**Name:Trusha Gondaliya Enrolment No:190310116009 Semester:7**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Definition** | **Date** | **Signature** | **Remarks** |
| **1** | To Briefly introduction about SCILAB. |  |  |  |
| **2** | Write a program to calculate cluster size N for given value of signal to interference ratio. |  |  |  |
| **3** | To determine the freespace loss and the power received using SCILAB program. |  |  |  |
| **4** | To write a SCILAB program to calculate the link budget for satellite communication. |  |  |  |
| **5** | To write a SCILAB program to calculate the median path loss for Hata model for outdoor propagation. |  |  |  |
| **6** | To study the BER performance of DS-CDMA using mixed codes in multipath channel  using RAKE receiver for single user case. |  |  |  |
| **7** | To study the BER performance of MRC combining and equal combining varying with SNR. |  |  |  |
| **8** | To stimulate wireless channel including Rayleigh Fading using SCILAB |  |  |  |
| **9** | To stimulate Orthogonal Frequency Division Multiplexing (OFDM) using SCILAB |  |  |  |
| **10** | 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 SCILAB

**Aim:**

To Briefly introduction about SCILAB.

# Theory:

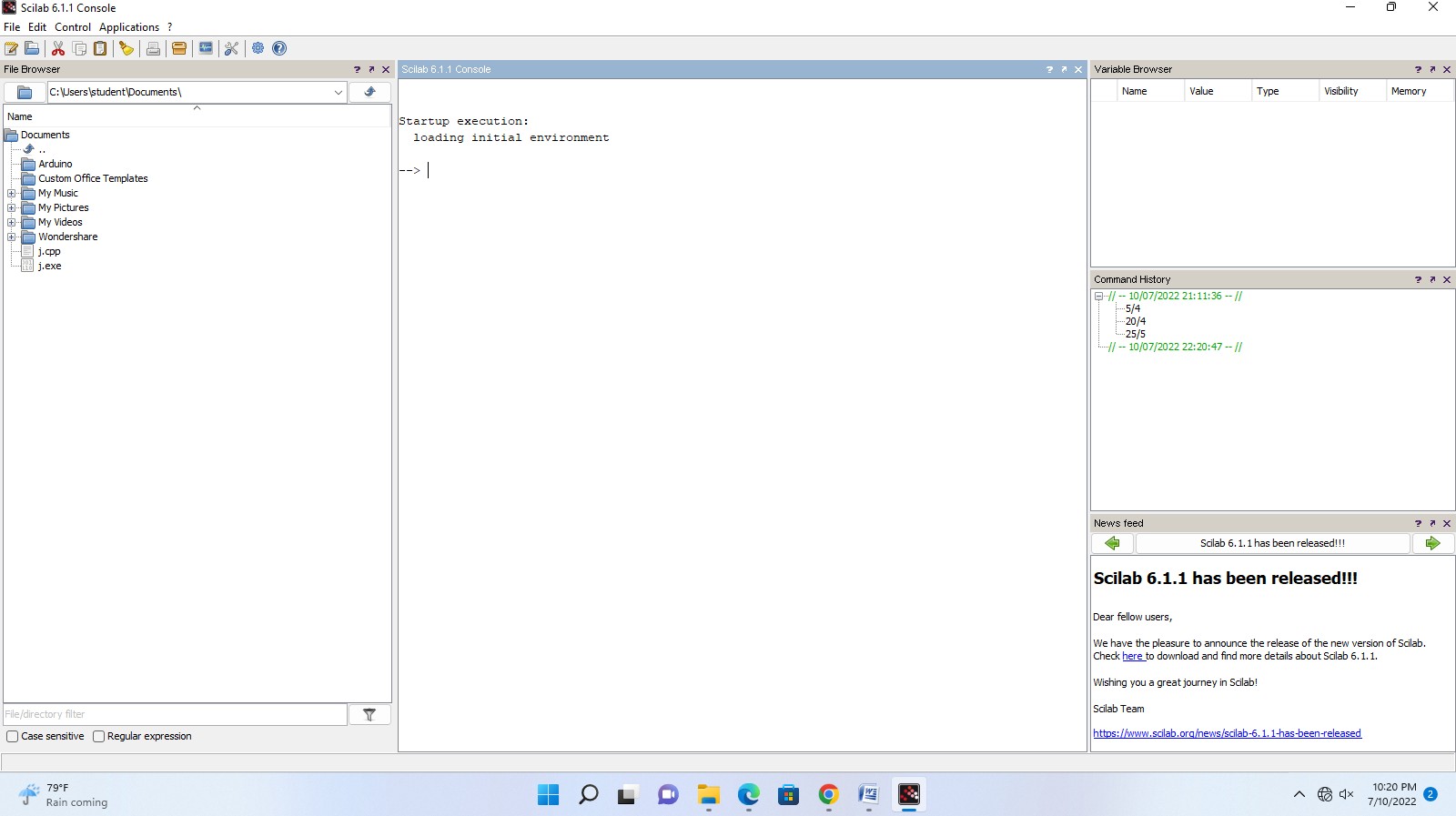
INTRODUCTION TO SCILAB:

The useful workspace in Scilab consists of several windows:

* The console for making calculations,
  + The editor for writing programs,
  + The graphics windows for displaying graphics,
  + The embedded help.

The general environment and the console

After double-clicking the icon to launch Scilab, Scilab environment by default consists of the following docked windows – console, files and variables browsers, command history.



1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

In the console after the prompt “ --> “, just type a command and press the Enter key on the keyboard to obtain the corresponding result.

--> 57/4

ans = 14.25

--> (2+9)^5

ans = 161051.

It is possible to come back at any moment with the keyboard's arrow keys ← ↑ → ↓ or with the mouse. The left and right keys are used to change the instructions and the up and down keys are used to come back on a previously executed command.

**Simple numerical calculations**

All computations done with Scilab are numerical. Scilab performs computations with matrices(see chapter 2, page 23).

Operations are written with “ **+** “ for addition, “ **–** “ for subtraction, “ **\*** “ for multiplication, “ **/** “ for division, “ **^**“ for exponents. For example:

-->2+3.4

Ans = 5.4

The case is sensitive. It is thus necessary to respect uppercase and lowercase for the calculations to be performed properly. For example, with **sqrt** command (which calculates the square root):

-->sqrt(9) while: SQRT(9) Ans= 3 !--error 4

Undefined

variable:

1. **Enrollment Number:** 190310116009 **Name:Trusha Gondaliya**

SQRT

## Particular numbers

**%e** and **%pi** represent respectively *e* and *π*:

--> %e --> %pi

%e =2.7182818 %pi=3.1415927

**%i** represents the **i** of complexes in input and is displayed **i**in output:

--> 2+3\*%i

Ans= 2+3i

## For not displaying the results

In adding a semi colon “ **;** “ at the end of a command line, the calculation is done but the result is not displayed.

-->(1+sqrt(5))/2; --> (1+sqrt(5))/2

Ans= 1.618034

To remind the name of a function

The names of commonly used functions are summarized in Chapter 3 of this document (page 32). For example:

--> exp(10)/factorial(10)

ans = 0.0060699

The tab key →│ on the keyboard can be used to complete the name of a function or a variable by giving its first few letters.

For example, after typing in the console the command:

-->fact

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

and then pressing the tab key, a window is displayed with all the functions and variables names beginning with **fact**, such as **factorial** and **factor**. Just double click on the required function or select it with the mouse or with the keys ↑ ↓ and press Enter (Windows and Linux) or Return (Mac OS X) to insert it in the command line.

**The menu bar**

The menus listed below are particularly useful.

## Applications

* The command history allows you to find all the commands from previous sessions to the current session.
* The variables browser allows you to find all variables previously used during the current session.

