

Trabalho 2

3) Noise and Signal

3.1) Noise Statistics

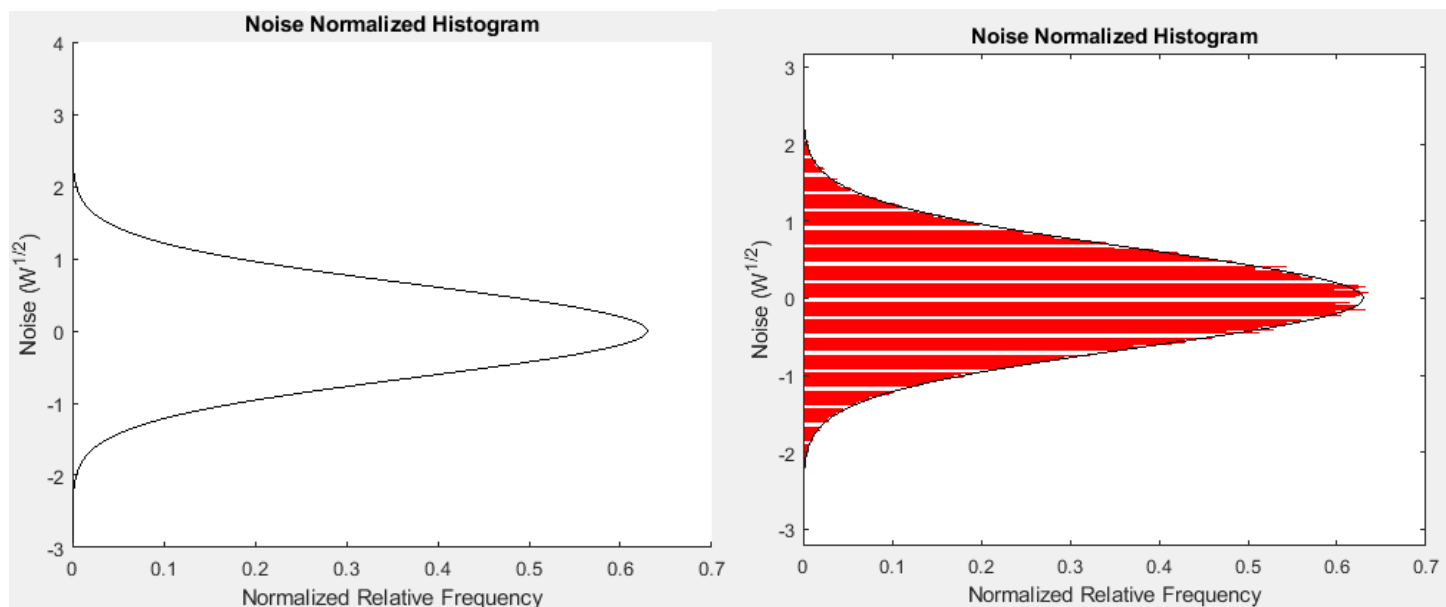
3.1.1)

O ruído vai ser um Gaussian White Noise, com uma densidade espectral de ruído -66 dBm/Hz, bilateral. O código é o seguinte:

```
fs = N*R; % Sampling Frequency
noisePower = noisePSD + 10*log10(fs) - 30; % dBm to dBW
sigma2_n = 10^(noisePower/10);
figure(1);
signal = randi([0 1],1,NSYM); % information signal
[signalTransmitted,t] = rcosflt(signal,R,N*R,'fir/normal',F); % formatted
noise = wgn(length(signalTransmitted),1,noisePower); % noisePower in dBW
[n, x] = hist(noise,1000);
dx = abs(x(2)-x(1));
bar(x,n/(sum(n)*dx),'r');
hold on
plot(x, 1/sqrt(2*pi*sigma2_n)*exp(-x.^2/(2*sigma2_n)),'k'); % sigma2_n is the
theoretical value of the noise variance (average power)
hold off;
title('Noise Normalized Histogram');
xlabel('Noise (W^{1/2})');
ylabel('Normalized Relative Frequency')
view([90 -90]);
```

3.1.2)

As imagens obtidas são:



As linhas de código são necessárias para termos acesso às barras de histograma, a primeira linha de código ajuda no cálculo do número de barras que queremos.

3.1.3)

Foi obtido a média do ruído ($\text{mean}(\text{noise})$) e a potência média do ruído ($\text{mean}(\text{noise}^2)$). O valor obtido foi de 9.4×10^{-4} e 0.4015 respetivamente.

