needed and executed.
- Controlling the appearance of floating point number: MATLAB by default displays only 4 decimals in the result of the calculations, for example 163:6667, as shown in above examples. However, MATLAB does numerical calculation in double precision, which is 15 digits. The command format controls how the results of computations are displayed. Here are some examples of the di®erent formats together withthe resulting outputs.

```
>> format short

>> x=-163.6667
```

If we want to see all 15 digits, we use the command format long

```
>> format long

>> x= -1.636666666666667e+002
```

To return to the standard format, enter format short, or simply format.There are several other formats. For more details, see the MATLAB documentation,or type help format.Note - Up to now, we have let MATLAB repeat everything that we enter at theprompt (>>). Sometimes this is not quite useful, in particular when the output is pages enlength. To prevent MATLAB from echoing what we type, simply enter a semicolon (;) atthe end of the command. For example,

```
>> x=-163.6667;
```

and then ask about the value of x by typing,

```
>> x
```

```
x =-163.6667
```

- Managing the workspace: The contents of the workspace persist between the executions of separate commands. Therefore, it is possible for the results of one problem to have an e®ect on the next one. To avoidthis possibility, it is a good idea to issue a clear command at the start of each new independent calculation.

```
>> clear
```

The command clear or clear all removes all variables from the workspace. Thisfrees up system memory. In order to display a list of the variables currently in the memory,type

```
>> who
```

while, whos will give more details which include size, space allocation, and class of thevariables.

- Keeping track of your work session: It is possible to keep track of everything done during a MATLAB session with the diary command.

```
>> diary
```

or give a name to a created file,

```
>> diary FileName
```

where FileName could be any arbitrary name you choose.The function diary is useful if you want to save a complete MATLAB session. Theysave all input and output as they appear in the MATLAB window. When you want to stopthe recording, enter diary off. If you want to start recording again, enter diary on. The file that

is created is a simple text ¯le. It can be opened by an editor or a word processingprogram and edited to remove extraneous material, or to add your comments. You canuse the function type to view the diary ¯le or you can edit in a text editor or print. Thiscommand is useful, for example in the process of preparing a homework or lab submission.

- Entering multiple statements per line:It is possible to enter multiple statements per line. Use commas (,) or semicolons (;) toenter more than one statement at once. Commas (,) allow multiple statements per linewithout suppressing output.

```
>> a=7; b=cos(a), c=cosh(a) b =0.6570

c =548.3170
```

 Miscellaneous commandsHere are few additional useful commands:

1. To clear the Command Window, type clc
2. To abort a MATLAB computation, type ctrl-c
3. To continue a line, type . . .

- Getting help: To view the online documentation, select MATLAB Helpfrom Help menu of MATLAB Helpdirectly in the Command Window. The preferred method is to use the Help Browser. TheHelp Browser can be started by selecting the ? icon from the desktop toolbar. On the otherhand, information about any command is available by typing

```
>> help Command
```

Another way to get help is to use the lookfor command. The lookfor command di®ersfrom the help command. The help command searches for an exact function name match,while the lookfor command searches the quick summary information in each function fora match. For example, suppose that we were looking for a function to take the inverse ofa matrix. Since MATLAB does not have a function named inverse, the command helpinverse will produce nothing. On the other hand, the command lookfor inverse willproduce detailed information, which includes the function of interest, inv.

```
>> lookfor inverse
```

Note - At this particular time of our study, it is important to emphasize one main point.Because MATLAB is a huge program; it is impossible to cover all the details of each functionone by one. However, we will give you information how to get help. Here are some examples:

- Use on-line help to request info on a specific function

```
>> help sqrt
```

- In the current version (MATLAB version 7), the doc function opens the on-line versionof the help manual. This is very helpful for more complex commands.

```
>> doc plot
```

- Use look for to find functions by keywords. The general form is

```
>> lookfor FunctionName
```

- Programming in MATLAB: So far in these lab sessions, all the commands were executed in the Command Window.The problem is that the commands entered in the Command Window cannot be savedand executed again for several times. Therefore, a different way of executing repeatedlycommands with MATLAB is:

1. to create a file with a list of commands,
2. save the file, and
3. run the file.

If needed, corrections or changes can be made to the commands in the file. The files thatare used for this purpose are called script files or scripts for short.

This section covers the following topics:

- M-File Scripts: A script file is an external file that contains a sequence of MATLAB statements. Scriptfiles have a filename extension .m and are often called M-files. M-files can be scripts thatsimply execute a series of MATLAB statements, or they can be functions that can acceptarguments and can produce one or more outputs.

Example

Consider the system of equations:

$x + 2y + 3z = 1$ $3x + 3y + 4z = 1$ $2x + 3y + 3z = 2$

Find the solution x to the system of equations.

Solution:

Use the MATLAB editor to create a file: File New M-file. Enter the following statements in the file:

```
A = [1 2 3; 3 3 4; 2 3 3];

b = [1; 1; 2];

x = A\b Save the file, for example, example1.m.
```

Run the file, in the command line, by typing:

```
>> example1 x =

-0.5000

1.5000

-0.5000
```

When execution completes, the variables (A, b, and x) remain in the workspace. To see a listing of them, enter whos at the command prompt.

Note: The MATLAB editor is both a text editor specialized for creating M-files and agraphical MATLAB debugger. The MATLAB editor has numerous menus for tasks such assaving, viewing, and debugging. Because it performs some simple checks and also uses color to differentiate between various elements of codes, this text editor is recommended as thetool of choice for writing and editing M-files.

There is another way to open the editor:

or

to open filename.m.

```
>> edit
```

```
>> edit filename.m
```

- M-File Functions: As mentioned earlier, functions are programs (or routines) that accept input arguments andreturn output arguments. Each M-file function (or function or M-file for short) has its ownarea of workspace, separated from the MATLAB base workspace.
- Anatomy of a M-File function

This simple function shows the basic parts of an M-file.

```
function f = factorial(n)  (1)

% FACTORIAL(N) returns the factorial of N.        (2)

% Compute a factorial value      (3)

f = prod(1:n);     (4)
```

The first line of a function M-file starts with the keyword function. It gives the functionname and order of arguments. In the case of function factorial, there are up to one outputargument and one input argument. Table given below summarizes the M-file function.

As an example, for n = 5, the result is,

```
>> f = factorial(5) f =120
```

Table: Anatomy of a M-File function

Part No	M-File Element	Description
1	Function Definition Line	Define the function name, and the definition number and order of input and line output arguments
2	H1 Line	A one line summary description of the program, displayed when you request Help
3	Help Text	Help text A more detailed description of the program
4	Function body	Function body Program code that performs the actual computations

Both functions and scripts can have all of these parts, except for the function definition line which applies to function only. In addition, it is important to note that function name must begin with a letter, andmust be no longer than the maximum of 63 characters. Furthermore, the name of thetextfile that you save will consist of the function name with the extension .m. Thus, theabove example file would be factorial.m.

Table: Difference between scripts and function

SCRIPTS	FUNCTION
• Do not accept input arguments. • Store variables in a workspace that is shared with other scripts. • Are useful for automating a series of commands	• Can accept input arguments and return output arguments. • Store variables in workspace that is internal to the function • Are useful for extending the MATLAB a series of commands language for your application

Enrollment no :-

EXPERIMENT: 2

FREESPACE PROPAGATION – PATH LOSS MODEL

Aim:

To determine the freespace loss and the power received using MATLAB program.

Theory:

The free space path loss, also known as FSPL is the loss in signal strength that occurs when an electromagnetic wave travels over a line-of-sight path in free space. In these circumstances there are no obstacles that might cause the signal to be reflected refracted, or that might cause additional attenuation.

The free space path loss calculations only look at the loss of the path itself and do not contain any factors relating to the transmitter power, antenna gains or the receiver sensitivity levels.

To understand the reasons for the free space path loss, it is possible to imagine a signal spreading out from a transmitter. It will move away from the source spreading out in the form of a sphere. As it does so, the surface area of the sphere increases. As this will follow the law of the conservation of energy, as the surface area of the sphere increases, so the intensity of the signal must decrease.

As a result of this it is found that the signal decreases in a way that is inversely proportional to the square of the distance from the source of the radio signal.

Free Space Path Loss Formula

The free space path loss formula or free space path loss equation is quite simple to use. Not only is the path loss proportional to the square of the distance between the transmitter and receiver, but the signal level is also proportional to the square of the frequency in use.

$$\mathbf{FSPL = (4\pi d/ \lambda)2 = (4\pi df/ c)2}$$

FSPL is the Free space path loss

d is the distance of the receiver from the transmitter (metres)

λ is the signal wavelength (metres)

f is the signal frequency (Hertz)

c is the speed of light in a vacuum (metres per second)

The free space path loss formula is applicable to situations where only the electromagnetic wave is present, i.e., for far field situations. It does not hold true for near field situations.

Decibel version of free space path loss equation

Most RF comparisons and measurements are performed in decibels. This gives an easy and consistent method to compare the signal levels present at various points. Accordingly, it is very convenient to express the free space path loss formula, FSPL, in terms of decibels.

$$\textbf{FSPL (dB)} = 20 \log10 (d) + 20 \log10 (f) + 32.44$$

Where:

d is the distance of the receiver from the transmitter (km)

f is the signal frequency (MHz)

Effect of antenna gain on path loss equation

The equation above does not include any component for antenna gains. It is assumed that the antenna gain is unity for both the transmitter. In reality, though, all antennas will have a certain amount of gain and this will affect the overall affect. Any antenna gain will reduce the "loss" when compared to a unity gain system. The figures for antenna gain are relative to an isotropic source, i.e. an antenna that radiates equally in all directions.

$$\textbf{FSPL (dB)} = 20 \log10 (d) + 20 \log10 (f) + 32.44 - Gtx - Grx$$

Where:

Gtx is the gain of the transmitter antenna relative to an isotropic source (dBi)

Grx is the gain of the receiver antenna relative to an isotropic source (dBi)

The free space path loss equation or formula given above, is an essential tool that is required when making calculations for radio and wireless systems either manually or within applications such as wireless survey tools, etc. By using the free space path loss equation, it is possible to determine the signal strengths that may be expected in many scenarios. While the free space path loss formula is not fully applicable where there are other interactions, e.g. reflection, refraction, etc as are present in most real life applications, the equation can nevertheless be used to give an indication of what may be expected. It is obviously fully applicable to satellite systems where the paths conform closely to the totally free space scenarios.

