

UNIVERSIDADE FEDERAL DE SANTA CATARINA - UFSC  
CENTRO TECNOLÓGICO - CTC  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA -  
DEEL  
EEL7522 - PROCESSAMENTO DIGITAL DE SINAIS

TRABALHO 2  
ANÁLISE ESPECTRAL

ANDRÉ MARTINS PIO DE MATTOS  
GUSTAVO SIMAS DA SILVA

FLORIANÓPOLIS  
OUTUBRO, 2020

## **SUMÁRIO**

<b>INTRODUÇÃO</b>	<b>2</b>
<b>TAREFAS</b>	<b>3</b>
<b>CONSIDERAÇÕES FINAIS</b>	<b>12</b>
<b>REFERÊNCIAS</b>	<b>13</b>
<b>ANEXO I</b>	<b>14</b>
<b>ANEXO II</b>	<b>16</b>

## **INTRODUÇÃO**

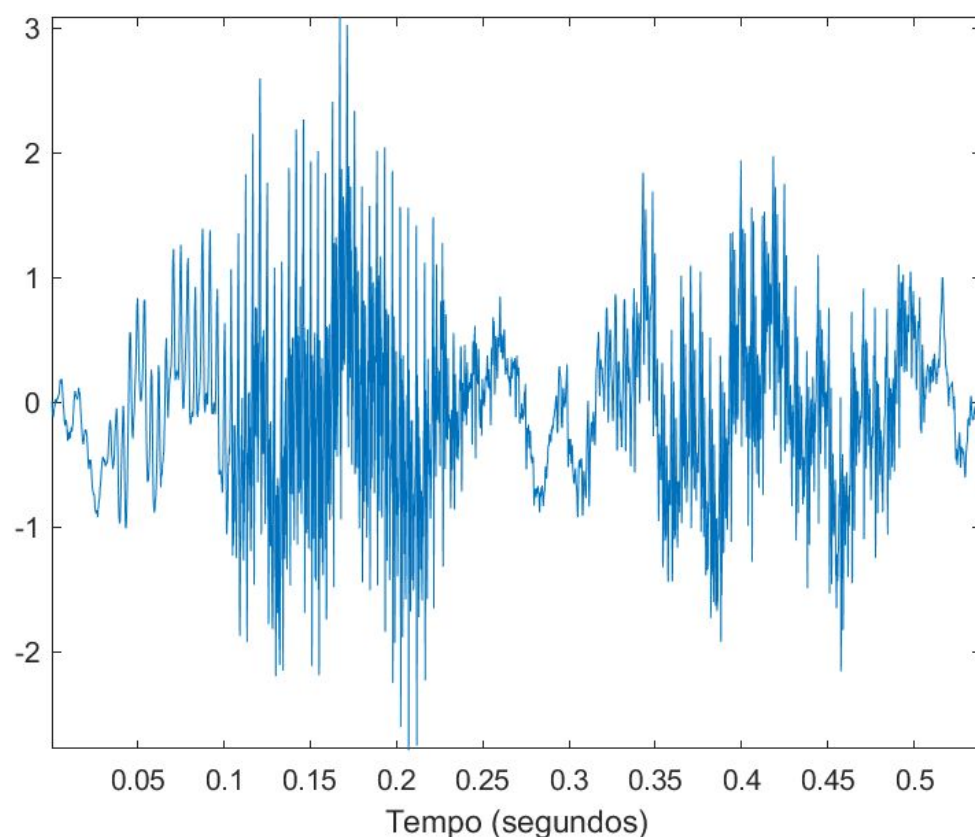
A série de Fourier foi desenvolvida por Jean Baptiste Joseph Fourier (1768-1830) e decompõe uma função periódica ou sinal periódico numa soma de funções sinusoidais ou de exponenciais complexas. O cálculo e estudo da série de Fourier são denominados análise de harmônicos [1]. A série de Fourier tem muitas aplicações em diversas áreas do conhecimento, tais como, engenharia elétrica, análise de vibrações, acústica, processamento de sinais, entre outros problemas físicos e matemáticos da engenharia [2].

O objetivo desta experiência é realizar a análise espectral de sinais no domínio da frequência, mais especificamente sinais de áudio na faixa de frequência da voz com ruído adicionado. E, então, verificar o uso de filtro Butterworth em diferentes ordens para avaliar o desempenho do sistema de forma geral e suas condições apresentadas mediante variação de parâmetros.

## 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?

Primeiramente se organizou o código implementado, com diferentes seções a serem analisadas. O código completo para esta parte está presente na seção ANEXO I. O gráfico temporal do sinal de voz integrado do Matlab de comprimento total 0,5 s, é apresentado na Figura 1.