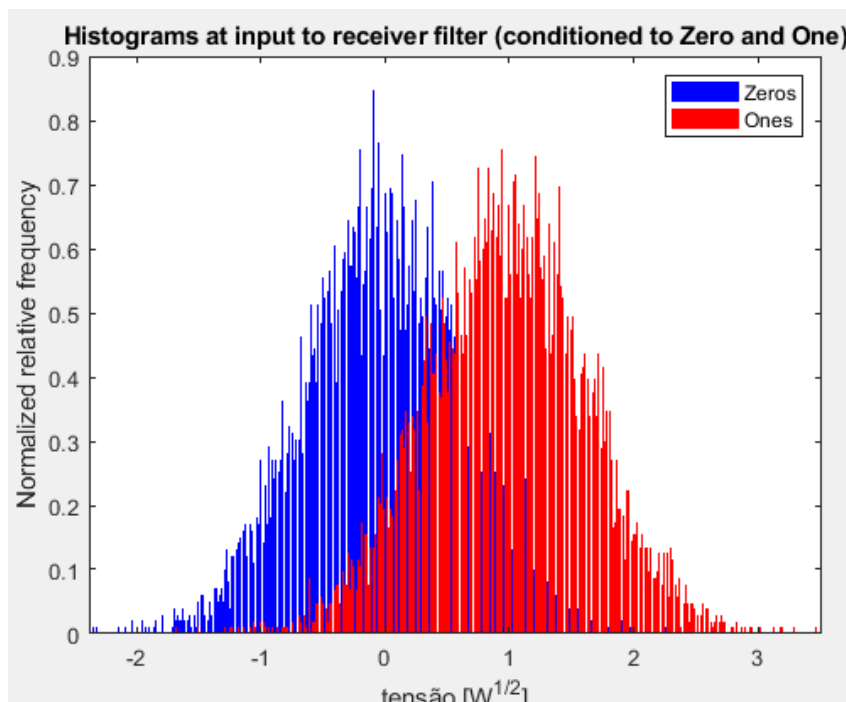
3.2) Symbol conditioned signal histograms

```
signalReceived = signalTransmitted + noise;
signalReceived_amost = zeros(1,NSYM);
[~,atraso1] = correlacao(signal,signalTransmitted,N,16,16);
for k=1:NSYM % Align and sampled the received signal plus noise
    signalReceived_amost(k) = signalReceived(N*(k-1)+atraso1);
end

%Zero conditioned histogram
signalReceived_amost_0 = signalReceived_amost(signal==0; %Samples that are ZEROs
[n0, x0] = hist(signalReceived_amost_0,1000);
dx0 = abs(x0(2)-x0(1));
bar(x0,n0/(sum(n0)*dx0),'b');
hold on

%ONE conditioned histogram
signalReceived_amost_1 = signalReceived_amost(signal==1); % Samples that are ONES
[n1, x1] = hist(signalReceived_amost_1,1000);
dx1 = abs(x1(2)-x1(1));
bar(x1,n1/(sum(n1)*dx1),'r');
title('Histograms at input to receiver filter (conditioned to Zero and One)');
xlabel('tensão [ $W^{1/2}$ ]');
ylabel('Normalized relative frequency');
hold off
```

A figura obtida foi a seguinte:



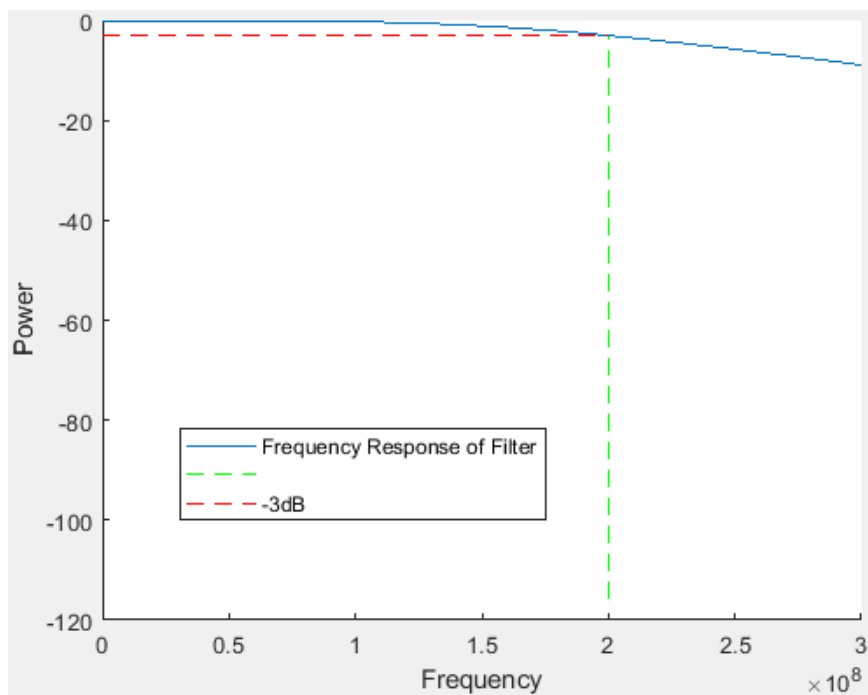
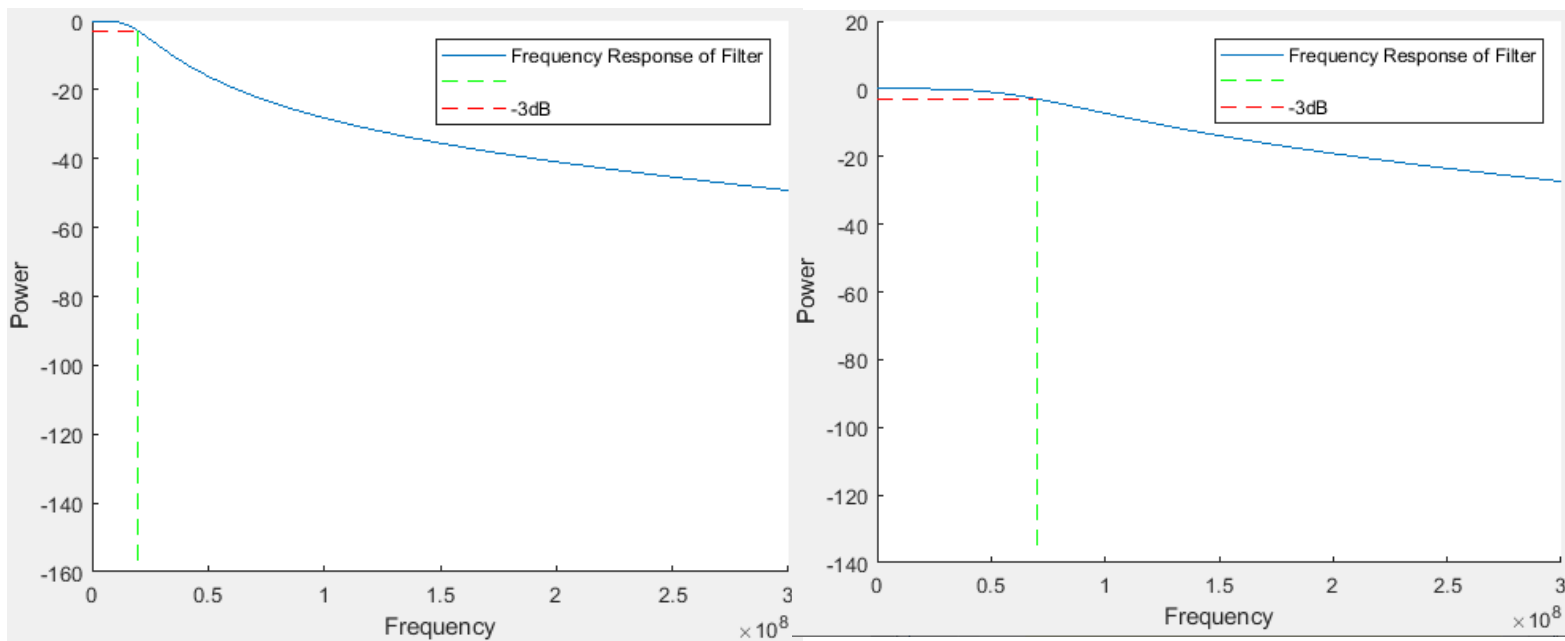
É o esperado pois com a adição de ruído os respetivos '0' e '1', não vão ter um valor fixo, mas sim uma gama de valores (por causa do ruído mesmo), porém podemos ver que em média a tensão usada para representar o '1' lógico é mais elevada que a tensão usada para representar o '0' lógico tal como esperado.