Power Received:

[Pr] = [pt] + [Gt] + [Gr] – [FSPL]

Pr – Received power

Pt – Transmitted power

Gt – Gain of the transmitting antenna

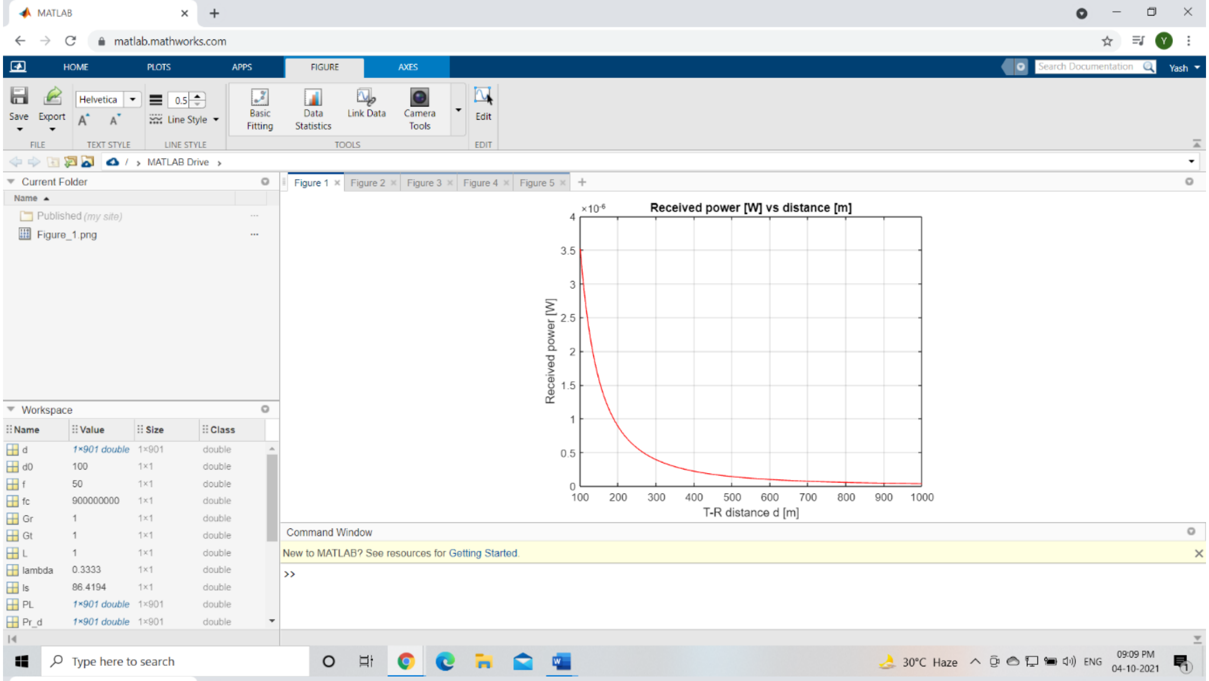
Gr – Gain of the receiving antenna

Program:

```
%% Free Space Propagation Model
%% Received power, based on ex. 3.2 from Wireless Communication Principle and
Practice, S.Rappaport
% Friis space equation, dBm, dBW
fc = 900*10^6; % Carrier frequency [Hz]
lambda = 3*10^8/fc; % Wavelength [m]
Pt = 50; % Transmitter power [W]
Gt = 1; % Transmitter gain (unity gain)
Gr = 1; % Receiver gain (unity gain)
L = 1; % System losses (lossless)
d0 = 100; % Close-in distance [m]
d = 100:10^3; % Operation distance up to 1 kilometer [m]
% Using Friis equation we can find P received at close-in distance d0 and use it as a
% reference for any d > d0
Pr_d0 = Pt*Gt*Gr*lambda^2/(16*pi^2*d0^2*L); % Friis space equation for received power
at close-in (reference) distance [W]
Pr_d0_dBm = 10*log10(Pr_d0/0.001);          % Close-in received power in dBm
Pr_d0_dBW = 10*log10(Pr_d0);                % Close-in received power in dBW
Pr_d = Pt*Gt*Gr*lambda^2./(d.^2*16*pi^2*L); % Friis space equation for received power
along the operation distance [W]
Pr_d_dBm = Pr_d0_dBm+20*log10(d0./d);       % Received power calculated using
reference power in dBm
Pr_d_dBW = Pr_d0_dBW+20*log10(d0./d);       % Received power calculated using
reference power in dBW
PL = -10*log(Gr*Gt*lambda^2./(16*pi^2*d.^2)); % Path loss
figure(1) % Received power in Watts
plot(d,Pr_d,'r')
grid on
title('Received power [W] vs distance [m]')
xlabel('T-R distance d [m]')
ylabel('Received power [W]')
figure(2) % Received power in dBm
plot(d,Pr_d_dBm,'g')
grid on
```

```
title('Received power [dBm] vs distance [m]')
xlabel('T-R distance d [m]')
ylabel('Received power [dBm]')
figure(3) % Received power in dBW
plot(d,Pr_d_dBW,'b')
grid on
title('Received power [dBW] vs distance [m]')
xlabel('T-R distance d [m]')
ylabel('Received power [dBW]')
figure(4) % Path loss
plot(d,PL)
grid on
title('Path loss vs distance')
xlabel('T-R distande d [m]')
ylabel('Path loss')
figure(5) % Here we see that the power in Watts is near 0 compared to visible arc of
dB description
plot(d,Pr_d,'r')
grid on
title('Received power [W] vs distance [m]')
xlabel('T-R distance d [m]')
ylabel('Received power [W]')
hold on
plot(d,Pr_d_dBm,'g')
grid on
title('Received power [dBm] vs distance [m]')
xlabel('T-R distance d [m]')
ylabel('Received power [dBm]')
hold on
plot(d,Pr_d_dBW,'b')
grid on
title('Received power [dBW] vs distance [m]')
xlabel('T-R distance d [m]')
ylabel('Received power [dBW]')
```

Output:

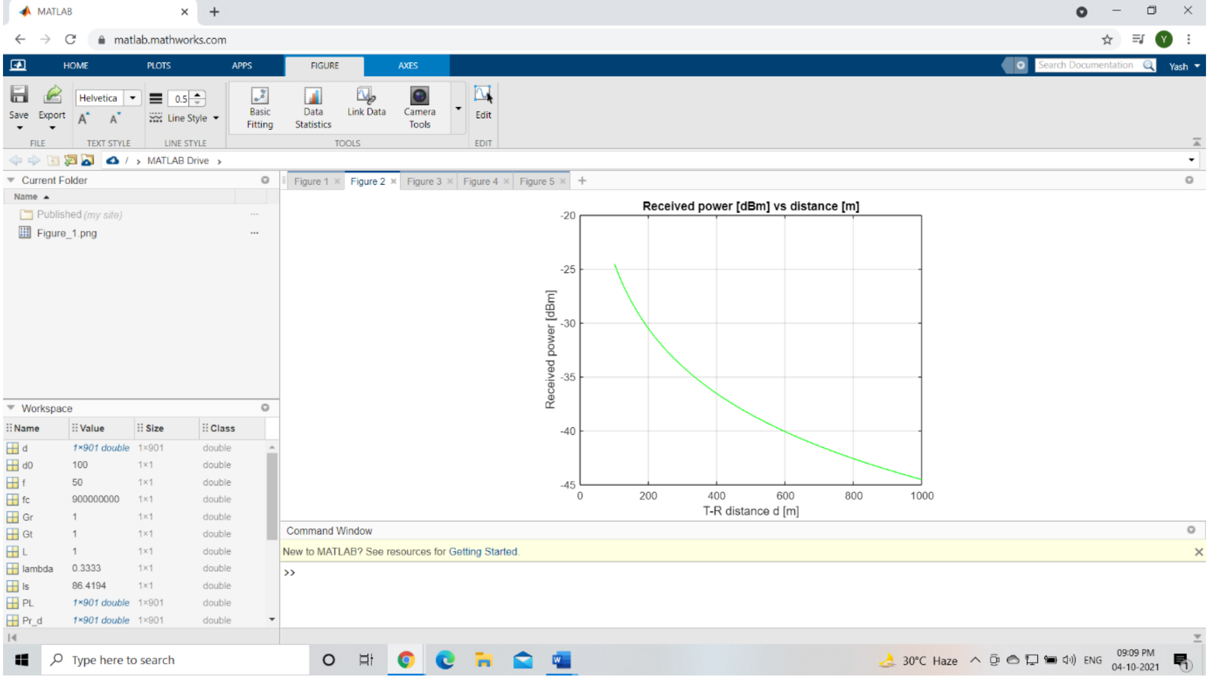


Workspace

Name	Value	Size	Class
d	1×901 double	1×901	double
d0	100	1×1	double
f	50	1×1	double
fc	900000000	1×1	double
Gr	1	1×1	double
Gt	1	1×1	double
L	1	1×1	double
lambda	0.3333	1×1	double
ls	86.4194	1×1	double
PL	1×901 double	1×901	double
Pr_d	1×901 double	1×901	double

Command Window

```
New to MATLAB? See resources for Getting Started.
>>
```

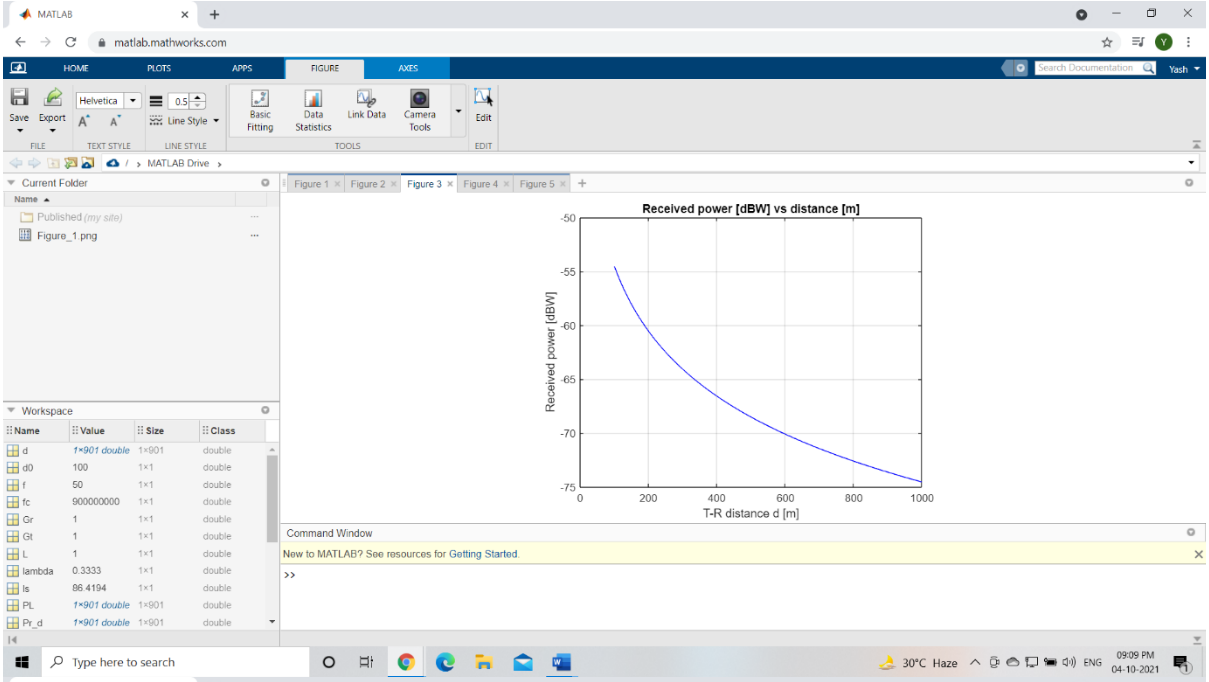


Workspace

Name	Value	Size	Class
d	1×901 double	1×901	double
d0	100	1×1	double
f	50	1×1	double
fc	900000000	1×1	double
Gr	1	1×1	double
Gt	1	1×1	double
L	1	1×1	double
lambda	0.3333	1×1	double
ls	86.4194	1×1	double
PL	1×901 double	1×901	double
Pr_d	1×901 double	1×901	double

