

Relatório III - Laboratório III

Schaiana Sonaglio

Sistemas de Comunicação I

<schaiana.s@aluno.ifsc.edu.br>

1 Introdução

Neste relatório, serão apresentadas as soluções para duas questões propostas no terceiro laboratório da disciplina Sistemas de Comunicação I (COM1), além de uma breve explicação sobre os conceitos teóricos necessários para a resolução de cada questão, análise dos resultados e considerações finais sobre o conteúdo revisado ou aprendido. Todas as questões foram resolvidas utilizando o software Matlab.

Nas questões um e dois, foram utilizados conceitos de transmissão binária de uma informação utilizando sinalização NRZ (Nonreturn to Zero) unipolar e bipolar com adição de ruído branco gaussiano (AWGN - Additive White Gaussian Noise). Para detectar o sinal, foi utilizado um filtro casado, aumentando a eficácia em relação a probabilidade de erro de bit.

2 Conceitos Teóricos

2.1 Representação Binária da Informação

O PCM (Pulse Code Modulation) é uma forma de se transmitir uma informação e se divide em três etapas principais: amostragem, quantização e codificação. Para recuperação do sinal, se faz uma decodificação e uma filtragem. O PCM, quando aplicado a um símbolo binário, gera uma forma de onda PCM, que pode ser representada como NRZ (Nonreturn to Zero), RZ (Return to Zero), codificação em fase e sinalização multinível. Nos exercícios aqui resolvidos, foi utilizada a NRZ unipolar (amplitude entre 0V e 1V) e bipolar (amplitude varia entre -1V e 1V).

2.2 Canal AWGN

O movimento aleatório dos elétrons em elementos que dissipam energia é chamado de ruído térmico, é constante em toda faixa de frequência utilizada por sistemas de comunicação e possui densidade espectral de potência definida como N_0 ; como esses movimentos dos elétrons não possuem correlação entre si, o ruído é chamado de ruído gaussiano. Para separar o ruído da informação do canal, usa-se o SNR (Razão Sinal Ruído), que divide a energia de bit do sinal (E_b) pela densidade espectral de potência do ruído (N_0). Para detectar os sinais binários no canal AWGN a probabilidade de detecção dos símbolos s_1 e s_2 é verificada, através de um limiar de decisão (gama zero), sendo que um ou o outro foi transmitido, conforme ilustra a Figura 1.

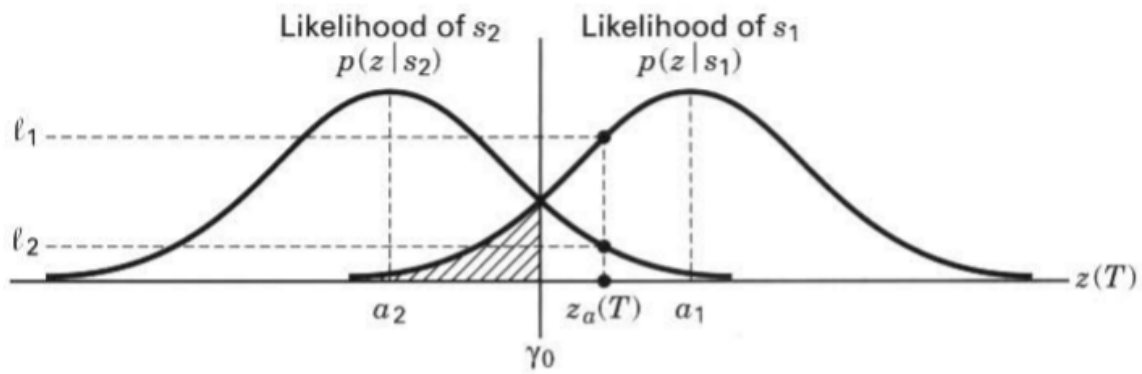


Figura 1 - Exercício 1: AWGN

2.3 Filtro Casado e Probabilidade de Erro de Bit

O filtro casado é utilizado para ressaltar a detecção do pulso de entrada de um sistema, minimizando o ruído. Ele é projetado para que a potência instantânea do sinal de saída seja a maior possível, quando comparada com o sinal do ruído da saída, maximizando o SNR.

