



UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO TECNOLÓGICO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA  
CURSO DE ENGENHARIA ELÉTRICA  
EEL7522 - PROCESSAMENTO DIGITAL DE SINAIS

Projeto de Processamento de Voz

MATHEUS LENZI DOS SANTOS (19100420)

FLORIANÓPOLIS - SC, 2021

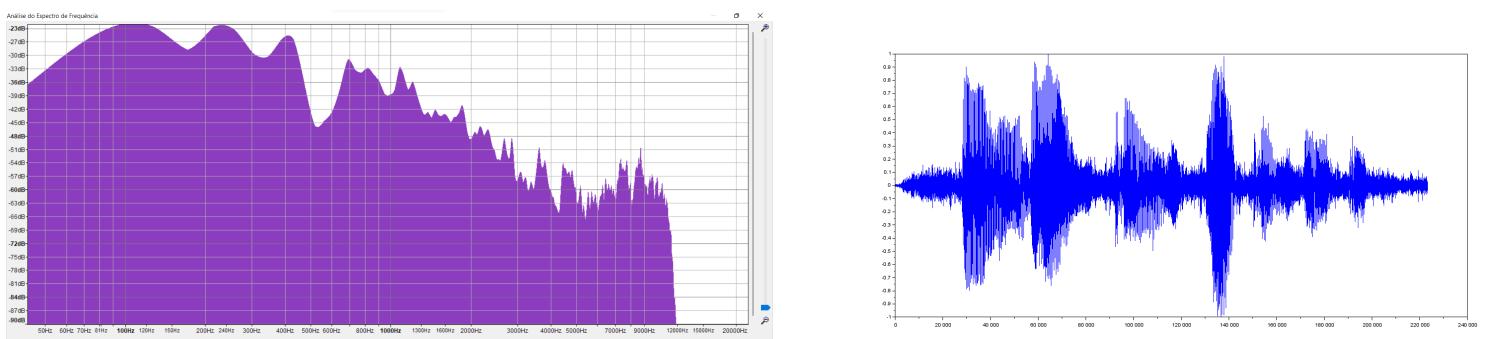


Figura 01: Análise do Espectro da Frequência e Amplitude do Áudio Original

```

1 //·Matheus·Lenzi·dos·Santos -- 19100420
2
3 //·Indicando·o·caminho·onde·está·salvo·o·áudio·original·no·formato·.wav
4 wave = "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio.wav";
5 wavread(wave,"Size")
6 wavread(wave,"Info")
7 //·X·->·Amplitude·do·som·entre·[-1,·1],·com·uma·linha·por·canal·de·gravação
8 //·Fs·->·Frequência·de·amostragem·em·Hz
9 //·bits·->·Número·de·bits·em·X
10 [X,Fs,bits] = wavread(wave)
11
12 //·Recuperando·o·tamanho·do·áudio·original·no·formato·.wav
13 wave_size = wavread(wave,"Size") (2)
14
15 //·Recuperando·apenas·um·canal·de·gravação·(Mono)·áudio·original
16 x = X(1,:);
17
18 x_norm = x/ (max(abs(x)))
19
20 //·Plotando·o·áudio·original
21 plot(x_norm)
22

```

Figura 02: Código para Análise da Amplitude do Áudio Original

Comentário: Áudio com duração de 5 segundos, no qual é feito a contagem de 1 até 5. É possível perceber pelo espectro de amplitude a proeminência de valores em 7 pontos do áudio, mas apenas em 5 deles isso se deve a fala de fato, no restante deles se deve ao ruído captado pelo microfone.

