

**UNIVERSIDADE FEDERAL DE SANTA MARIA
COMUNICAÇÃO DE DADOS**

VINÍCIUS HARDT SCHREINER

RELATÓRIO DO TRABALHO PRÁTICO FINAL - TRANSMISSÃO E RECEPÇÃO
DE DADOS ATRAVÉS DE MODEM ACÚSTICO

SANTA MARIA, 11 DE FEVEREIRO DE 2021

1. INTRODUÇÃO

O processo de transmissão e recepção de dados é muito importante para o estudo de comunicação de dados, sendo utilizado amplamente em todos os meios de comunicação.

O presente trabalho busca demonstrar o processo de simulação, adaptação e programação de um enlace de comunicação utilizando sincronismo temporal e sincronismo de portadora em um modem acústico utilizando a ferramenta GNU Octave, a fim de realizar o último trabalho prático da disciplina de Comunicação de Dados.

2. ROTEIRO

1) Os arquivos de implementação do modem ("modem_acustico.zip") estão disponíveis no Moodle. Faça o download dos arquivos, e execute ambos os arquivos de transmissor e receptor no Octave (é necessário executar o Octave em duas janelas separadas).

2) Salvar graficamente o sinal transmitido e recebido. Escrever um relatório comentando as diferenças percebidas entre os sinais.

3) Modifique o código para programar alguma funcionalidade estudada na disciplina. Pode optar por alguma das sugestões listadas abaixo ou propor outra funcionalidade que desejar programar.

3. SIMULAÇÕES

Para realizar as simulações do modem acústico, foi utilizado no software Octave as simulações de transmissor e receptor. O arranjo utilizado para o teste foi feito ligado à saída de som do computador à entrada do microfone.



Imagem 1: conexão para teste utilizando um computador.

e posteriormente para fins de teste foram utilizados dois computadores, um como transmissor e outro como receptor, nesses dois casos foi utilizado um cabo P2 para fazer as ligações entre entrada e saída.



Imagem 2: conexão para teste utilizando dois computadores.

Primeiramente foi baixado o material disponibilizado pelo professor no ambiente Moodle, então esse código foi analisado e modificado para iniciar a tarefa e preparação para o processo de transmissão e recepção real. Para isso, foi criado um teste ideal, onde o sinal não é transmitido efetivamente, ele só passa pelos filtros de transmissão e recepção, e após isso foi gerado o seguinte gráfico.

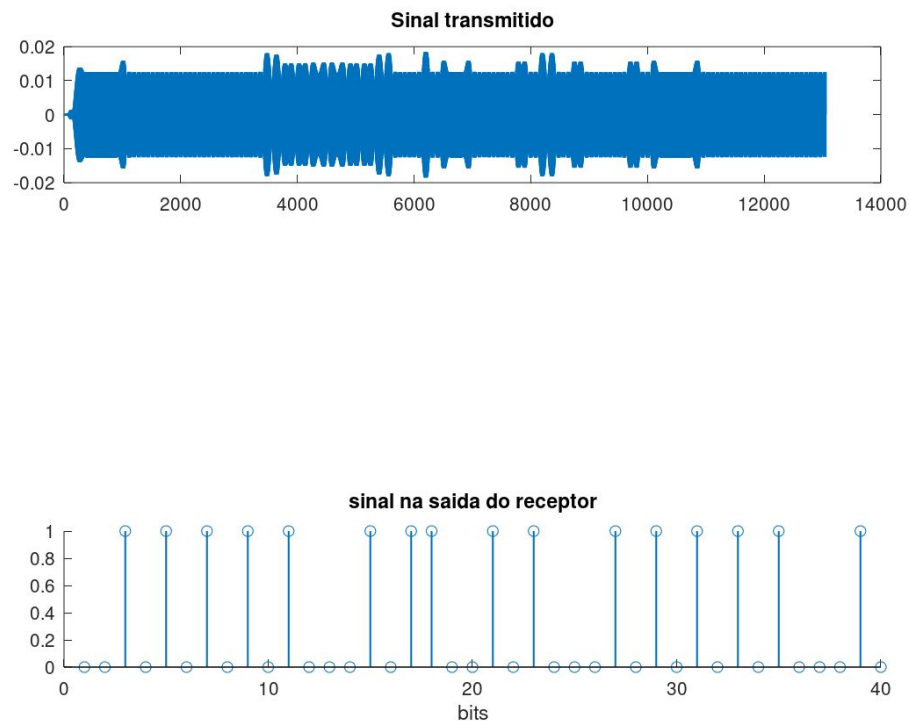


Imagem 3: simulação inicial sem transmissão efetiva.

