

**A black text on a white background

Description automatically generated**

**گزارش تمرین کامپیوتری 4**

**سیستم های مخابراتی**

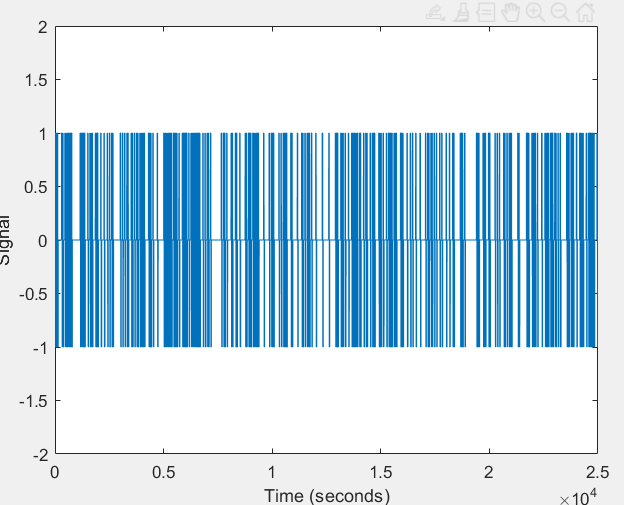
**تهیه­کننده:**

**زهرا ملکی**

**تابستان 1403**



Hear is the generated signal:



effective\_bandwidth = 1 / Tb;

Effective Bandwidth: 0.02 Hz



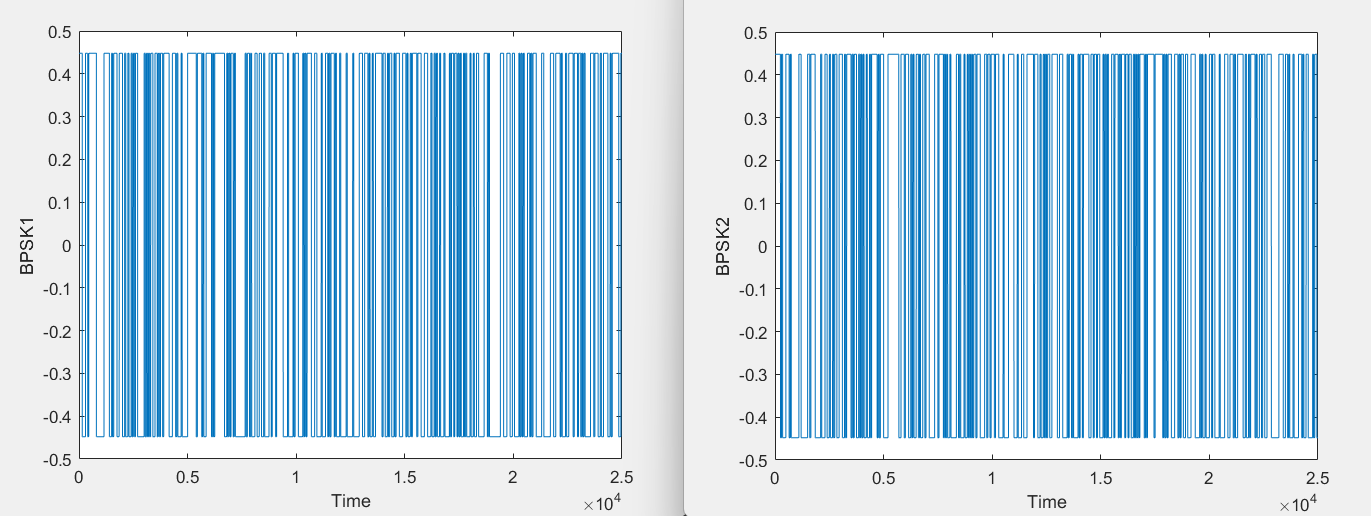
power\_noise = (norm(white\_noise)^2) / (eta\*Ts);

Power of Sampled White Noise: 0.39585

3/4.