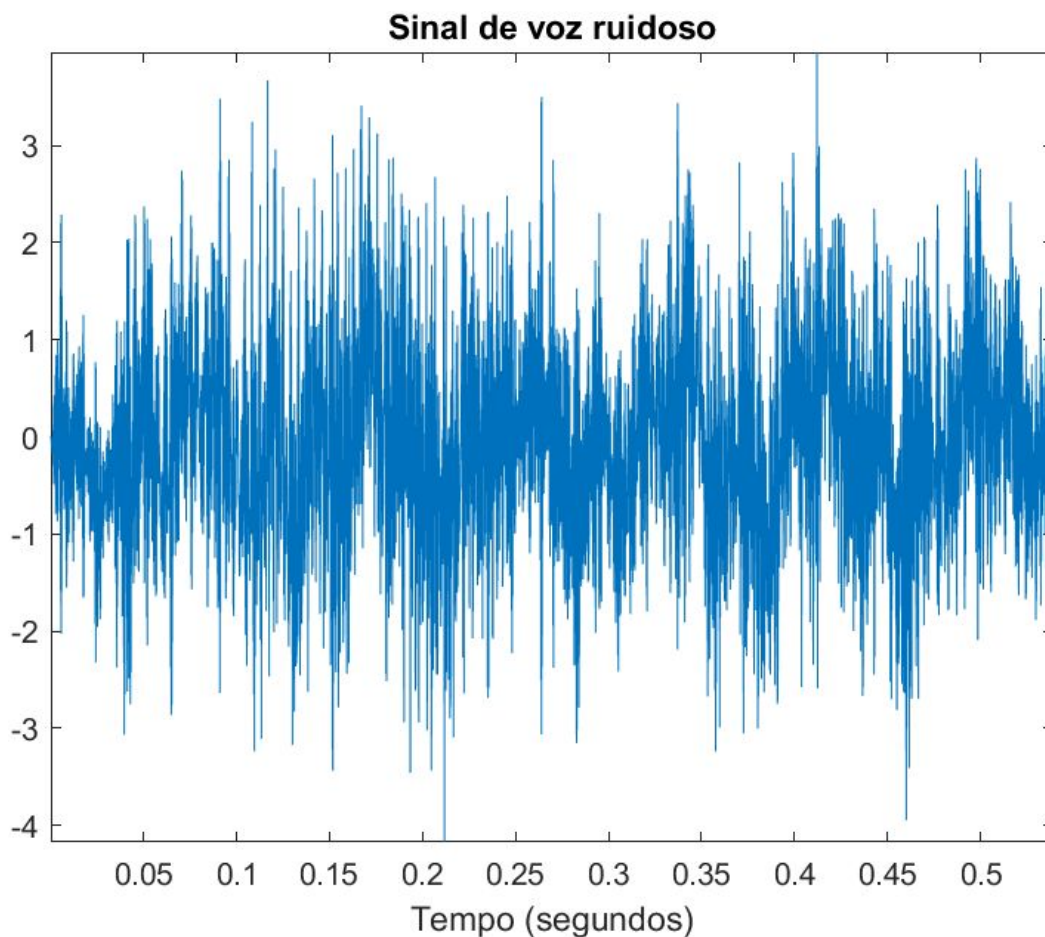


**Figura 1 - Sinal de voz com pessoa pronunciando “matlab”**

Pela reprodução do sinal de voz “puro” pronunciando a palavra “matlab”, percebe-se que a voz é aguda sendo que, provavelmente, apresentará componentes de

elevada frequência em termos da capacidade do trato vocal humano (aproximadamente de 50 Hz até 3400 Hz).

Em seguida, verificou-se o sinal de voz acrescido de ruído. O gráfico resultante (temporal/transitório) para o sinal de voz ruidoso do arquivo NoisySpeech é apresentado na Figura 2.



