

Teoria das Comunicações

ENE 167126

Profs. André Noll Barreto

Exercício de Simulação Nº 2

Breno Rodrigues Brito

08/25379

Parte 1

1.1 Obtenha duas amostras de voz em formato .wav

1.2 Faça a multiplexação dos dois sinais utilizando modulação SSB. Um dos sinais pode ser transmitido em banda base (sem modulação), enquanto o outro pode ser transmitido com modulação USB com frequência de portadora de 4kHz. Explique os passos desta multiplexação.

Se obteve o sinal analítico discreto no tempo através do código *hilbert()* no MatLab e a partir dele se encontrou a transformada de Hilbert e os sinais y_{1+} e y_{1-} e foi modulado USB usando a fórmula

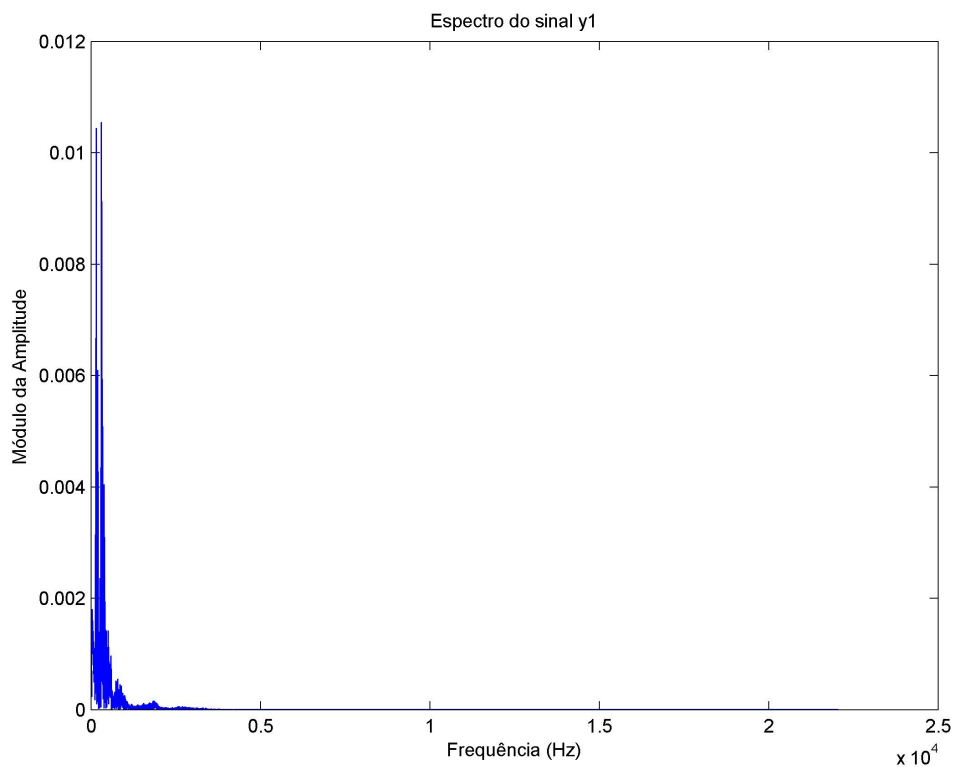
$$y_{1\text{USB}} = y_{1+} * \exp(i * 2 * \pi * f_c * t_1) + y_{1-} * \exp(-i * 2 * \pi * f_c * t_1) \text{ com } f_c = 4\text{kHz}.$$