Para realizar a detecção dos símbolos transmitidos, usamos o limiar de decisão γ_0 . Cada tipo de sinalização para transmissão de informação possui uma probabilidade diferente de os símbolos serem detectados com erro. Nesse laboratório, utilizamos duas sinalizações, a NRZ unipolar e a NRZ bipolar, com as seguintes informações:

NRZ Unipolar: $s_1(t) = A$ e $s_2(t) = 0$ NRZ Bipolar: $s_1(t) = A$ e $s_2(t) = -A$

Se as amplitudes dos símbolos A , 0 e $-A$ forem $1V$, $0V$ e $-1V$, o limiar de decisão unipolar é $\gamma_0 = 1/2$ e o bipolar $\gamma_0 = 0$. Para a NRZ bipolar, a probabilidade de erro de bit é menor. Para conseguir o mesmo desempenho que a bipolar, a unipolar precisa ter o dobro de energia de bit na transmissão.

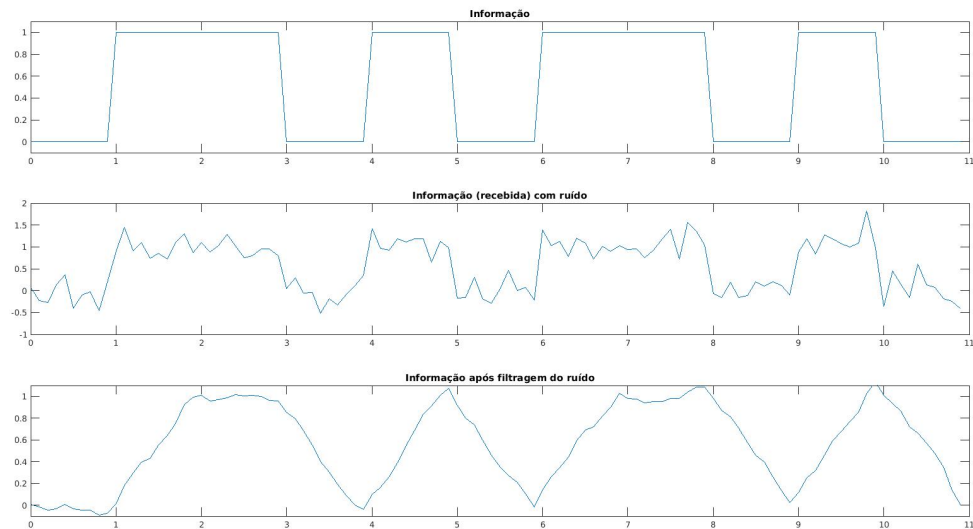
3 Apresentação dos Resultados

3.1 Exercício 1

Simular uma linha de transmissão com os seguintes parâmetros:

- Sequência de informação [0 1 1 0 1 0 1 1 0 1 0];
- Sinalização NRZ unipolar com nível de amplitude de $1V$;
- Canal AWGN com $SNR = 10dB$;
- Recepção com e sem filtro casado.

Apresentar os gráficos de todos os estágios da transmissão, comentar e concluir os resultados.



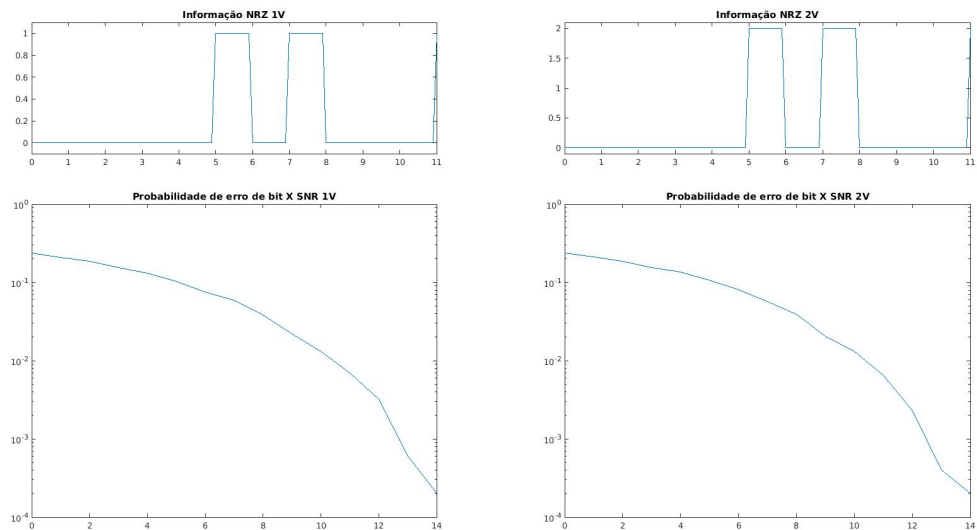
Analisando o sinal filtrado e o sinal com ruído, nota-se que é possível recuperá-lo das duas maneiras.