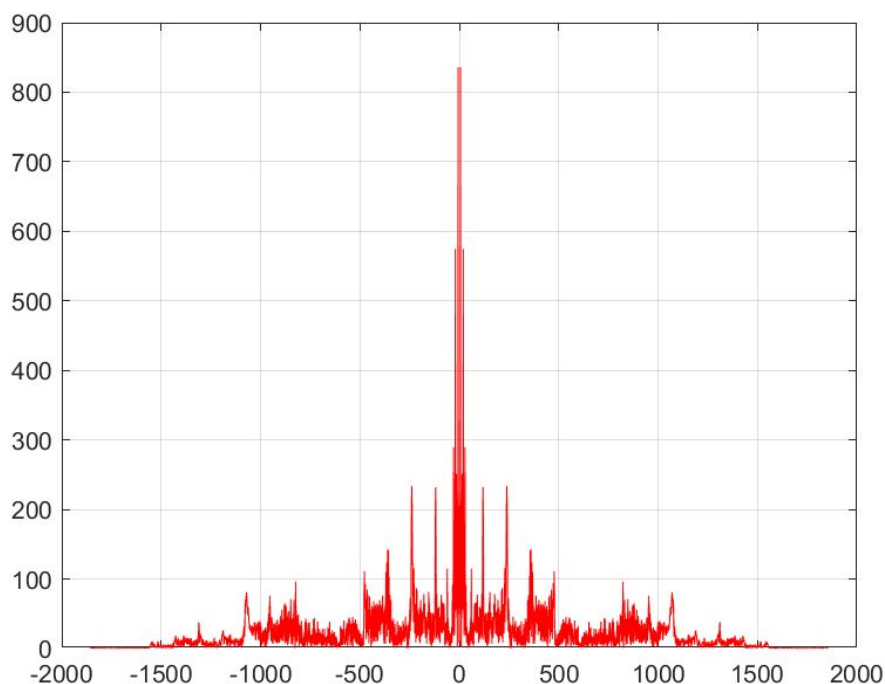
**Figura 2 - Sinal de voz ruidoso temporal arquivo Noisy Speech**

Com a reprodução do sinal, utilizando a função `soundsc(x)`, foi possível identificar o som vocalizado, apesar do proeminente ruído semelhante a um apito. Por apresentar tal característica de similaridade a um apito, estima-se (antes de se visualizar o espectro de frequências do sinal resultante), que o ruído estará de maior forma presente em frequências elevadas, quiçá acima da extensão vocal verificada, ou seja, sem sobreposição em demasia ao sinal original.

Portanto, prosseguiu-se com a análise em domínio da frequência, utilizando as funções `fft` e `fftshift`.

Observa-se que 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).

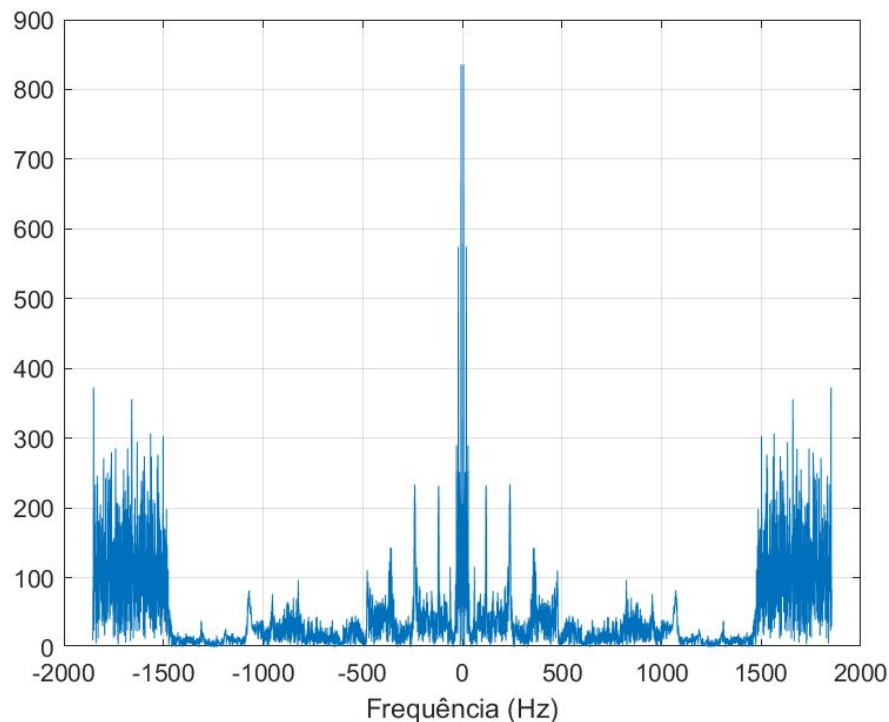
O espectro de frequências do sinal original (sem ruído) é apresentado na Figura 3.



**Figura 3 - Espectro de Frequência do sinal “matlab” original**

Neste espectro identificam-se valores de frequência representativos do som ouvido. O sinal possui extensão aparente até cerca de 1500 Hz (não ocupando a faixa total de produção vocal humana, nem muito dos 20 kHz possíveis da audição). A unidade dos valores do eixo-y não são relevantes para esta análise no momento.