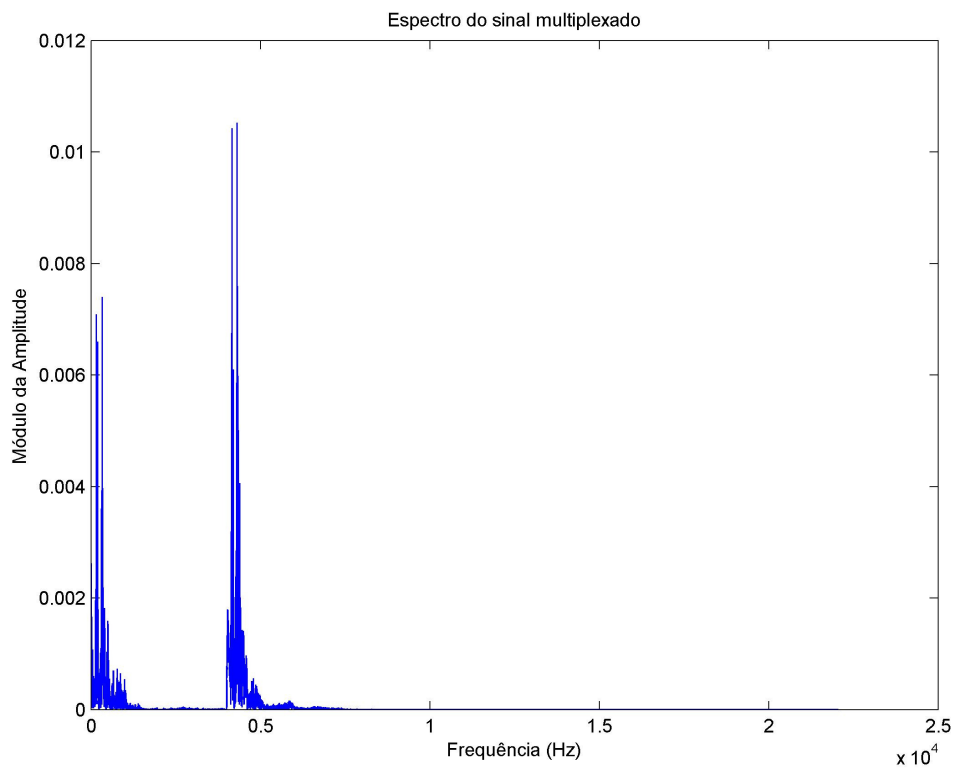
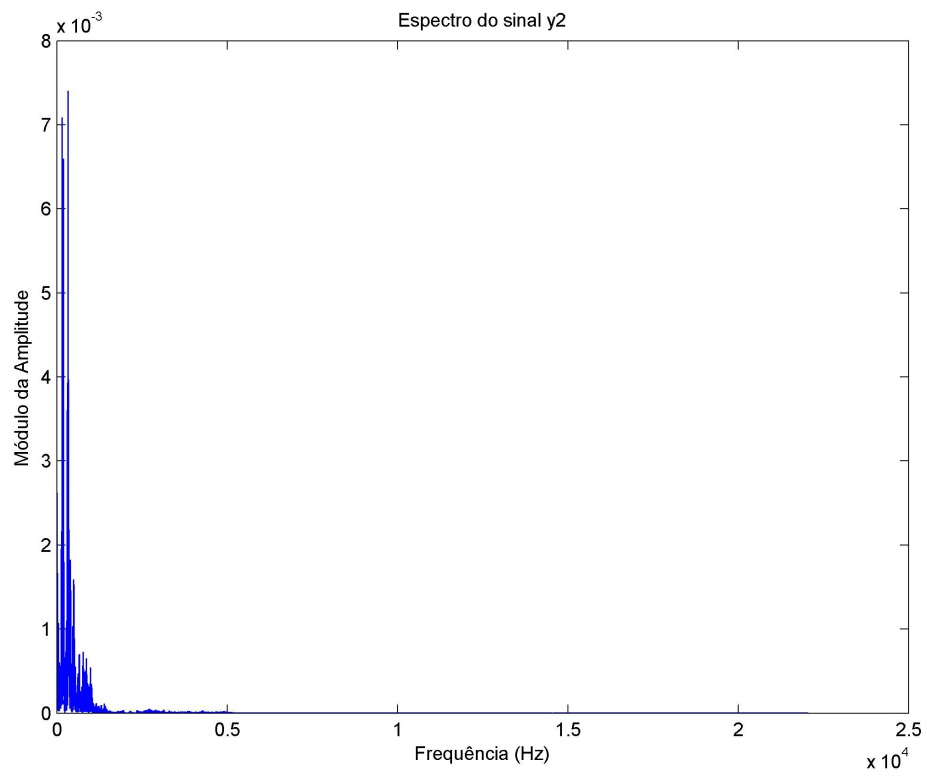
Para se obter o sinal multiplexado, bastou-se somar os outros dois