3.2 Exercício 2

Fazer a simulação de desempenho de erro para comparar, através de um gráfico de Probabilidade de erro de bit (P_b) vs SNR, os seguintes sistemas:

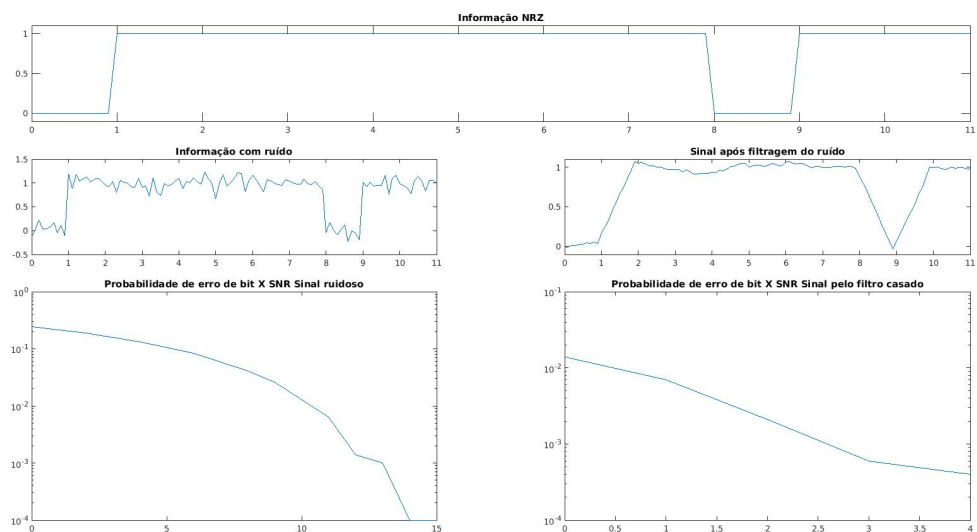
- Transmissão utilizando sinalização NRZ unipolar com amplitude de 1V e 2V, ambos sem a utilização do filtro casado;
- Transmissão utilizando sinalização NRZ unipolar com amplitude de 1V, com e sem filtro casado;
- Transmissão utilizando sinalização NRZ unipolar e bipolar, ambos com a utilização de filtro casado;
- Plote as expressões teóricas de P_b das sinalizações Polar e Bipolar (eq. 3.73 e 3.76) e compare-as com os resultados da simulação do item 3. Observe que as simulações anteriores estão em função de SNR e as expressões em função de E_b/N_0 !
- Comentar e concluir todos os resultados.

Item 1:



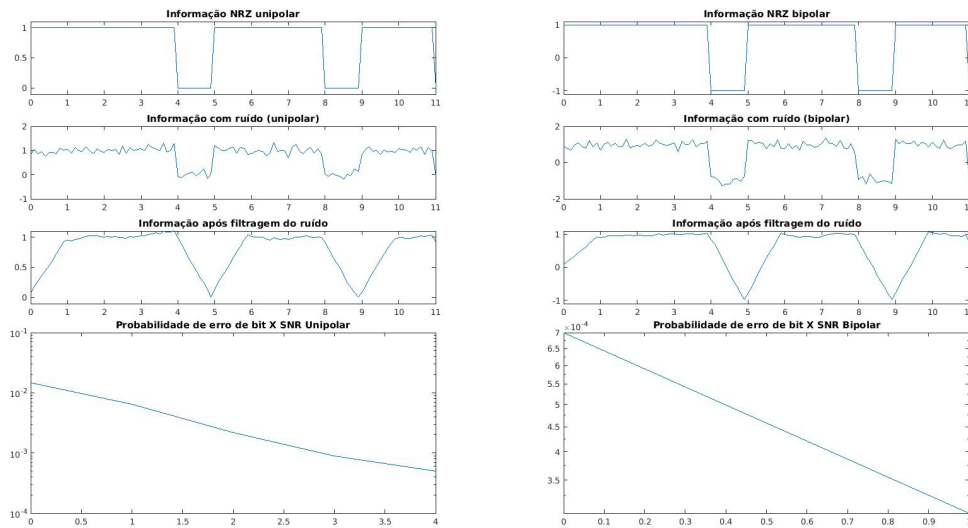
A sinalização unipolar com amplitude de 2V é teoricamente mais eficaz, uma vez que o limiar de decisão tem uma distância maior entre seus extremos. Pelo resultado do Matlab, não foi tão clara a visualização para confirmar a teoria.