## Edit

**Preferences** (in **Scilab** menu under Mac OS X) allows you to set and customize colors, fonts and font size in the console and in the editor, which is very useful for screen projection.

Clicking on **Clear Console** clears the entire content of the console. In this case, the command history is still available and calculations made during the session remain in memory. Commands that have been erased are still available through the keyboard’s arrow keys.

## Control

To interrupt a running program, you can:

* Type **pause** in the program or click on **Control> Interrupt**

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

in the menu bar (Ctrl X under Windows and Linux or Command X under Mac OS X), if the program is already running. In all cases, the prompt “ **-->** “ will turn into “ **-1-**

**>** “, then into “ **-2->** “…, if the operation is repeated.

* + To return to the time prior to the program interruption, type **resume** in the console or click on **Control > Resume**. To quit for good a calculation without any possibility of return, type **abort** in the console or click on **Control > Abort** in the menu bar.

**The editor**

Typing directly into the console has two disadvantages: it is not possible to save the commands and it is not easy to edit multiple lines of instruction. The editor is the appropriate tool to run multiple instructions.

## Opening the editor

To open the editor from the console, click on the first icon in the toolbar  or on **Applications**

**> SciNotes** in the menu bar.

The editor opens with a default file named “ **Untitled 1**“.

## Writing in the editor

Typing in the editor is like as in any word processor.

In the text editor, opening and closing parentheses, end loops, function and test commands are added automatically. However, these features can be disabled in **Options > Auto-•‐ completion on** menu, in clicking on the two below entries enabled by default:

# (,[,…

* **if,function,…**

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

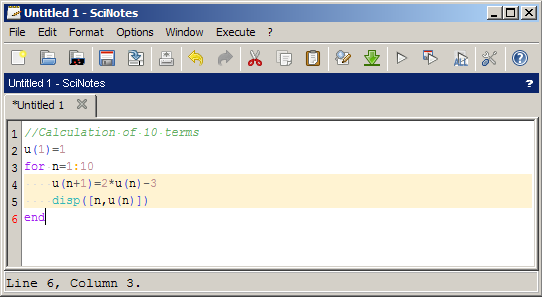
While in principle each instruction should be entered on a separate line, it is possible to type multiple statements on a same line separating each statement with a semicolon “ **;**“.

A space at the start of the line called indentation is automatic when a loop or a test is started.

In the following example, we calculate 10 terms of the sequence (𝑢𝑛) defined by:

𝑢1 = 1

𝑢𝑛!1 = 2𝑢𝑛 – 3



## Saving

Any file can be saved by clicking on **File > Save as**.

The extension “ .sce “ at the end of a file name will launch automatically Scilab when opening it (except under Linux and Mac OS X).

Copying into the console, executing a program

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

In clicking on Execute in the menu bar, three options are available:

* + Execute “ **…file with no echo** “ (Ctrl Shift E under Windows and Linux, Cmd Shift E under Mac OS X): the file is executed without writing the program in the console (saving the file first is mandatory).
  + Execute **“ … file with echo** “ (Ctrl L under Windows and Linux, Cmd L under Mac OS X): rewrite the file into the console and executes it.
  + Execute “ **…until the caret, with echo** “ (Ctrl E under Windows and Linux, Cmd E under Mac OS X): rewrite the selection chosen with the mouse into the console and executes it or execute the file data until the caret position defined by the user.

Standard copy/paste can also be used.

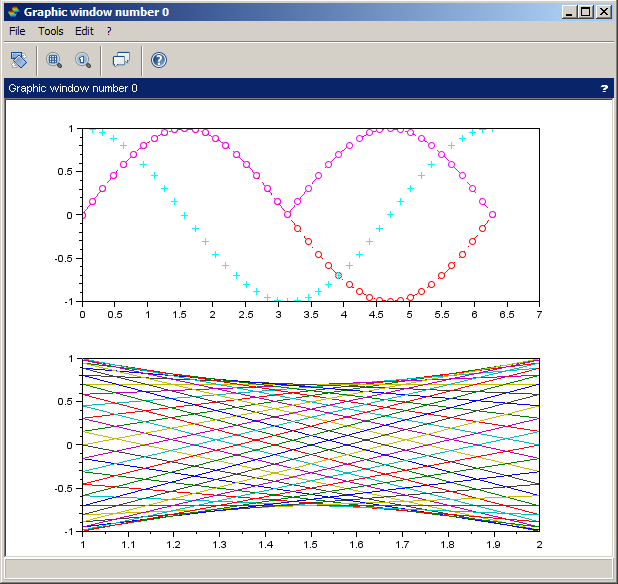
**The graphics window**

## Opening a graphics window

A graphics window opens automatically when making any plot. It is possible to plot curves, surfaces, sequences of points (see chapter 2, page 18).

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

To obtain an example of curve, type in the console:



## Modifying a plot

The magnifying glass  allows zooming. To zoom in two dimensions, click on the tool and with the mouse create a rectangle which will constitute the new enlarged view. To zoom in three dimensions, click on the tool and create a parallelepiped which will constitute the new enlarged view. It is also possible to zoom in using the mouse wheel. To return to the initial screen, click on the other magnifying glass .

The icon enables rotation of the figure (particularly useful in 3-•‐D) with right click actions which are guided by messages in the bottom of the graphics window.



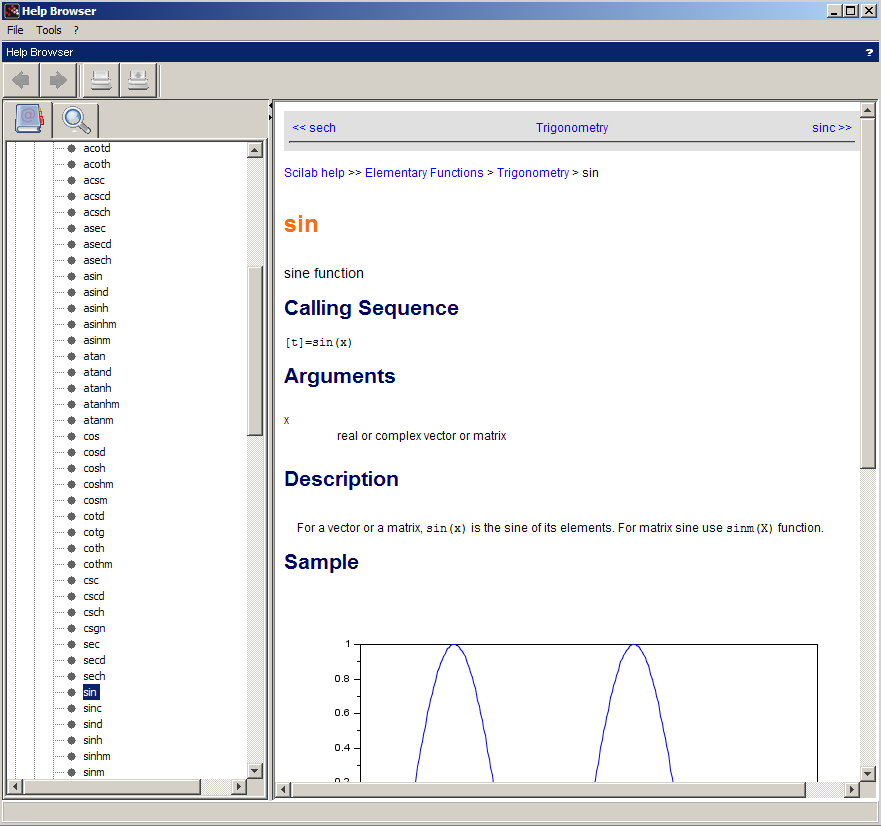
For more precise modifications, click on **Edit > Figure properties** or **Axes properties** and let yourselves be guided (this option is not yet available under Mac OS X).