Assim, foi executado o mesmo procedimento agora para o sinal com ruído acrescido, tendo como resultante o gráfico da Figura 4.



**Figura 4 - Espectro de Frequência do sinal com ruído acrescido**

Neste gráfico há a distinção de haver componentes espectrais de ruído predominantes acima dos 1500 Hz, sendo até aproximadamente 2kHz conforme avaliado na extensão do gráfico. Dessa forma é possível distinguir entre os componentes de frequência referentes à voz e aqueles referentes ao ruído, pois não há aliasing ou mascaramento do sinal original. Tal fato comprova a hipótese proposta inicialmente ao se ouvir o som produzido do arquivo Noisy Speech, de forma tal que foi possível identificar e distinguir pela audição a voz pronunciando “matlab” do ruído em semelhança com um “apito”, sem distorção aparente do sinal original, apenas simultaneidade de reprodução.

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

A resposta em frequência de um filtro Butterworth é dada por

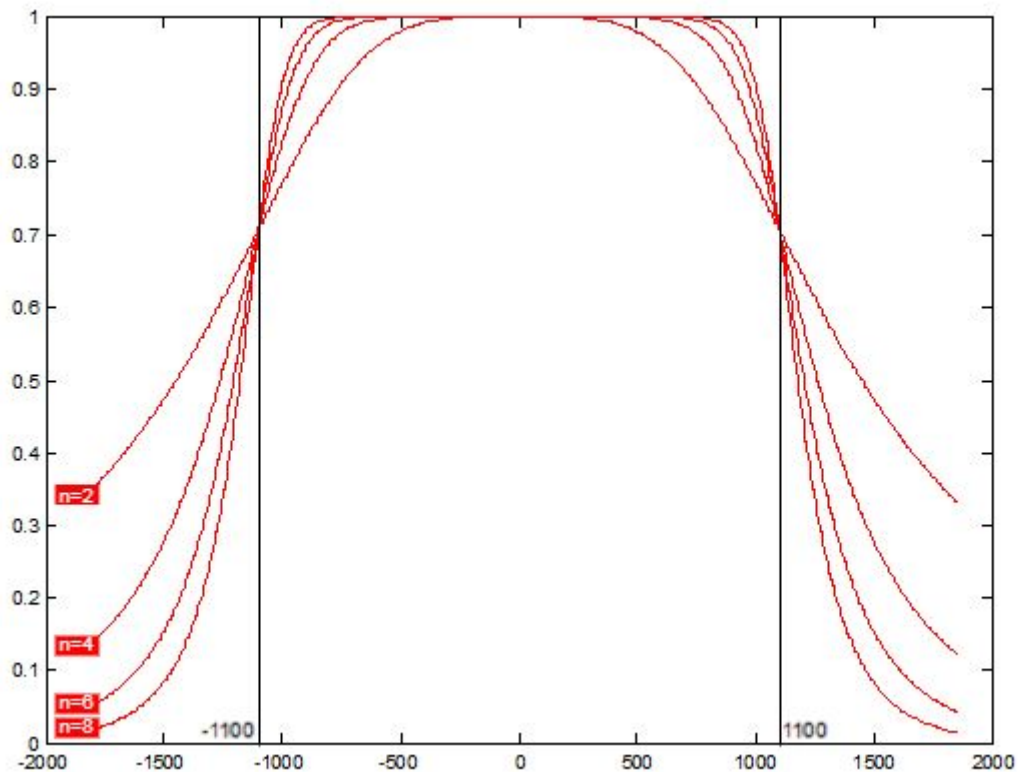
$$H(f) = \frac{1}{1+j(\frac{f}{f_p})^n}$$

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 no código em ANEXO II.