4) Receiver

4.1) Filtering

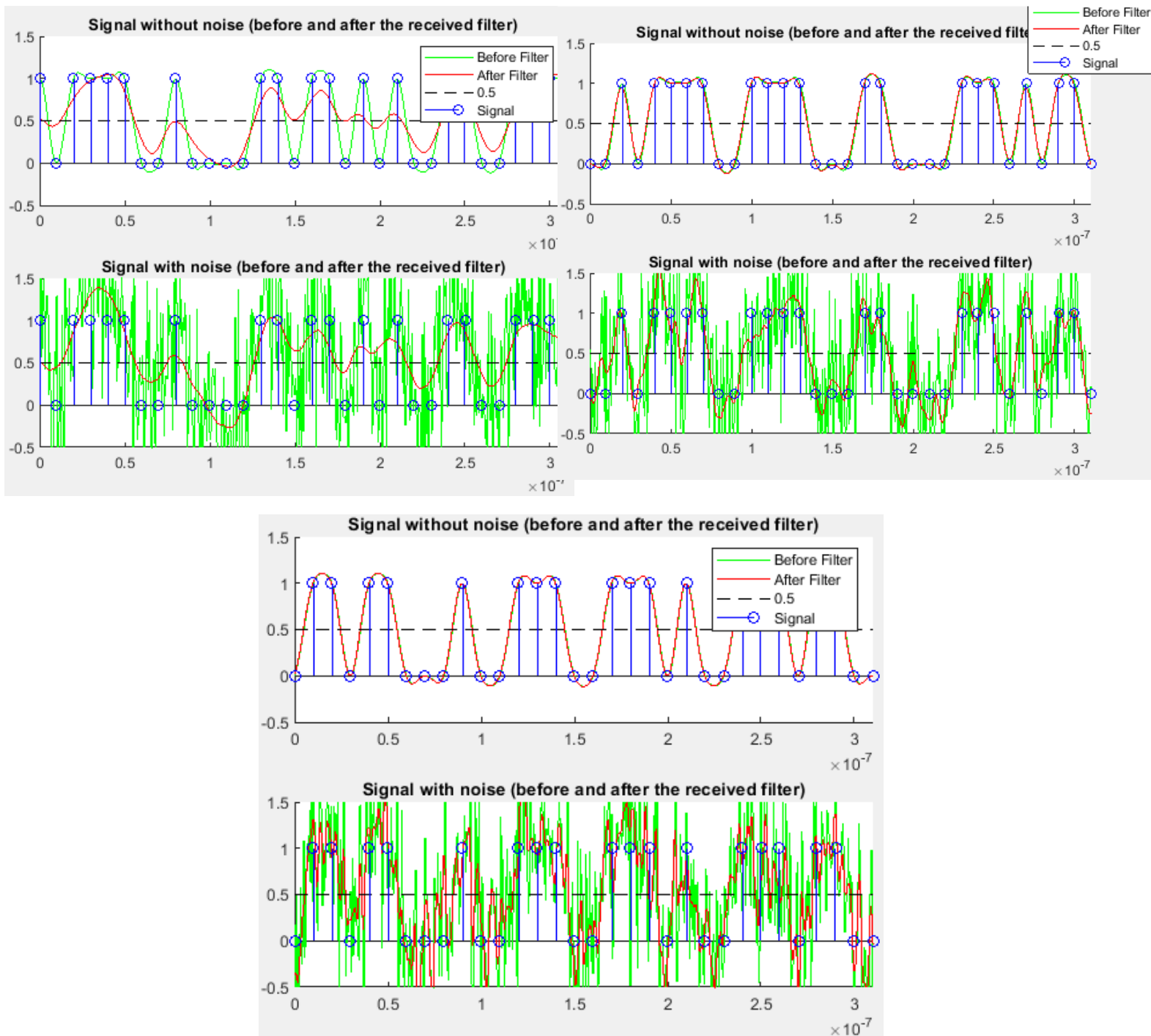
a) Frequency Domain

A imagem obtida nos 3 casos foi a seguinte:



b) Time Domain

A imagem obtida nos 3 casos foi a seguinte:



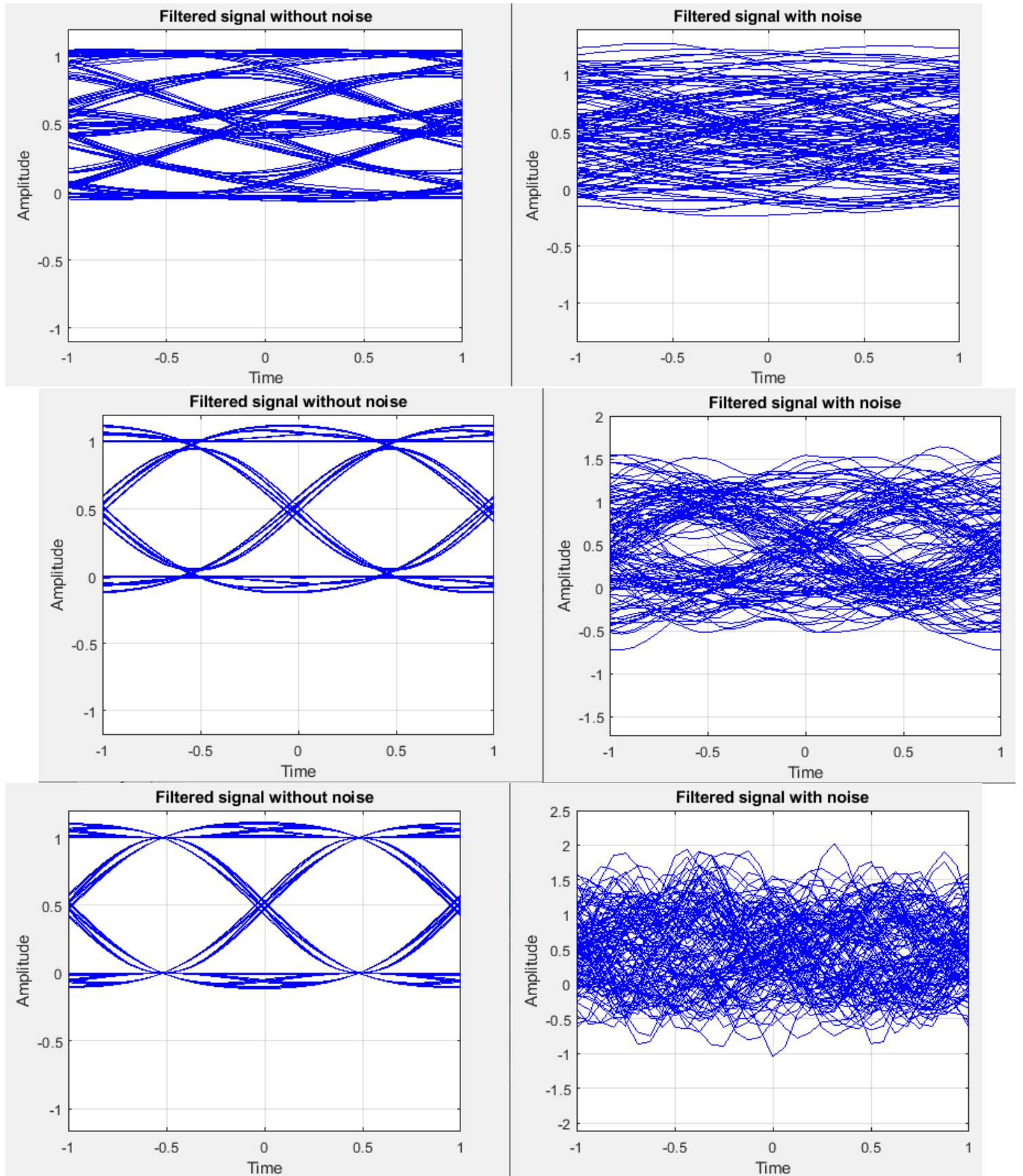
0.7 -> Melhor ISI mas mais ruído.

2 -> ISI mantém mas pioramos muito o ruído

Se aumentarmos a LB, podemos melhorar um pouco a ISI mas pioramos e muito o ruído.

4.2) Eye Diagram

A imagem obtida nos 3 casos foi a seguinte:



4.5) Probability of Error

a) Monte-Carlo Estimator

Os valores obtidos foram os seguintes:

```
N° de bits errados = 2587, num total de 40000  
BER (REAL) = 0.064675  
N° de bits errados = 257, num total de 4000  
BER (Monte-Carlo) = 0.064250, Erro Associado = 1%
```

c) Analytical Approach

Os valores obtidos foram os seguintes:

```
Potência de Ruído depois do Filtro (Analítico) = 0.111600 W  
Potência de Ruído depois do Filtro (Numérico) = 0.105875 W  
BER (Analítico) = 0.067234, Erro Associado = 4%
```

d) Q Factor (Gaussian Approximation)

O valor obtido foi o seguinte:

```
BER (Q-Factor) = 0.061483, Erro Associado = 5%
```

e) Conclusions

A imagem obtida foi a seguinte:

