

Universidade Federal de Santa Catarina
Departamento de Engenharia Elétrica e Eletrônica

Trabalho 2- Processamento Digital de Sinais

O software Matlab vem com um arquivo de áudio de alguém dizendo “Matlab”. O comando `load (mtlb)` carrega este arquivo. Ouça o som original com o comando `soundsc(mtlb)`. Este áudio foi obtido utilizando uma frequência de amostragem de $F_s = 7418$ Hz.

Faça um gráfico desse sinal utilizando os seguintes comandos:

```
>> clear all, close all
```

```
>> load mtlb
```

```
>> who
```

Your variables are:

```
Fs      mtlb
```

```
>> L=length(mtlb);
```

```
>> plot([1:L]/Fs,mtlb)
```

```
>> axis tight
```

```
>> xlabel('Tempo (segundos)')
```

O arquivo `NoisySpeech.txt` disponibilizado no moodle, contém o sinal `mtlb` corrompido por ruído aditivo. Trace este sinal utilizando os seguintes comandos:

```
>> load('NoisySpeech.txt');
```

```
>> x= NoisySpeech;
```

```
>> L=length(x);
```

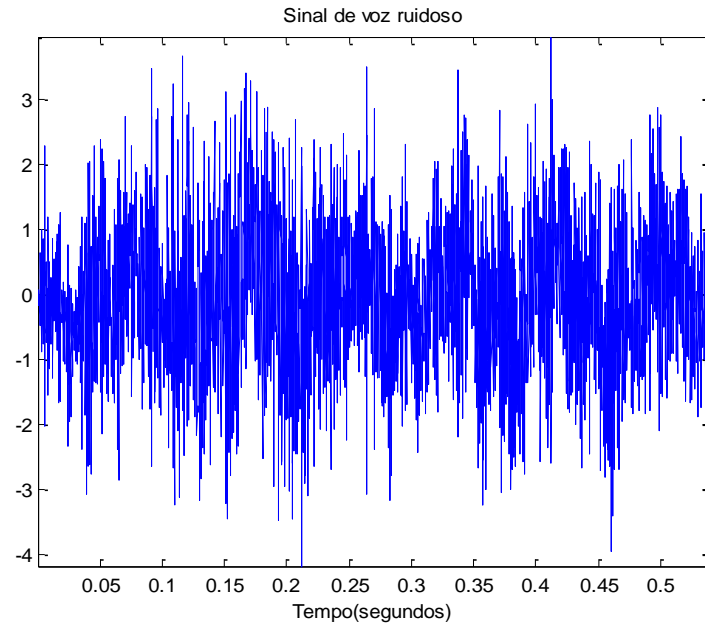
```
>> plot([1:L]/Fs,x)
```

```
>> axis tight
```

```
>> title('Sinal de voz ruidoso')
```

```
>> xlabel('Tempo (segundos)')
```

Você deverá obter o gráfico mostrado abaixo.



Use o comando `soundsc(x)` e escute o sinal.

Tarefas:

1 – Trace o espectro do sinal sem e com ruído (utilize os comandos `fft` e `fftshift` com um número de pontos conveniente). Quais são as frequências predominantes no sinal sem ruído? Você pode identificar que parte do espectro corresponde ao sinal e qual corresponde ao ruído?

Obs: O comando `fft(y,N)` (Fast Fourier Transform) calcula a transformada de Fourier do sinal $y(t)$ amostrado a cada $1/F_s$ segundos. **y** é um vetor que contém os valores das amostras. **N** é o número de pontos espectrais. Normalmente, o valor de **N** é uma potência de 2. Note que, como o comando `fft` calcula *pontos espectrais*, o espectro é discreto. Porém, espectro discreto é característico de sinais periódicos, assim o espectro calculado pela `fft` diz respeito a um sinal periódico com um determinado período T . Evidentemente, este período deve ser maior do que a duração do sinal original (não periódico). A duração do nosso sinal de áudio é de aproximadamente $d = 0,54s = 4010/7418 s$ (número de amostras do sinal dividido pela frequência de amostragem). Isso pode ser verificado pela figura acima. Dessa maneira, T deve ser maior do que $0,54s$. Vamos considerar $T=2d = 1,08 s$. Assim, o vetor de sinal utilizado para calcular a `fft` deve ter um número de amostras de, no mínimo, 8020. Considerando que a potência de 2, logo acima de 8020 é 8192, utilize esse número de amostras no comando `fft`. Então, o vetor utilizado para o cálculo da transformada de Fourier será composto das 4010 amostras do sinal de áudio seguidas de 4182 zeros (isto é denominado de *zero padding*). Neste caso, os comandos utilizados para gerar o espectro dos sinais **mtlb** e **x** (NoisySpeech) são:

```
>> M=fft(mtlb,8192); % espectro entre 0 e  $F_s$  Hz
```

```
>> M=fftshift(M); % espectro entre  $-F_s/2$  e  $F_s/2$ 
```

```
>> R=fft(x,8192);
```

```
>> R=fftshift(R);
```

```
>> f=(-4096:4095)/8192* $F_s/2$ ;
```

```
>> plot(f,abs(R))
>> xlabel('Frequência (Hz)')
>> grid
>> hold on
>> plot(f,abs(M), 'r')
```

2 – Projete um filtro passa-baixas Butterworth para atenuar o ruído de maneira adequada.

Obs: A resposta em frequência de um filtro Butterworth é dada por

$$H(f) = \frac{1}{1 + j \left(\frac{f}{f_p} \right)^n}$$

onde n é a ordem do filtro e f_p é a frequência de corte. A partir do espectro do sinal e do ruído, escolha uma f_p . Escolha aquela que você achar a mais adequada. Filtre e escute o sinal para $n = 2, 4, 6$ e 8 . Trace a magnitude do espectro do sinal filtrado para cada n . Um exemplo de comandos Matlab é dado abaixo.

```
>> n = 5;
>> fp = 500;
>> H=1./(1+j*(f/fp).^n);
>> Y=H.'*R; % espectro do sinal de saída do filtro
>> plot(f,abs(Y))
>> xlabel('Frequência (Hz)')
>> Y=fftshift(Y);
>> y=real(ifft(Y));
>> soundsc(y)
```

Faça uma discussão sobre os resultados que você obteve!