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

## Online help

To access the online help, click on **? > Scilab Help** in the menu bar, or type in the console:

-->help



To get help with any function, type help in the console followed by the name of the appropriate function. For example:

-->help sin

displays help for **sin** (sine) function.

**Windows management and workspace customization**

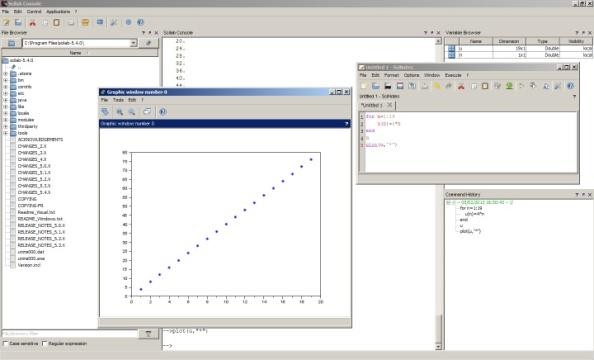
As in the default Scilab environment, where the console, files and variables browsers and command history are all together docked windows, all other windows in Scilab can be repositioned in a single one. For example, the user can choose to position the editor in the default environment of Scilab.

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

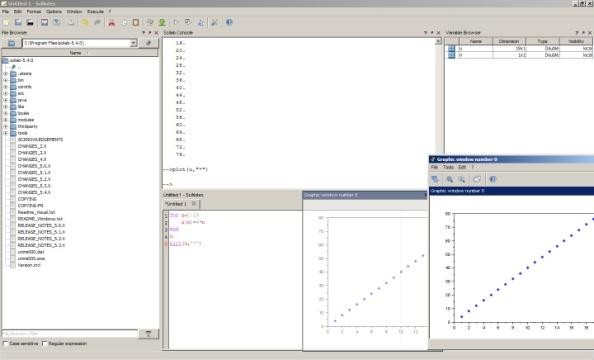
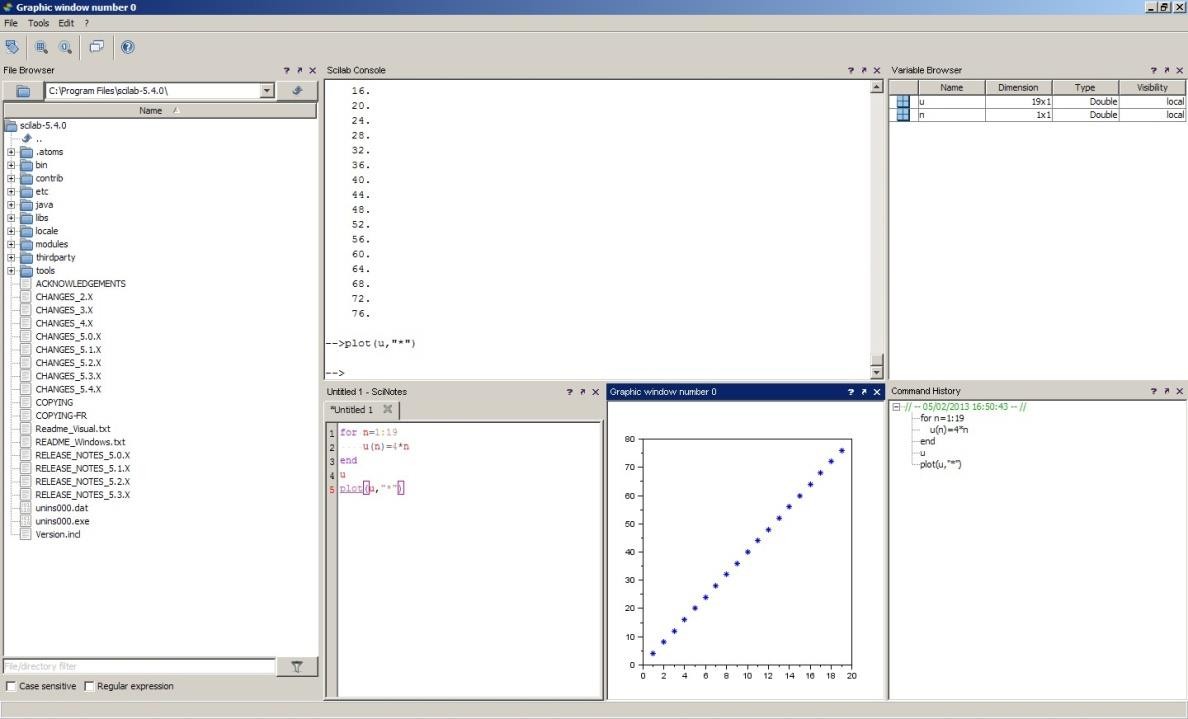
To dock a window in another one, first identify the blue horizontal bar under Windows, or black under Mac OS X and Linux, at the top of the window in the toolbar containing a question mark on the right.

* + Under Windows and Linux, click on this bar with the left mouse button and, while maintaining the click, move the mouse pointer in the desired window.
  + Under Mac OS X, click on this bar and while maintaining the click, move it in the desired window.

A rectangle appears indicating the future positioning of the window. When the position is the one you want, release the mouse button. To cancel and bring out the window, click on the small arrow on the right of the same bar.



1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**



1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# EXPERIMENT: 2

**Aim:**

Write a program to calculate cluster size N for given value of signal to interference ratio.

# Program:

clc; clear all;

SIRdb=input('Enter the SIRdb='); SIR=10^(SIRdb/10)

N=input('Enter the number of cell per cluster, N='); i=input('Enter the number of cochannel cells,i='); n=input('Enter the path loss exponent,n='); N=(1/3)\*((SIR\*i)^(2/n))

if N > 1 && N <=3 N=3;

elseif N >3 && N<=7 N=7;

elseif N>7 && N<=12

N12;

elseif N>12 && N<=13 N=13;

end; N

*//for this value of N the given SIR is obtained or not* SIRnew=(sqrt(3\*N)^n)/i SIRnewdb=10\*log10(SIRnew)

if SIRnewdb>SIRdb

disp('The new cluster size can be implemented'); end;

N

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# Output:

Enter the SIRdb=20

Enter the number of cell per cluster, N=12 Enter the number of cochannel cells,i=10 Enter the path loss exponent,n=6

"The new cluster size can be implemented"

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# EXPERIMENT: 3

**Radio Propagation and Propagation PathLoss Models**

# Aim: Scilab code To find free space and reflected surface attenuations.

**Program:**

// To de t e rmine f r e e spa c e and r e f l e c t e d s u r f a c e a t t e n u a t i o n s .

clc ; clear all;

hb =100; // i n f e e t s ( h e i g h t o f BS ant enna ) hm =5; // i n f e e t s ( h e i g h t o f mobi l e ant enna ) f =881.52; // i n MHz

lamda =1.116; // i n f e e t d =5000; // i n f e e t

Gb =10^0.8; // 8dB(BS ant enna g a in )

Gm =10^0; // 0dB ( Mobi l e ant enna g a in )

// s o l u t i o n

free\_atten =(4\* %pi\*d/ lamda ) ^2\*( Gb\*Gm)^ -1; y= round (10\* log10 ( free\_atten ));

disp ( “ Fr e e spa c e a t t e n u a t i o n i s %d dB nn “ ,y);

reflect\_atten = (d ^4/( hb\*hm) ^2) \*( Gb\*Gm)^ -1; x= round (10\* log10 ( reflect\_atten ));

disp ( “ Re f l e c t i n g s u r f a c e a t t e n u a t i o n i s %d dB nn

“ ,x);

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# C:\Users\student\Downloads\WhatsApp Image 2022-08-22 at 9.42.44 AM.jpegOutput:

**Aim: Scilab code To find received signal power and SNR. Program:**

d =8000; // Di s t a n c e between ba s e s t a t i o n and mobi l e s t a t i o n

f =1.5\*10^9; // i n Hz lamda =0.2; // i n me t r e s

Pt =10; //BS t r a n smi t t e d power i n wa t t s Lo =8; // To t a l sys t em l o s s e s i n dB