$$y_{\text{mux}} = y_{1\text{USB}} + y_2$$

1.3 Mostre o espectro dos sinais originais, assim como o do sinal multiplexado.

Para isso se fez a transformada rápida de Fourier de y_1 , y_2 e y_{mux} usando o código *fft()* e se plotou nos gráficos a seguir:

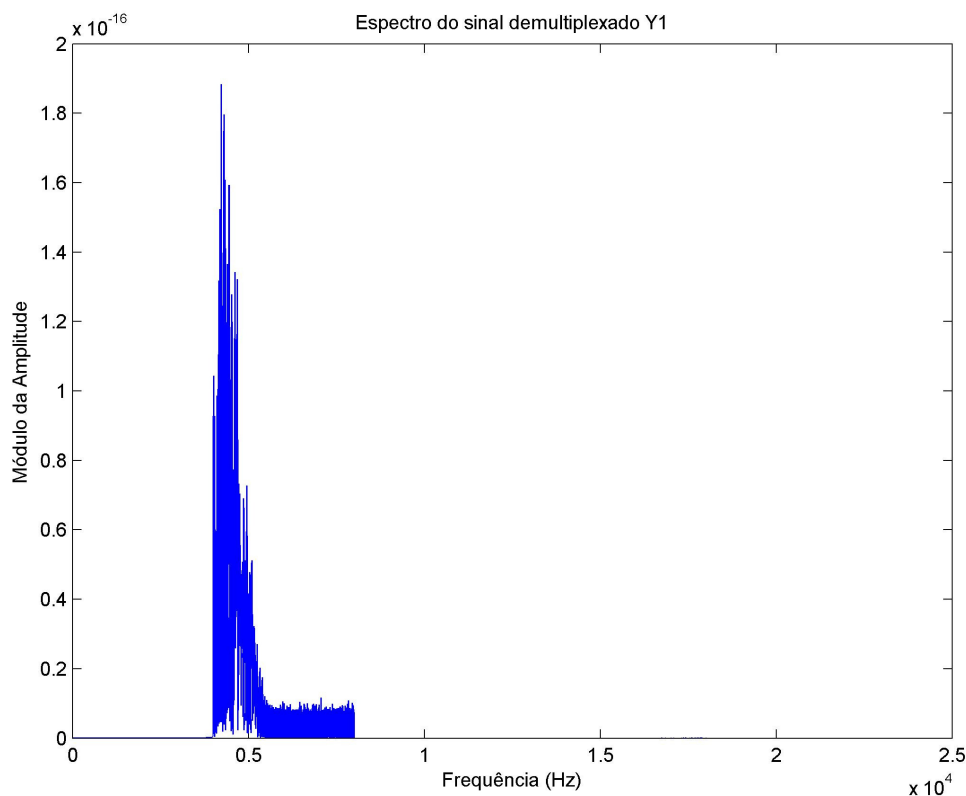


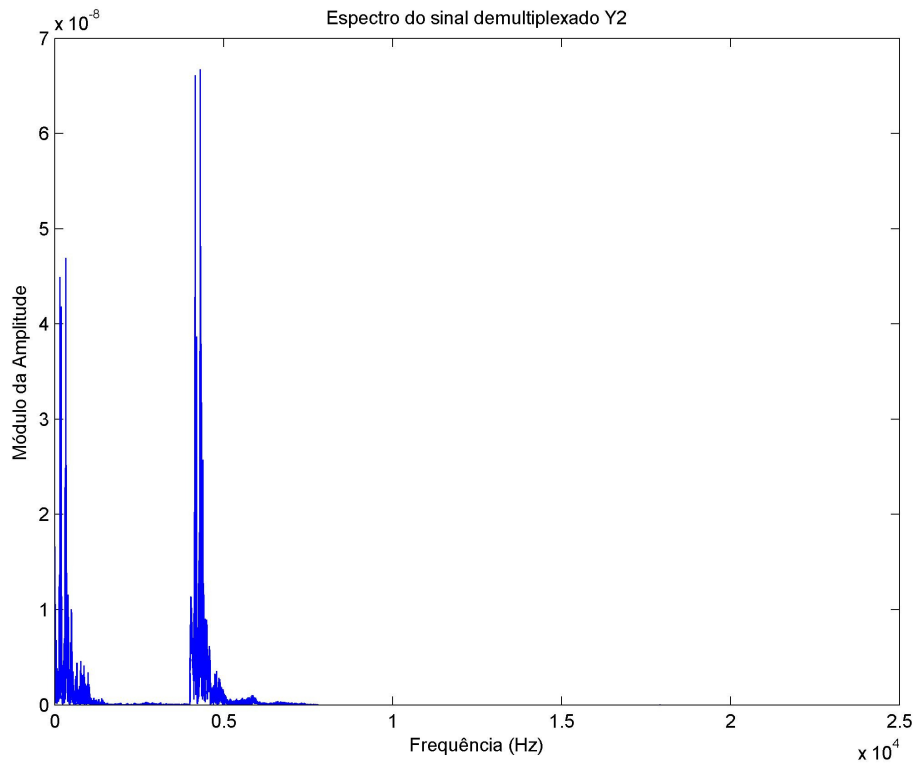


1.4 Faça a demultiplexação do sinal, ou seja, recupere os sinais originais a partir do sinal multiplexado. Expliquem o processo. Como vocês avaliam a qualidade de voz?

Primeiro de tudo, usou-se um filtro passa baixas digital para se extrair o sinal y_2 . Tomou-se o devido cuidado para não se apagar a parte simétrica do sinal. O sinal resultante foi normalizado para não haver clipagem do som ao se converter no formato wave. A parte filtrada do sinal foi justamente y_{1USB} . Aproveitando o processo, obteve-se este sinal modulado. Para demodulá-lo, bastou-se multiplicá-lo por $\cos(2\pi f_c t)$ para seu espectro retornar à origem do gráfico. Porém, isso gera harmônicos indesejados. Bastou um outro filtro passa baixas digital para “limpar” o espectro e gravar o som com a função *wavwrite()* não se esquecendo da normalização para não houver clipagem do sinal.

