**Tribhuvan University**

**Bhaktapur Multiple Campus**

**Dudhpati-17, Bhaktapur**

**Wireless Networking**

**“Lab Report”**

****

11/29/2021

**Submitted To: Submitted By:**

**Department of B.Sc.CSIT Name: Arjun Shrestha**

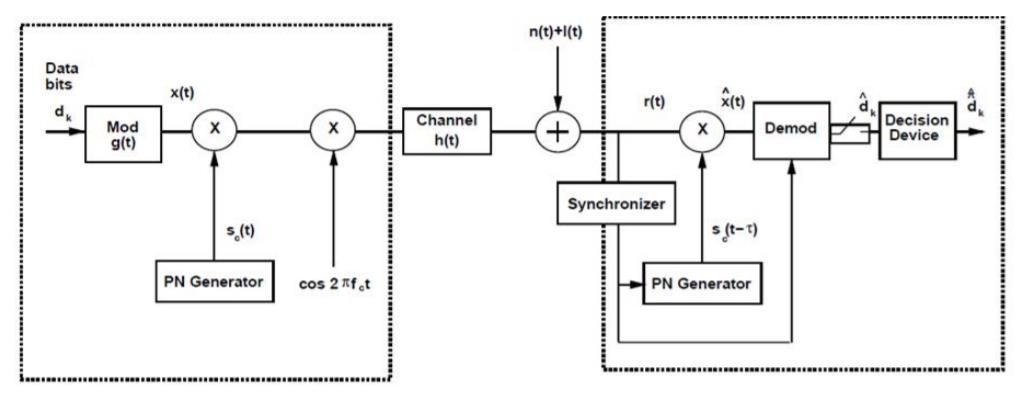
**Bhaktapur Multiple Campus Roll No: Six (06)**

**IMPLEMENTATION OF DIRECT SEQUENCE SPREAD SPECTRUM(DSSS)**

**INTRODUCTION:**

**A. Introduction**

The aim of our project is to make end to end simulation of Direct Sequence Spread Spectrum (DSSS) technique. It’s a technique that message signal is multiplied by a pseudorandom sequence to spread message bandwidth so energy of message signal expand a wider spectrum, and it appears as noise. The motivation of DSSS technique comes from channelcapacity theorem. It says that we can have good communication performance by increasing bandwidth even when signal-to-noise power is low. [1] DSSS increases transmit signal bandwidth to remove intersymbol interference (ISI) and narrowband interference. It provides security of transmission because receiver must know pseudo-random sequence to get transmitted data signal. Otherwise, receiver can’t detect original message signal, and it only sees transmitted signal as a noise. This property of DSSS technique make it suitable to use for military communication systems. DSSS also enables to usage of a bandwidth by multiple users because each user multiply its message signal by different pseudo-random sequence, and a receiver gets its message signal if it has transmitter pseudo-random sequence. The process of DSSS is shown below.



**Figure 1:** *Block diagram of Spread Spectrum System* [2]

**B. Experimental Work**

In our project, end to end simulation of DSSS modulation started with generation of message signal at transmitter. We generated 8 bits message signal for better understanding of given figures below. Each bit has 0.1 seconds bit duration. Then, we generated pseudorandom code that has 4 times higher frequency than message signal. This means that each message bit coded by 4 randomly generated bit at DSSS signal. To represent binary data, we used non-return-to-zero (NRZ) line code scheme because we chose BPSK modulation to transmit our signal. BPSK modulation has 180 degree phase shift between carriers of binary 0 and 1 so binary-1 represented as 1 while binary-0 represented as -1. We obtained DSSS signal with multiplying message signal and pseudo-random code. The receiver must be aware of this pseudo-random code. Otherwise, it can’t get transmitted message signal, and transmitted signal appears as a noise at receiver. Figure 2 shows generated 8 bits message signal, pseudo-random code and modulated DSSS signal.

**Lab Sheet 1:**

**Implement DSSS and line coding in MATLAB or equiv. tool**

**MATHLAB**

**Theory:**

MATLAB is a software package for high-performance mathematical computation, visualization, and programming environment. It provides an interactive environment with hundreds of built-in functions for technical computing, graphics, and animations. MATLAB stands for Matrix Laboratory. MATLAB was written initially to implement a simple approach to matrix software developed by the LINPACK (Linear system package) and EISPACK (Eigen system package) projects. MATLAB is a modern programming language environment, and it has refined data structures, includes built-in editing and debugging tools, and supports object-oriented programming. MATLAB is Multi-paradigm. So, it can work with multiple types of programming approaches, such as Functional, Object-Oriented, and Visual.

The following are some of the basic comments used in MATHLAB:

clc Clears command window.

clear Removes variables from memory.

exist Checks for existence of file or variable.

global Declares variables to be global.

help Searches for a help topic.

cd Changes current directory.

date Displays current date.

delete Deletes a file.

disp Displays contents of an array or string.

fscanf Read formatted data from a file.

format Controls screen-display format.