Command Window

```
New to MATLAB? See resources for Getting Started.
>>
```



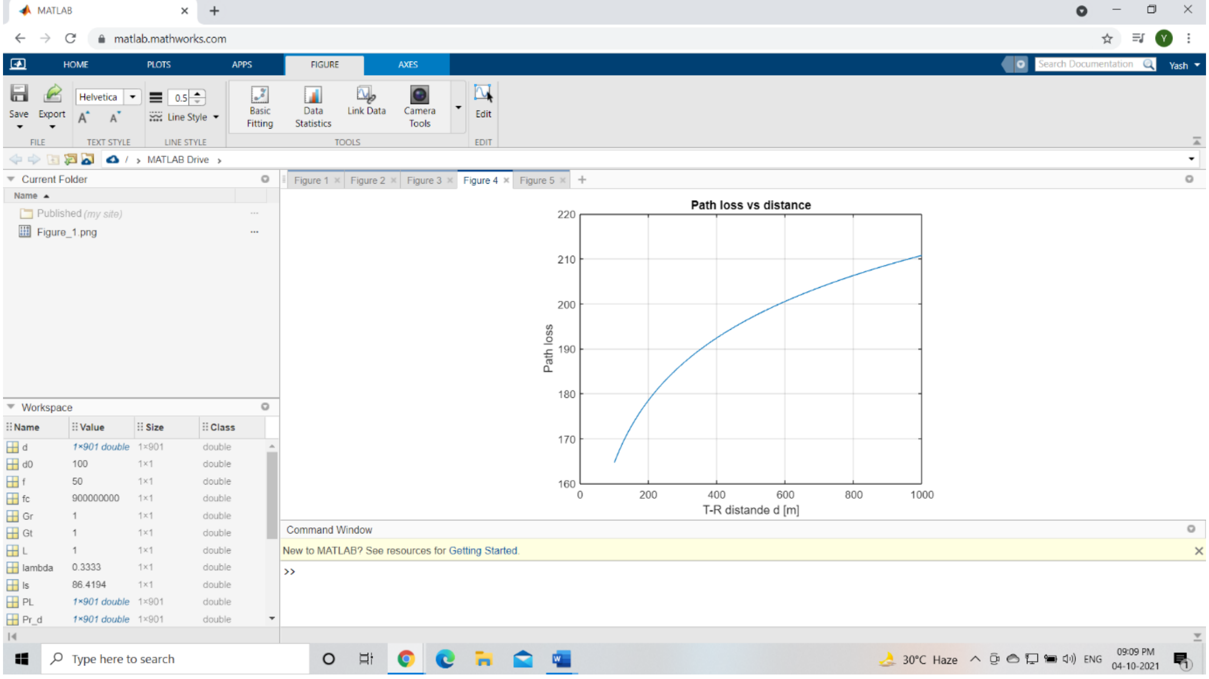
Workspace

Name	Value	Size	Class
d	1×901 double	1×901	double
d0	100	1×1	double
f	50	1×1	double
fc	900000000	1×1	double
Gr	1	1×1	double
Gt	1	1×1	double
L	1	1×1	double
lambda	0.3333	1×1	double
ls	86.4194	1×1	double
PL	1×901 double	1×901	double
Pr_d	1×901 double	1×901	double

Command Window

New to MATLAB? See resources for Getting Started.

```
>>
```



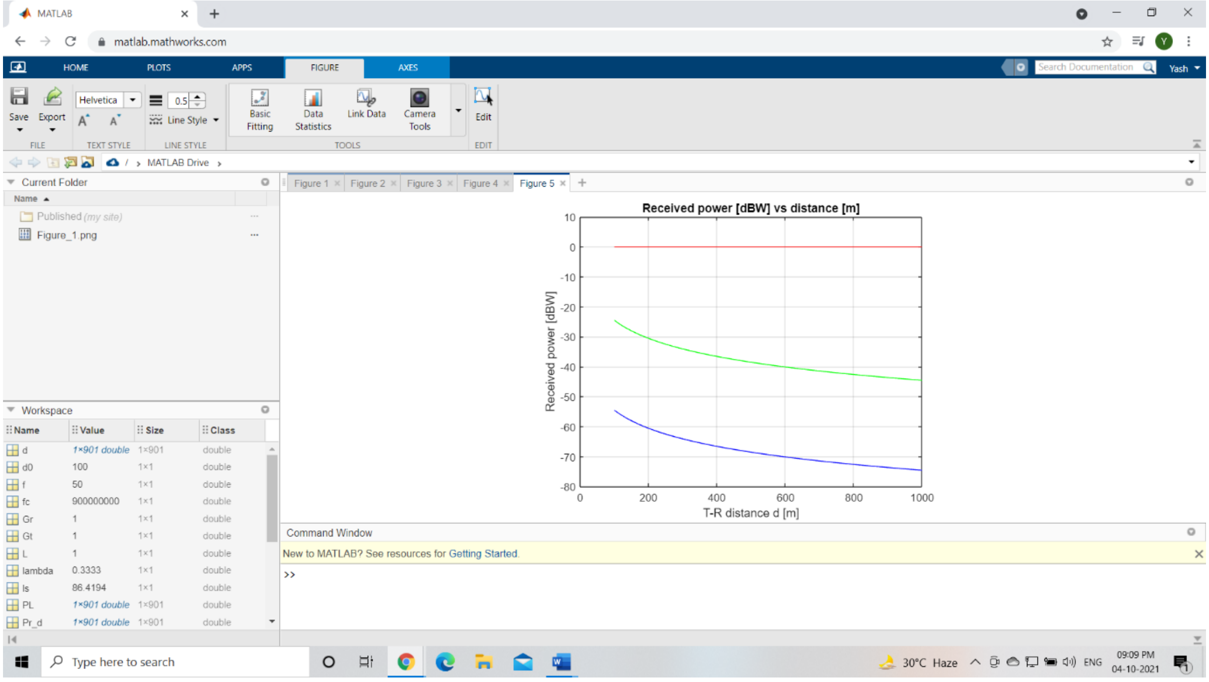
Workspace

Name	Value	Size	Class
d	1×901 double	1×901	double
d0	100	1×1	double
f	50	1×1	double
fc	900000000	1×1	double
Gr	1	1×1	double
Gt	1	1×1	double
L	1	1×1	double
lambda	0.3333	1×1	double
ls	86.4194	1×1	double
PL	1×901 double	1×901	double
Pr_d	1×901 double	1×901	double

Command Window

New to MATLAB? See resources for Getting Started.

```
>>
```



Workspace

Name	Value	Size	Class
d	1×901 double	1×901	double
d0	100	1×1	double
f	50	1×1	double
fc	900000000	1×1	double
Gr	1	1×1	double
Gt	1	1×1	double
L	1	1×1	double
lambda	0.3333	1×1	double
ls	86.4194	1×1	double
PL	1×901 double	1×901	double
Pr_d	1×901 double	1×901	double

Command Window

New to MATLAB? See resources for Getting Started.

```
>>
```

Result:

The program for power received by an antenna and path loss in Free space propagation was simulated successfully.

Enrollment no :-

EXPERIMENT: 3

LINK BUDGET EQUATION- SATELLITE COMMUNICATION

Aim:

To write a MATLAB program to calculate the link budget for satellite communication.

Theory:

A link budget is an accounting of all the gains and losses in a transmission system. The link budget looks at the elements that will determine the signal strength arriving at the receiver. The link budget may include the following items:

- Transmitter power.
- Antenna gains (receiver and transmitter).
- Antenna feeder losses (receiver and transmitter).
- Path losses.
- Receiver sensitivity (although this is not part of the actual link budget, it is necessary to know this to enable any pass fail criteria to be applied.)

Where the losses may vary with time, e.g., fading, and allowance must be made within the link budget for this - often the worst case may be taken, or alternatively an acceptance of periods of increased bit error rate (for digital signals) or degraded signal to noise ratio for analogue systems.

Received power (dBm) = Transmitted power (dBm) + gains (db) - losses (dB)

The basic calculation to determine the link budget is quite straightforward. It is mainly a matter of accounting for all the different losses and gains between the transmitter and the receiver.

Losses = FSL + AML + RFL + PL + AA

FSL = Freespace loss

AML = Antenna Misalignment loss

RFL=Receiver Feeder loss

PL=Polarization Loss

AA = Atmospheric Absorption.