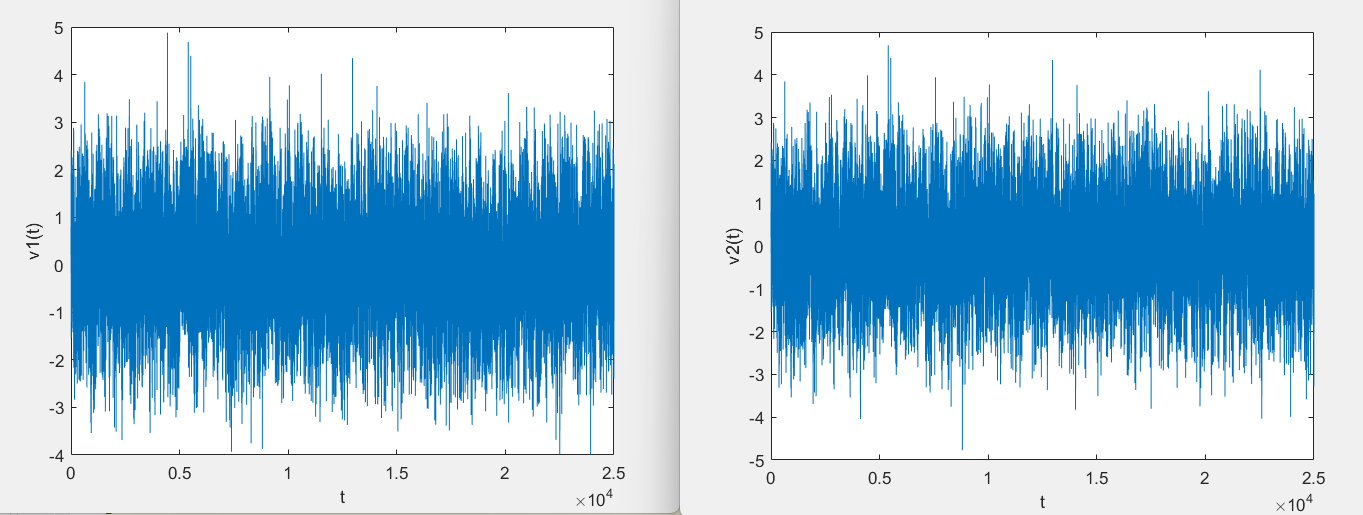
For the previous binary sequence and a new one:

SNRe: -6.9745



5.

noise\_sequence = randn(1, 500\*Tb/Ts);



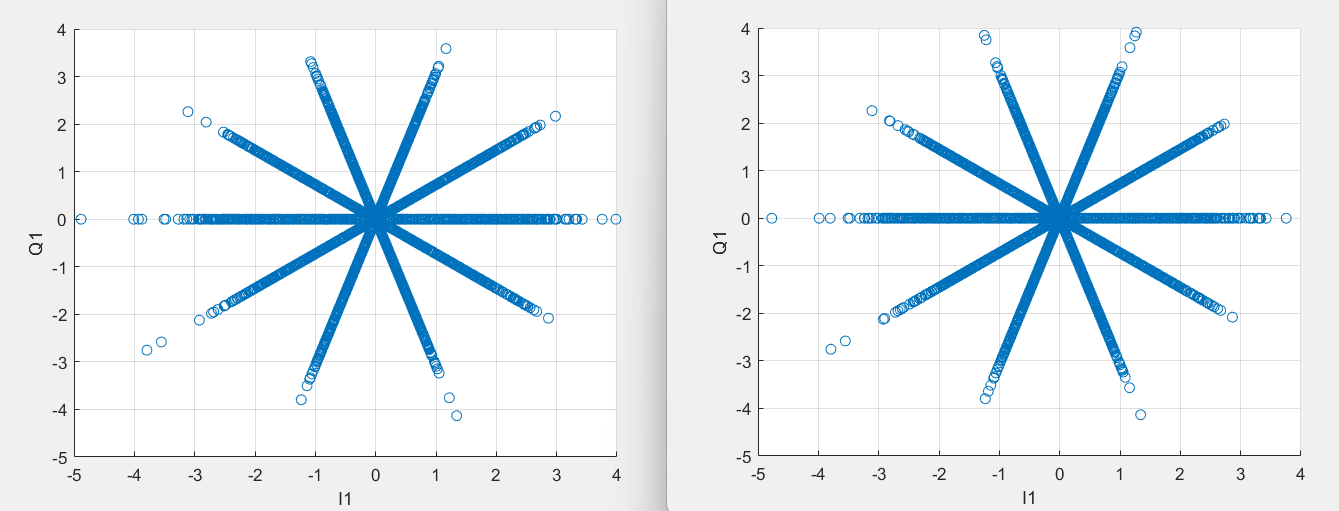
6.