1.5 Mostre o espectro dos sinais demultiplexados.





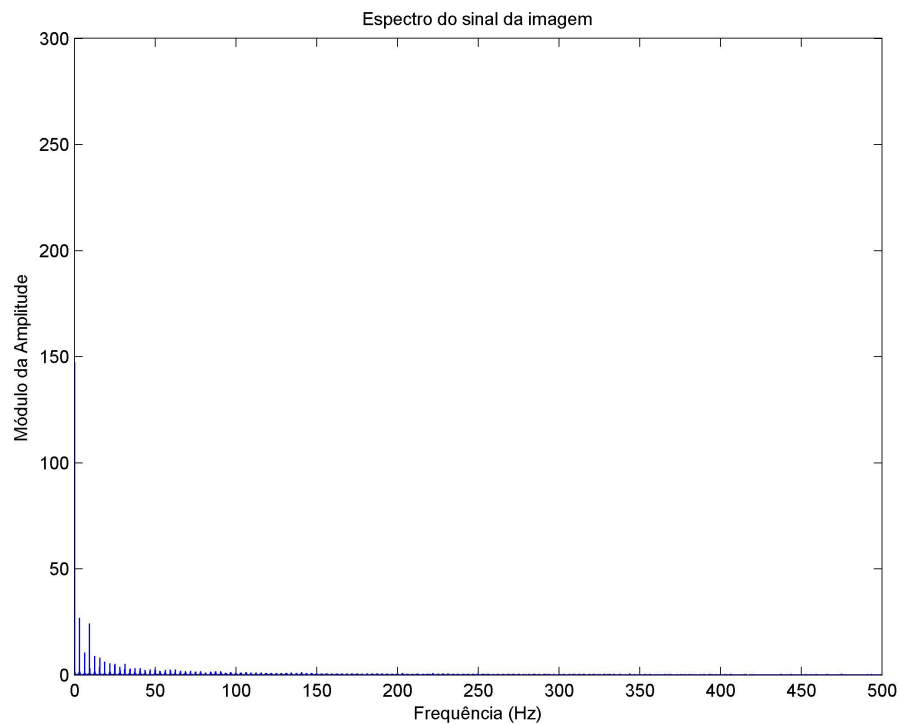
Observação: O gráfico da demultiplexação de Y1 deveria estar “colado” na origem e o gráfico da demultiplexação de Y2 não deveria ter este segundo pico. Não consegui arrumar isso no código.

Parte 2

2 Modulação de sinais de imagem

2.1 Importe uma imagem qualquer em formato bitmap. Mostre seu espectro, supondo que as amostras (pixels) são transmitidas a intervalos de 1ms.

Foi usada para isso a função *imread()* para ler a imagem e *reshape()* para transformar a matriz 3D em matriz coluna. Sua frequência será de 1kHz. Seu espectro está mostrado logo abaixo da imagem importada.