Carrier to Noise Ratio – Uplink

CNRu=EIRPu+GTRu-Lossu+228.6

Carrier to Noise Ratio – Uplink

CNRd=EIRPd+GTR-Lossd+228.6

Overall Carrier to Noise Ratio

CNRoverall=CNRu X CNRd / (CNRu+CNRd)

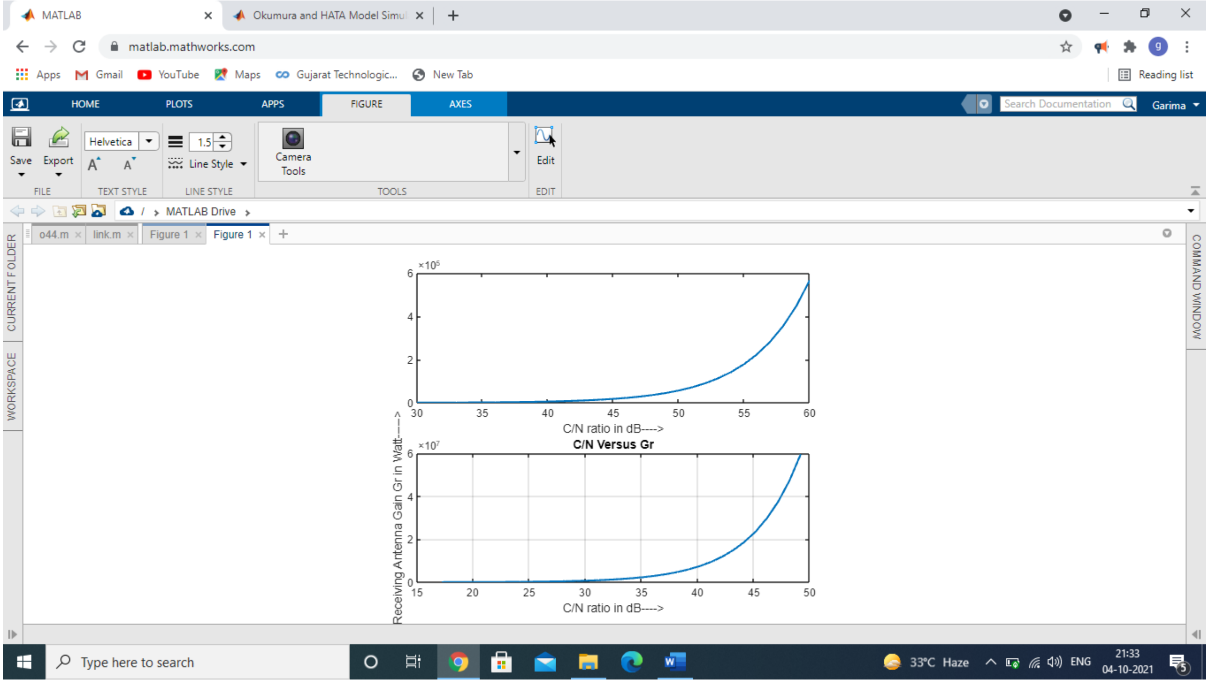
Program:

```
clc;
close all;
clear all;
% Boltzman Constant K = -228dBW/K/hz
% Satellite at 40,000Km distance from Earth station
% Satellite Antenna Gain = 31dB
% Receiver System Noise Temp = 500K
% Transponder saturated output Power = 80W
% Earth Station Antenna Diameter = 5m
% Earth Station Aperture Efficiency = 68%
% Uplink Frequency = 14.15Ghz
% Required C/N in transponder = 30dB
% Transponder HPA Output Backoff = 1 dB
% Location: -2dB contour of satellite receiving antenna
% Downlink Frequency = 11.45Ghz
% Receiver IF Noise BW = 43.2Mhz
% Antenna Noise Temp = 30K
% LNA Noise Temp = 110K
% Required Overall (C/N)o in clear air = 17dB
CNup=30:1:60;    %C/N required at transponder is 30dB
ln=length(CNup);
% Uplink Noise Power Budget
k=-228.6;  % Boltzman Constant
T1=500;
Ts1=10*log10(T1);     % noise temp 500K in dBK
B1=10*log10(43.2*10^6);;    % noise BW 43.2 Mhz in dBhz
```

```
N1=k+Ts1+B1;  %in dBW   % Transponder Noise power N=kTsB
Pr1=N1+CNup;   % received power must be 30dB graeter than noise power
% Uplink Power Budget
R=4*10^7;
D=5;       % Antenna Diameter
Ae=0.68;   % Aperture Efficiency
lmb=0.0212; % operating wavelength
Gt1=10*log10(Ae*(pi*D/lmb)^2);  % Earth Satation Antenna gain
Lp1=10*log10((4*pi*R/lmb)^2);   % pathlosss in dB
Gr1=31;     % Satellite Antenna Gain in dB
Lp1=-Lp1;    % pathloss is negative
Lant=-2;    % due to 2dB contour
% Pr=Pt+Gt+Gr+Lp+Lant; As all in dB so simply addition
Pt1=Pr1-(Gt1+Gr1+Lp1+Lant);
for i=1:ln
Ptw(i)=10^(Pt1(i)/10);
end
subplot(2,1,1);
plot(CNup,Ptw,'LineWidth',1.5);
xlabel('C/N ratio in dB---->');
ylabel('Transmitted Power Pt in Watt----->');
title('C/N Versus Pt');
grid on;
%DOWNLINK DESIGN
CNdwn=17.2:1:50; % C/N downlink is 17.2 dB as C/N air is 17dB
lnn=length(CNdwn);
%Downlink noise Power Budget
k=-228.6;
T2=30+110; % noise temp
Ts2=10*log10(T2); % Noise Temp in dB
B2=10*log10(43.2*10^6); % Noise BW in dB
N2=k+Ts2+B2;
Pr2=N2+CNdwn;  %power at earthstation receiver input
Lp2=207.2-20*log10(14.15/11.45);
Pt2=10*log10(80)-1; % Output power is 1dB below saturated power 80W
%Downlink Power Budget
Gt2=31;
Lp2=-Lp2;
La=-3;
% Pr=(Pt*Gt*Gr)/(Lp*La)  in Watt
% Pr=Pt+Gt+Gr-Lp-La      in dB
Gr2=Pr2-(Pt2+Gt2+Lp2+La);
for i=1:lnn
Grw(i)=10^(Gr2(i)/10);
end
subplot(2,1,2);
plot(CNdwn,Grw,'LineWidth',1.5);
xlabel('C/N ratio in dB---->');
ylabel('Receiving Antenna Gain Gr in Watt----->');
```

```
title('C/N Versus Gr');
grid on;
```

Output:



Result:

The MATLAB program for calculating the Link budget was stimulated successfully.

Enrollment no :-

EXPERIMENT: 4

OUTDOOR PROPAGATION MODEL- HATA MODEL

Aim:

To write a MATLAB program to calculate the median path loss for Hata model for outdoor propagation.

Theory:

In wireless communication, the Hata Model for Urban Areas, also known as the Okumura-Hata model for being a developed version of the Okumura Model, is the most widely used radio frequency propagation model for predicting the behaviour of cellular transmissions in built up areas. This model incorporates the graphical information from Okumura model and develops it further to realize the effects of diffraction, reflection and scattering caused by city structures. This model also has two more varieties for transmission in Suburban Areas and Open Areas. Hata Model predicts the total path loss along a link of terrestrial microwave or other type of cellular communications.

This particular version of the Hata model is applicable to the radio propagation within urban areas.

This model is suited for both point-to-point and broadcast transmissions and it is based on extensive empirical measurements taken.

PCS is another extension of the Hata model. The Walfisch and Bertoni Model is further advanced.

Coverage

Frequency: 150 MHz to 1500 MHz

Mobile Station Antenna Height: between 1 m and 10 m

Base station Antenna Height: between 30 m and 200 m

Link distance: between 1 km and 20 km.

Mathematical formulation

Hata Model for Urban Areas is formulated as:

LU = 69.55 + 26.16 log f – 13.82 log hB – CH + [44.9 – 6.55 log hB] log d.

For small or medium sized city,

CH = 0.8 + (1.1 log f – 0.7) hM – 1.56 log f.

and for large cities,

CH = 8.29 (log (1.54 hM))2– 1.1, if 150 ≤ f ≤ 200

CH = 3.2 (log (11.75 hM))2 – 4.97, if 200 ≤ f ≤ 1500

Where,

LU = Path loss in Urban Areas (dB)

hB= Height of base station Antenna. (m)

hM = Height of mobile station Antenna. (m)

f= Frequency of Transmission (MHz).

CH = Antenna height correction factor

d= Distance between the base and mobile stations (km).

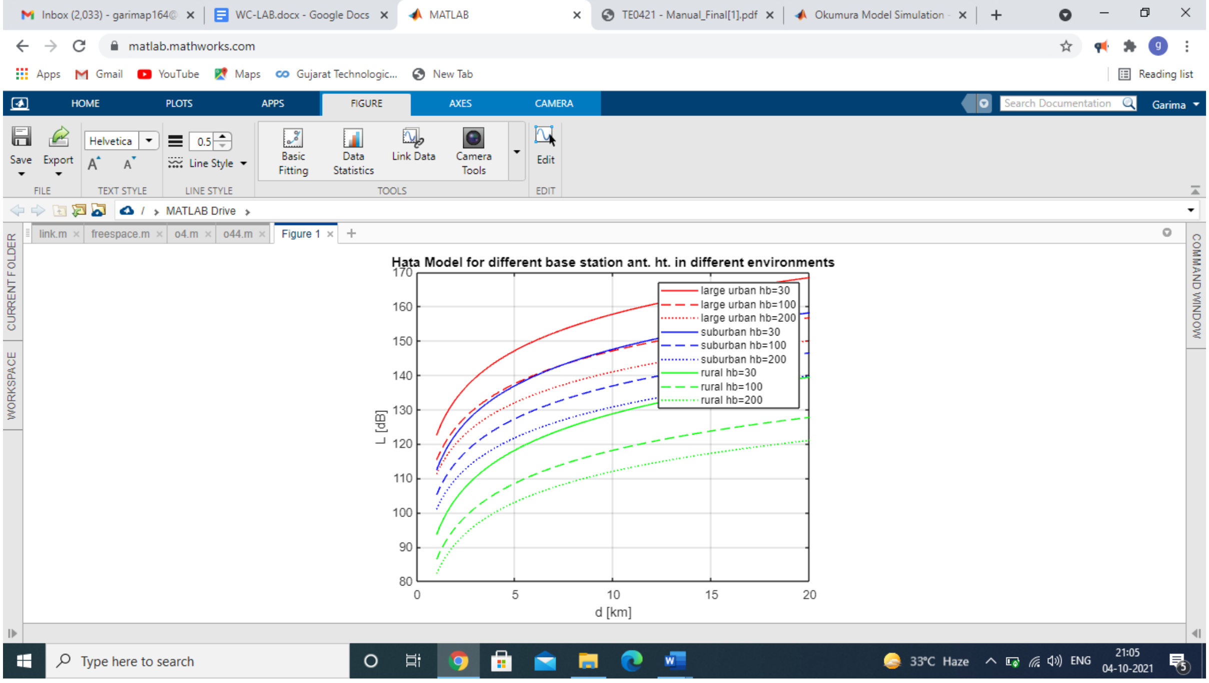
The term "small city" means a city where the mobile antenna height not more than 10 meters. i.e.