Using scatter:

symb\_I2 = bpsk\_noise\_sequence2 .\* cos(10\*pi\*t\_noise/Tb);

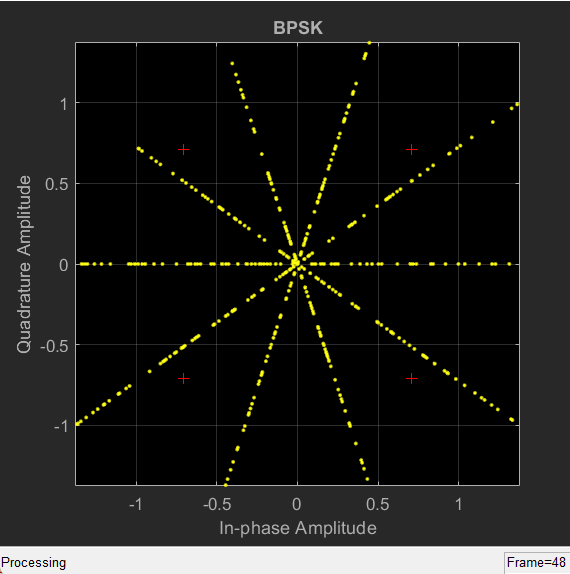
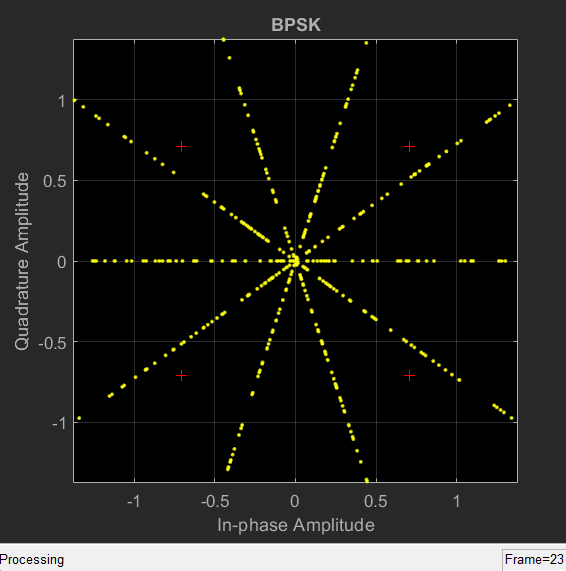
symb\_Q2 = bpsk\_noise\_sequence2 .\* sin(10\*pi\*t\_noise/Tb);

scatter(symb\_I2, symb\_Q2);