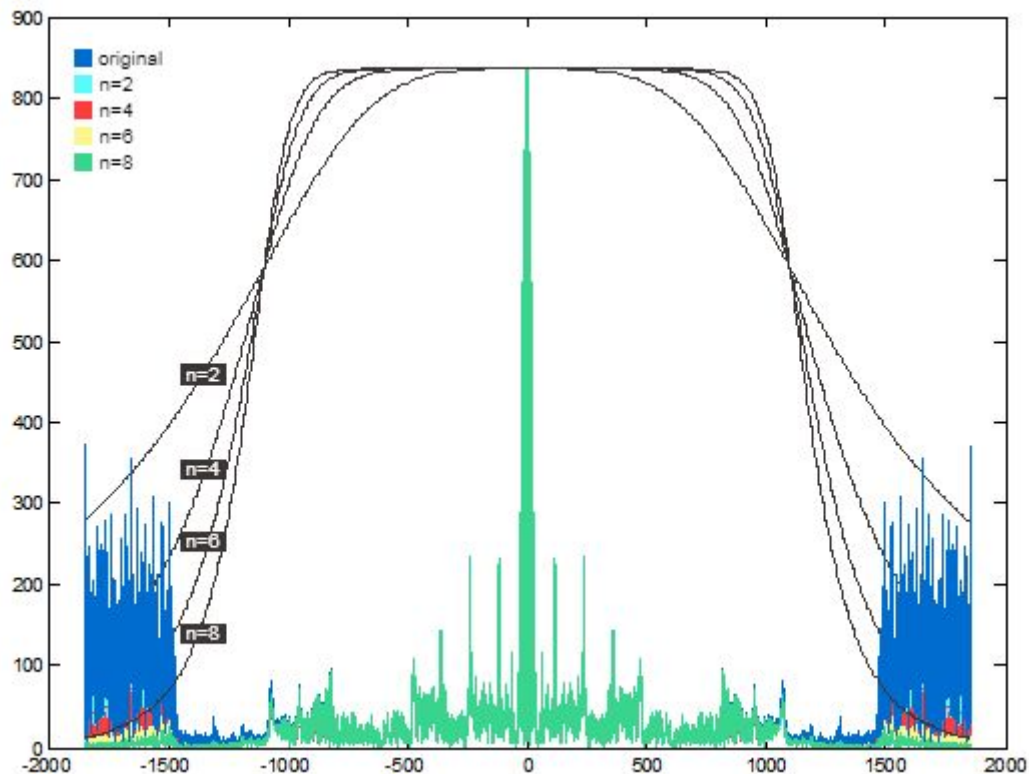
Para a construção desse filtro foram analisadas alguns parâmetros importantes do sinal original e com ruído, como: faixa de frequência do sinal de interesse, faixa de frequência do ruído e contribuição de amplitudes para identificação audível do sinal. Também, foi importante avaliar graficamente a resposta do filtro Butterworth de diferentes ordens para identificar qual frequência de corte ( $f_p$ ) produziria o melhor resultado, isto é, menor distorção do sinal e maior atenuação do ruído. Após algumas tentativas empíricas (usando o filtro com diferentes ordens), o melhor valor de frequência de corte encontrado foi em aproximadamente 1100Hz, que resulta em uma melhor filtragem para o Butterworth de 8ª ordem. O gráfico da Figura 5 apresenta a resposta do filtro com  $f_p=1100\text{Hz}$  para as diferentes ordens ( $n=2$  a  $n=8$ ). Como esperado, a resposta do Butterworth é mais seletiva quanto maior for a ordem do filtro. Vale ressaltar que o filtro Butterworth tem uma ótima resposta em frequência comparado a outros mais simples, uma vez que apresenta um ganho uniforme para a banda passante e uma atenuação acentuada na banda rejeitada.





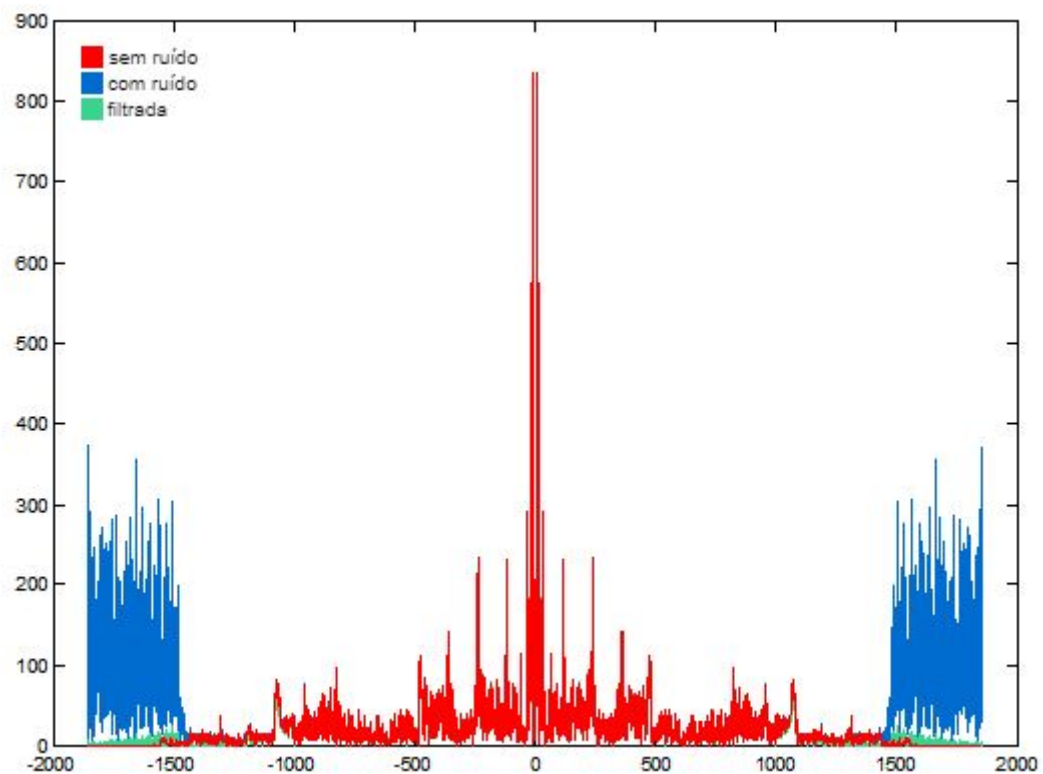
**Figura 5 - Resposta em frequência do filtro Butterworth de diferentes ordens**