1 ≤ hM ≤ 10m

Program:

```
clc
close all;
clear
d = 1:0.01:20;
hm = 5;
hb1 = 30;
hb2 = 100;
hb3= 200;
fc = 1000;
%a. For Large Cities.
% fc >= 400MHz
ahm=3.2(log10(11.75 hm)).^2 - 4.97;
% A. Typical Urban
L50urban1= 69.55+ 26.16 log10 (fc) + (44.9 6.55 log10 (hb1))*log10 (d) 13.82
log10 (hb1) - ahm;
L50urban2= 69.55+ 26.16 log10 (fc) + (44.9 6.55 log10 (hb2)) log10 (d) 13.82
log10 (hb2) - ahm;
L50urban3 = 69.55 + 26.16 log10 (fc) + (44.9 6.55 log10 (hb3))*log10 (d)
13.82 log10 (hb3) - ahm;
% B. Typical Suburban
L50suburban1= L50urban1 2(log10 (fc/28)).^2 - 5.4;
L50suburban2 =L50urban2 2* (log10 (fc/28)).^2 - 5.4;
L50suburban3 =L50urban32(log10 (fc/28)).^2 - 5.4;
% C. Typical Rural
L50rural1 = L50urban14.78 (log10 (fc)).^2 + 18.33 log10 (fc) - 40.94;
L50rural2 = 15eurban2-4.78* (log10 (fc)).^2 + 18.33 log10(fc) - 40.94;
L50rural3 = L50urban3 4.78* (log10 (fc)).^2 + 18.33*log10 (fc) - 40.94;
figure(1);
plot(d, L5eurban1, 'r', d, L50urban2, -r', d, L50urban3, ':r');
hold on;
plot(d, 150suburban1, 'b', d, L50suburban2, '-b', d, L50suburban3, ':b');
hold on;
plot(d, L50rurali, 'g', d, L50rural2,-g', d, L50rural3, g);
hold on;
legend ('large urban hb-38', 'large urban hb-100', 'large urban hb-200',
'suburban hb-30', 'suburban hb-100', 'suburban hb-200', 'rural hb=30',
'rural hb=100','rural hb=200');
grid on;
xlabel('d [km]');
ylabel('L [dB]');
title('Hata Model for different base station ant. ht. in different
environments');
```

Output:



Result:

The program for Hata Model – Outdoor Propagation was simulated successfully.

Enrollment no :-

EXPERIMENT: 5

CODE DIVISION MULTIPLE ACCESS - MULTIPATH

Aim

1. To study the BER performance of DS-CDMA using mixed codes in multipath channel using RAKE receiver for single user case.

2. To study the BER performance of MRC combining and equal combining varying with SNR.

Theory

1. What is Multipath?

2. Effect of Multipath on the performance of CDMA

3. What is RAKE Receiver?

4. Maximum Ratio Combining Technique

5. Equal Gain Combining Technique

Description

What is Multipath?

Multipath occurs when a radio signal is split into two or more signals causing the receiving antenna to receive multiple copies of the same signal. The radio signal can be split by obstacles such as walls, chairs, tables and other objects. As the signal bounces off an object it causes a longer path to the receiver. Some signals may bounce off several objects before reaching the receiver. The longer the path, the greater the amount of delay. As radio signals are delayed, they reach the receiving antenna at different times sometimes overlapping. The receiver becomes confused by the signals and is unable to interpret them correctly which causes data errors requiring retransmission of the signal. Performance can be significantly reduced by the delayed signals and retransmissions.

Effect of Multipath on the performance of DS-CDMA

CDMA is inherently tolerant to multipath delay spreading signals as any signal that is delayed by more than one chip time becomes uncorrelated to the PN code used to decode the signal. This results in the multipath simply appearing as noise. This noise leads to an increase in the amount of interference seen by each user subjected to the multipath and thus increases the received BER. The BER is essentially flat for delay spreadings of greater than one chip time (0.8 ms), which is to be expected as the reflected signal becomes uncorrelated. Also the multipath delay spreading leads to an increase in the equivalent number of users in the cell, as it increases the amount of interference seen by the receiver.

RAKE Receiver

A RAKE receiver is a radio receiver designed to nullify the effect of multipath fading. It uses number of sub-receivers called fingers. Each finger is a correlator and is designed to a different multipath component. Each finger independently decodes a single multipath component. The output of all the correlators is combined to increase the SNR in a multipath environment. The multipath channel through which a radio wave transmits can be viewed as transmitting the line of sight wave plus a number of multipath components. Multipath components are delayed copies of the original transmitted wave traveling through a different echo path, each with a different magnitude and time of arrival at the receiver. Since each component contains the original information, if the magnitude and phase of each component is computed at the receiver through a method called channel estimation then all the components can be added coherently to improve the information reliability. The RAKE receiver is so named because it looks like a garden rake, each finger collecting the symbol energy similar to how the fingers in a garden rake collects leaves. To minimize the distortions introduced in the DS-CDMA systems, RAKE receiver uses a technique called diversity.

RAKE Receiver

In our case, RAKE receiver has 2 fingers. Each finger of the receiver process one path of the composite multipath signal. All the processing in the RAKE fingers should be done at chip level. Here c(k)indicates the spreading code used for that particular user. h0 and hL are the multipath channel coefficients. LTc is the delay that is used in the multipath channel model.

Maximum Ratio Combining (MRC)

Here the signal from different paths is weighted according to their individual signal voltage to noise power ratios and then summed. Hence maximum ratio combining produces an output SNR equal to the sum of the individual SNRs which is the acceptable SNR even when none of the individual signals are themselves acceptable. In MRC, the receiver corrects the phase rotation caused by a fading channel and then combines the received signals of different paths proportionally to the strength of each path. Since each path undergoes different attenuations, combining them with different weights yield an optimum solution under an AWGN channel.

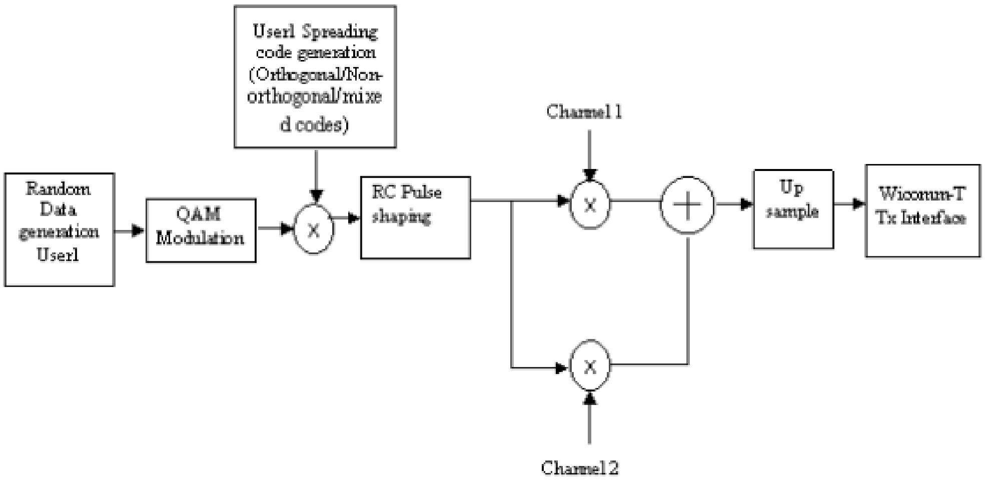
MRC is the optimum way (in the sense of the least BER) to use information from different paths to achieve decoding in an additive white Gaussian channel (AWGN)

Equal Gain Combining (EGC)

In certain cases, it is not convenient to provide for the variable weighting capability required for true maximal ratio combining. In such cases, the branch weights are all set to unity, but the signals from each branch are co-phased to provide equal gain combining diversity. This allows the receiver to exploit signals that are simultaneously received on each branch. The possibility of producing an acceptable signal from a number of unacceptable inputs is still retained, and performance is only marginally inferior to MRC The receiver corrects the phase rotation of the received signals caused by the fading channel and combines the received signals of different paths with equal weight. In a Rayleigh fading channel, the MRC performance is the best, followed by EGC. The performance of MRC is the same as that of EGC if signals from each path are of equal strength.

MATLAB Code Implementation

Transmitter


User1 Spreading code generation (Orthogonal/Non-orthogonal/mixe d codes)
Channel 1
Random Data generation User1
QAM Modulation
RC Pulse shaping
Up sample
Wicomm-T Tx Interface
Channel 2

Block diagram of CDMA- Multipath Transmitter

1. Random data to be transmitted for User1 is generated.

2. Random data of User1 is QAM modulated.

3. The QAM modulated User1 data is convolved with its spreading code.

4. The convolved data of User1 is RC Pulse shaped.

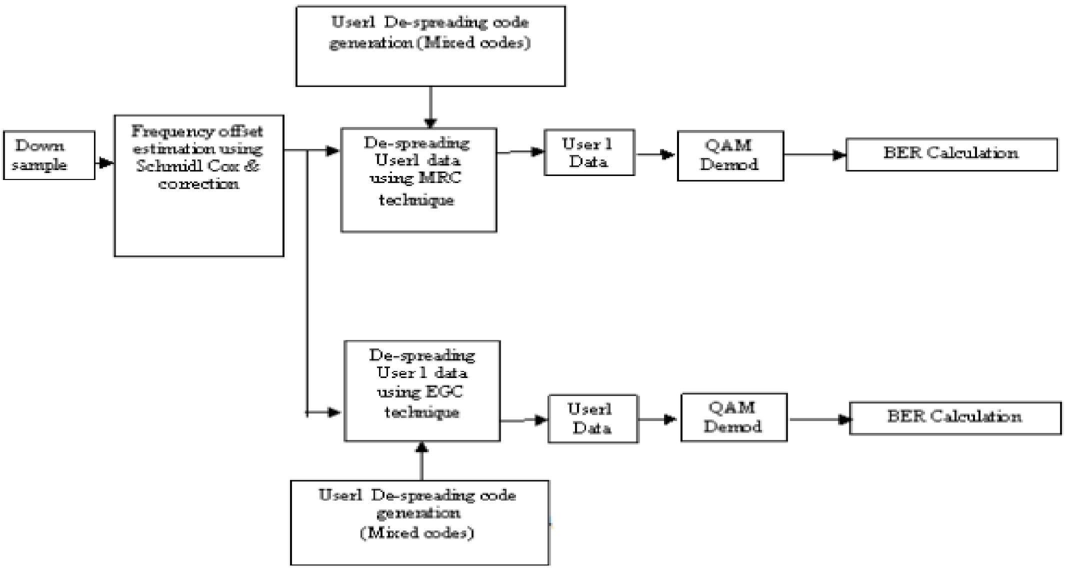
5. The RC pulse shaped data is multiplied with different channels to show the multipath effect.