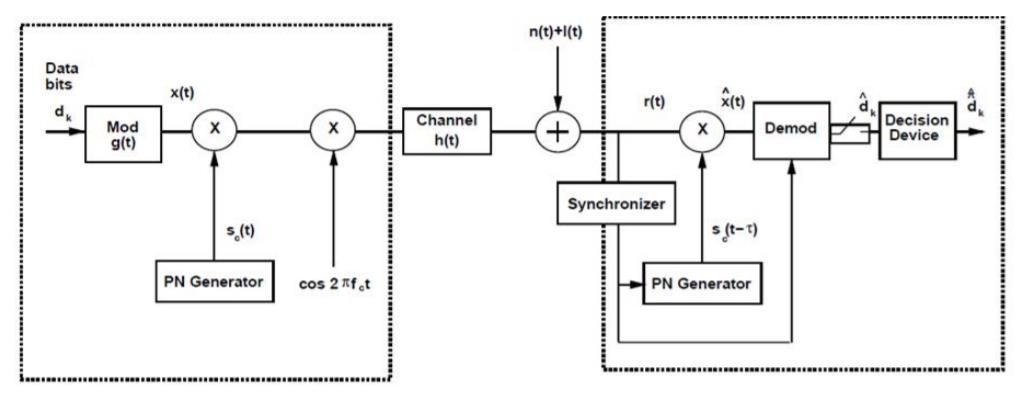
fprintf Performs formatted writes to screen or file.

input Displays prompts and waits for input.

**DSSS**

**Theory:**

The Direct Sequence Spread Spectrum (DSSS) is a spread-spectrum modulation technique primarily used to reduce overall signal interference in telecommunication. The Direct Sequence Spread Spectrum modulation makes the transmitted signal wider in bandwidth than the information bandwidth. In DSSS, the message bits are modulated by a bit sequencing process known as a spreading sequence. This spreading-sequence bit is known as a chip. It has a much shorter duration (larger bandwidth) than the original message bits.DSSS increases transmit signal bandwidth to remove intersymbol interference (ISI) and narrowband interference. It provides security of transmission because receiver must know pseudo-random sequence to get transmitted data signal. Otherwise, receiver can’t detect original message signal, and it only sees transmitted signal as a noise. This property of DSSS technique make it suitable to use for military communication systems. DSSS also enables to usage of a bandwidth by multiple users because each user multiply its message signal by different pseudo-random sequence, and a receiver gets its message signal if it has transmitter pseudo-random sequence. The process of DSSS is shown below.



**Figure 1:** *Block diagram of Spread Spectrum System*

**Program/Code in MATHLAB:**

clear all;

%% parameters

Fs = 1000;

fc = 100;

fp = 4;

bit\_t = 0.1;

%% message generation with BPSK

m = [0 0 1 1 1 1 0 0];

for bit = 1:length(m)

if(m(bit)==0)

m(bit) = -1;

end

end

message = repmat(m,fp,1);

message = reshape(message,1,[]);

%% PN generation and multiply with message

pn\_code = randi([0,1],1,length(m)\*fp);

for bit = 1:length(pn\_code)

if(pn\_code(bit)==0)

pn\_code(bit) = -1;

end

end

DSSS = message.\*pn\_code;

%% create carrier and multipy with encoded sequence

t = 0:1/Fs:(bit\_t-1/Fs);

s0 = -1\*cos(2\*pi\*fc\*t);

s1 = cos(2\*pi\*fc\*t);

carrier = [];

BPSK = [];

for i = 1:length(DSSS)

if (DSSS(i) == 1)

BPSK = [BPSK s1];

elseif (DSSS(i) == -1)

BPSK = [BPSK s0];

end

carrier = [carrier s1];

end

%% demodulation

rx =[];

for i = 1:length(pn\_code)

if(pn\_code(i)==1)

rx = [rx BPSK((((i-1)\*length(t))+1):i\*length(t))];

else

rx = [rx (-1)\*BPSK((((i-1)\*length(t))+1):i\*length(t))];

end

end

demod = rx.\*carrier;

result = [];

for i = 1:length(m)

x = length(t)\*fp;

cx = sum(carrier(((i-1)\*x)+1:i\*x).\*demod(((i-1)\*x)+1:i\*x));

if(cx>0)

result = [result 1];

else

result = [result -1];

end

end

pn\_codeWrong = randi([0,1],1,length(m)\*fp);

resultWrong = [];

rx2 =[];

for i = 1:length(pn\_code)

if(pn\_codeWrong(i)==1)

rx2 = [rx2 BPSK((((i-1)\*length(t))+1):i\*length(t))];

else

rx2 = [rx2 (-1)\*BPSK((((i-1)\*length(t))+1):i\*length(t))];

end

end

demod2 = rx2.\*carrier;

for i = 1:length(m)

x = length(t)\*fp;

cx = sum(carrier(((i-1)\*x)+1:i\*x).\*demod2(((i-1)\*x)+1:i\*x));

if(cx>0)

resultWrong = [resultWrong 1];

else

resultWrong = [resultWrong -1];

end

end

message1 = repmat(result,fp,1);

message1 = reshape(message1,1,[]);

message2 = repmat(resultWrong,fp,1);

message2 = reshape(message2,1,[]);

%% Draw original message, PN code , encoded sequence on time domain

pn\_size = length(pn\_code);

tpn = linspace(0,length(m)\*bit\_t-bit\_t/fp,pn\_size);

tm = 0:bit\_t/fp:length(m)\*bit\_t-bit\_t/fp;

figure

subplot(311)

stairs(tm,message,'linewidth',2)

title('Message bit sequence')

axis([0 length(m)\*bit\_t -1 1]);

subplot(312)

stairs(tpn,pn\_code,'linewidth',2)

title('Pseudo-random code');

axis([0 length(m)\*bit\_t -1 1]);

subplot(313)

stairs(tpn,DSSS,'linewidth',2)

title('Modulated signal');

axis([0 length(m)\*bit\_t -1 1]);

figure

subplot(311)

stairs(tm,message,'linewidth',2)

title('Message bit sequence')

axis([0 length(m)\*bit\_t -1 1]);

subplot(312)

stairs(tm,message1,'linewidth',2)

title('Received message using true pseudo-random code')

axis([0 length(m)\*bit\_t -1 1]);

subplot(313)

stairs(tm,message2,'linewidth',2)

title('Received message using wrong pseudo-random code')

axis([0 length(m)\*bit\_t -1 1]);