Nf =5; //Mobi l e r e c e i v e r n o i s e f i g u r e i n dB T =290; // t empe r a tur e i n d e g r e e k e l v i n

BW =1.25\*10^6; // i n Hz Gb =8; // i n dB

Gm =0; // i n dB

Hb =30; // i n me t r e s Hm =3; // i n me t r e s

B =1.38\*10^ -23; //Boltzmann “ s c o n s t a n t

// s o l u t i o n

Free\_Lp =20\* log10 (Hm\*Hb/d ^2) ;

Pr= Free\_Lp -Lo+Gm+Gb+Pt; // i n dBW Te=T \*(3.162 -1) ;

Pn=B\*( Te+T)\*BW;

disp ( “ Re c e ived s i g n a l power i s %d dBW nn “ ,10\* log10

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

(Pn));

SNR =Pr -10\* log10 (Pn);

disp ( “ SNR r a t i o i s %d dB nn “ ,round ( SNR ));

# C:\Users\student\Downloads\WhatsApp Image 2022-08-22 at 9.44.06 AM.jpegOutput:

**Aim: Scilab code To find the allowable path loss Program:**

d =3\*1000; // i n me t r e s

Y =4; // path l o s s exponent

Pt =4; // Transmi t t ed power i n wa t t s f =1800\*10^6; // i n Hz

Shadow =10.5; // i n dB d0 =100; // i n me t r e s P0 = -32; // i n dBm

// s o l u t i o n

disp ("Us ing e q u a t i o n 3 . 1 1 and i n c l u d i n g shadow e f f e c t we g e t ")

Pr=P0 +10\* Y\* log10 (d0/d)+ Shadow ;

disp ( “ Re c e ived power i s %. 1 f dBm nn “ ,Pr); path\_loss =10\* log10 (Pt \*1000) -Pr;

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

disp ( “ Al l owabl e path l o s s i s %. 1 f dB nn “ , path\_loss );

# Output:

**Aim: Scilab code to find the distance between transmitter and receiver.**

# Program:

shadow =10; // i n dB Lp =150; // i n dB

// s o l u t i o n

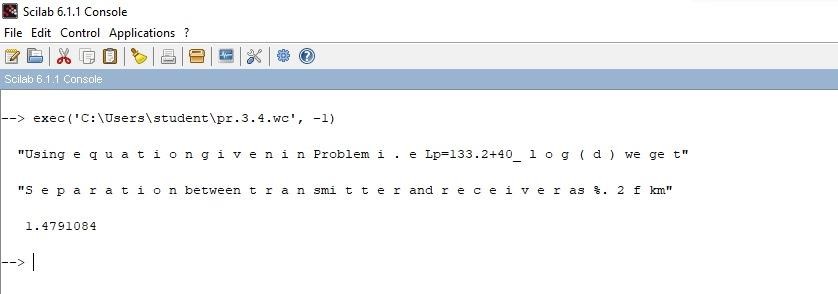
disp (" Us ing e q u a t i o n g i v e n i n Problem i . e Lp

=133.2+40\_ l o g ( d ) we ge t , "); d =10^(( Lp -10 -133.2) /40) ;

disp (" S e p a r a t i o n between t r a n smi t t e r and r e c e i v e r as %. 2 f km” , d ) ;

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# Output:



1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# EXPERIMENT: 4

**LINK BUDGET EQUATION- SATELLITE COMMUNICATION**

# Aim:

To write a SCILAB 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**

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

**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; pi=3.14;

*// 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;

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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;

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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);

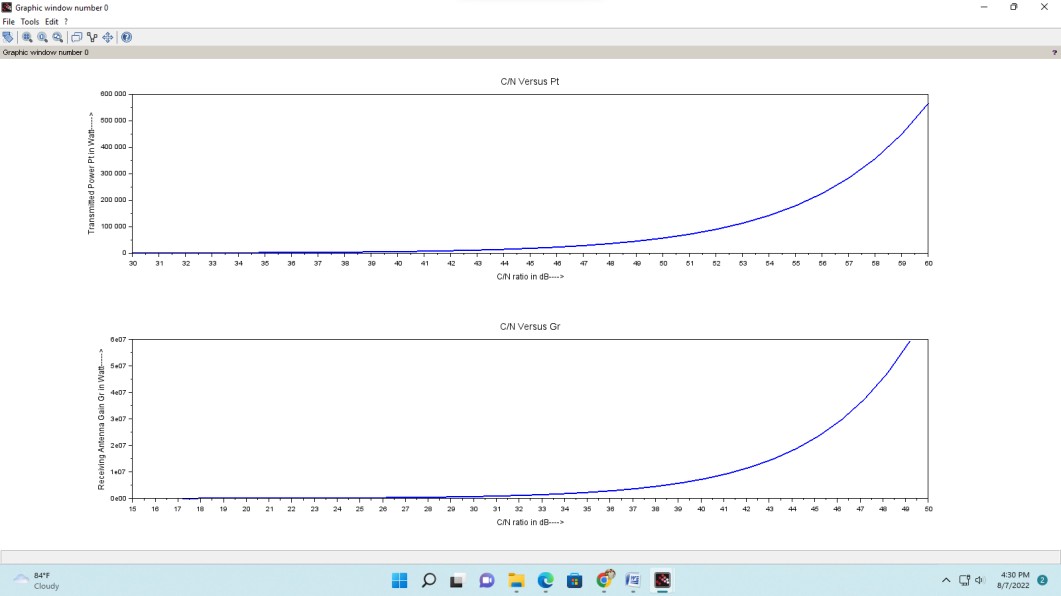
for i=1:lnn Grw(i)=10^(Gr2(i)/10); end

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:**



1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# EXPERIMENT: 5

**Aim:** To write a SCILAB 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.

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# 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

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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; clear all; close all;

f=input('enter the frequency of transmisson in mhz:'); Hb=input('enter the height of base station Antenna in meter:'); Hm=input('enter the height of mobile station Antenna in meter:'); d=input('enter the distance between the base and mobile stations:'); n=input('enter 0 for small city and 1 for large city:'); ch=0.8+(1.1\*log10(f)-0.7)\*Hm-1.56\*log10(f);

if n==0

ch=0.8+(1.1\*log10(f)-0.7)\*Hm-1.56\*log10(f); else

if f>=150 && f<=200 ch=8.29\*(log10(1.54\*Hm))^.2-1.1;

else

if f>=200 && f<=1500 ch=3.2\*(log10(11.75\*Hm))^.2-4.97;

end; end; end;

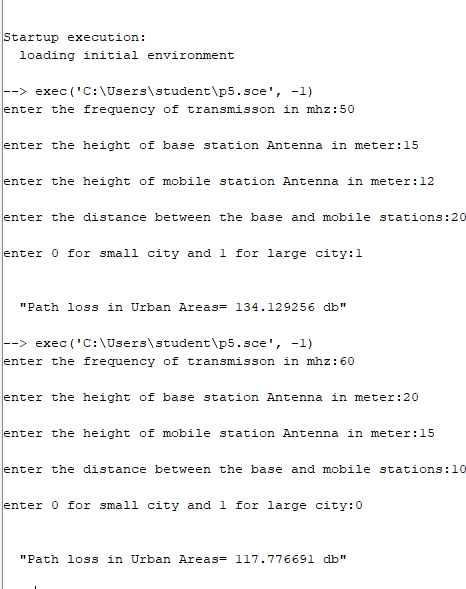
Lu=69.55+26.26\*log10(f)-13.82\*log10(Hb)-ch+(44.9-

6.55\*log10(Hb))\*log10(d);

disp(sprintf('%s %f %s','Path loss in Urban Areas=',Lu,'db'));

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# Output:



1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# EXPERIMENT: 6

**Aim:** To study the BER performance of DS-CDMA using mixed codes in multipath channel

using RAKE receiver for single user case.