6. The data convolved with channel 1 and channel 2 are summed together.

7. The summed up data is upsampled.

8. The upsampled data is then given to the WiCOMM-T Tx interface block to send through the WiCOMM-T.

Receiver


User1 De-spreading code generation (Mixed codes)
Down sample
Frequency offset estimation using Schmidl Cox & correction
De-spreading User1 data using MRC technique
User 1 Data
QAM Demod
BER Calculation
De-spreading User 1 data using EGC technique
User1 Data
QAM Demod
BER Calculation
User1 De-spreading code generation (Mixed codes)

1. The samples are received from the WiCOMM-T Rx interface block

2. The received samples are down sampled

3. The down sampled signals are de-spreaded using User1 de-spreading codes using MRC and EGC technique

4. The de-spreaded data are QAM demodulated for both MRC and EGC.

5. BER is calculated for the QAM demodulated data for both MRC and EGC.

Procedure

Note: Refer Appendix A on how to setup WiCOMM-T and Appendix B on how to generate the modem samples, vary the parameters, transmit, receive and analyzing the received modem samples etc. The following are the default values used for this experiment.

1. Connect WiCOMM-T for baseband loop back.

2. Select CDMAPART2 from the experiments list in EXPERIMNT window.

3. Select the SNR maximum and minimum value from pop up menu for generating the transmitter modem sample.

4. Transmit and receive the modem sample through WiCOMM-T.

5. Analyse the received modem samples.

6. Observe the BER plot generated by MATLAB for MRC and EGC techniques.

7. Connect WiCOMM-T in IF loop back and repeat steps 2 to 6

8. Connect 2 WiCOMM-Ts in baseband level and repeat steps 2 to 6

9. Connect the 2 WiCOMM-Ts in IF level and repeat steps 2 to 6

Note: For running this experiment between two WiCOMM-Ts such that one will be transmitter and other will be receiver, ‘data1.bin’, ‘data2.bin’generated by transmitter Matlab file under ‘C:\WiCOMM-T\EXPERIMENTS\CDMAPART2\REF_Data’ directory should be copied to receiver ‘C:\WiCOMM-T\EXPERIMENTS\CDMAPART2 \REF_Data’ directory since receiver Matlab code refers ‘data_1.bin’& ‘data_2.bin’ file for finding pilot symbols & BER calculations.

BER Calculation

The BER value for the two different types of rake receiver combining technique is obtained and tabulated. Find out the best combining technique from these results.

Enrollment no :-

EXPERIMENT: 6

STIMULATION OF WIRELESS CHANNEL – RAYLEIGH FADING

Aim:

To stimulate wireless channel including Rayleigh Fading using MATLAB

Theory:

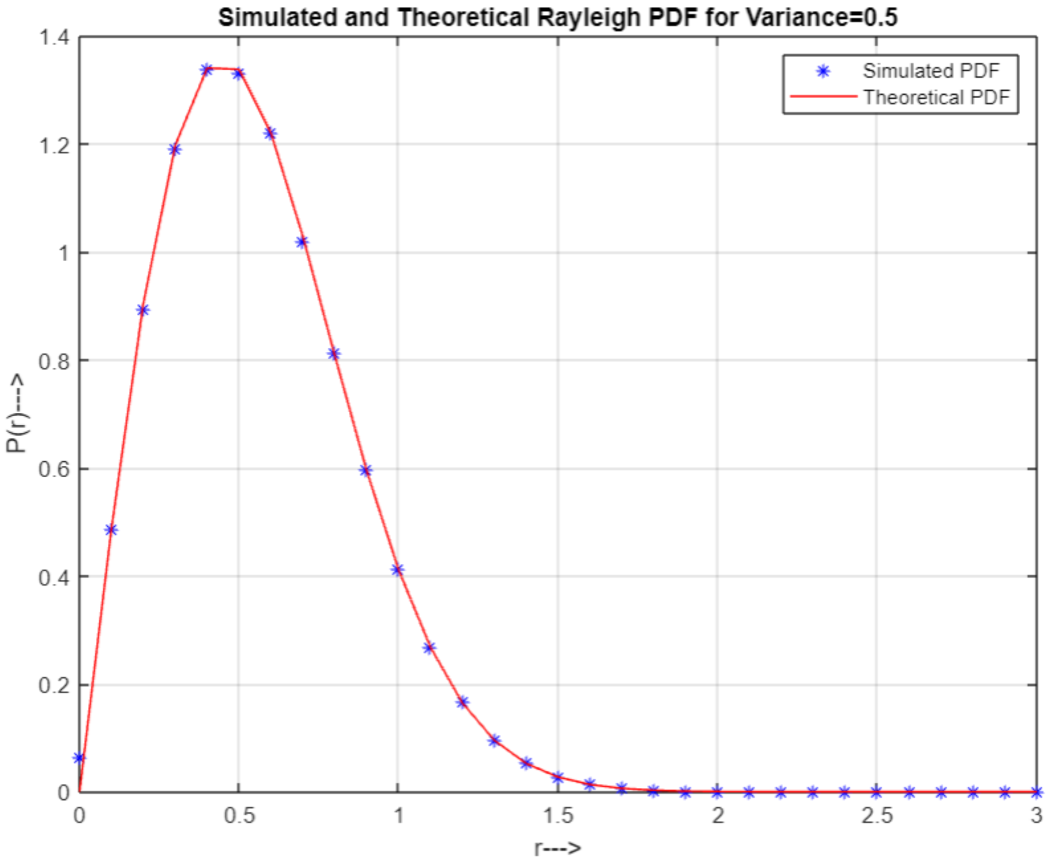
Rayleigh fading is a statical model for the effect of a propagation environment on a radio signal, such as that used by wireless devices. Rayleigh fading models assume that the magnitude of a signal that has passed through such a transmission medium (also called a communication channel will vary randomly, or fade, according to a Rayleigh distribution – the Radial component of the sum of the two uncorrelated Gaussian random variables. Rayleigh fading is viewed as a reasonable model for tropospheric and ionospheric signal propagation as well as theeffect of heavily built-up urban environments on radio signals.Rayleigh fading is most applicable when there is no dominant propagation along a line of sight between the transmitter and receiver.

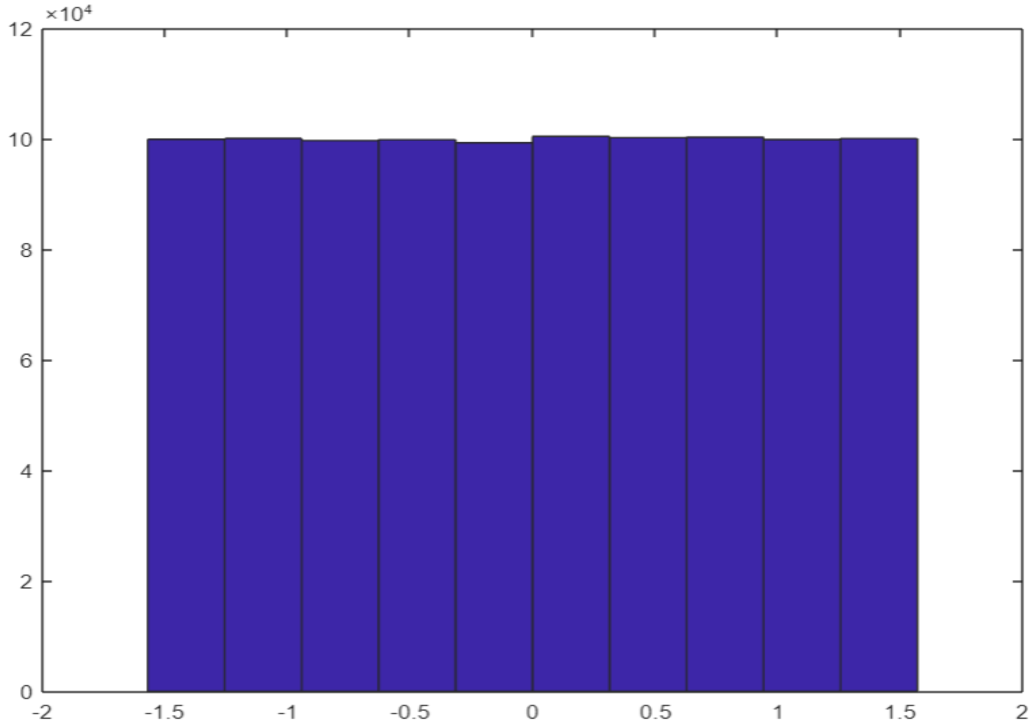
PROGRAM

```
 %----Rayleigh PDF-------
%----------Input Section------
N=1000000;% Number of samples to generate
Variance= 0.2; % Variance of underlying Gaussian random variables
%------------------------------------------------------
%Independent Gaussian random variables with zero mean and unit variance
x=randn(1, N);
y = randn(1, N);
%Rayleigh fading envelope with the desired variance
r= sqrt(Variance*(x.^2+y.^2));
%Define bin steps and range for histogram plotting
step=0.1; range=0:step:3;
%Get histogram values and approximate it to get the pdf curve
h=hist(r, range);
approxPDF=h/(step*sum(h));  %Simulated PDF from the x and y samples
%Theoritical PDF from the Rayleigh Fading equation
```

```
theoretical=(range/Variance).*exp(-range.^2/(2*Variance));
plot(range, approxPDF,'b*', range, theoretical,'r');
title('Simulated and Theoretical Rayleigh PDF for Variance=0.5')
legend('Simulated PDF', 'Theoretical PDF')
xlabel('r--->');
ylabel('P(r)--->');
grid;
%PDF of phase of the Rayleigh envelope
theta=atan(y./x);
figure(2)
hist(theta); %Plot histogram of the phase part
%Approximate the histogram of the phase part to a nice PDF curve
[counts.range] =hist(theta, 100);
step=range(2)-range(1);
approxPDF=counts/(step*sum(counts)); %Simulated PDF from the x and y samples
bar(range, approxPDF.'b');
hold on
plotHandle=plot(range, approxPDF.'r');
set(plotHandle, 'Line Width', 3.5);
axis([-2 2 0 max(approxPDF)+0.2])
hold off
title('Simulated PDF of Phase of Rayleigh Distribution');
xlabel('\theta --->');
ylabel('P(\theta) --->');
grid;
```

Stimulation Waveform





Result:

The program for Wireless channel – Rayleigh Fading was simulated successfully.