%% Draw original message, PN code , encoded sequence on frequency domain

f = linspace(-Fs/2,Fs/2,1024);

figure

subplot(311)

plot(f,abs(fftshift(fft(message,1024))),'linewidth',2);

title('Message spectrum')

subplot(312)

plot(f,abs(fftshift(fft(pn\_code,1024))),'linewidth',2);

title('Pseudo-random code spectrum');

subplot(313)

plot(f,abs(fftshift(fft(DSSS,1024))),'linewidth',2);

title('Modulated signal spectrum');

figure;

subplot(311)

plot(f,abs(fftshift(fft(BPSK,1024))),'linewidth',2);

title('Transmitted signal spectrum');

subplot(312)

plot(f,abs(fftshift(fft(rx,1024))),'linewidth',2);

title('Received signal multiplied by pseudo code');

subplot(313)

plot(f,abs(fftshift(fft(demod,1024))),'linewidth',2);

title('Demodulated signal spectrum before decision device ');

**Output:**

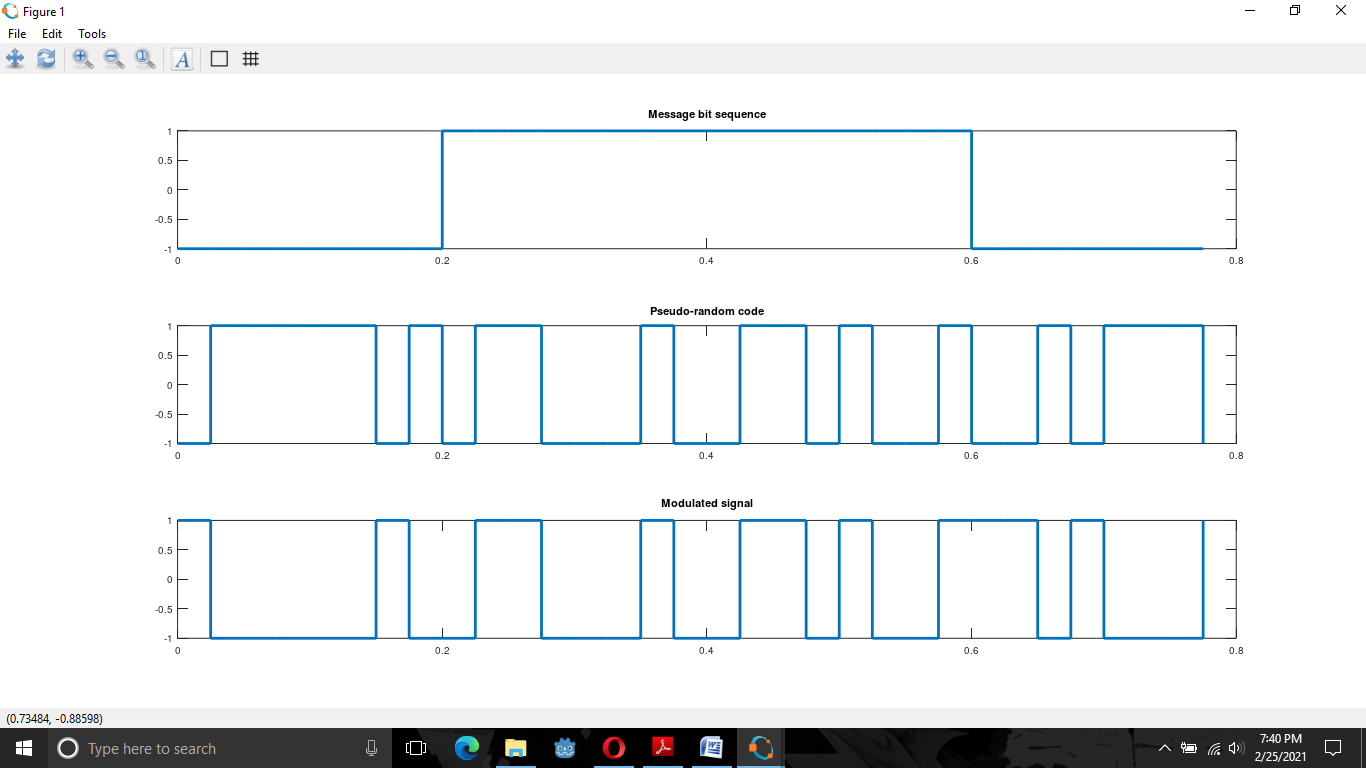


Fig 1 Message bit sequence, Pseudo random code and Modulated signal

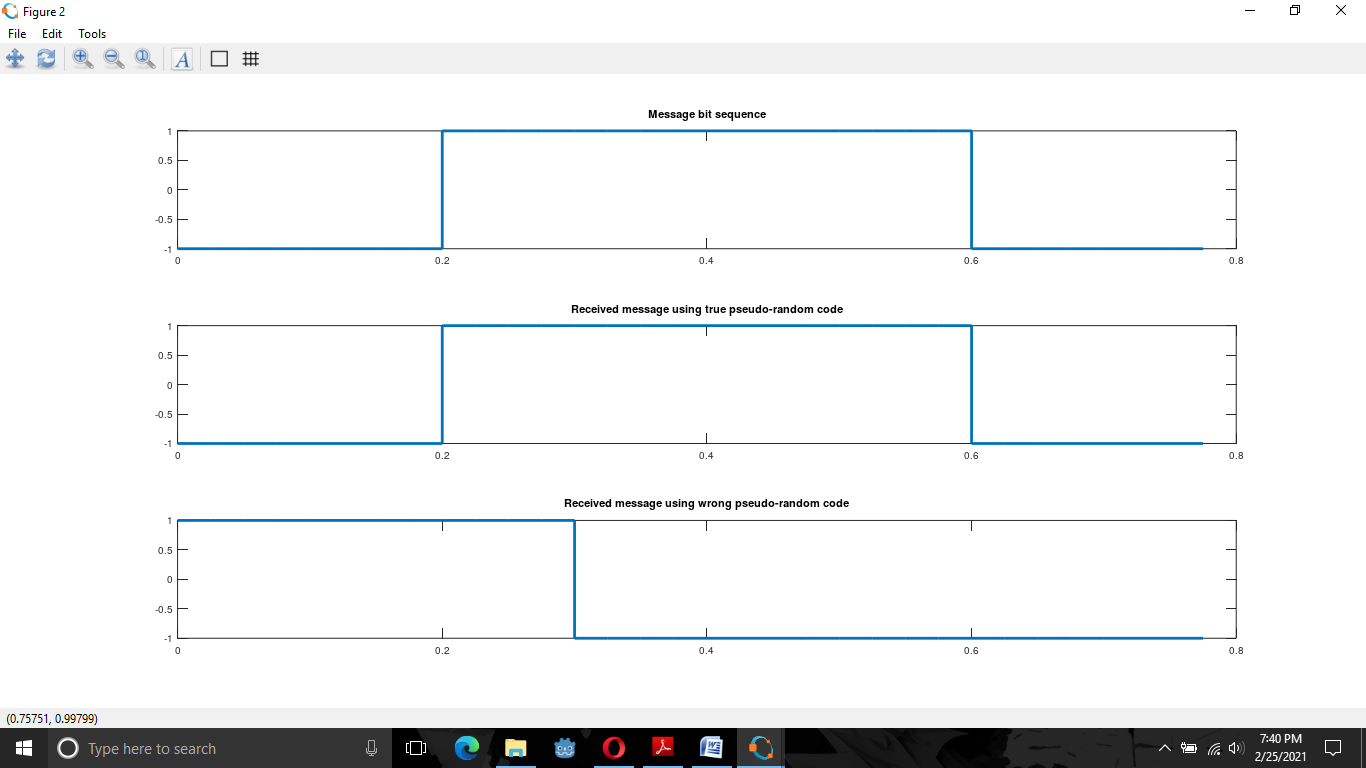


Fig 2 Draw original message, PN code , encoded sequence on time domain

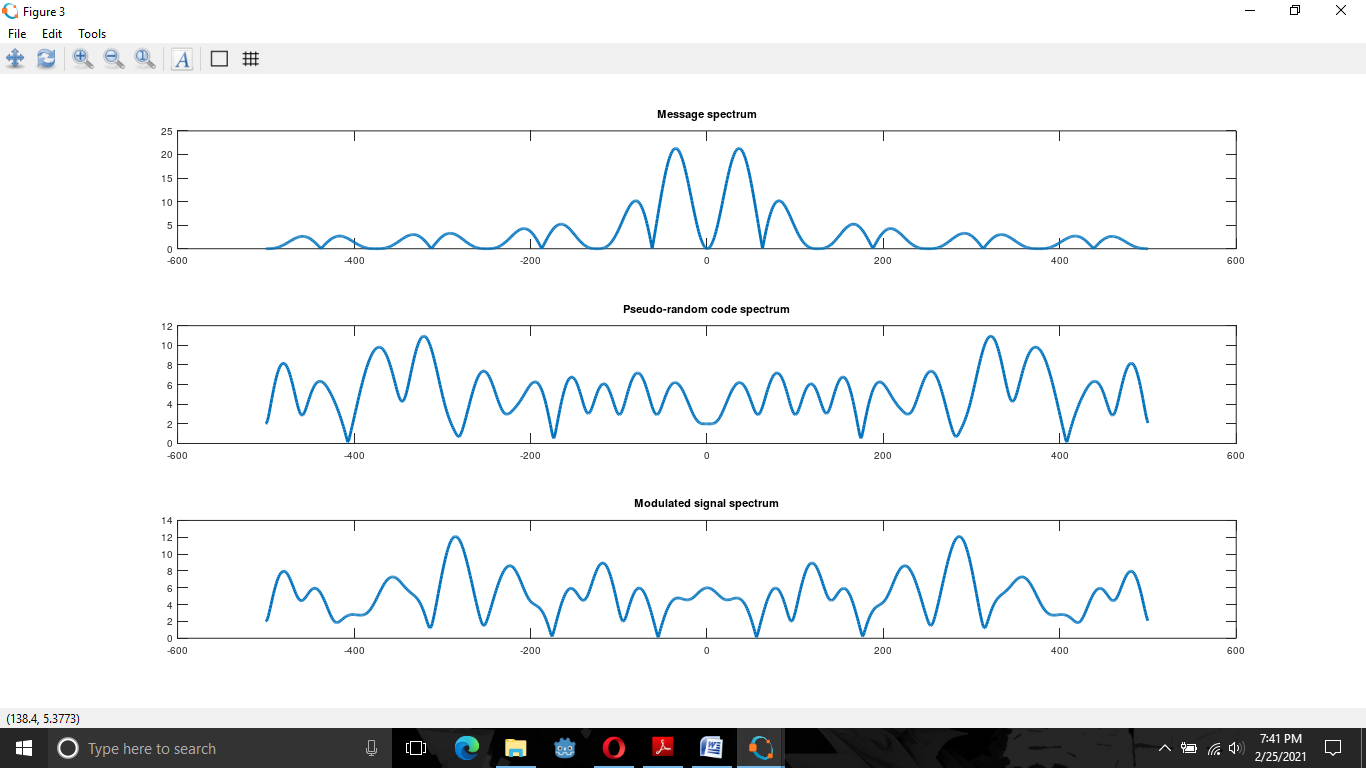


Fig 3: Message spectrum, Pseudo random code spectrum and Modulated signal spectrum

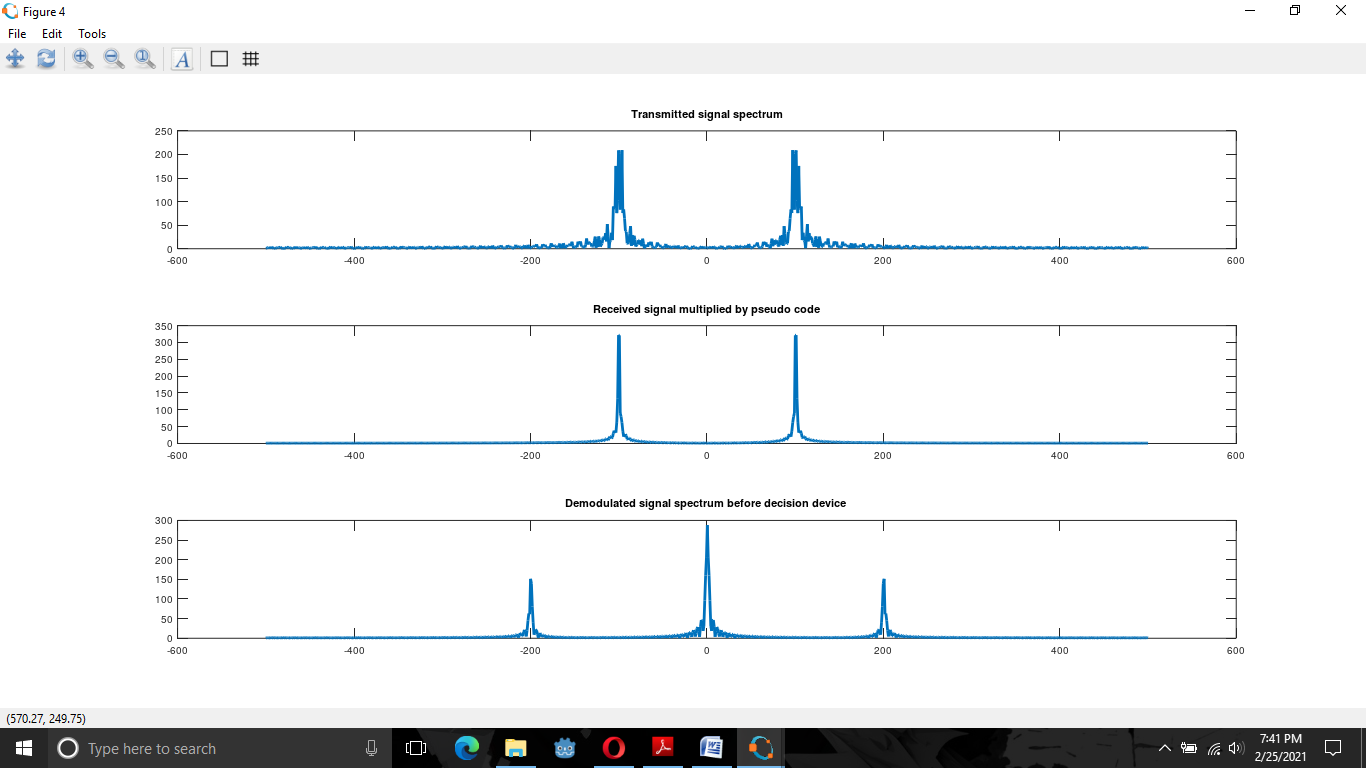


Fig 4 Draw original message, PN code , encoded sequence on frequency domain

**Line Coding**

**Theory:**

Data as well as signals that represents data can either be digital or analog. Line coding refers to the process of converting digital data into digital signals. Line coding is the process of converting digital data to digital signals. Whenever we transmit data it is in the form of digital signals, so with the help of line coding, we can convert a sequence to bits (or encoding) into a digital signal which then again converted into bits by the receiver (or can be said as decoded by the receiver). A line code is the code used for data transmission of a digital signal over a transmission line. This process of coding is chosen so as to avoid overlap and distortion of signal such as inter-symbol interference.