Com intuito de prover uma visualização mais completa da filtragem e justificar algumas escolhas, o código no ANEXO II foi alterado para compor alguns gráficos. Por exemplo, a Figura 6 mostra um gráfico completo representando em diferentes cores os resultados das filtragens para o filtro na frequência de corte escolhida com diferentes ordens e a resposta do respectivos filtros (como mostrado na Figura 5). Com esse gráfico é possível visivelmente analisar os resultados da filtragem e verificar comparativamente a eficiência de cada ordem do filtro, além de possibilitar a visualização completa do espectro para a melhor filtragem (em verde,  $f_p=1100\text{Hz}$  e  $n=8$ ). Vale ressaltar que, para ser possível visualizar a resposta do filtro que possui amplitude entre 0 e 1 (um fator multiplicativo de atenuação) foi necessária a multiplicação por um valor arbitrário para representação na mesma escala, que neste caso foi escolhido a máxima amplitude do espectro do sinal original ruidoso. Dessa forma, apesar de quantitativamente não ser efetivo, essa visualização permite uma análise qualitativa interessante, pois é possível correlacionar para cada frequência a intensidade de atenuação promovida pelo filtro.

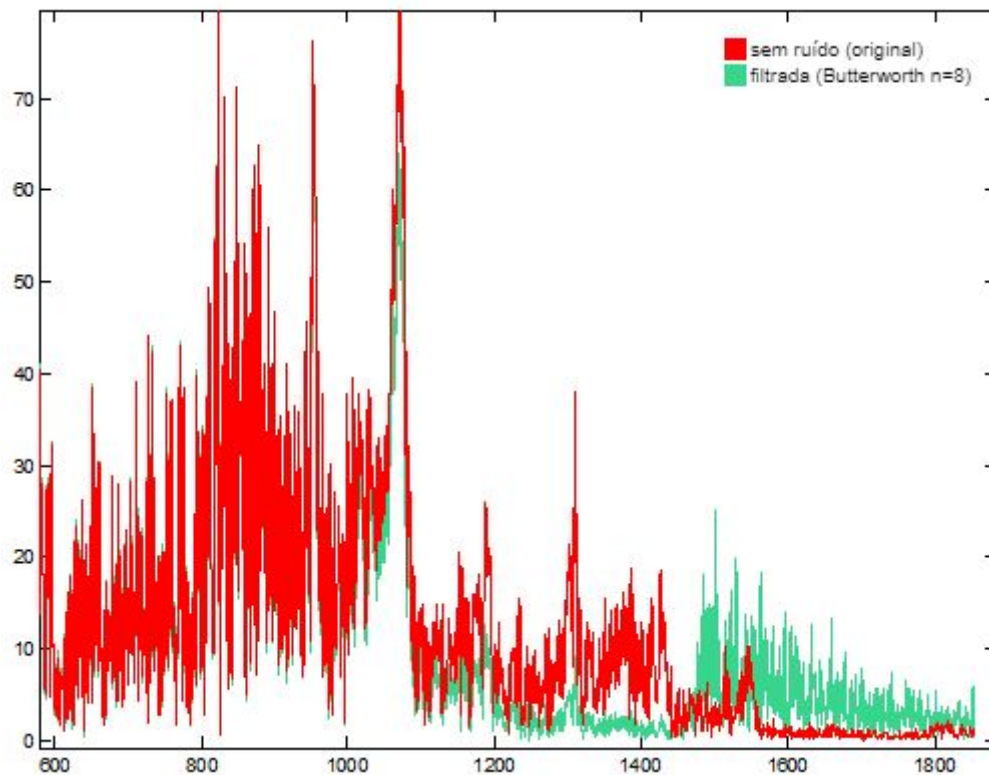


**Figura 6 - Comparação entre as diferentes filtragens com  $f_p=1100\text{Hz}$**

A Figura 7 apresenta uma comparação entre o espectro original (em vermelho), o espectro do sinal ruidoso (em azul) e o espectro resultante da filtragem com o Butterworth de 8ª ordem com  $f_p=1100\text{Hz}$  (em verde). Observa-se que a filtragem foi efetiva, pois as frequência do ruído presentes a partir de  $1500\text{Hz}$  foram fortemente atenuadas ao mesmo tempo que não se obteve uma distorção importante no espectro original. Além disso, usando o comando “soundsc()” do sinal filtrado, foi possível ouvir a palavra “matlab” sem grandes alterações se comparado ao áudio original. Apesar disso, é evidente que os sinais não serão idênticos devido a utilização de uma filtragem não ideal que apesar de escolhida para o melhor resultado mostra leves alterações. A Figura 8 exibe em maiores detalhes a diferença entre os dois sinais, que se resume a uma leve atenuação precoce do espectro a partir de  $1000\text{Hz}$  e uma atenuação não completa do ruído a partir de  $1500\text{Hz}$ . Isso fica evidente ao se observar que de  $1000\text{Hz}$  a  $1500\text{Hz}$  o espectro filtrado tem menor amplitude que o original e a partir de  $1500\text{Hz}$  o espectro filtrado ainda apresenta alguma contribuição do ruído, o que o posiciona acima da amplitude original.

