**LAB 07**

**Implementation of Spread Spectrum technique**

Objectives:

1. Implementation of Direct Sequence Spread Spectrum in MATLAB.
2. Implementation of the Frequency Hopping Spread Spectrum in MATLAB.

Introduction

There is a tradeoff that can be made when transmitting information through a channel. It was discovered by Claude Shannon and is described by a theorem given by bits/sec. What this means is that the channel capacity C [symbols/sec] can be increased either by increasing the bandwidth B [Hz] or increasing the SNR (NB: here, SNR is expressed in linear scale!). This is a theoretical limit and is independent of the type of modulation and other channel parameters.

One method of increasing the bandwidth is to implement spread spectrum techniques. There are two fundamental techniques that fall within the normal definition of spread spectrum: **direct sequence** and **frequency hopping**.

## Direct Sequence Spread Spectrum (DSSS)

In DSSS, the energy in a relatively narrowband information signal is spread over a much larger bandwidth.

Consider the following communication system:

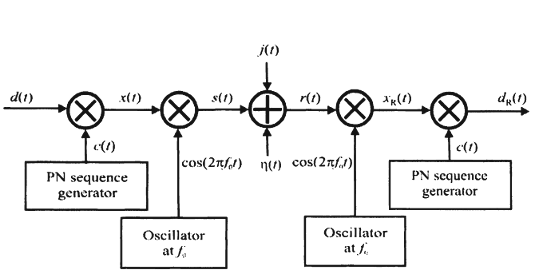


Figure 1. DSSS based communication system.

The *symbol j(t)* represents an interfering signal, such as a jammer while *η(t)* represents noise contributed by the communication channel.

A data sequence , is multiplied by a chip sequence , as illustrated in the following figure:

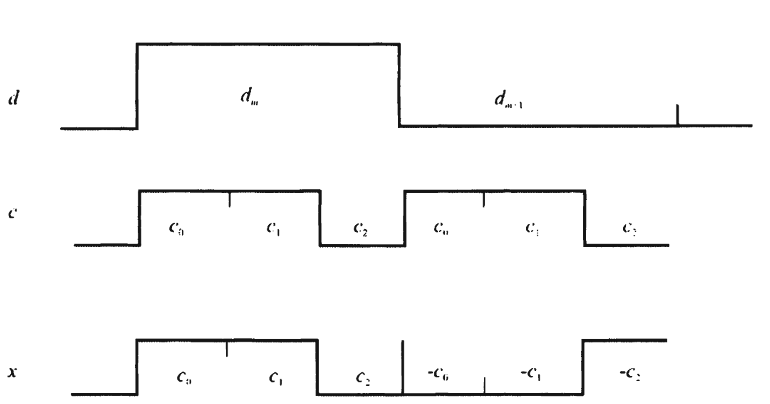


Figure 2 DSS data spreading

The chip rate, 1/Tc*,* is much higher than the data rate 1/Tb. Furthermore, the number of chips per data bit is an integer, *N = Tb/Tc.* The product *d(t)c(t)* multiplies the carrier signal *cos(2πf0t),* forming the transmitted signal:

 ,

Where S is the power in the signal; the energy per bit is thus Eb=STb, k = 0,1,2, ..., N -1, n is an integer, and nTb + kTc ≤ t < nTb + (k+ 1)Tc.

The processing gain of a spread spectrum system, denoted by Gp, is given by the ratio of the bandwidth of the spread signal to that of the non-spread signal. It represents the advantage of a spread spectrum system. Thus if the bandwidth of a non-spread signal is Bs and this signal is spread by some means to occupy a bandwidth of Bss*,* then .

Note that, in order to get the modulated and encoded signal to transmit, we use the XOR (exclusive OR) truth table to calculate the transmitted signal as depicted in the following example:

A diagram of a number and a line

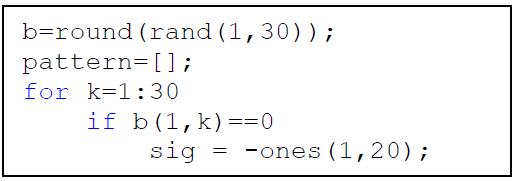
Description automatically generated with medium confidence

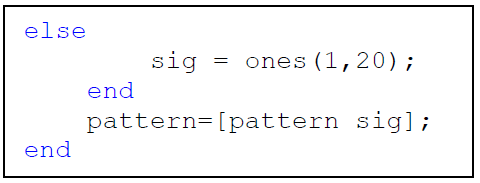
*PN code has many radio pulses that are way shorter than the original signal. The shorter this duration, the larger the bandwidth of the resulting transmitted signal.*