**Code Implementation:**

N=10

n=randi([0,1],1,N)

for m=1:N

if n(m)==1

nn(m)=1;

else

nn(m)=0;

endif

endfor

i=1

t=0:0.01:length(n);

for j=1:length(t)

if t(j)<=i

y(j)=nn(i);

else

i=i+1

y(j)=nn(i);

endif

endfor

plot(t,y);

axis([1,N,-2,2])

**Output:**

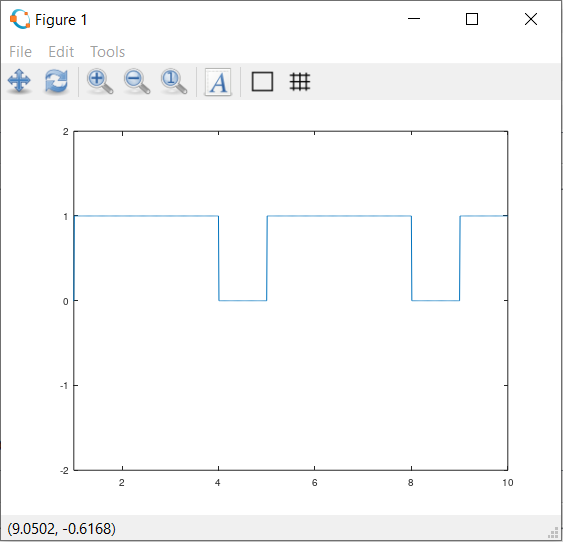
****

Fig. Line coding

**Lab Sheet 2:**

Analyze the performance of WiMAX/WiFi network using NetSim or equiv. tool.

**Netsim**

**Theory:**

A (netsim) or network simulator is software that predicts the behavior of a computer network. Since communication networks have become too complex for traditional analytical methods to provide an accurate understanding of system behavior, network simulators are used. In simulators, the computer network is modeled with devices,links, applications etc. and the network performance is reported. Simulators come with support for the most popular technologies and networks in use today such as 5G, Internet of Things (IoT), Wireless LANs, mobile ad hoc networks, wireless sensor networks, vehicular ad hoc networks, cognitive radio networks, LTE etc. NetSim is an end-to-end, full stack, packet level network simulator and emulator. It provides network engineers with a technology development environment for protocol modeling, network R&D and military communications. The behavior and performance of new protocols and devices can be investigated in an virtual network within NetSim at significantly lower cost and in less time than with hardware prototypes.

**WiFi**

**Theory:**

WiFi stands for Wireless Fidelity. WiFi It is based on the IEEE 802.11 family of standards and is primarily a local area networking (LAN) technology designed to provide in-building broadband coverage.Current WiFi systems support a peak physical-layer data rate of 54 Mbps and typically provide indoor coverage over a distance of 100 feet. WiFi offers remarkably higher peak data rates than do 3G systems, primarily since it operates over a larger 20 MHz bandwidth, but WiFiWiFi systems are not designed to support high-speed mobility.

One significant advantage of WiFi over WiMAX and 3G is its wide availability of terminal devices. A vast majority of laptops shipped today have a built-in WiFi interface. WiFi interfaces are now also being built into a variety of devices, including personal data assistants (PDAs), cordless phones, cellular phones, cameras, and media players.

The WiFi standards define a fixed channel bandwidth of 25 MHz for 802.11b and 20 MHz for either 802.11a or g networks. primary channel access mechanism is CSMA/CA

It has support for infrastructure and ad hoc mode of operation. Also is has support for authentication, roaming, security, etc standards are backward compatible

Performance of a WiFi link and system depends on signal quality, interference,etc

**Signal Quality:** Quality is how much interference there is between the satellite and your decoder. Signal quality is how good the information you are recieving actually is. It is how good the signal is that you receive or how good it will keep during rainy / cloudy conditions. It means that quality is how good the information that you receive, is. All signals hold reflections/ghosting or interference that we call noise. Noise will cause the loss of data. The noise will be low if the quality is high or good. Signal quality is the quality of the signal and not pixel not picture quality. If the the signal is suffering due to reflections/ghosting or interfernce from electrical installations switches/motors etc then the quality will be poor as these problems will cause the loss of data in the signal as the effect of them will block/overpower the signal and your box won't be able to "read" the true data. Without any of these problems the your quality will be high.

**Interference:** an interference is that which modifies a signal in a disruptive manner, as it travels along a communication channel between its source and receiver. The term is often used to refer to the addition of unwanted signals to a useful signal. Interference occurs when unwanted signals disrupt wireless communication, including the use of your television, radio etc. Interference may prevent reception altogether, may cause only a temporary loss of a signal, or may affect the quality of the audio or video produced by your equipment. Interference is the primitive nature of the wireless communication system. It is the case in which multiple transmission occurs many times simultaneously over a common communication medium.