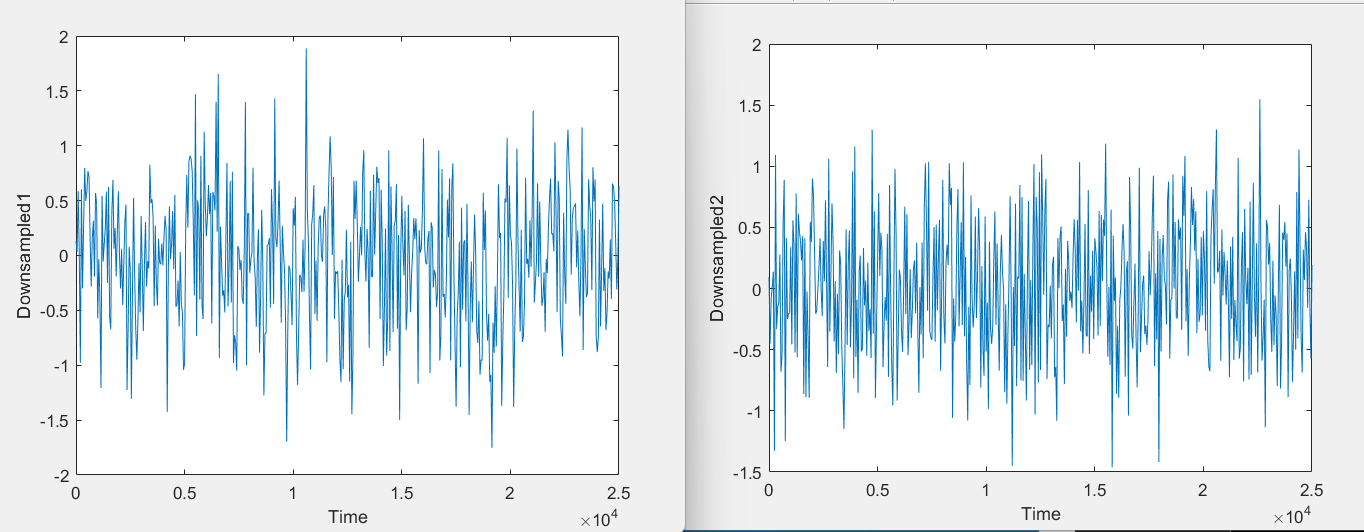


constellation = comm.ConstellationDiagram('Title','BPSK','ShowTrajectory',false,'SamplesPerSymbol',1);

and the result is two gifs like below:



7.



filteredSignal1 = conv(symb\_I1 + 1j \* symb\_Q1, filterCoeff);

filteredNoise1 = conv(bpsk\_noise\_sequence1 .\* cos(10\*pi\*t\_noise/Tb), filterCoeff);

n=round(Tb / Ts);

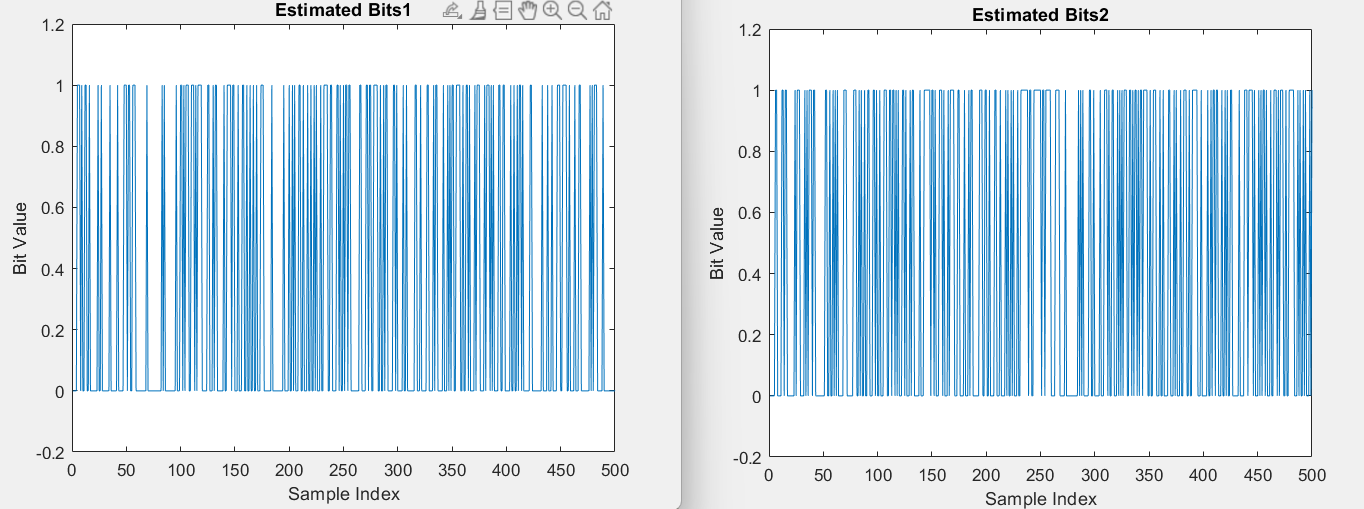
downsampledSignal1 = filteredSignal1(1:n:end);