# Concepts

1. What is Multipath?
2. Effect of Multipath on the performance of CDMA
3. What is RAKE Receiver?

# 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.

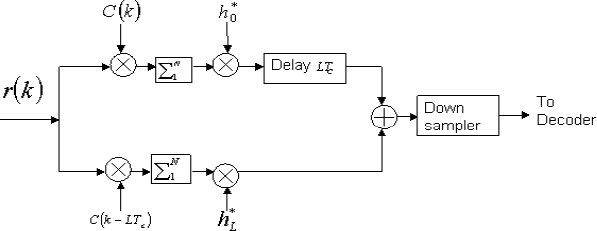
# Effect of Multipath on the performance of DS-CDMA

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

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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. h0and hLare the multipath channel coefficients. LTcis the delay that is used in the multipath channel model.

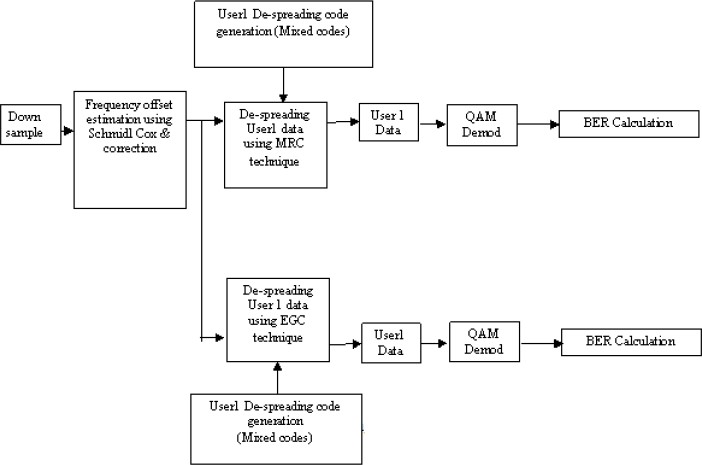
# C:\Users\jimesh\Desktop\WC-SEm-7\2.pngSCILAB Code Implementation Transmitter

**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. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**
8. The summed up data is upsampled.
9. The upsampled data is then given to the WiCOMM-T Tx interface block to send through the WiCOMM-T.

# Receiver

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,

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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.
5. Observe the BER plot generated by MATLAB for MRC and EGC techniques.
6. Connect WiCOMM-T in IF loop back and repeat steps 2 to 6.
7. Connect 2 WiCOMM-Ts in baseband level and repeat steps 2 to 6.
8. 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.

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# EXPERIMENT: 7

**Aim**: To study the BER performance of MRC is combining and equal combining varying with SNR.

# Description

**Why Spread Spectrum Technique?**

Shannon’s formula for channel capacity is a relationship between achievable bit rate, signal bandwidth and Signal to Noise Ratio (SNR).

Channel Capacity = Bandwidth\*log2(1+SNR)

When the signal is much smaller than the noise or under very low SNR condition the above relationship becomes much simpler as given below.

Channel Capacity / Bandwidth **= 1.44\*SNR** From the above relationship we can conclude that SNR can be traded for Bandwidth or vice versa. If there is a way to encode our data into a large signal bandwidth, then error free transmission is possible in a very low SNR condition. This is the reason why Spread Spectrum technique is used.

# Advantages of Spread Spectrum Technique Ability to selectively address

If the signal is spread and encoded properly, then the signal can only be decoded by a receiver which knows the transmitting code and hence a specific receiver in a group can be targeted. This is termed as Code Division Multiple Access

# Bandwidth Sharing

If the proper modulation codes are selected, it is feasible to have multiple pairs of receivers and transmitters occupying the same bandwidth

# Security

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

It is very difficult to intercept the signal if the modulation code of spread spectrum transmission is not known. If the proper spreading code is not known to demodulate, the signal will be seen as random electrical noise and not as useful signal. And also spread spectrum link puts out much less power per bandwidth than a conventional radio link, having spreading it over a wider bandwidth and hence a knowledge of the link’s spreading code is required to demodulate. Hence it is very difficult to detect.

# Immunity to Interference

If an external radio signal interferes with the spread spectrum signal, it will be rejected by the demodulator much as random noise and hence provide excellent error rate even with faint signals.

# Direct Sequence Spread Spectrum

The Spread Spectrum technique can be divided into Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS). In DSSS the Pseudo Random sequence is applied directly to baseband data entering the carrier modulator. The modulator therefore sees a much larger bit rate, which corresponds to the chip rate of the PN sequence. This code sequence is typically Pseudo random binary code or PN specially chosen for desirable statistical properties. In effect, we are transmitting a wideband noise like signal which contains embedded message data. The time period of a single bit in the PN code is termed as chip and the bit rate of the PN code is termed Chip rate.

# Spreading codes

The spreading code or the PN sequence should be ideally balanced with equal number of ones and zeros over the length of the sequence as well as cryptographically secure. Some of the most popular PN sequences are Barker, M – Sequence, Gold and Walsh. More complex sequences provide a more robust link but the implementation becomes very expensive. We have Orthogonal spreading codes, Non-Orthogonal spreading codes and Mixed spreading codes. Orthogonal codes are generated using Walsh-Hadamard series and the Non-orthogonal codes

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

are generated using Linear Feedback Shift Register (LFSR). The mixed codes are generated by multiplying the orthogonal and non-orthogonal codes. The orthogonality property of the orthogonal codes is very important for any communication system. Because of the orthogonality property, two orthogonal signals can be transmitted at the same time and will not interfere with each other. But the auto correlation function of the Walsh – Hadamard matrix can have more than one peak and therefore it is not possible for the receiver to detect the beginning of code word without an external synchronization scheme. Also the cross correlation can also be non-zero for a number of time shifts and un- synchronous users can interfere with each other. The spreading is not over the entire bandwidth instead it is over a number of discrete frequency component. Orthogonality is affected by multi-path effect.

Gold sequences are popular for Non-orthogonal codes. Here the transmission can be asynchronous. The receiver can synchronize using the auto correlation property of the Gold Sequence.

# Orthogonal Spreading Code

An important set of Orthogonal codes is the Walsh set. Walsh functions are generated using an iterative process of constructing a Hadamard matrix starting withH1 =(1). The Hadamard matrix is built by where Hn is the inverse of Hn. For example Walsh – Hadamard matrix of length 2 and 4 are given as Type equation here.

From the above matrix the Walsh-Hadamard spreading codes are given by the rows of the matrix. For example one spreading code can be generated from the 2nd row of H4which is given asH4.2 = 1 1 1 1 and the other spreading code can be generated from the 4th row of H4 H4.4 = 1 1 1 1 . Computing the orthogonality between H4.2 and H4.4 by multiplying element by element ((1x1)+(-1x-1)+(1x-1)+(-1x1))

= (1+ 1-1-1) = 0.This shows that both the spreading codes are orthogonal to each other. In our experiment we used H6 and generated a code of length 32. Then the generated code is normalized by dividing it by√32.

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# Non-Orthogonal Spreading Code

To give up the orthogonality property among users and to reduce the interference between users by using spread spectrum technique, Non- orthogonal spreading codes are used. Non-orthogonal codes are generated using LFSR method. The LFSR is a shift register whose input bit is a linear function of the previous state. The initial value of the LFSR is called Seed. It has number of flip-flops termed as registers. For the feedback mechanism the bits contained in the selected positions in the shift registers are XORed and then fed as input to the first shift register. The bit positions selected for feedback is termed as taps. The Linear Feedback Shift Register is shown below. If the length of the LFSR is taken as n then the repetition rate of the PN sequence generated will be 2n-1. The repetition rate is 2n-1 because if the contents of all the registers are zero then the shift registers won’t change their states. So the condition of all zeros in the PN sequence output is forbidden.