Item 2:



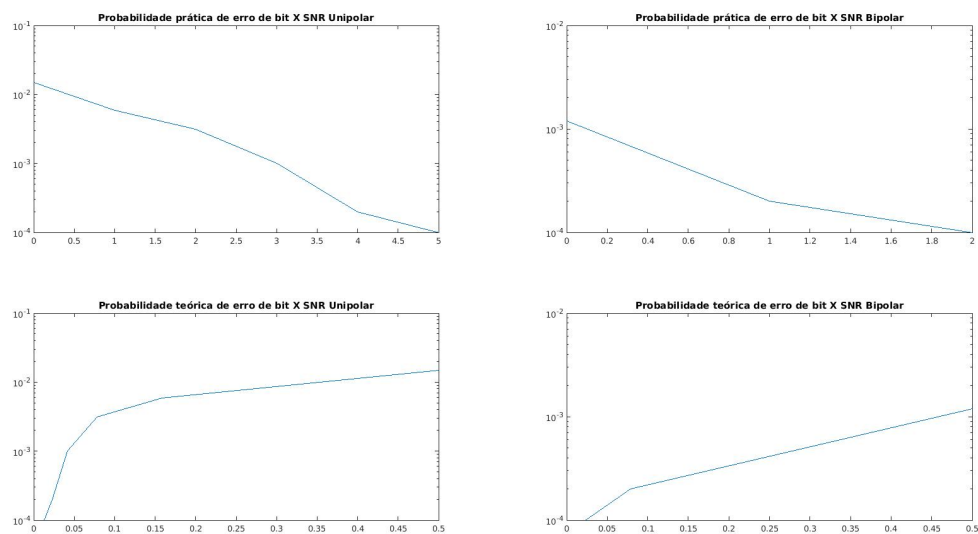
Houve diferença na probabilidade de erro de bit. Após a filtragem, menos erros ocorreram na detecção de cada símbolo.

Item 3:



Utilizando o filtro casado com a sinalização NRZ bipolar foi necessário apenas metade da energia do que com a NRZ unipolar, para se ter a mesma probabilidade de erro de bit.

Item 4:



4 Considerações Finais

Essa atividade demonstrou a geração e formatação de um sinal através da sinalização NRZ unipolar e bipolar, que foi submetido a um canal AWGN e posteriormente passado por um filtro casado. Foi possível demonstrar as diferentes formações do sinal, quando passado pelo filtro e quando com ruído.

5 Anexos

5.1 Anexo 1 - Exercício 1: código feito no Matlab

```
clear all; close all; clc;

N = 10; %numero de amostras
SNR = 10;
filtro_nrz = ones(1,N);
Rb = 1; %taxa de transmissão de bits

info = [0 1 1 0 1 0 1 1 0 1 0];
info_up = upsample(info, N); %superamostragem
info_format = filter(filtro_nrz, 1, info_up); %formatação do sinal

t = [0:length(info_format)-1]/(N*Rb);

Rx = awgn(info_format, SNR, 'measured'); %adicionando ruído branco ao sinal formatado
filtro_casado = filtro_nrz;
Rx_filtrado = filter(filtro_casado, 1, Rx)/N;

figure(1)
subplot(311)
plot(t, info_format); axis([0 11 -0.1 1.1]);

subplot(312)
plot(t, Rx); xlim([0 11])
title('Informação (recebida com ruído)');
subplot(313)
plot(t, Rx_filtrado); axis([0 11 -0.1 1.1]);
title('Informação após filtragem do ruído');

amostra_filtrado = Rx_filtrado(N:N:end); %metade dos bits até o final
info_hat_filtrado = amostra_filtrado > 0.5; %limiar de decisão
[n_erro_f, taxa_f] = biterr(info, info_hat_filtrado); %compara o sinal original e taxa de erro de bit

amostra_rx = Rx(N:N:end);
info_hat = amostra_rx > 0.5;
[n_erro, taxa] = biterr(info, info_hat);
```