downsampledNoise1 = filteredNoise1(1:n:end);

8.

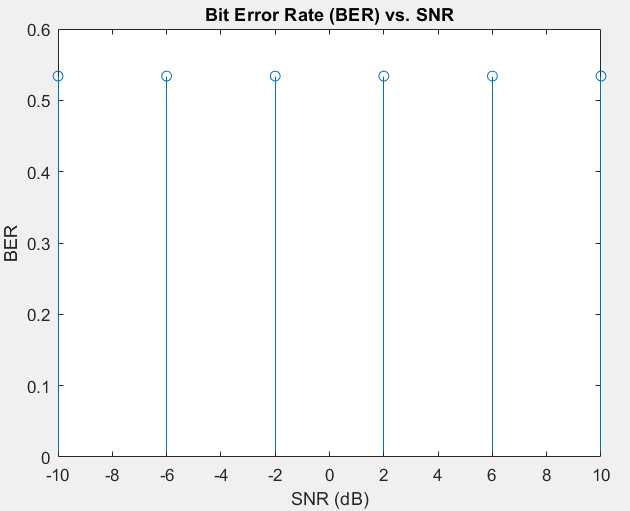
Threshold Estimate 1: 0.8000

Threshold Estimate 2: 0.7000



9.

SNRe: -10 -6 -2 2 6 10



error\_count = sum(abs(estimated\_bits9 - binary\_sequence));

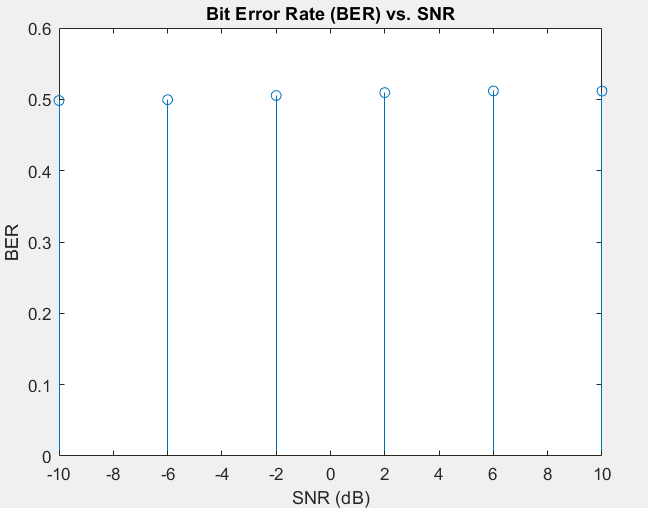
BER(j) = error\_count / 500;

10.

error\_count = sum(abs(estimated\_bits9 - binary\_sequence));

error\_count\_total = error\_count\_total + error\_count;

BER(j) = error\_count\_total / (500 \* 50);



The theoretical BER for a BPSK (Binary Phase Shift Keying) modulation scheme in an AWGN (Additive White Gaussian Noise) channel can be calculated using the following formula:

BER = 0.5 \* erfc(sqrt(10^(SNR/10)))

Which is compatible with the shown plot.