2.2 Com base no espectro obtido, estime a largura de banda essencial B do sinal (99,9999% da energia). Observação: originalmente, se pedia o valor de 99%, mas como a componente DC é muito maior que as outras, a banda acabava sendo zero.

A banda $B=0,0038\text{Hz}$. Ainda assim é muito pequena graças ao tamanho da componente DC. Não dá para ver direito, mas há uma grande componente em $f=0\text{Hz}$.

2.3 Faça a modulação do sinal em uma portadora de frequência 8B, usando os seguintes esquemas. Descreva o que foi feito.

2.3.1 AM-SSB-USB

Novamente foi feita a transformada de Hilbert encontrada no o sinal analítico discreto no tempo através do código *hilbert()* no MatLab e foi modulado USB conforme a fórmula já exposta:

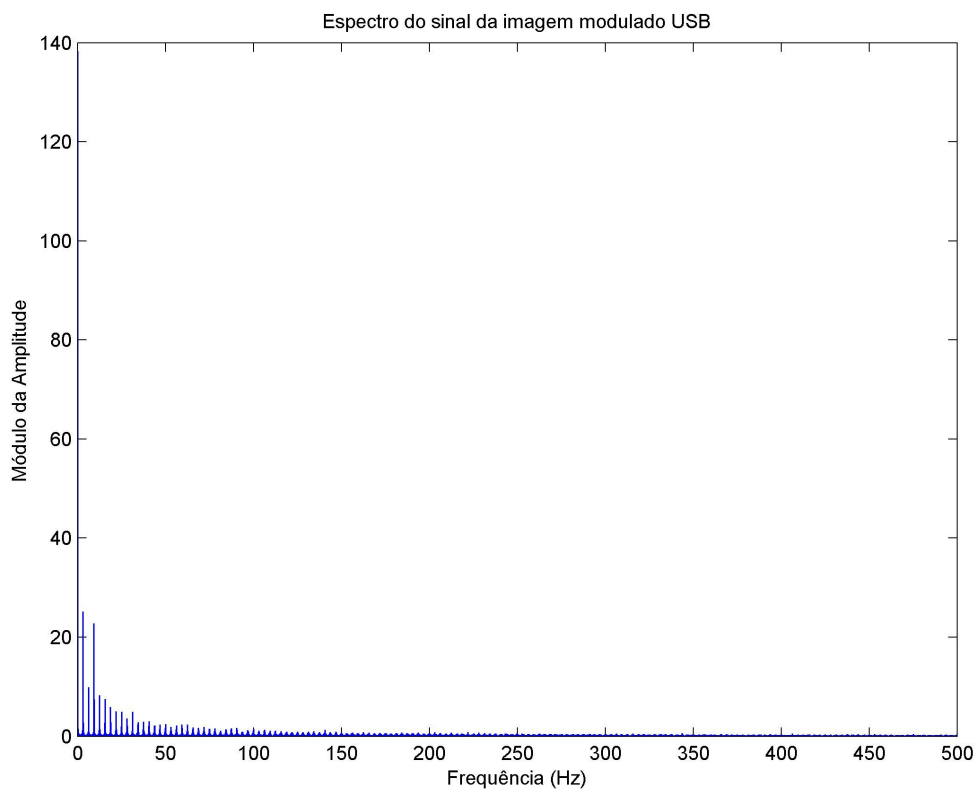
$$y_{1\text{USB}} = y_{1+} * \exp(i * 2 * \pi * f_c * t_1) + y_{1-} * \exp(-i * 2 * \pi * f_c * t_1) \text{ com } f_c = 4\text{kHz}.$$