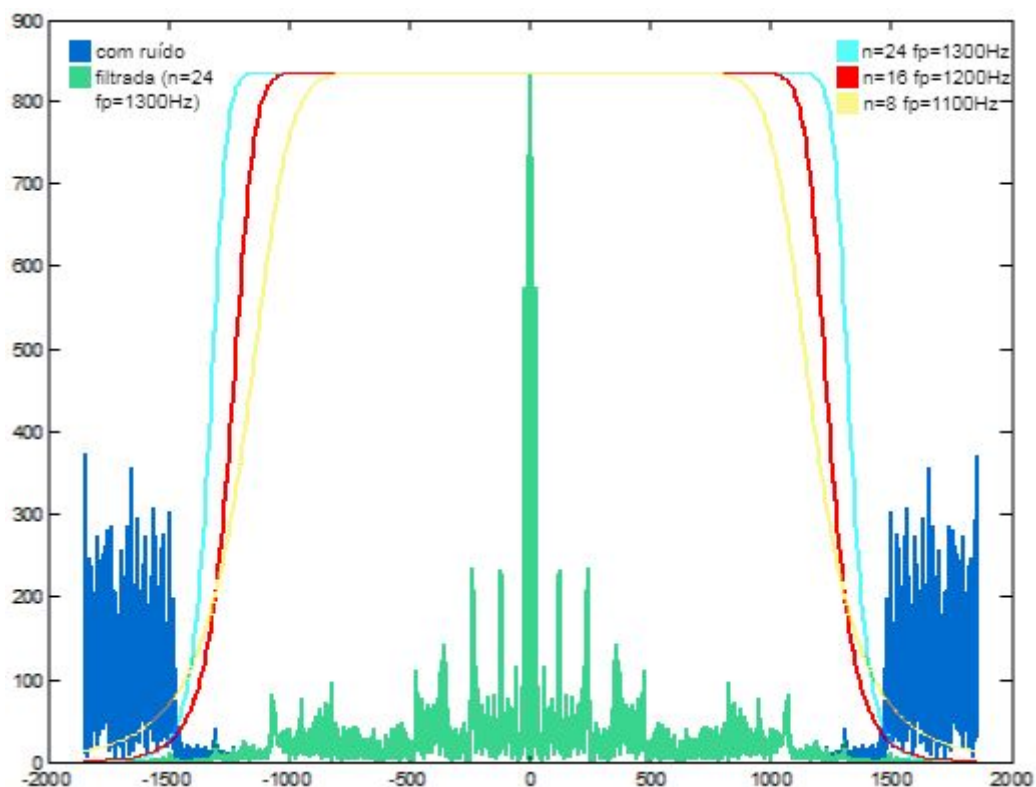


**Figura 7 - Comparação entre os espectros original, ruidoso e filtrado ( $n=8$ ,  $f_p=1100\text{Hz}$ )**



**Figura 8 - Comparação entre as diferentes filtrações com  $f_p=1100\text{Hz}$**

Apenas para se obter uma análise complementar, a Figura 9 tem como propósito exibir a filtragem por um Butterworth de maior ordem (mais seletivo) e como a frequência de corte deve ser ajustada para comportar essa nova resposta em frequência. Para isso, nesse gráfico é possível identificar três variações de filtro: em azul claro,  $n=24$ ,  $f_p=1300\text{Hz}$ ; em vermelho,  $n=16$ ,  $f_p=1200\text{Hz}$ ; e em amarelo,  $n=8$ ,  $f_p=1100\text{Hz}$  (para comparação). Além disso, mostra-se o espectro ruidoso e o filtrado com a melhor configuração. Apesar de uma filtragem muito melhor, à medida que se aumenta a ordem, a complexidade do filtro também aumenta e o benefício em clareza do áudio é cada vez menor, uma vez que a diferença de amplitude das componentes em frequência mais importantes com as restantes do ruído é muito alta (elevada relação sinal-ruído).



**Figura 9 - Comparação entre as diferentes filtrações em termos de  $n$  e  $f_p$**

## **CONSIDERAÇÕES FINAIS**

Com as atividades deste experimento, foi possível analisar o espectro de frequência de sinais em sistemas lineares com representação por série de Fourier. Foi realizada a análise teórica e simulação com software especializado Matlab. Identificou-se experimentalmente que é possível propor hipóteses de estimativa de espectro de frequência de sinais tendo como base a audição dos sinais sonoros reproduzidos. Salienta-se, obviamente, que tal método é impreciso e inexato, servindo apenas como proposição empírica.

Para que o sinal seja detectável, é desejável que sua amplitude esteja acima da do (modelado estatisticamente) ruído gaussiano branco aditivo (Additive White Gaussian Noise - AWGN), além de que o espectro de frequência não estejam sobrepostos (sem mascaramento/aliasing). Observa-se que estas flutuações aleatórias podem ser classificadas como ruído térmico (ruído de Johnson-Nyquist), ruído de tiro (ou ruído de Schottky, ruído de Poisson/shot noise), ruído de cintilação (flicker noise), entre outros [3].