In our experiment the LFSR length is taken as 15 and hence the repetition rate of the PN sequence is 215-1 = 32767. The initial seed is taken as 110000000100001 and the tap positions are taken as 5,7,8,13,15. From the generated sequence any 32 length sequence can be picked up as spreading codes. The above LFSR is implemented in Matlab codes in our experiment.

# Mixed Spreading Code

Mixed codes are generated to combine the advantages of both Orthogonal and non-orthogonal codes. This is generated by multiplying the orthogonal and non-orthogonal codes.

# Near - Far Problem

The main problem with CDMA is the Near-Far effect. Consider a receiver and two transmitters; one close to the receiver; the other far away. If both transmitters transmit simultaneously and at equal powers, then the receiver will receive more power from the nearer transmitter than the farther transmitter. This makes the farther transmitter more difficult, if not impossible, to be understood. Since the signal from one transmitter is the noise for the other transmitter, the Signal-to-noise ratio

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

(SNR) for the nearer transmitter is much higher. If the nearer transmitter transmits a signal of higher power than the farther transmitter, then the SNR for the farther transmitter may be below the detectable level and the farther transmitter may look as if that it didn’t transmit at all. This effectively jams the communication channel. In CDMA systems or other cellular phone-like networks, this is commonly solved by dynamic output power adjustment of the transmitters by the base stations. That is the closer transmitters use less power so that the SNR for all transmitters at the receiver is roughly the same.

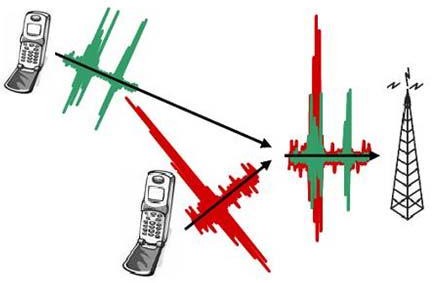


Fig 3: Near- Far problem in CDMA

This near-far problem is actually an uplink problem in reality. But in this experiment it is assumed as a downlink problem for the ease of implementation. Single Base station transmits the data at different powers to the two Users and thus the effect of one user data on other user is studied. The constellation plots for the two users are provided for ease of understanding of this phenomenon.

# Effect of delay in spreading the data

If the signal from one user arrives little delayed than the other user, the orthogonality between user1 and user2 will be lost. When a CDMA receiver de-spreads a signal, it effectively computes the cross- correlation between the signal and a locally generated PN sequence. If this PN sequence is identical to the one used to spread the signal at the transmitter (ie. the message intended for the receiver) cross-correlation computations restore the original information. Otherwise, due to

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

nonzero cross-correlation, some part of the other user data affects the desired user data depending on the power levels of the users.

This effect of delay is studied for three types of spreading codes discussed earlier. The delayed arrival of one user with respect to the other user is done by introducing delay between two users. It is observed that the mixed codes have a better constellation plot than the other two codes.

# C:\Users\jimesh\Desktop\WC-SEm-7\1.pngSCILAB Code Implementation Transmitter

* Random data to be transmitted for User1 and User2 are generated.
* Random data of User1 and User2 are QAM modulated.
* The QAM modulated User1 and User2 data are convolved with their corresponding spreading codes.
* The convolved data of User1 and User2 are RC pulse shaped.
* Sync sequence is generated and inserted in pulse shaped User1 data.
* The Pulse shaped data of User1 and User2 are multiplied with their corresponding powers.
* Two user data are added and then upsampled.

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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

WiCOMM-T.

# Receiver

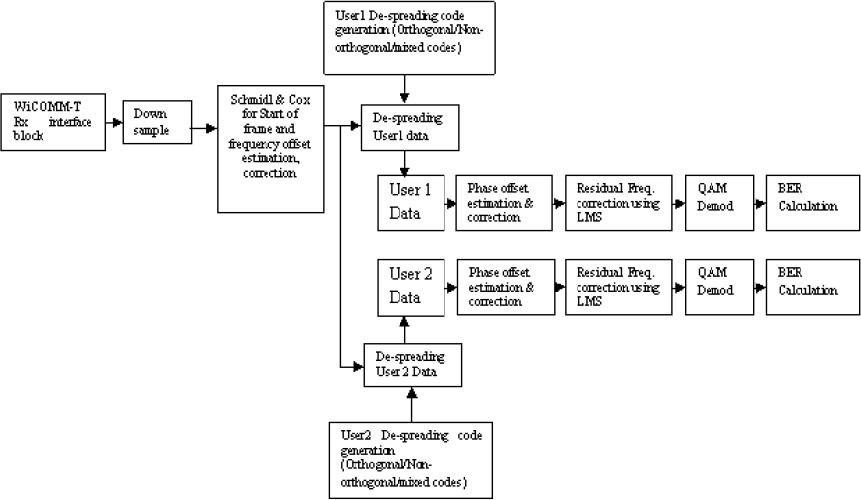
1. The samples are received from the WiCOMM-T Rx interface block
2. The received samples are down sampled
3. The start of the frame information is found out using Schmildl & Cox. Frequency offset in the down sampled signals are estimated using Schmidl Cox. and the estimated offset is corrected.
4. These frequency offset corrected samples are de-spreaded using User1 and User2 de-spreading codes
5. The phase offset in the de- spreaded data is estimated and corrected.
6. Then the residual frequency error is corrected by using LMS algorithm.
7. The residual frequency offset corrected data are QAM demodulated
8. BER is calculated for the QAM demodulated data

Fig : Block diagram of CDMA Receiver

# Procedure

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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. The following are the default values used in this experiment.

Default values are given below:

Chip length = 32 Samples per symbol = 8 Data length = 12

No of frames = 10

Sync sequence length = 800 Spreading code : Orthogonal Delay : zero

User Power level : User1 as one & User2 as zero.

* Connect WiCOMM-T in baseband loop back.
* Select CDMAPART1 from the Experiments popup menu in EXPERIMENT window.
* Choose the orthogonal spreading code.
* Keep the power of User1 (A1) as 1 and power of User2 (A2) as 0 for single user condition.
* Generate the transmitter modem sample.
* Transmit and receive the modem sample through WiCOMM-T.

1. Analyze the received modem samples.
2. Observe the various plots generated by MATLAB and tabulate the BER Value.
3. Introduce delay between two users.
4. Vary the power of User1 and User2 and repeat steps 5 to 8.
5. Change the spreading code to Non-orthogonal & Mixed code and repeat steps 3 to 10.
6. Connect WiCOMM-T in IF loop back.
7. Repeat steps 2 to 11 for IF loop back
8. Connect 2 WiCOMM-Ts in baseband level.
9. Repeat steps 2 to 11 for baseband communication
10. Connect the 2 WiCOMM-Ts in IF level
11. Repeat steps 2 to 11 for IF communication
12. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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\CDMAPART1\REF\_Data’ directory should be copied to receiver ‘C:\WiCOMM-T\EXPERIMENTS\CDMAPART1

\REF\_Data’ directory since receiver Matlab code refers ‘data1.bin’& ‘data2.bin’ file for finding pilot symbols & BER calculations.

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# EXPERIMENT: 8

**Aim**: To stimulate wireless channel including Rayleigh Fading using SCILAB.

# 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.

*//----------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=rand(1, N);

y = rand(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=histplot(r,range);

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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)

histplot(theta); *//Plot histogram of the phase part*

*//Approximate the histogram of the phase part to a nice PDF curve*

[counts.range] =histplot(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

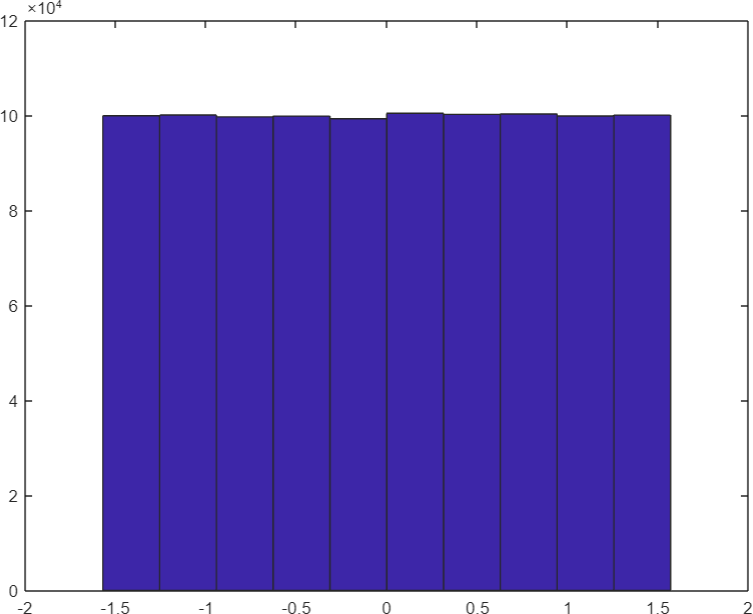
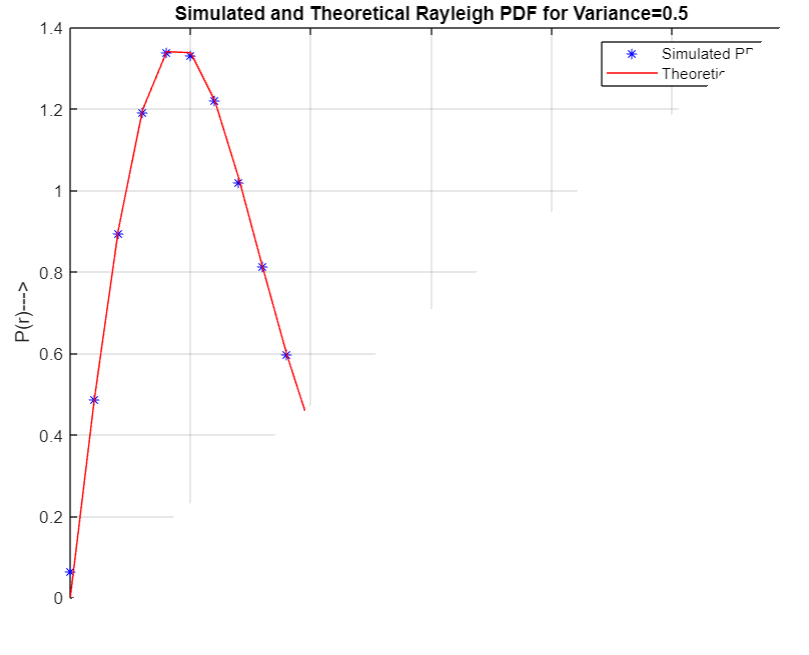
plotHandle=plot(range, approxPDF,'r'); set(plotHandle, 'Line Width', 3.5); axis([-2 2 0 max(approxPDF)+0.2]) hold off

title('Simulated PDF of Phase of Rayleigh Distribution'); xlabel('\theta --->');

ylabel('P(\theta) --->'); grid;

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# Stimulation Waveform



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

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# EXPERIMENT: 9

**Aim**: To stimulate Orthogonal Frequency Division Multiplexing (OFDM) using SCILAB.

# 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.

*////scilab simulation code*

*//code for OFDM signal transmission and reception in AUGN channel*

*//code*

n=256; *//Number of bits to process*

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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 (1^j\*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*

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

z=fft(z, nmax); *//FFT*

zsym=demodulate(modem.qamdemod (M), z); *// demodulation of bandpass*

z=de2bi (zsym, 'left-msb'); *//Convert integers to bits.*

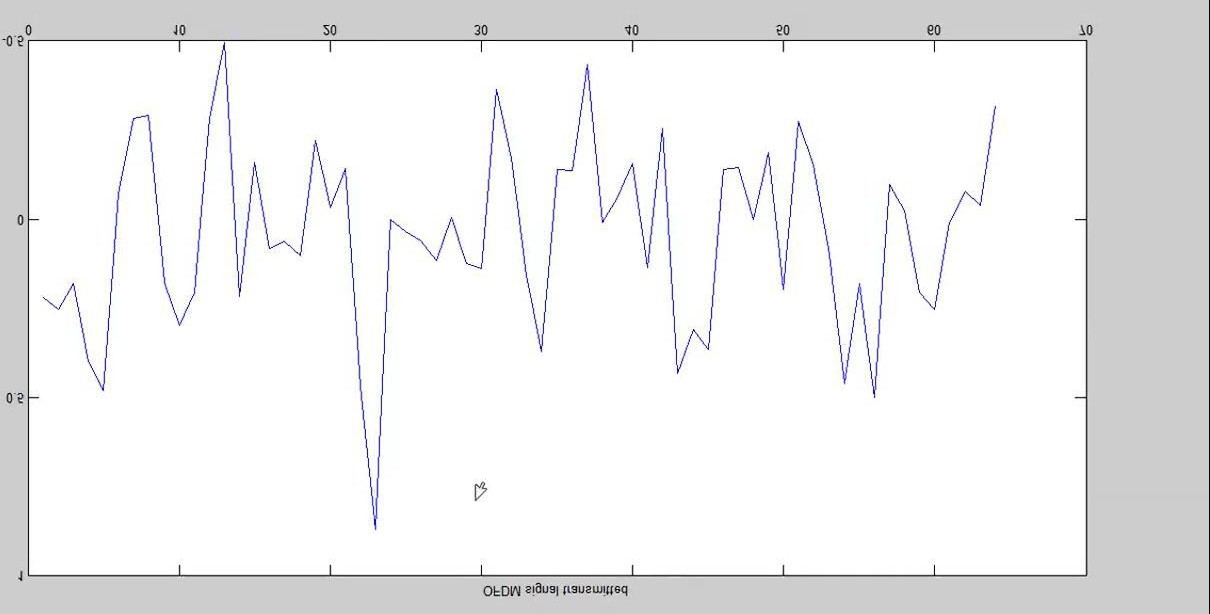
z = reshape (z.',prod (size (z)),1); *//matrix to vector conversion*

[noe, ber] = biterr (x,z); *//BER calculation figure;*

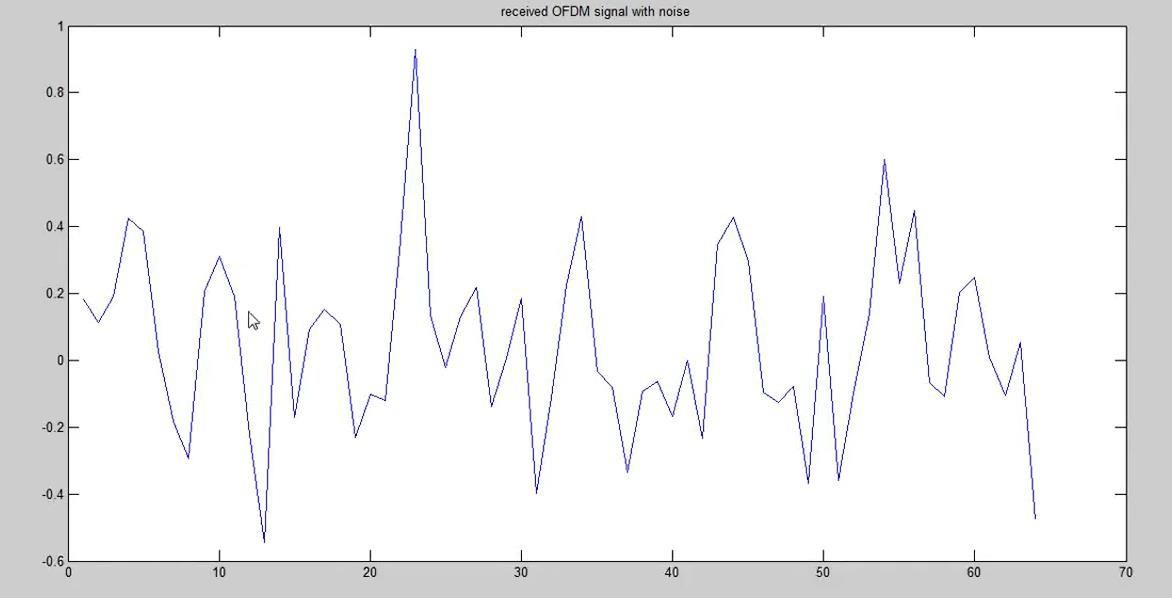
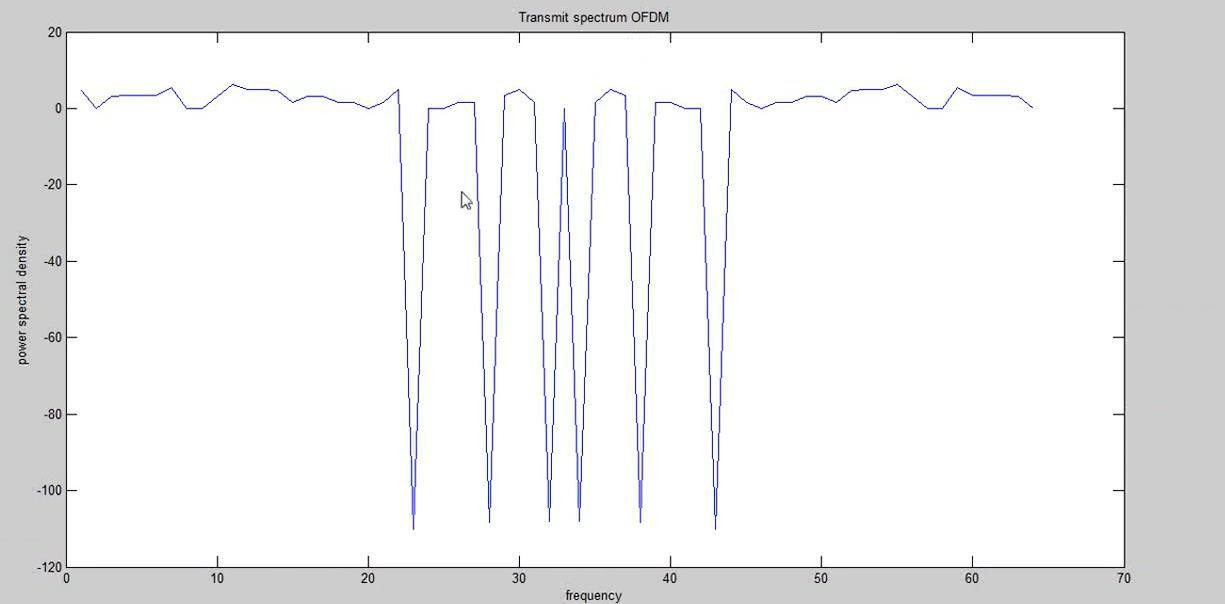
figure;

subplot (211); stem (x (1:256)) ;title('Original Message'); subplot (212); stem (z (1:256)) ;title ('recovered Message');

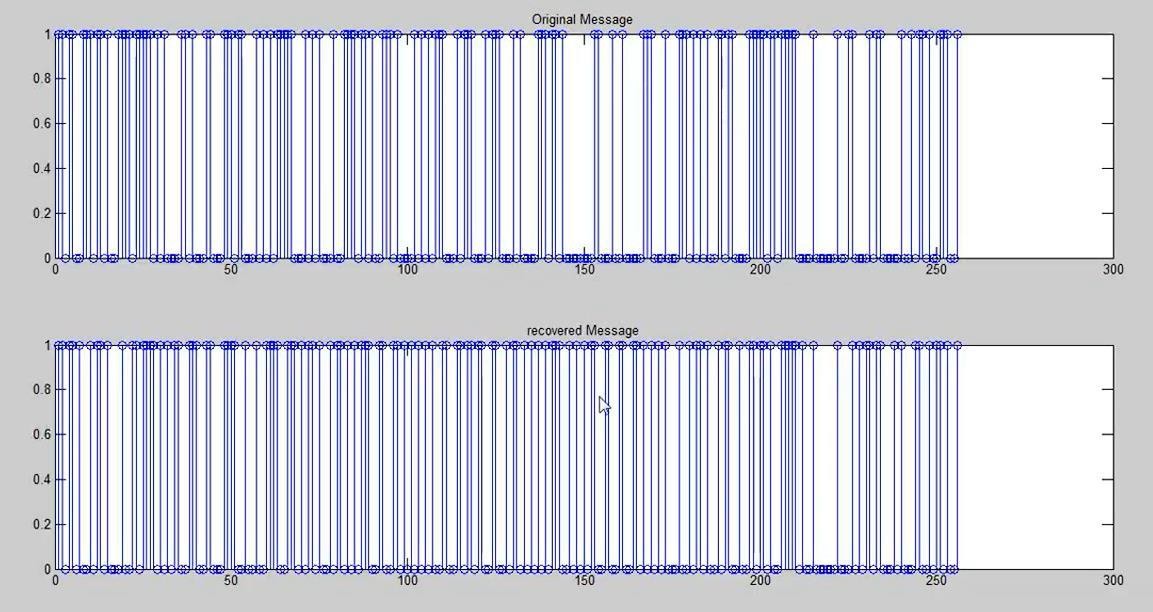
# Stimulation Waveform



1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**



1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**



# Result:

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

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# EXPERIMENT: 10

**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

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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.

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

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

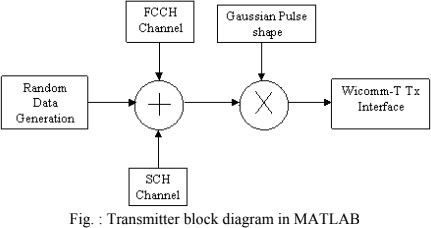
Carrier phase offset estimation is done with the help of FCCH channel. The received FCCH channel, previously identified through the frame synchronization, is decimated by a factor of 8 and the sequence S4(n) is chosen [i.e. the samples S(4), S(12), S(20) are selected]. Deterministic autocorrelation is 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.

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# SCILAB 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

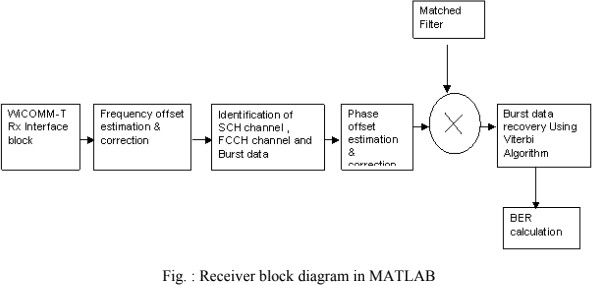
through the WiCOMM-T.



1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**

# 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.



1. **Enrollment Number:** 190310116009 **Name :Trusha Gondaliya**

# 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.

1. Observe various plots generated by SCILAB.
2. Connect WiCOMM-T in IF loop-back and repeat steps 2 to 4
3. 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 SCILAB 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 SCILAB code refers ‘bits.bin’ file for synchronization & BER calculation.

# Result:

Thus, the experiment was performed successfully.

1. **Enrollment Number:** 190310116009 **Name: Trusha Gondaliya**