Sendo que $f_c = 8\text{B}$.

2.3.2 AM-VSB (use o filtro que achar mais adequado)

2.4 Mostre o espectro dos sinais modulados.

Novamente foi realizada a FFT para a obtenção dos espectros.



2.5 Faça a demodulação destes sinais e compare a imagem demodulada nos dois casos com a imagem original. Mostre-as no relatório.

2.6 Mostre também o espectro das mensagens demoduladas nos dois casos.

Código

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Simulação 2 de TECOM
%Modulação SSB de sinais de voz
%1.1 Obtenha duas amostras de voz em formato .wav

[y1,Fs1,nbits1]=wavread('BRB');
[y2,Fs2,nbits2]=wavread('simulacao_de_tecom');

%igualando os tamanhos dos vetores
if length(y1)>length(y2)
    diferenca=length(y1)-length(y2);
    coluna=zeros(diferenca,1);
    y2=[y2;coluna];
else
    if length(y1)<length(y2)
        diferenca=abs(length(y1)-length(y2));
        coluna=zeros(diferenca,1);
        y1=[y1;coluna];
    end
end

L=length(y1);
t1=0:1/Fs1:(L-1)/Fs1;
t1=t1';
if Fs1~=Fs2
    t2=0:1/Fs2:(L-1)/Fs2;
    t2=t2';
end

%1.2 Faça a multiplexação dos dois sinais utilizando modulação SSB.
% Um dos sinais pode ser transmitido em banda base (sem modulação),
% enquanto o outro pode ser transmitido com modulação USB com
% frequência de portadora de 4kHz. Explique os passos desta
multiplexação.
```



```

%fazendo a transformada de Hilbert e pegando a parte USB
y1_mais=hilbert(y1)/2;
y1_menos=conj(y1_mais);
y1h=imag(y1_mais);
fc=4e3;

y1_USB=y1_mais.*exp(1i*2*pi*fc.*t1)+y1_menos.*exp(-1i*2*pi*fc.*t1);
mux_y1_USB_y2=y1_USB+y2;

```

%1.3 Mostre o espectro dos sinais originais, assim como o do sinal multiplexado.