Figura 2 - Exercício 1: Código - parte 1

5.2 Anexo 2 - Exercício 2: código feito no Matlab

```
clear all; close all; clc;

N = 10; %número de amostras
filtro_nrz = ones(1,N);
Rb = 1; %taxa de transmissão de bits

info = randi([0 1], 1, 10e3);
info_2 = info*2;

info_up_1 = upsample(info, N); %superamostragem
info_up_2 = upsample(info_2, N);
info_format_1 = filter(filtro_nrz, 1, info_up_1); %formatacao do sinal
info_format_2 = filter(filtro_nrz, 1, info_up_2);

t = [0:length(info_format_1)-1]/(N*Rb);

for SNR = 0:15
    Rx_1 = awgn(info_format_1, SNR, 'measured');
    Rx_2 = awgn(info_format_2, SNR, 'measured');
    amostra_1 = Rx_1(N:N:end);
    amostra_2 = Rx_2(N:N:end);
    info_hat_1 = amostra_1 > 0.5;
    info_hat_2 = amostra_2 > 1;
    [n_erro_1(SNR+1), taxa_1(SNR+1)] = biterr(info, info_hat_1);
    [n_erro_2(SNR+1), taxa_2(SNR+1)] = biterr(info, info_hat_2);
end

figure(1)
subplot(321)
plot(t, info_format_1); axis([0 11 -0.1 1.1]);
title('Informação NRZ 1V');
subplot(322)
plot(t, info_format_2); axis([0 11 -0.1 2.1]);
title('Informação NRZ 2V');
subplot(3,2,[3 5])
semilogy([0:15], taxa_1)
title('Probabilidade de erro de bit X SNR 1V');
subplot(3,2,[4 6])
semilogy([0:15], taxa_2)
title('Probabilidade de erro de bit X SNR 2V');
```

Figura 3 - Exercício 2: Código - item 1

```

clear all; close all; clc;

N = 10; %numero de amostras
filtro_nrz = ones(1,N);
Rb = 1; %taxa de transmissão de bit

info = randi([0 1], 1, 10e3);
info_up = upsample(info, N); %superamostragem
info_format = filter(filtro_nrz, 1, info_up); %formatação do sinal

t = [0:length(info_format)-1]/(N*Rb);

filtro_casado = filtro_nrz;
for SNR = 0:15
    Rx = awgn(info_format, SNR, 'measured');
    %Pb sem filtro
    amostra_Rx = Rx(N:N:end);
    info_hat = amostra_Rx > 0.5;
    [n_erro(SNR+1), taxa(SNR+1)] = biterr(info, info_hat)
    %Pb com filtro
    Rx_filtrado = filter(filtro_casado, 1, Rx)/N;
    amostra_filtrado = Rx_filtrado(N:N:end); %metade dos bits
    info_hat_filtrado = amostra_filtrado > 0.5; %limiar de decisão
    [n_erro_f(SNR+1), taxa_f(SNR+1)] = biterr(info, info_hat_filtrado); %comparando o sinal original e taxa de erro de bit
end

figure(1)
subplot(4,2,1:2)
plot(t, info_format); axis([0 11 -0.1 1.1]);
title('Informação NRZ');
subplot(4,2,3)
plot(t, Rx); xlim([0 11]);
title('Informação com ruído');
subplot(4,2,4)
plot(t, Rx_filtrado); axis([0 11 -0.1 1.1]);
title('Sinal após filtragem do ruído');
subplot(4,2,[5 7])
semilogy([0:15], taxa)
title('Probabilidade de erro de bit X SNR Sinal ruidoso');
subplot(4,2,[6 8])
semilogy([0:15], taxa_f)
title('Probabilidade de erro de bit X SNR Sinal pelo filtro casado');

```

Figura 4 - Exercício 2: Código - item 2

```

clear all; close all; clc;

N = 10;
filtro_nrz = ones(1,N);
filtro_casado = filtro_nrz;
Rb = 1;

unip = randi([0 1], 1, 10e3);
info_up_unip = upsample(unip, N);
unipolar_format = filter(filtro_nrz, 1, info_up_unip);
t = [0:length(unipolar_format)-1]/(N*Rb);

for SNR = 0:15
    Rx_unipolar = awgn(unipolar_format, SNR, 'measured');
    Rx_unipolar_filtrado = filter(filtro_casado, 1, Rx_unipolar)/N;
    amostra_unip = Rx_unipolar_filtrado(N:N:end);
    info_hat_unipolar = amostra_unip > 0.5;
    [n_erro_unipolar(SNR+1), taxa_unipolar(SNR+1)] = biterr(unip, info_hat_unipolar);
end

bipolar = unip*2 - 1;
info_up_bipolar = upsample(bipolar, N);
bipolar_format = filter(filtro_nrz, 1, info_up_bipolar);
t2 = [0:length(bipolar_format)-1]/(N*Rb);

%Normalização (-1 = 0)
bipolar_norm = bipolar;
find_0 = bipolar_norm == -1;
pos_0 = find(find_0);
bipolar_norm(pos_0) = 0;

for SNR = 0:15
    Rx_bipolar = awgn(bipolar_format, SNR, 'measured');
    Rx_bipolar_filtrado = filter(filtro_casado, 1, Rx_bipolar)/N;
    amostra_bipolar = Rx_bipolar_filtrado(N:N:end);
    info_hat_bipolar = amostra_bipolar > 0;
    [n_erro_bipolar(SNR+1), taxa_bipolar(SNR+1)] = biterr(bipolar_norm, info_hat_bipolar);
end