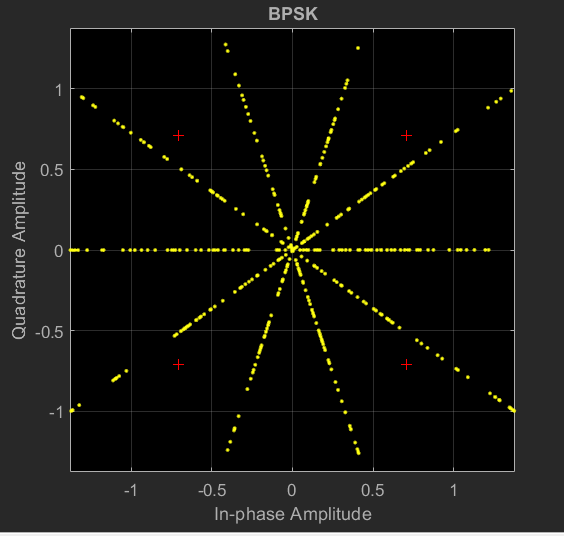
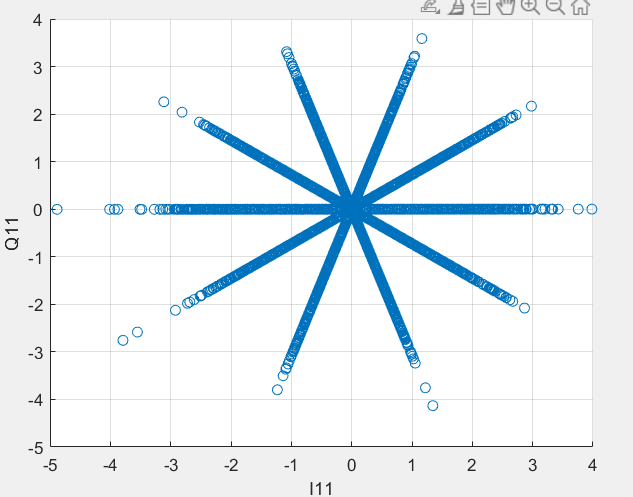
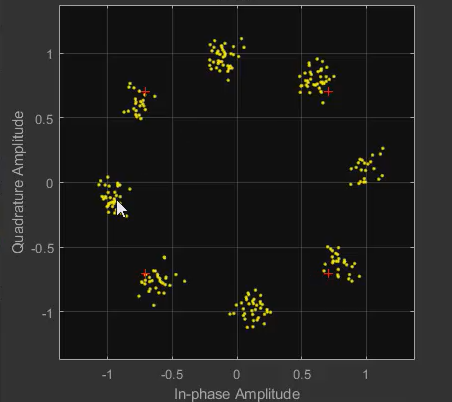
11.

offset\_frequency = 0.001 \* 1/Ts;

t\_noise\_offset = t\_noise + offset\_frequency;

symb\_I11 = bpsk\_noise\_sequence1 .\* cos(10\*pi\*t\_noise\_offset/Tb);

symb\_Q11 = bpsk\_noise\_sequence1 .\* sin(10\*pi\*t\_noise\_offset/Tb);



12.

Therefore, to determine an appropriate threshold for symbol detection in your specific system, you would need to perform empirical testing and conduct a more detailed analysis.

13.

 I suggest using a matched filter that is capable of handling frequency offset while maintaining stability.

To achieve stability against frequency offset, you can employ a Finite Impulse Response (FIR) filter. FIR filter is one of the stable filters that uses a sequence of coefficients (usually constant coefficients) to perform the filtering operation.

One of the methods for designing a frequency-matched FIR filter is through windowing. In this approach, we first determine the desired frequency response and then calculate the filter coefficients using a windowing function to apply appropriate weighting. After applying these coefficients to the input signal, the output signal will be frequency-matched to the desired frequency.