```

NFFT = 2^nextpow2(L);
Ys1 = fft(y1,NFFT)/L;
f = Fs1/2*linspace(0,1,NFFT/2+1);

```

```

plot(f,2*abs(Ys1(1:NFFT/2+1)));
title('Espectro do sinal y1');
xlabel('Frequência (Hz)');
ylabel('Módulo da Amplitude');
print -djpeg -r300 Espectroy1

```

```

NFFT = 2^nextpow2(L);
Ys2 = fft(y2,NFFT)/L;
f = Fs2/2*linspace(0,1,NFFT/2+1);

```

```

plot(f,2*abs(Ys2(1:NFFT/2+1)))
title('Espectro do sinal y2')
xlabel('Frequência (Hz)')
ylabel('Módulo da Amplitude')
print -djpeg -r300 Espectroy2

```

```

NFFT = 2^nextpow2(L);
Ysmux = fft(mux_y1_USB_y2,NFFT)/L;
f = Fs1/2*linspace(0,1,NFFT/2+1);

```

```

plot(f,2*abs(Ysmux(1:NFFT/2+1)))
title('Espectro do sinal multiplexado')
xlabel('Frequência (Hz)')
ylabel('Módulo da Amplitude')
print -djpeg -r300 EspectroYmux

```

%1.4 Faça a demultiplexação do sinal, ou seja, recupere os sinais originais a partir do
 % sinal multiplexado. Expliquem o processo. Como vocês avaliam a
 qualidade de
 % voz?

```

ft = Fs1/2*linspace(0,1,NFFT);

```

```

Ys2demux=Ysmux;
Ys1USBdemux=zeros(size(Ysmux));
for n=1:length(Ysmux)
    if (ft(n)>4e3)&&(ft(n)<(max(ft)-4e3))
        Ys1USBdemux(n)=Ysmux(n);
        Ys2demux(n)=0;
    end
end

```

```

end
y2demux=real(ifft(Ys2demux,NFFT));
convy2=(max(abs(y2))/(max(abs(y2demux))));
y2normdemux=y2demux*convy2;
wavwrite(y2normdemux,Fs2,nbits1,'simulacao_demux');

```

```

y1USBdemux=ifft(Ys1USBdemux,NFFT);
td=0:1/Fs1:(length(y1USBdemux)-1)/Fs1;
td=td';
y1demoddemux=y1USBdemux.*cos(2*pi*fc.*td);
Ys1demoddemux=fft(y1demoddemux,NFFT)/L;

```

```

W=Ys1demoddemux;
for n=1:length(Ys1demoddemux)
    if (ft(n)>4e3)&&(ft(n)<(max(ft)-4e3))
        W(n)=0;
    end
end
y1demux=real(ifft(W,NFFT));
convy1=(max(abs(y1))/(max(abs(y1demux))));
y1normdemux=y1demux*convy1;
wavwrite(y1normdemux,Fs2,nbits1,'BRB_demux');

```

```

%1.5 Mostre o espectro dos sinais demultiplexados.
% %

```

```

NFFT = 2^nextpow2(L);
f = Fs1/2* linspace(0,1,NFFT/2+1);

```

```

plot(f,2*abs(W(1:NFFT/2+1)))
title('Espectro do sinal demultiplexado Y1')
xlabel('Frequência (Hz)')
ylabel('Módulo da Amplitude')
print -djpeg -r300 EspectroDemuxY1

```

```

NFFT = 2^nextpow2(L);
f = Fs1/2* linspace(0,1,NFFT/2+1);

```

```

plot(f,2*abs(Ys2demux(1:NFFT/2+1)))
title('Espectro do sinal demultiplexado Y2')
xlabel('Frequência (Hz)')
ylabel('Módulo da Amplitude')
print -djpeg -r300 EspectroDemuxY2

```

```

% 2 Modulação de sinais de imagem
% 2.1 Importe uma imagem qualquer em formato bitmap. Mostre seu
espectro,
% supondo que as amostras (pixels) são transmitidas a intervalos de
lms.

```

```

img=imread('tsunami.bmp');
[x,y,z]=size(img);
imgcol=reshape(img,1,[],1);
Fsimg=1/1e-3;
Limg=x*y*z;

```

```

NFFT = 2^nextpow2(Limg);

```

```

Ysimg = fft(imgcol,NFFT)/Limg;
f = Fsimg/2*linspace(0,1,NFFT/2+1);

plot(f,2*abs(Ysimg(1:NFFT/2+1)))
title('Espectro do sinal da imagem')
xlabel('Frequência (Hz)')
ylabel('Módulo da Amplitude')
print -djpeg -r300 Yimagem

% 2.2 Com base no espectro obtido, estime a largura de banda essencial
B do
% sinal (99,9999% da energia).