Nesse gráfico, foi utilizado apenas uma filtragem gráfica para observar a saída do receptor, o qual nos dá a mensagem que foi transmitida originalmente em uma string de caracteres, e posteriormente convertida para binário, no receptor ocorre o processo inverso, com uma transformação de binário para decimal, e posteriormente de decimal para caractere novamente.

Em todos os testes, foi utilizada a string "TESTE" no processo de comunicação, nesse exemplo, pode-se perceber o número binário recebido.

Logo após, foi iniciado o processo de modificação do código para poder efetivamente realizar uma transmissão real entre computadores. Para realizar o enlace de dados, é utilizado um esquema para sincronização dos dados, com por exemplo, como o header para sincronização e quadros, além de filtro cossenóide.

Finalmente, o código foi modificado para utilizar também a *Fast Fourier Transform* como um meio para se obter o espectro das frequências do sinal enviado e recebido.

O Gráfico abaixo mostra o sinal enviado pelo transmissor, juntamente com o seu espectro.

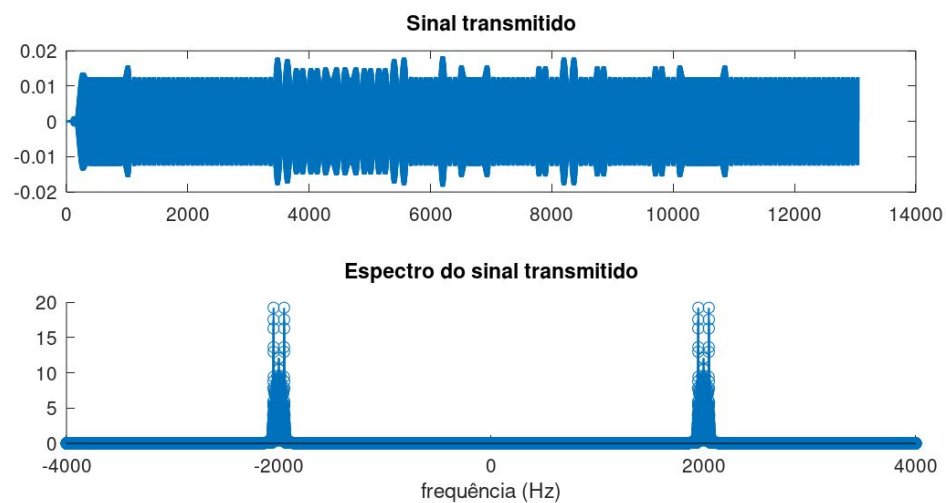


Imagem 4: sinal transmitido e seu espectro.

Nesse sinal, temos a string “TESTE” convertida e enviada com 100 bits, 2 KHz de frequência de portadora e 8000 amostras por segundo.

No no outro computador, recebe-se o mesmo sinal, que passa pela placa de som que atua como DAC e faz a conversão Analógico-Digital e faz o envio deste sinal pelo cabo P2. No outro lado do cabo, o sinal é recebido no outro computador pela entrada do microfone e a placa de som do mesmo efetua o processo inverso, fazendo a conversão do sinal analógico recebido para digital novamente, para ser processado.

Pode-se aprender durante o decorrer da disciplina que mesmo com altas taxas de bits e frequência de amostragem, sempre existem perdas no processo de conversão AD e transmissão e recepção de dados, mesmo que mínimas.

No computador receptor, foi utilizado uma implementação de receptor que recebe o sinal e mostra o mesmo e o seu espectro de frequências. Na imagem abaixo, pode-se observar o sinal “cru” recebido no computador receptor, antes de entrar efetivamente no receptor. Na próxima página pode-se comparar o sinal transmitido, com o recebido.

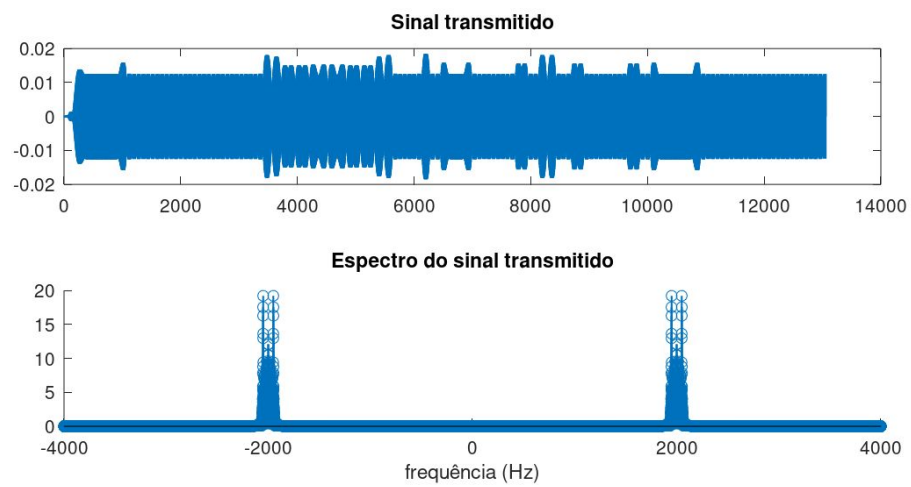


Imagem 5: sinal transmitido e seu espectro.

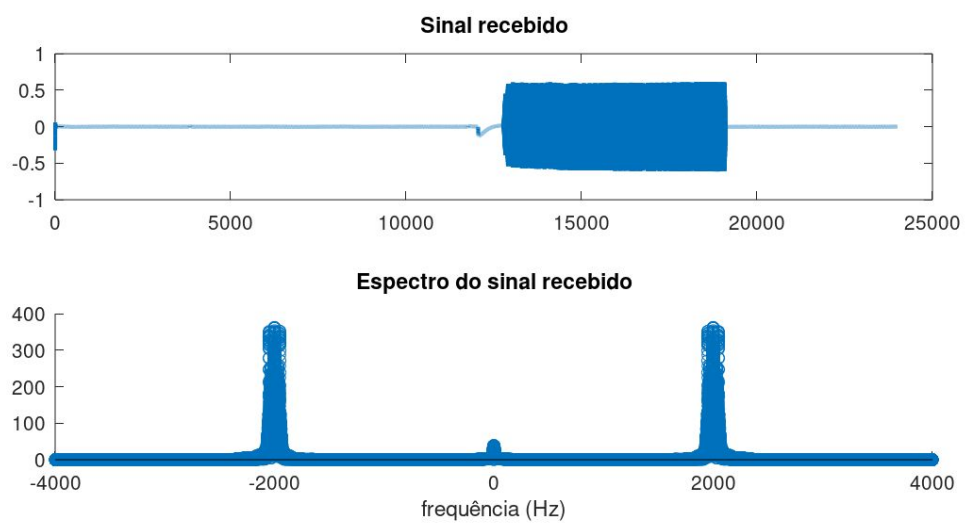


Imagem 6: sinal recebido e seu espectro.

Após o processo do sinal entrar no receptor, foi feita a conversão dele utilizando o algoritmo do receptor.

Como citado acima, em todo tipo de comunicação ocorrem erros, seja por meio de propagação, ou até mesmo pelo processo de amostragem e conversão Analógico digital.

Para testar os erros na comunicação, foi utilizada inicialmente a string “ABC” enviada com uma taxa de amostragem de 8000 amostras por segundo e uma frequência de portadora de 2KHz, abaixo pode-se observar o resultado da recepção.

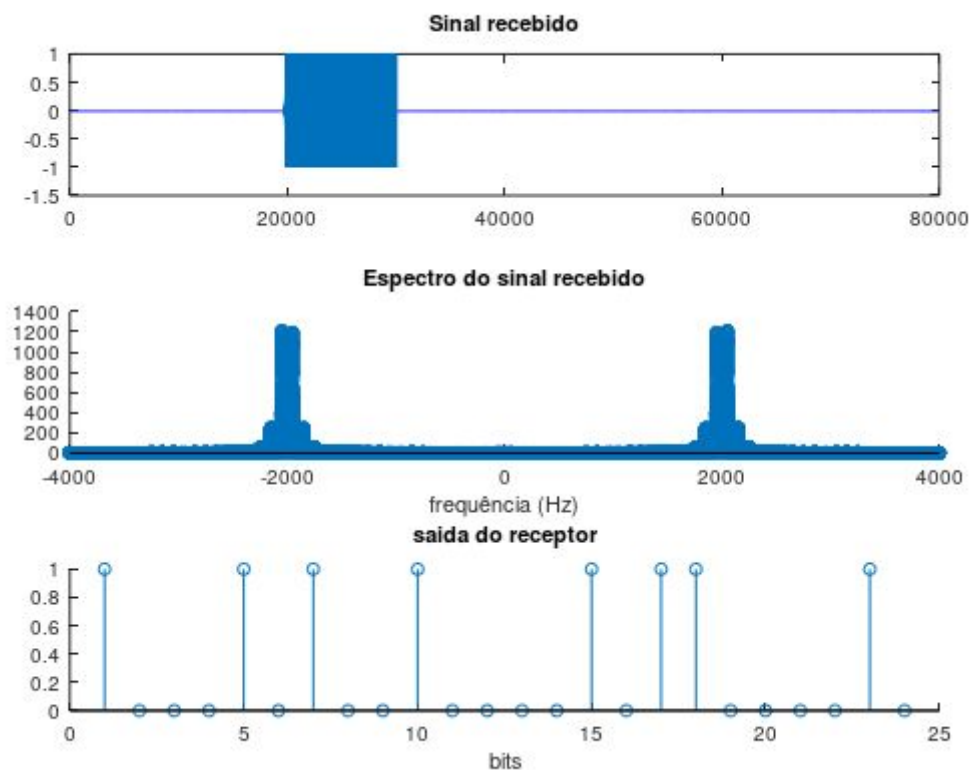


Imagem 7: sinal no receptor.

No exemplo acima, a string “ABC” convertida para binário pela tabela ASCII ficaria “01000001 01000010 01000011”. Pode-se notar no gráfico que o bit 5 apresenta erroneamente o valor 1, quando deveria apresentar 0, nesse caso a letra “A” foi transmitida como “Q”. Esse erro pode ter sido causado por diversos motivos, e por esse motivo é que protocolos de comunicação utilizam verificação e correção de erros.

Para o último teste, foi utilizado uma string maior contendo “Teste”, foi utilizada a mesma taxa de amostragem, frequência e taxa de bits, pois com a frequência igual a da placa de som causa muitos travamentos no software Octave.

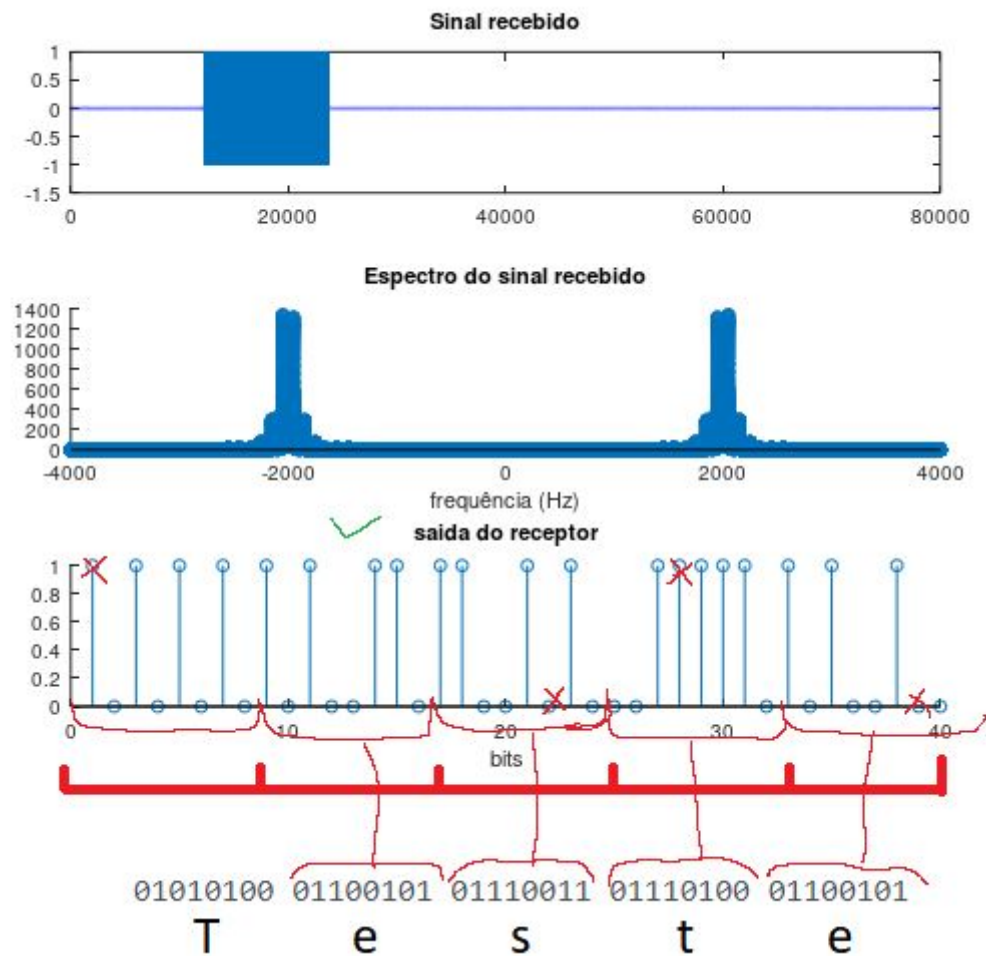


Imagem 8: segundo sinal no receptor, com bits errados

Após a simulação, foi dividido o sinal obtido na saída do receptor em bytes e analisado bit a bit, onde se pode observar que em todos os bytes um bit estava errado, com exceção do segundo byte que representa a primeira letra “e” em “Teste”. Para corrigir esses problemas, poderia ter sido utilizada uma taxa menor de bits, e uma maior taxa de amostragem, de forma semelhante às placas de som, além de se utilizar algum método de detecção e correção de erros no receptor.

4. CONCLUSÃO

Pode-se concluir que o processo de transmissão e recepção de dados é a base do estudo da Comunicação de Dados. Durante o semestre da disciplina pode-se estudar muitos processos e técnicas utilizadas para a comunicação. O trabalho final da disciplina culminou na utilização de muitas dessas técnicas aprendidas durante o semestre para a compreensão e simulação dos processos. Além do trabalho final utilizar os conceitos aprendidos no final da disciplina como transmissão de dados, enlaces e comunicações síncronas, o trabalho foi modificado para utilizar os conceitos básicos, como a *Fast Fourier Transform* para obter o espectro de frequências do sinal enviado e recebido. Pode-se perceber também, que a detecção e correção de erros é essencial nos processos de comunicação de dados tanto com fio, quanto sem fio.

5. ANEXO (CÓDIGOS UTILIZADOS PARA AS SIMULAÇÕES)

TX:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Codigo do trabalho pratico final ELC 1046
% Aluno Vinicius Hardt Schreiner
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load nos PKG necessários                                %
clear all;
pkg load signal;
pkg load communications;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Definição da mensagem a ser enviada e frequencias %
msg = "Teste";
TB = 24; % Taxa de bits
Fp = 2000; % Frequência da portadora
Fa = 48000; % Taxa de amostragem

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Geração do sinal a ser transmitido                    %
ytx = transmissor(msg, TB, Fp, Fa);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Printa o tamanho o sinal                                %
disp(['Tamanho do sinal: ' num2str(length(ytx)) ' amostras'])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Transmissão do sinal usando a placa de som             %
soundsc(ytx,Fa);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calcula o tamanho do sinal e define os tempos         %
Lsig=length(ytx);
```

```
t=td:td:1;
```

%%%%%%%%
%%%%%%%%

```
% Calcula a Transformada Rapida de Fourier
```

```
Lfft=2^ceil(log2(Lsig)+1);
```

$$F_{\max} = 1/(2 \cdot t_d);$$

```
Faxis=linspace(-Fmax,Fmax,Lfft);
```

```
Xsig=fftshift(fft(ytx,Lfft));
```

%%%%%%%%
%%%%%%%%

% Plota o grafico do sinal transmitido	%
<pre> % Plota o grafico do sinal transmitido figure; plot(t, s, 'b'); axis([0 10 0 1]); title('Sinal Transmitido'); xlabel('Tempo (s)'); ylabel('Amplitude'); </pre>	

```
figure(1);
```

```
subplot(311); sfig1a=plot(ytx);
```

```
set(sfig1a,'LineWidth',2);
```

```
xlabel("");
```

```
title('Sinal transmitido');
```

%%%%%%%%%%
 %%%%%%%%%%

```
% Plota o grafico do espectro do sinal transmitido %
```

```
subplot(312); sfig1b=stem(Faxis(1:Lfft),abs(Xsig));
```

```
set(sfig1b,'LineWidth',1);
```

```
xlabel('frequência (Hz)');
```

```
title('Espectro do sinal transmitido');
```

RX:

%%%%%%%%%%%%%%
 %%%%%%%%%%

% Código do trabalho pratico final ELC 1046

% Aluno Vinicius Hardt Schreiner

%%%%%%%%%%%%%%
 %%%%%%%%%%

% Load nos PKG necessários	%
0	0
10	0
20	0
30	0
40	0
50	0
60	0
70	0
80	0
90	0
100	0

clear all:

pkg load signal;

pkg load communications;

%%%%%%%%%%%%%%
 %%%%%%%%%%

```

% Definição das taxas e frequencias da recepcao    %
TB = 16; % Taxa de bits
Fp = 2000; % Frequência da portadora
Fa = 48000; % Taxa de amostragem

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Captura do sinal de audio                        %
NS = 0;
NS_min = 0.1; % nível de sensibilidade - ajustar conforme necessário
t_captura = 10; % tempo de captura do sinal
while NS < NS_min
    y = record(t_captura, Fa);
    if ~isempty(y)
        NS = max(y);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calcula o tamanho do sinal e define os tempos    %
Lsig=length(y);
td=1/8000;
t=td:td:1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calcula a Transformada Rapida de Fourier          %
Lfft=2^ceil(log2(Lsig)+1);
Fmax=1/(2*td);
Faxis=linspace(-Fmax,Fmax,Lfft);
Xsig=fftshift(fft(y,Lfft));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Printa o valor de deteccao do sinal                %
disp(['Valor de detecção do sinal: ' num2str(NS)]);

yrx = receptor(y, TB, Fp, Fa);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plota o grafico do sinal recebido                  %
figure(1);

```

```
subplot(311); sfig1a=plot(y);
set(sfig1a,'LineWidth',2);
xlabel("");
title('Sinal recebido');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Plota o grafico do espectro do sinal recebido %
subplot(312); sfig1b=stem(Faxis(1:Lfft),abs(Xsig));
set(sfig1b,'LineWidth',1);
xlabel('frequência (Hz)');
title('Espectro do sinal recebido');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Plota a saida digital recebida %
subplot(313); sfig1b=stem(yrx);
set(sfig1b,'LineWidth',1);
xlabel('bits');
title('saida do receptor');
```

```
%x = bi2de(reshape(yrx,8,l))';
%msg = char(x);
%disp(["Mensagem recebida: " msg]);
```