**Implementation:**

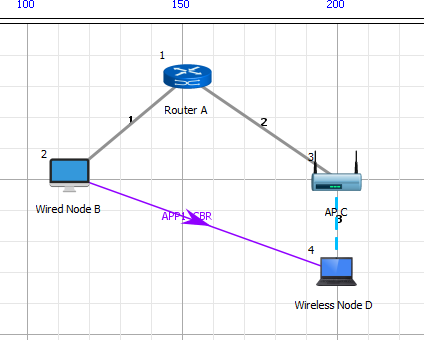
**Network Setup:** An IEEE 802.11n link between an access point and a client downlink UDP traffic between a server and the client is generated as below:

Fig. IEEE 802.11n network topology

IEEE 802.11n supports a fixed set of data rates among which we only use BPSK and QPSK both having coding rate of ½

|  |  |  |  |
| --- | --- | --- | --- |
| MCS | Modulation  Type | Coding  Rate | Data  Rate1 |
| 0 | BPSK | 1/2 | 7.2 |
| 1 | QPSK | 1/2 | 14.4 |

fig. Modulation type and it’s Data rate

**Rate Adaptation: Experiment Configuration**

|  |  |
| --- | --- |
| **WiFi Radio** | |
| Standard | 802.11n |
| Band | 2.4 GHz |
| Channel | 1 |
| Bandwidth | 20 MHz |
| Transmit Power | 20 dBm and 10 dBm |
| Rate Adaptation | False |

|  |  |
| --- | --- |
| **Wireless Channel and Link** | |
| Channel Characteristics | Path Loss Only |
| Path Loss Model | Log Distance |
| Path Loss Exponent | 3.5 |
| Mobility Model | Constant |
| AP - Client Distance | Variable |

|  |  |
| --- | --- |
| **Application Properties** | |
| Application | CBR |
| Transport Protocol | UDP |
| Packet Size | 1450 bytes |
| Interarrival Time | 116 microseconds |
| TCP | Disabled |

|  |  |
| --- | --- |
| **Miscellaneous** | |
| Wired Link Capacity  Wired Link Delay Simulation Time | 1 Gbps  10 microseconds  10 seconds |

**Performance Measure:**

1. Data rate (from Packet Trace):

DATARATE= PHY LAYER PAYLOAD∗8

(PHY LAYER END TIME −PHY LAYER ARRIVAL TIME −40)

1. Average throughput (from Packet Trace):

AVERAGE THROUGHPUT = APPLICATION BYTES RECEIVED ∗8.0

SIMULATION TIME

**Output:**

**Data Rate and Throughput with Distance**

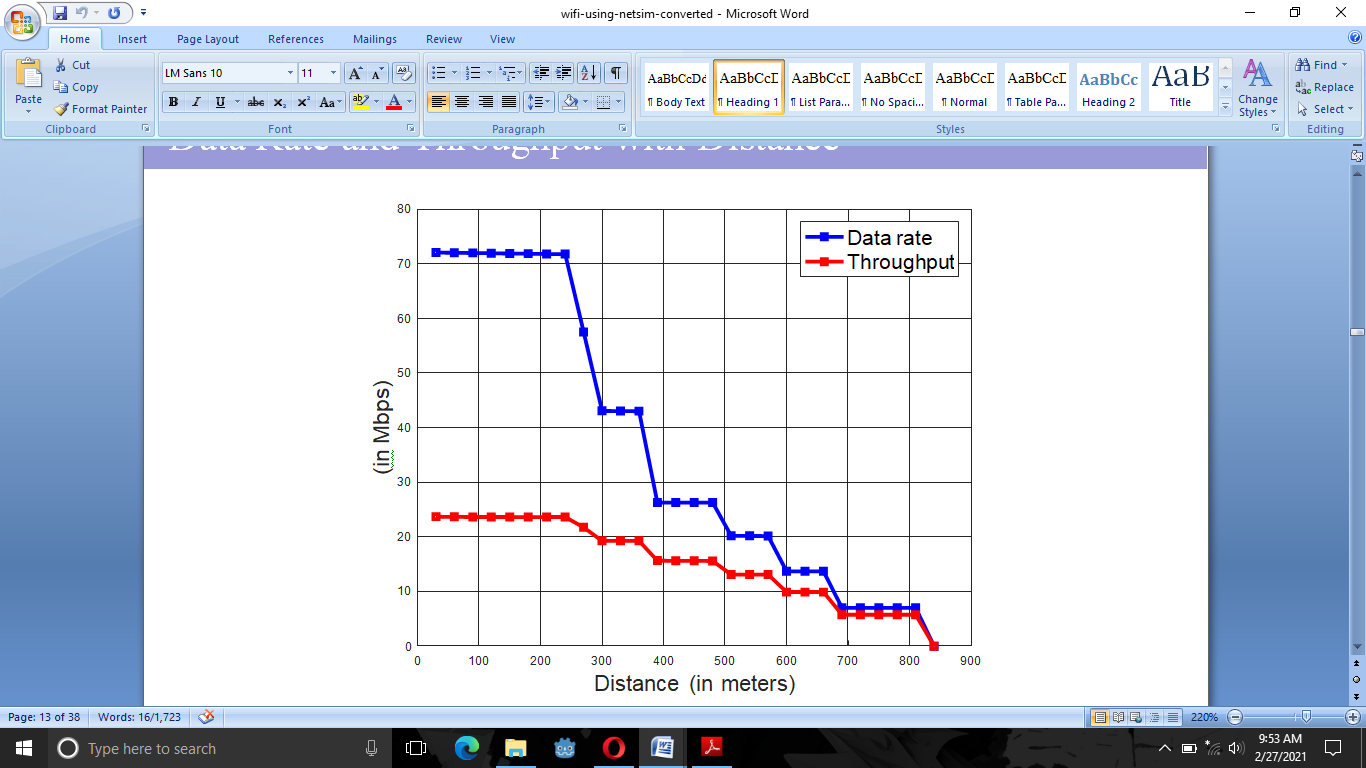
****

Fig. Relation between Data rate and throughput with distance

data rate and throughput decreases with distance data rates correspond to recommend rates of 802.11n

**Data Rate with Distance and transmit power**

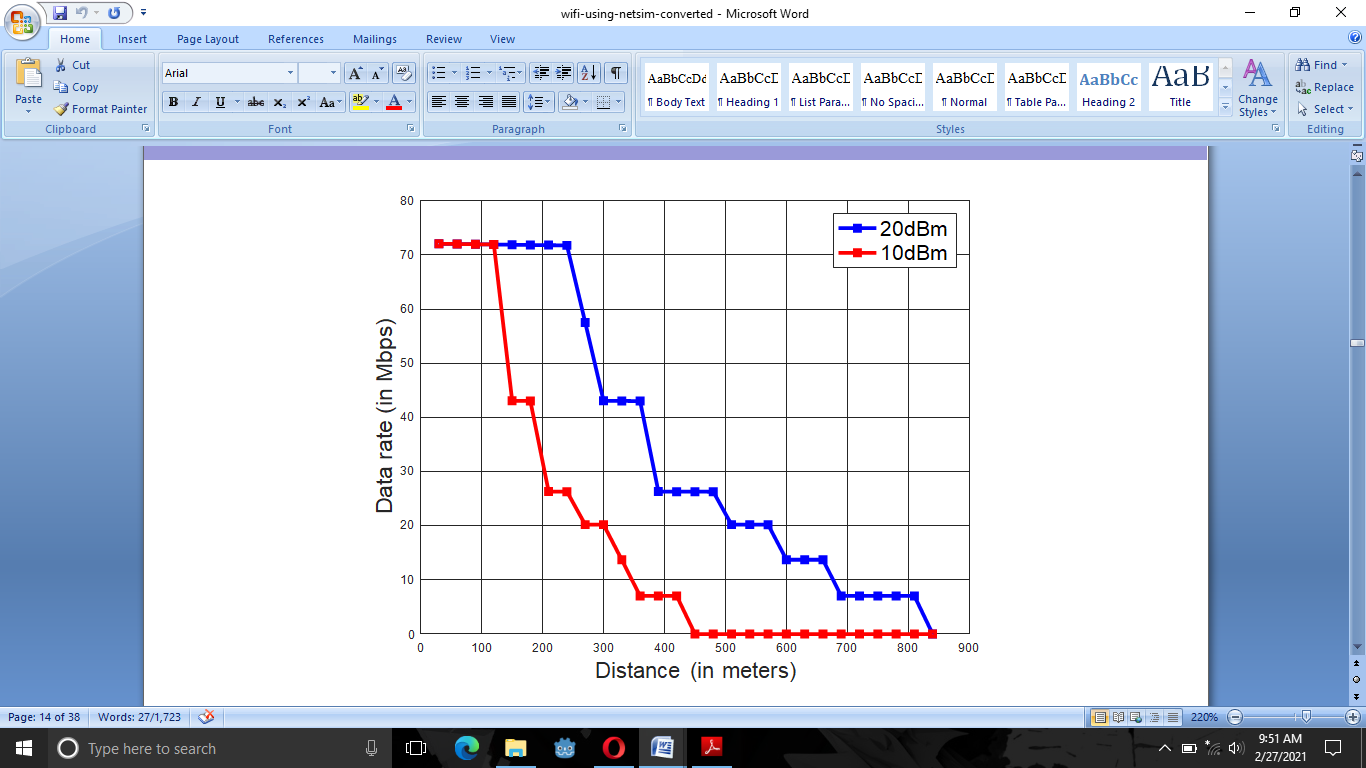
****

Fig. Relation between data rate with distance and transmit power

data rate decreases as transmit power decreases coverage (range) decreases as transmit power decreases

**Lab Sheet 3:**

Develop QPSK detector and understand the relation between BER and SNR.

**Theory:**

**Modulation:** modulation is the process of varying one or more properties of a periodic waveform, called the carrier signal, with a separate signal called the modulation signal that typically contains information to be transmitted. For example, the modulation signal might be an audio signal representing sound from a microphone, a video signal representing moving images from a video camera, or a digital signal representing a sequence of binary digits, a bitstream from a computer. The carrier is higher in frequency than the modulation signal. The purpose of modulation is to impress the information on the carrier wave, which is used to carry the information to another location.

In digital modulation, an analog carrier signal is modulated by a discrete signal. Digital modulation methods can be considered as digital-to-analog conversion and the corresponding demodulation or detection as analog-to-digital conversion. The changes in the carrier signal are chosen from a finite number of M alternative symbols (the modulation alphabet). Digital Modulation provides more information capacity, high data security, quicker system availability with great quality communication. Hence, digital modulation techniques have a greater demand, for their capacity to convey larger amounts of data than analog modulation techniques.

**SNR:** Signal-to-noise ratio (SNR or S/N) is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. SNR is defined as the ratio of signal power to the noise power, often expressed in decibels. Signal-to-noise ratio (SNR or S/N) is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. SNR is defined as the ratio of signal power to the noise power, often expressed in decibels.

If Vs = Vn, then S/N = 0. In this situation, the signal borders on unreadable, because the noise level severely competes with it. In digital communications, this will probably cause a reduction in data speed because of frequent errors that require the source (transmitting) computer or terminal to resend some packets of data.

A ratio higher than 1:1 (greater than 0 dB) indicates more signal than noise.

**BER:**