Pot=abs(sum(Ysimg.^2));
Pot_aprox=0;
n=0;
while (Pot_aprox/Pot) < 0.999999
    n=n+1;
    Pot_aprox=Pot_aprox+Ysimg(n).^2;
end
B=f(n+1);

% 2.3 Faça a modulação do sinal em uma portadora de frequência 8B,
usando
% os seguintes esquemas. Descreva o que foi feito.
% 2.3.1 AM-SSB-USB
%ajustando o tamanho para a FFT
% W=zeros(1,NFFT);
% W(1:length(imgcol))=imgcol(1:length(imgcol));
% imgcol=W;
% Limgcol= length(imgcol);

timg=0:1/Fsimg:(length(imgcol)-1)/Fsimg;

img_mais=hilbert(imgcol)/2; %Transformada de Hilbert
img_menos=conj(img_mais); %
imggh=imag(img_mais); %

%modulação
img_USB=img_mais.*exp(1i*2*pi*8*B.*timg)+img_menos.*exp(-
1i*2*pi*8*B.*timg);

% 2.3.2 AM-VSB (use o filtro que achar mais adequado)

% Gerando espectro DSB
% fc=8*B; %Frequencia da portadora
%
% Fsm=2*(fc+Fsimg/2);
%
% Tsm=1/Fsm;
%
% t=0:Tsm:(Limgcol)/Fsm-Tsm;
% imgdsb=imgcol.*cos(2*pi.*fc*t); %Modulando o sinal

% Limgdsb=length(imgdsb);
%
% NFFT = 2^nextpow2(Limgdsb); %N elementos FFT DSB
% espectro_dsb = fftshift(fft(imgdsb,NFFT)/Limgdsb);

```

```

%
% 2.4 Mostre o espectro dos sinais modulados.

NFFT = 2^nextpow2(length(img_USB));
Ysimg_USB = fft(img_USB,NFFT)/length(img_USB);
f = Fsimg/2*linspace(0,1,NFFT/2+1);

plot(f,2*abs(Ysimg_USB(1:NFFT/2+1)))
title('Espectro do sinal da imagem modulado USB')
xlabel('Frequência (Hz)')
ylabel('Módulo da Amplitude')
print -djpeg -r300 ImgUSB

% 2.5 Faça a demodulação destes sinais e compare a imagem demodulada
nos
% dois casos com a imagem original. Mostre-as no relatório.

td=0:1/Fs1:(length(img_USB)-1)/Fs1;
% td=td';
imgdemod=img_USB.*cos(2*pi*fc*td);
Limg=length(imgdemod);
NFFT = 2^nextpow2(Limg);
Yimgmod = fft(imgdemod,NFFT)/Limg;
f = Fsimg/2*linspace(0,1,NFFT/2+1);

Yimgdemod=Yimgmod;

ft = Fsimg/2*linspace(0,1,NFFT);
%filtro
for n=1:length(Yimgdemod)
    if (ft(n)>fc)&&(ft(n)<(max(ft)-fc))
        Yimgdemod(n)=0;
    end
end

img_demod=ifft(Yimgdemod,NFFT);
imgfinal=reshape(img_demod,[],y,z);
imwrite(imgfinal,'imgSSBdemod','jpeg');

% 2.6 Mostre também o espectro das mensagens demoduladas nos dois
casos.
plot(f,2*abs(Yimgdemod(1:NFFT/2+1)))
title('Espectro do sinal de imagem demodulado')
xlabel('Frequência (Hz)')
ylabel('Módulo da Amplitude')
print -djpeg -r300 Espectroimgdemod

```