Em relação aos códigos implementados, observa-se que a função `fftshift` realiza a centralização do espectro de frequências para a origem, sendo útil para facilitar a visualização do sinal analisado. Assim como a função `soundsc` possibilita uma avaliação qualitativa importante para construção do filtro.

O filtro Butterworth em ordens mais elevadas apresentou uma filtragem satisfatória do sinal, pois permitiu a eliminação do ruído sem distorções importantes para o sinal original. Além disso, foi observado que apesar da filtragem não ser ideal e nem capaz de obter o sinal original perfeitamente, ela é suficientemente eficaz para em termos do sinal de áudio a resultante seja identificada como a original.

Assim, o objetivo de estudar e realizar análise espectral de sinais, funções representativas de sinais de áudio com ruído adicionado, tão como explorar modificação de parâmetros em sistemas lineares, foi alcançado.

## **REFERÊNCIAS**

- [1] LATHI, B.P. Sinais e Sistemas Lineares. Bookman, 2ª Edição (17 agosto 2006).
- [2] Notas de Aula e materiais disponibilizados no Moodle da disciplina
- [3] STEER, Michael. Microwave and RF Design: Modules. Volume 4. (Third Edition), NC State University, 2019. doi: <https://doi.org/10.5149/9781469656977> Steer.

## ANEXO I

### **analise\_espectral.m**

```
%UNIVERSIDADE FEDERAL DE SANTA CATARINA - UFSC
%CENTRO TECNOLÓGICO - CTC
%DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA
%GUSTAVO SIMAS & ANDRÉ MATTOS

%ANÁLISE ESPECTRAL

clear all, close all
load mtlb
who
L=length(mtlb);
plot([1:L]/Fs,mtlb)
axis tight
xlabel('Tempo (segundos)')
load('NoisySpeech.txt');
x= NoisySpeech;
L=length(x);
plot([1:L]/Fs,x)
axis tight
title('Sinal de voz ruidoso')
xlabel('Tempo (segundos)')
soundsc(x)
M=fft(mtlb,8192); % espectro entre 0 e Fs Hz
M=fftshift (M); % espectro entre -Fs/2 e Fs/2
R=fft(x,8192);
R=fftshift (R);
f=(-4096:4095)/8192*Fs/2;
plot(f,abs(R))
```

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



## **ANEXO II**

```
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)
```