Enrollment no :-

EXPERIMENT: 7

OFDM SIGNAL TRANSMISSION AND RECEPTION

Aim:

To stimulate Orthogonal Frequency Division Multiplexing (OFDM) using MATLAB

Theory:

In telecommunications, the orthogonal frequency - division multiplexing (OFDM) is a type of digital transmission and a method of encoding digital data on multiple carrier frequencies. OFDM has developed into a popular scheme for wideband digital communication, used in applications such as digital television and audio broadcasting, DSL internet access, wireless networks, power line networks. and 4G/5G mobile communications. The main advantage of OFDM over single-carrier schemes is its ability to cope with severe channel conditions (for example, attenuation of high frequencies in a long copper wire, narrowband interference and frequency selective fading due to multipath) without complex equalization filters. Channel equalization is simplified because OFDM may be viewed as using many slowly modulated narrowband signals rather than one rapid modulated wideband signal.

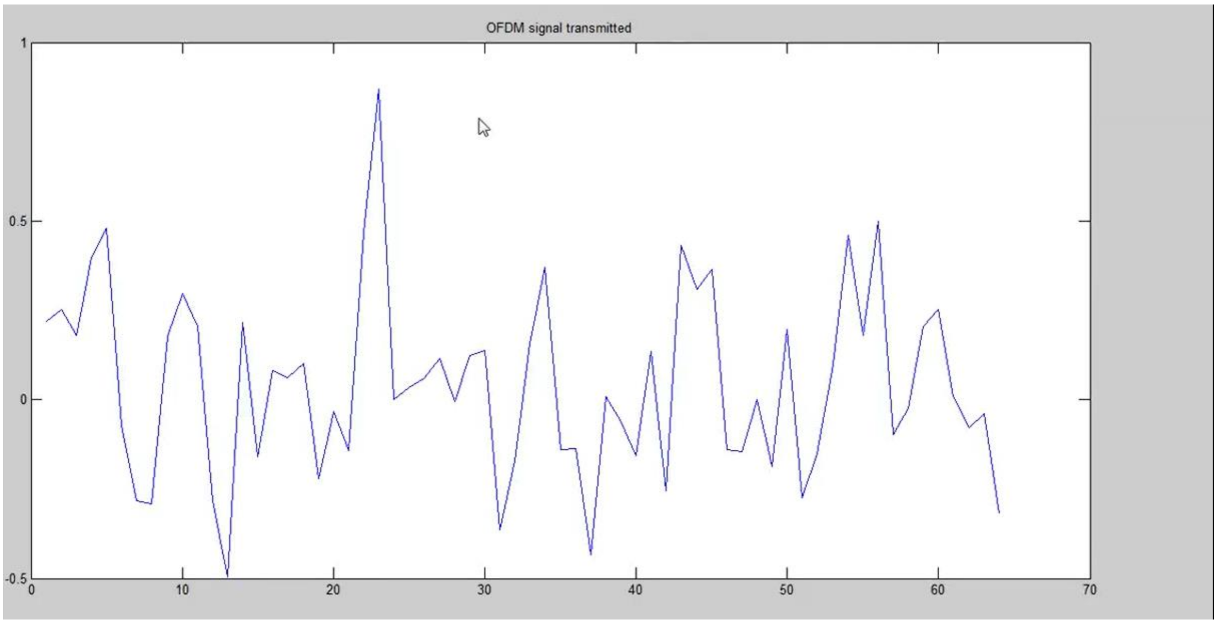
The low symbol rate makes the use of a guard interval between symbols affordable, making it possible to eliminate intersymbol interference (ISI) and use echoes and time spreading (in analog television visible as ghosting and blurring, respectively) to achieve a diversity gain, i.e., a signal-to-noise ratio improvement. This mechanism also facilitates the design of single frequency networks (SFNs) where several adjacent transmitters send the same signal simultaneously at the same frequency, as the signals from multiple distant transmitters may be re-combined constructively, sparing interference of a traditional single-carrier system.

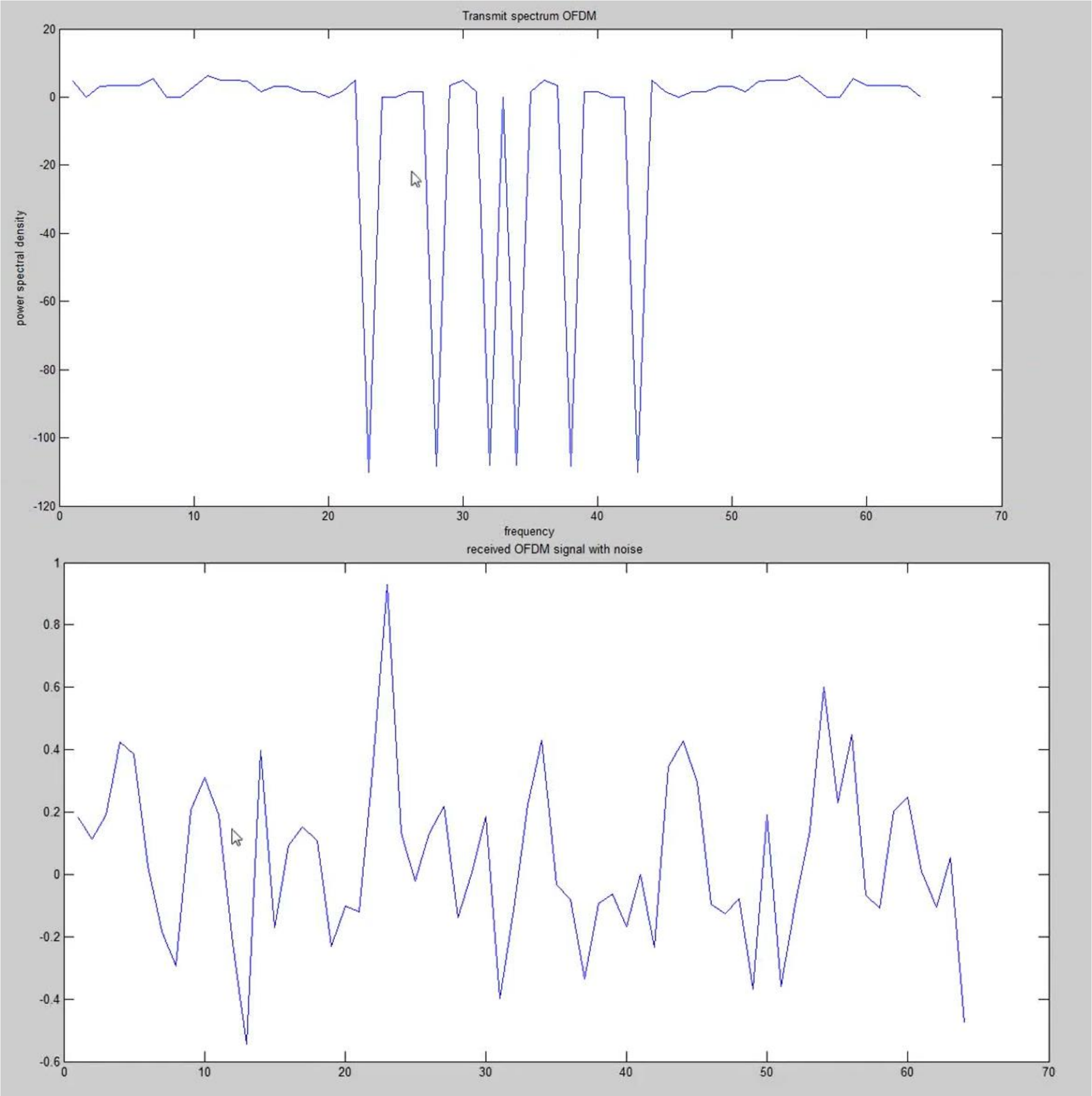
PROGRAM:

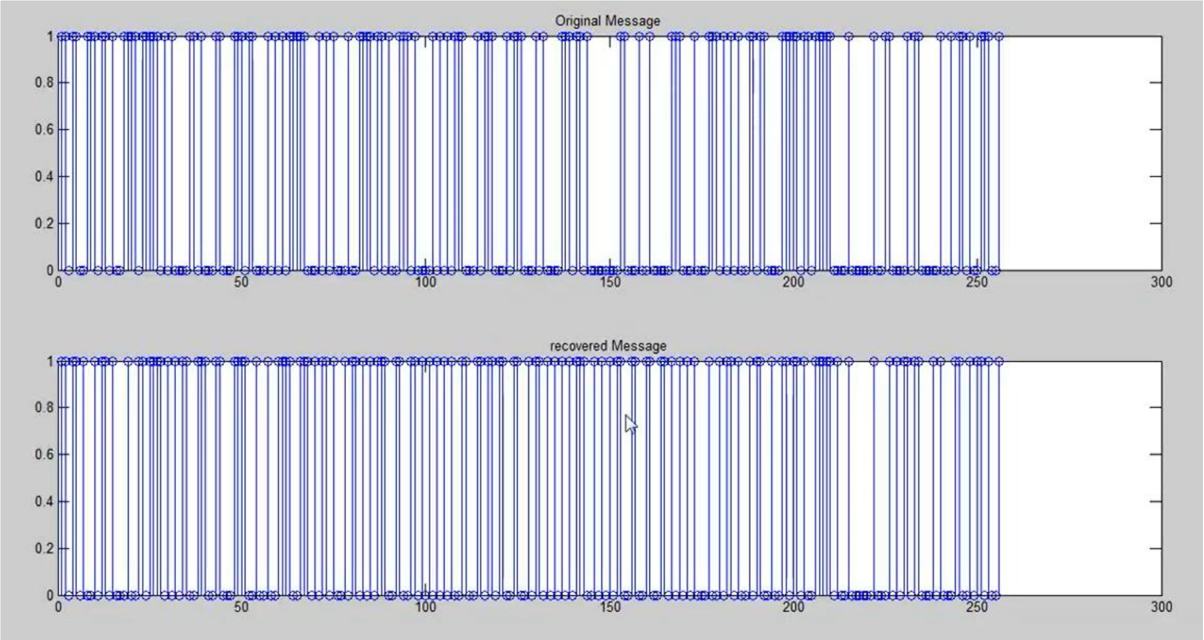
```
%%matlab simulation code
%code for OFDM signal transmission and reception in AUGN channel
%code
n=256; %Number of bits to process
x=randint(n,1); % Random binary data stream
M = 16; % Size of signal constellation
% code for OFDM signal transmission and reception in AUGN channel
```

```
%code
n =16; %Size of signal constellation
k= log2 (M); %Number of bits per symbol
xsym= bi2de (reshape (x, k, length(x)/k).', 'left-msb');
%Convert the bits in x into k-bit symbols.
y = modulate (modem.qammod (M),xsym); % Modulate using QAM
tu-3.2e-6; % useful symbol period
tg=0.8e-6; % guard interval length
ts-tu+tg; % total symbol duration
nmin=0;
nmax=64; % total number of subcarriers
scb=312.5e3; % sub carrier spacing
fc-3.6e9; %carrier frequency
Rs=fc;
tt=0: 6.2500e-008: ts-6.2500e-008;
c=ifft (y, nmax); % IFFT
s=real (c'.*(exp (1j*2 *pi*fc*tt))) ; %bandpass modulation
figure;
plot (real (s));
title('OFDM signal transmitted');
figure;
plot (10* log10 (abs (fft (s, nmax)))) ;title ('OFDM spectrum ');
xlabel('frequency')
ylabel('power spectral density')
title( 'Transmit spectrum OFDM');
snr=10; %signal to noise ratio
ynoisy = awgn(s,snr, 'measured'); %awgn channel
ynoisy augn (s, snr, 'measured'); tawgn channel
figure;
plot (real (ynoisy), 'b');title('received OFDM signal with noise');
z=ynoisy.*exp(j*2*pi*fc*tt); %Bandpass demodulation
z=fft(z, nmax); %FFT
zsym=demodulate(modem.qamdemod (M), z); % demodulation of bandpass
z=de2bi (zsym, 'left-msb'); %Convert integers to bits.
z = reshape (z.',prod (size (z)),1); %matrix to vector conversion
[noe, ber] = biterr (x,z); %BER calculation figure;
figure;
subplot (211); stem (x (1:256)) ;title('Original Message');
subplot (212); stem (z (1:256)) ;title ('recovered Message');
```