```

Figura 5 - Exercício 2: Código - item 3 - parte 1


```

figure(1)
subplot(5,2,1)
plot(t, unipolar_format); axis([0 11 -0.1 1.1]);
title('Informação NRZ unipolar');
subplot(5,2,2)
plot(t, bipolar_format); axis([0 11 -0.1 1.1]);
title('Informação NRZ bipolar'); axis([0 11 -1.1 1.1]);
subplot(5,2,3)
plot(t, Rx_unipolar); xlim([0 11]);
title('Informação com ruído (unipolar)');
subplot(5,2,4)
plot(t, Rx_bipolar); xlim([0 11]);
title('Informação com ruído (bipolar)');
subplot(5,2,5)
plot(t, Rx_unipolar_filtrado); axis([0 11 -0.1 1.1]);
title('Informação após filtragem do ruído');
subplot(5,2,6)
plot(t, Rx_bipolar_filtrado); axis([0 11 -1.1 1.1]);
title('Informação após filtragem do ruído');
subplot(5,2,[7 9])
semilogy([0:15], taxa_unipolar)
title('Probabilidade de erro de bit X SNR Unipolar');
subplot(5,2,[8 10])
semilogy([0:15], taxa_bipolar)
title('Probabilidade de erro de bit X SNR Bipolar');

```

Figura 6 - Exercício 2: Código - item 3 - parte 2

```

clear all; close all; clc;
N = 10;
filtro_nrz = ones(1,N);
filtro_casado = filtro_nrz;
Rb = 1;

unip = randi([0 1], 1, 10e3);
info_up_unip = upsample(unip, N);
unipolar_format = filter(filtro_nrz, 1, info_up_unip);
t = [0:length(unipolar_format)-1]/(N*Rb);

for SNR = 0:15
    Rx_unipolar = awgn(unipolar_format, SNR, 'measured');
    Rx_unipolar_filtrado = filter(filtro_casado, 1, Rx_unipolar)/N;
    amostra_unip = Rx_unipolar_filtrado(N:N:end);
    info_hat_unipolar = amostra_unip > 0.5;
    [n_erro_unipolar(SNR+1), taxa_unipolar(SNR+1)] = biterr(unip, info_hat_unipolar);
end

bipolar = unip*2 - 1;
info_up_bipolar = upsample(bipolar, N);
bipolar_format = filter(filtro_nrz, 1, info_up_bipolar);
t2 = [0:length(bipolar_format)-1]/(N*Rb);

%Normalização (-1 = 0)
bipolar_norm = bipolar;
find_0 = bipolar_norm == -1;
pos_0 = find(find_0);
bipolar_norm(pos_0) = 0;

```

Figura 7 - Exercício 2: Código - item 4 - parte 1

```

for SNR = 0:15
    Rx_bipolar = awgn(bipolar_format, SNR, 'measured');
    Rx_bipolar_filtrado = filter(filtro_casado, 1, Rx_bipolar)/N;
    amostra_bipolar = Rx_bipolar_filtrado(N:N:end);
    info_hat_bipolar = amostra_bipolar > 0;
    [n_erro_bipolar(SNR+1), taxa_bipolar(SNR+1)] = biterr(bipolar_norm, info_hat_bipolar);
end

figure(1)
subplot(2,2,1)
semilogy(0:15, taxa_unipolar)
title('Probabilidade prática de erro de bit X SNR Unipolar');
subplot(2,2,2)
semilogy(0:15, taxa_bipolar)
title('Probabilidade prática de erro de bit X SNR Bipolar');
subplot(2,2,3)
semilogy(qfunc(sqrt(0:15)), taxa_unipolar)
title('Probabilidade teórica de erro de bit X SNR Unipolar');
subplot(2,2,4)
semilogy(qfunc(sqrt(2*(0:15))), taxa_bipolar)
title('Probabilidade teórica de erro de bit X SNR Bipolar');

```

Figura 8 - Exercício 2: Código - item 4 - parte 2