1. Implementation of Spread Spectrum communication model in MATLAB

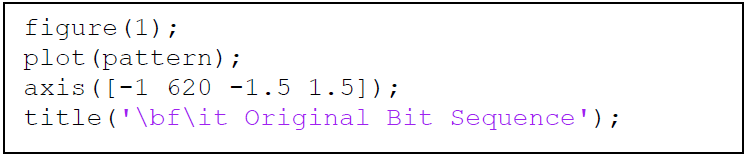
Now model a communication system using DS-SS:

1. Start Matlab.
2. Create a new Script named as: DSSS\_model
3. Let’s generate the bit pattern that we will be applying DSSS on. Execute the following code:





The bit pattern is generated so that each bit is actually 20 samples long.

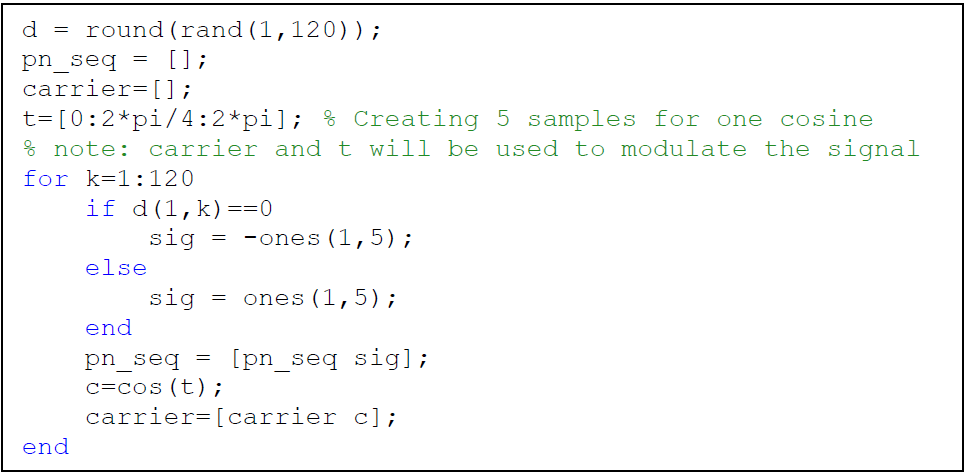
1. We can plot the generated pattern by executing:

The figure represents the randomly generated bit stream.

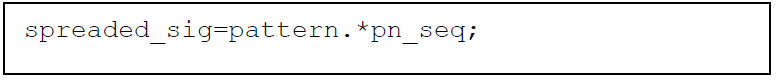
A screenshot of a cell phone

Description automatically generated

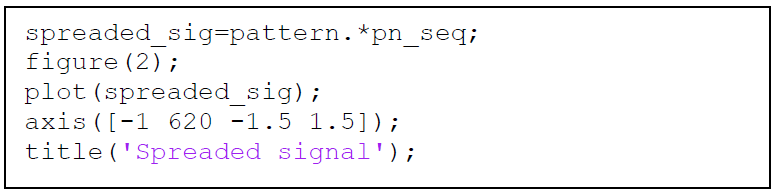
1. Now, let’s generate the pseudo-random bit pattern for spreading:



1. Spreading of the sequence:

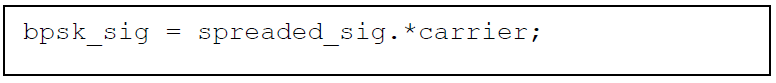


1. Let’s plot the spreaded sequence:

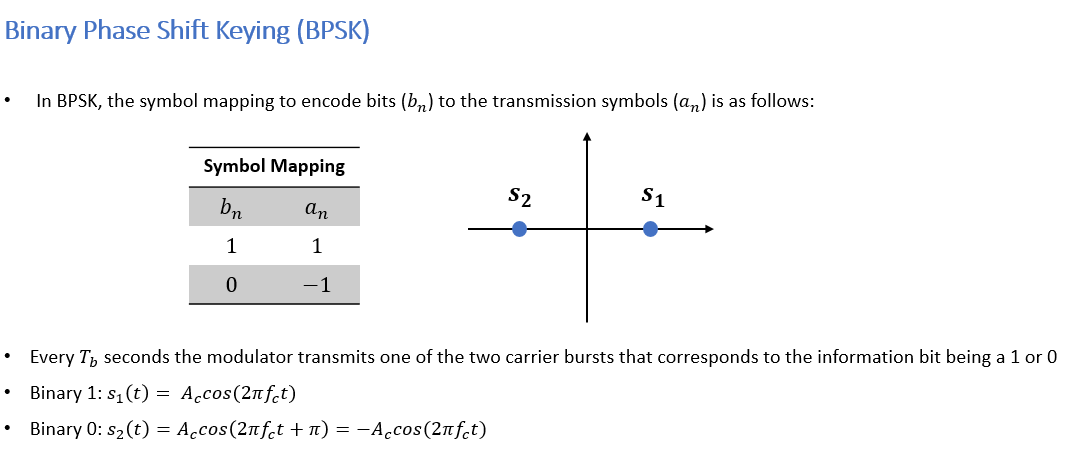


A screenshot of a cell phone

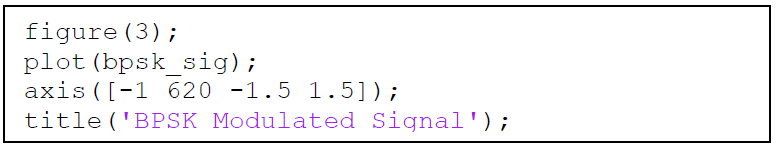
Description automatically generated

1. Let’s perform the BPSK modulation of the *spreaded\_sig,* Execute:

Regarding BPSK modulation:



After this brief intro of BPSK, you understand why we assign (-1) to 0 bits and (1) to 1 bits (in point 6).

1. We can plot the modulated signal by executing:

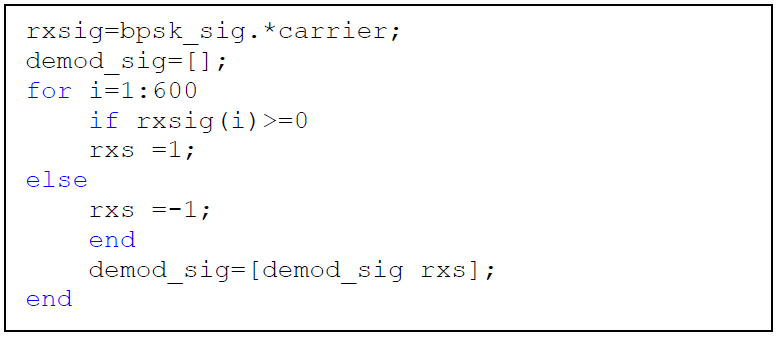
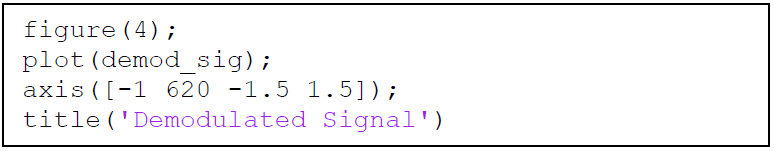
A screenshot of a cell phone

Description automatically generated

**DSSS at the transmitter side is done.**

Let’s go to the **receiver side**.

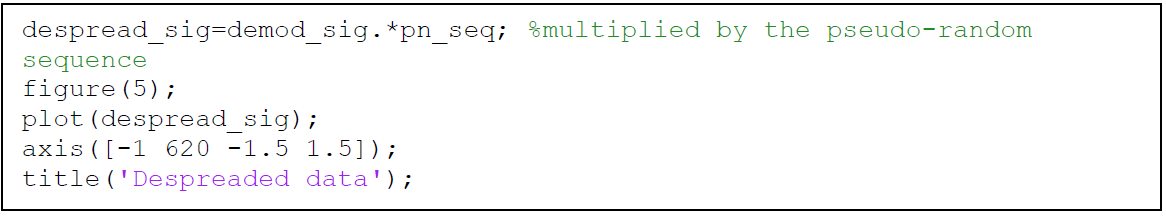
The receiver will receive bpsk\_sig, so we must perform BPSK demodulation and then dispreading after that we can compare the transmitted bits and the received bits to see if our system is working well.

1. demodulation of the received signal (bpsk\_sig):
2. Let’s plot the demodulated signal :

A screenshot of a cell phone

Description automatically generated

1. Let’s perform the de-spreading and plot it:



A close up of a piece of paper

Description automatically generated

After observing the figures, you must note that “Demodulated Signal” must be identical to “Spreaded Signal” and “Despreaded data” must be identical to “Original Bit Sequence”. This means that the communication system is modeled in a correct way.

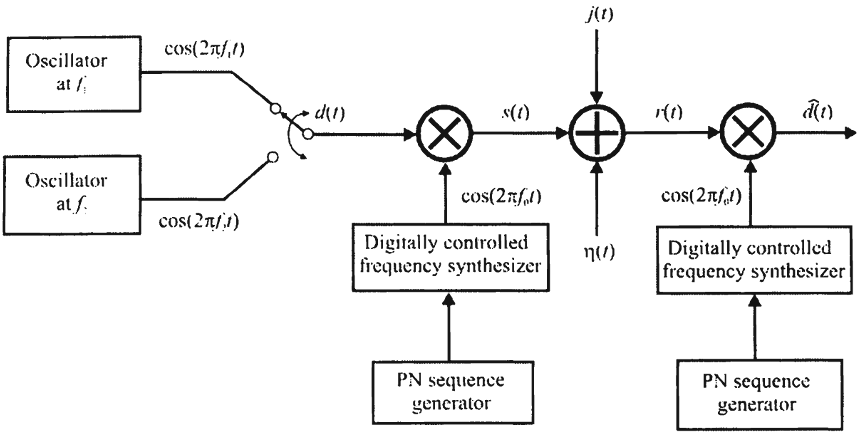
Exercise 1:

Repeat the procedure by generating a pattern consisting of 60 bits (instead of 30, refer to point 3) and each bit 40 samples long (instead of 20 as mentioned in point 3).

## Frequency Hopping Spread Spectrum (FHSS)

In the FHSS, the carrier frequency of the signal is changed periodically, or "hopped."

Consider the spread spectrum communication system shown in the following figure:

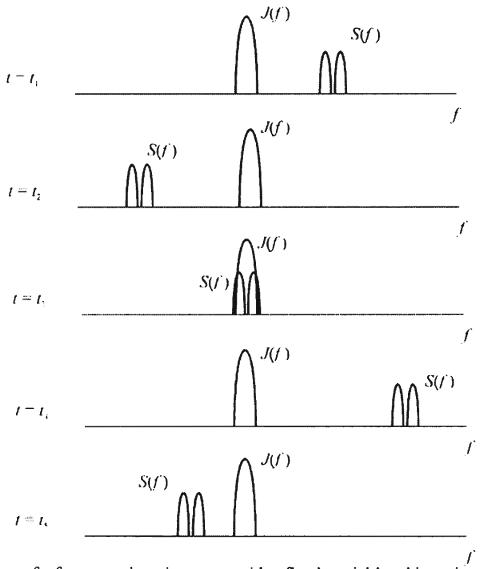
.

At the transmitter, the modulating signal is imposed by some modulation method on a carrier frequency that is varied on a regular, frequent basis. Again, amplifiers, filters, and other components are omitted from the diagram for clarity. A digitally controlled frequency synthesizer generates the carrier frequency. The PN sequence generator determines the particular frequency at any moment. This is known as an FHSS system.

The information signal *d(t)* can be digital or analog but here it is shown as BFSK. BFSK (Binary Frequency Shift Keying) is a popular form of modulation for frequency hopping. Although for FSK the two complementary frequencies f1 and f2 need not be close together, typically in the VHF frequency range they are two frequencies in the same 25-kHz channel. In many realistic applications of FHSS, noncoherent detection is employed for simplicity and economics. Therefore, even though the PN sequences must be synchronized for proper operation, the local oscillators are not phase-synchronized. Like other noncoherent communication systems, this typically results in about a 3 dB loss in performance.

At the receiver there is an equivalent frequency generator controlled by an equivalent PN sequence generator. When synchronized, the PN sequence generators change frequency control words simultaneously and in synchronization with each other. Synchronizing the PN sequence generators is typically accomplished by using a subset of the available frequency channels to which all of the communication devices tune when synchronization is required (such as when a new member joins the communication network). Synchronization information is always available on these channels.

The functions *j(t)* represents an interfering signal, such as a jammer. The advantage of FHSS systems in the presence of jammers and other interference is illustrated in the following figure:

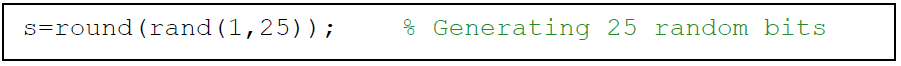


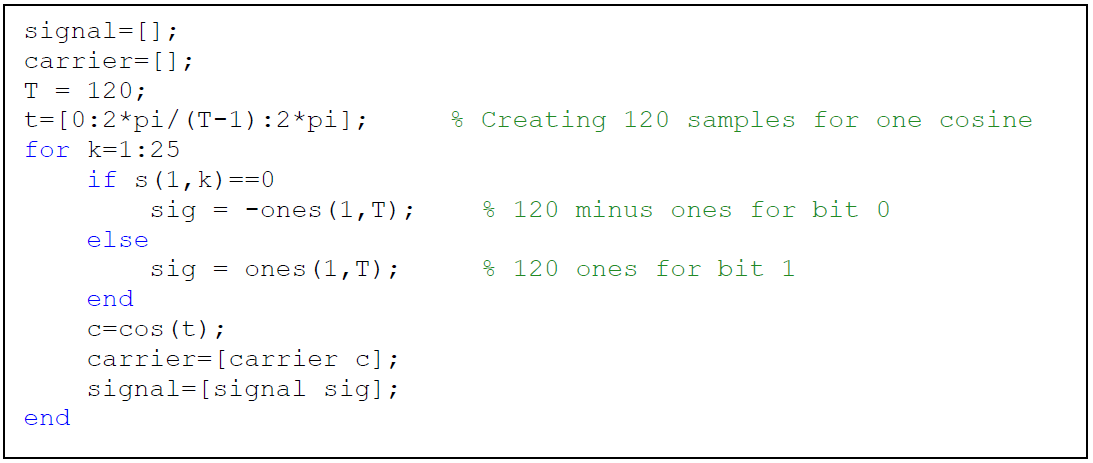
Here BFSK signals are transmitted, and the frequency of transmission is varied at the hop rate. For illustrative purposes, a narrowband jammer is shown, although jammer waveforms targeted against frequency hoppers would typically cover many more than just the one channel shown here. The hopping signal can hop into the jammer for the duration of the hop. In that case the information during that hop would typically be lost to the receiver. Normally coding, interleaving, and other forms of redundancy are built into the signaling scheme so that if the hopping signal hops into the jamming signal, the data is not lost. Of course, if the signal is digitized voice, some tolerance for lost data bits is inherent anyway. In any case, since the jammer is fixed in frequency, any lost information is only lost for the duration of that hop, and for prior and subsequent hop intervals, the information is transferred unimpeded.

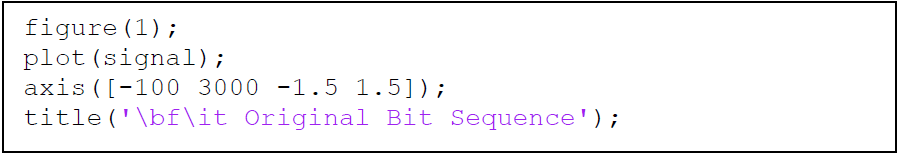
If there is more than one data bit sent per hop frequency, then the system is said to be slow frequency hopping. On the other hand, if a single data bit is sent over several hops in sequence, then the system is a fast frequency hopping system. The hop rate is denoted as *Rh*, and the data rate is denoted as *Rd*. Therefore, in a slow frequency hopping system *Tb = 1/Rd << Th = 1/Rh*.

**After introducing FHSS, we are going to do some practice by modeling a communication system based on FHSS.**

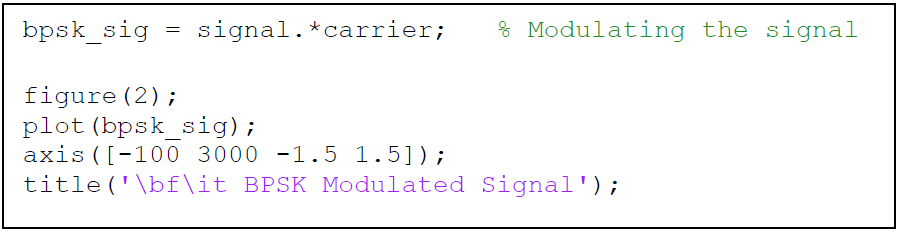
1. Create a new Script named as FHSS.m
2. First, we must generate the bit pattern that we are going to send. Generate N bits (N= 25).

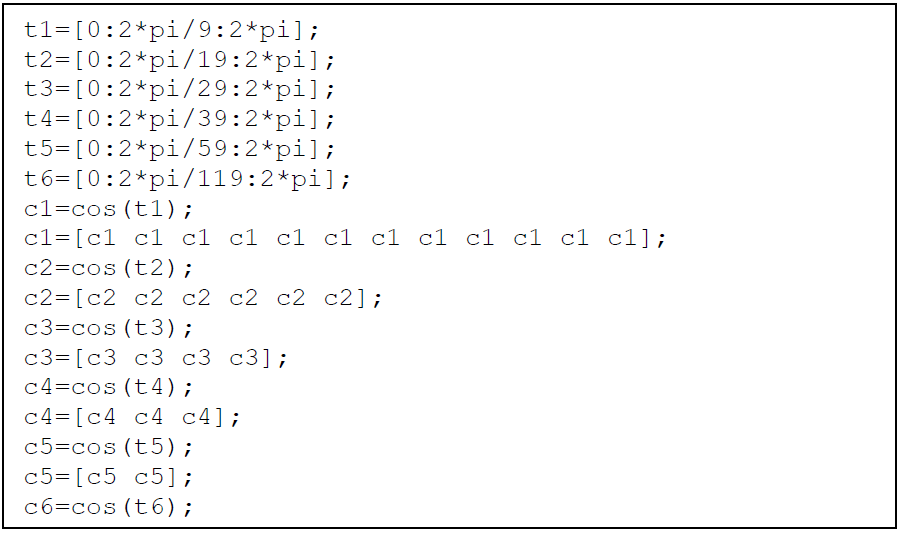
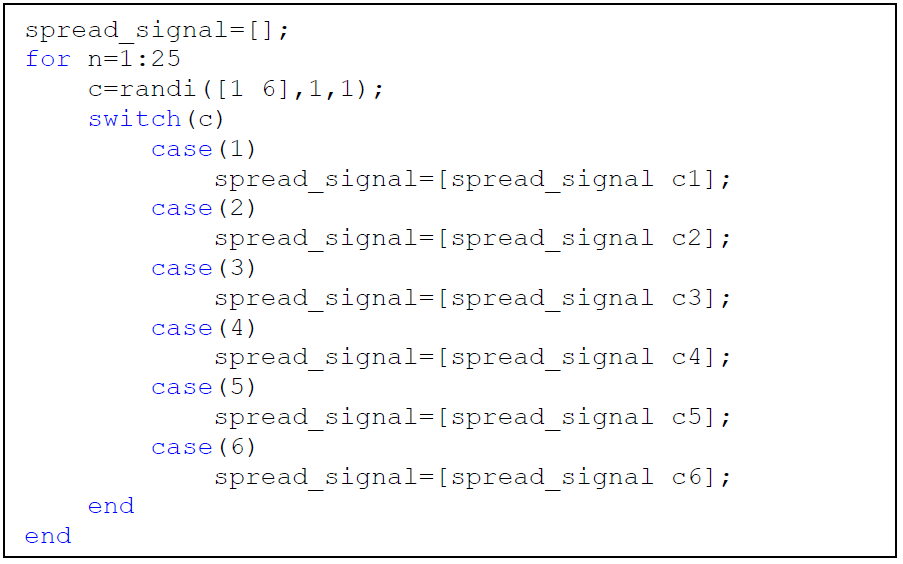


1. Generate signal and carriers in order to perform the modulation later.
2. Plot the generated bit sequence.



1. Let’s perform the BPSK modulation of the signal.



1. Let’s Prepare 6 new different carrier frequencies.
2. Random frequency hops to form a spread signal.
3. Plot the spread signal. For better understanding select manually c=1, c=2, c=3, c=4, c=5, c=6 (instead of selecting randomly) and plot after each selected value.
4. Try to perform Spreading of the BPSK signal into wider band, to obtain the freq\_hopped\_signal.

Use: freq\_hopped\_sig=bpsk\_sig.\*spread\_signal;

1. At the receiver side. Try to demodulate the BPSK Signal.

Use: demod\_psk=freq\_hopped\_sig.\*spread\_signal;

1. Demodulate the Binary Signal and compare with the transmitted bit sequence generated in point 3.

Pay attention to the demodulation process.

recovered\_s = [];

for i = 1:120:3000

if demod\_psk(i) == -1

bit = 0;

else

bit = 1;

end

recovered\_s = [recovered\_s bit];

end

1. Plot the signals obtained in point 9,10 and 11 and observe.

**You can upload your reports on our channel on Teams:**

**LAB - 2024/2025 Physical Layer for software radio Files Lab08 student reports**

**Please try to compress your MATLAB scripts (DSSS, FHSS) into one file called: Lab07\_yourName**