Stimulation Waveform




Transmit spectrum OFDM
power spectral density
frequency
20
0
-20
-40
-60
-80
-100
-120
0
10
20
30
40
50
60
70
received OFDM signal with noise
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6



Result:

The program for Orthogonal Frequency Division Multiplexing (OFDM) was simulated successfully.

Enrollment no :-

EXPERIMENT: 8

GLOBAL SYSTEM FOR MOBILE COMMUNICATION (GSM)

Aim

To study Gaussian Minimum Shift Keying (GMSK) modulation technique

To design a receiver using Viterbi algorithm

To study the BER using Viterbin

Theory

- GMSK modulation
- Why GMSK modulation for GSM
- GMSK signal generation
- GSM transmitter
- GSM Receiver using Viterbi

Description

GMSK Modulation

Offset QPSK (OQPSK) is obtained from QPSK by delaying the Q data stream by 1 bit with respect to the I data stream. MSK is derived from OQPSK by replacing the rectangular pulses in amplitude with a half cycle sinusoidal pulse. MSK modulation makes the phase change linear and limited to +π /2 over a bit interval of T. Because of this linear phase change, the power spectral density has low side lobes that help to control adjacent channel interference. In MSK when the half sinusoidal pulse is replaced by Gaussian Pulse shape then the modulation is Gaussian Minimum Shift Keying (GMSK)

Why GMSK Modulation for GSM?

The phase of the transmitted signal in GMSK scheme is continuous and smoothed by a Gaussian filter. This results in more compact spectrum which enables better utilization of the available frequency spectrum. The side lobe energy for GMSK is less and hence channel spacing can be tighter. The compact spectrum is beneficial in a mobile communication scenario where the operators pay premium for bandwidth. Phase modulation, further, makes the transmitted signal to have constant envelope. The constant envelope property enables employing lower cost class C power amplifiers at the receiver end thereby reducing the overall cost.

GMSK Signal generation

To generate the GMSK signals the input data stream is first passed through a Gaussian Low pass filter with a Time-Bandwidth product (BT) of 0.3. This filter deliberately introduces ISI

spreading the bits over a period of 3 bits. The impulse response of the Gaussian low pass filter is given by:

GSM Transmitter

Each GSM transmitter frame consists of 156.25 symbols. Six such frames constitute a hyper frame. Ten hyper frames repeated one after the other constitute the transmitted information. Total number of samples transmitted is N samples = 8 x 156.25 x 6 x 10 = 75000. The frame structure of the GSM transmitter consists of first 2 frames for the identification. They are the FCCH (Frequency Control Channel) and the SCH (Synchronization Channel). The remaining 4 frames carry the actual data to be transmitted.

- The FCCH consists of a 148 '0' bits followed by 8.25 random guard bits. It is mainly used to estimate the frequency difference between received and transmitted frequencies S4(n).

- The SCH channel has a known 64 bit sequence with good correlation properties. Hence this channel is used for frame synchronization. (In our case we use the whole SCH frame for synchronization)
- The traffic channel contains the data to be decoded.

In this experiment, the parameters are estimated under noise free conditions.

GSM Receiver

GMSK signals can be detected in many ways. Optimal GMSK detection can be performed using MLSE, which is nonlinear and highly complex. Here for bit recovery Viterbi algorithm is used

Frequency Synchronization

Take samples of the received data, and calculate the FFT. The difference between the most dominant frequency component of the transmitted and received spectrum will give us the frequency offset between the transmitter and receiver. Necessary corrections are performed on the received data.

Frame Synchronization

Correlate the received data with the actual transmitted SCH channel and look for the peaks. The location of peak helps in identifying the beginning of the SCH channel. The beginning of the FCCH and the traffic channels are also identified.

Offset Phase Estimation

Carrier phase offset estimation is done with the help of FCCH channel. The received FCCH channel, previously identified through the frame synchronization, is decimated by a factor of 8 and the sequence S4(n) is chosen [i.e. the samples S(4), S(12), S(20) are selected]. Deterministic autocorrelation is performed over this set of data to estimate the carrier phase offset. The necessary phase corrections are made to the received data. The received data is now ready for demodulation of the traffic channels.

Traffic Channel Demodulation

The demodulation algorithm previously described is applied individually to each of the traffic channels to receive the transmitted data.

MATLAB Code implementation

1. Random data to be transmitted is generated.
2. FCCH and SCH channel are generated and then added to the random data.

3. The added data is then sent through Gaussian filter.

4. The Gaussian pulse shaped data is given to the WiCOMM-T Tx interface block to send through the WiCOMM-T.

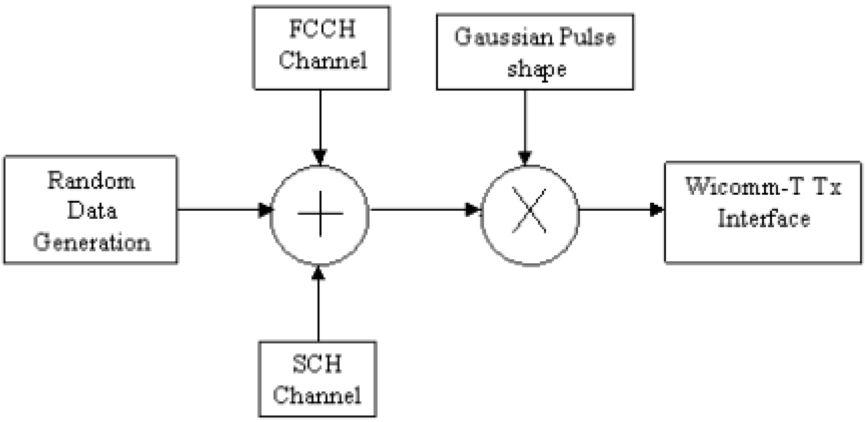


Fig. : Transmitter block diagram in MATLAB

Receiver

1. The samples are received from the WiCOMM-T Rx interface block

2. Frequency offset of the received samples are estimated and then corrected.

3. Identification of the SCH, FCCH channels and the burst data are done in the frequency offset corrected samples.

4. The phase offset is estimated and corrected.

5. The phase offset corrected samples are convolved with the Matched filter to recover the Burst data.

6. BER is calculated for various values of SNR and is plotted against the theoretical value.

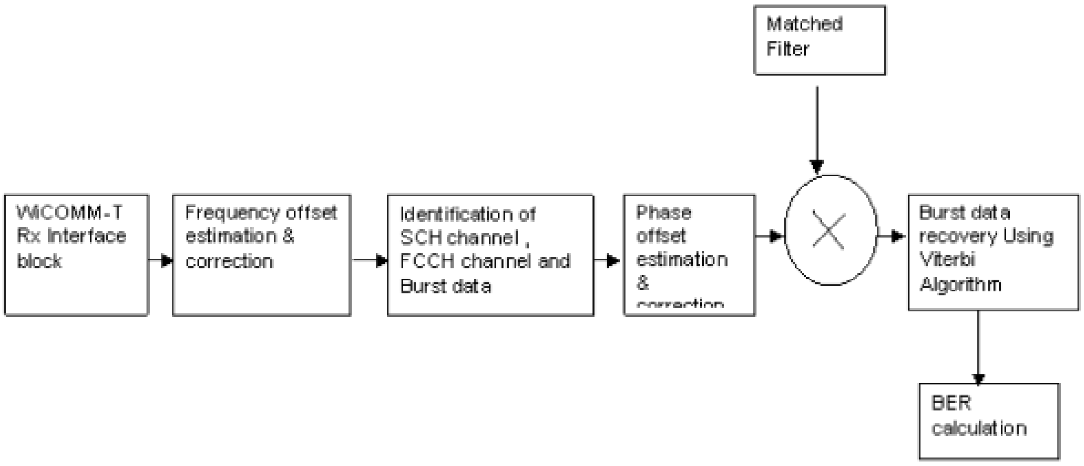

Matched Filter
WiCOMM-T Rx Interface block
Frequency offset estimation & correction
Identification of SCH channel , FCCH channel and Burst data
Phase offset estimation & correction
Burst data recovery Using Viterbi Algorithm
BER calculation

Fig. : Receiver block diagram in MATLAB

Procedure

Note: Refer Appendix A on how to setup WiCOMM-T and Appendix B on how to generate the modem samples, vary the parameters, transmit, receive and analyze the received modem samples etc.

1. Connect WiCOMM-T in baseband loop back with the sampling rate set to 2MBps.

2. Generate the transmitter modem sample.

3. Transmit and receive the modem sample through WiCOMM-T and analyse the received modem samples.

4. Observe various plots generated by MATLAB.

5. Connect WiCOMM-T in IF loop-back and repeat steps 2 to 4

6. Connect 2 WiCOMM-Ts such that one as transmitter and other as receiver in baseband and in IF and repeat steps 2 to 4

Note: For running this experiment between two WiCOMM-Ts such that one will be transmitter and other will be receiver, 'bits.bin', generated by transmitter Matlab file under 'C:\WiCOMM-T\EXPERIMENTS\GMSK\REF_Data' directory should be copied to receiver 'C:\WiCOMM-T\EXPERIMENTS\GMSK\REF_Data' directory since receiver Matlab code refers 'bits.bin' file for synchronization & BER calculation.

Result:

Thus the experiment was performed successfully.

Enrollment no :-