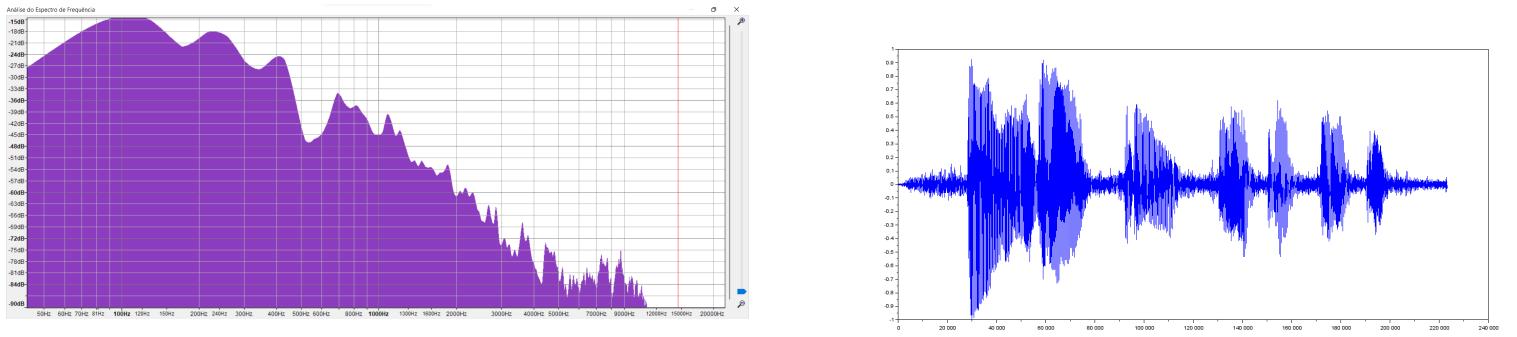


Figura 03: Análise do Espectro da Frequência e Amplitude do Filtro de Esquecimento com *Alpha* igual a 0.98

```

1 //-.Matheus-Lenzi-dos-Santos -.-19100420
2
3 //-.Indicando-o-caminho-onde-está-salvo-o áudio-original-no-formato-.wav
4 wave = "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio.wav";
5 wavread(wave,"Size")
6 wavread(wave,"Info")
7 //-.X--> Amplitude-do-som-entre-[-1,.1],-com-uma-linha-por-canal-de-gravação
8 //-.Fs-->-Frequência-de-amostragem-em-Hz
9 //-.bits-->-Número-de-bits-em-X
10 [X,Fs,bits] = wavread(wave)
11
12 //-.Recuperando-o-tamanho-do-áudio-original-no-formato-.wav
13 wave_size = size(X) (2)
14
15 //-.Recuperando-apenas-um-canal-de-gravação-(Mono)-áudio-original
16 x = X(1,:);
17
18 //-.Definindo-os-valores-iniciais-dos-vetores-indexados
19 y = zeros(wave_size,1);
20 y(1) = .0;
21
22 //-.Definindo-o-valor-de-alpha-do-filtro-de-esquecimento
23 alpha = 0.98
24
25 //-.Implementando-o-filtro-de-esquecimento-(y[n]:=alpha.^x[n-1].*x[n])
26 for n = 2:wave_size
27 ... y(n) = alpha.* y(n-1) + x(n)
28 end
29
30 //-.Normalizando-os-valores-de-y
31 y_norm = y/(max(abs(y)))
32
33 //-.Plotando-os-valores-de-y-normalizados
34 plot(y_norm)
35
36 //-.Escrevendo-os-valores-de-y-no-formato-.wav
37 wavwrite(y_norm, Fs, bits, "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio_filtro_esquecimento_098.wav")
38

```

Figura 04: Código para Filtro de Esquecimento com *Alpha* igual a 0.98

Comentário: Com a aplicação do filtro de esquecimento utilizando um *alpha* igual a 0.98, a principal diferença que se tem para o áudio original é que o tom de voz se torna um pouco mais grave. Pelo espectro de amplitude é possível perceber uma ampliação do sinal nas primeiras proeminências, que referem se a fala dos números 1 e 2, já para os números 3, 4 e 5 é possível ver no gráfico do espectro a atenuação do valor do sinal, mas na percepção o áudio se manteve, ou seja, com um tom de voz mais grave. No gráfico do espectro da frequência é possível perceber uma atenuação para o original dos valores mais altos de frequência.

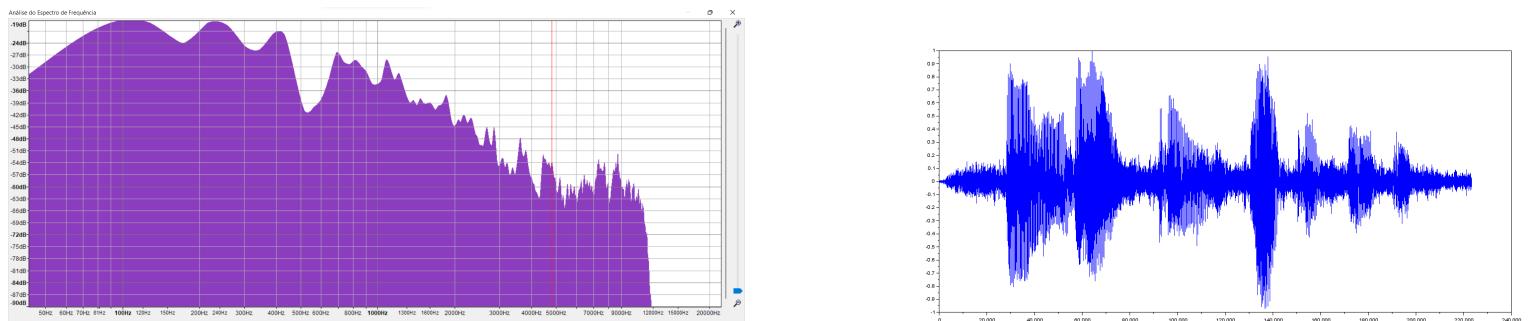


Figura 05: Análise do Espectro da Frequência e Amplitude do Filtro de Esquecimento com *Alpha* igual a 0.50

```

filtrado_de_esquecimento.sce (C:\Users\Matheus\Desktop\Projeto Processamento de Voz\filtro_de_esquecimento...)
Arquivo Editar Formatar Opções Janela Executar ?
filtrado_de_esquecimento.sce (C:\Users\Matheus\Desktop\Projeto Processamento de Voz\filtro_de_esquecimento.sce) - SciNotes
filtrado_de_esquecimento.sce
1 //-.Matheus.Lenzi.dos.Santos.-.19100420
2
3 //-.Indicando.o.caminho.onde.está.salvo.o áudio.original.no.formato..wav
4 wave = "C:\Users\Matheus\Desktop\Projeto.Processamento.de.Voz\audio.wav";
5 wavread(wave,"Size")
6 wavread(wave,"Info")
7 //-.X.->.Amplitude do som entre. [-1,.1], com uma linha por canal de gravação
8 //-.Fs .-> Frequência de amostragem em Hz
9 //-.bits .-> Número de bits em X
10 [X,Fs,bits] = wavread(wave)
11
12 //-.Recuperando.o.tamanho do áudio.original.no.formato..wav
13 wave_size = size(X) (2)
14
15 //-.Recuperando.apenas.um.canal.de.gravação.(Mono). áudio.original
16 x = X(1,:);
17
18 //-.Definindo.os.valores.iniciais.dos.vetores.indexados
19 y = zeros(wave_size,1);
20 y(1) = 0;
21
22 //-.Definindo.o.valor.de.alpha do filtro de esquecimento
23 alpha = 0.5
24
25 //-.Implementando.o.filtro.de.esquecimento -(y[n] =.alpha.^k.y[n-1].^k.x[n])
26 for n = 2:wave_size
27 ... y(n) = alpha ^ k.y(n-1) + x(n)
28 end
29
30 //-.Normalizando.os.valores.de.y
31 y_norm = y/(max(abs(y)))
32
33 //-.Plotando.os.valores.de.y.normalizados
34 plot(y_norm)
35
36 //-.Escrevendo.os.valores.de.y.no.formato..wav
37 wavwrite(y_norm, Fs, bits, "C:\Users\Matheus\Desktop\Projeto.Processamento.de.Voz\audio_filtro_esquecimento_050.wav")
38

```

Figura 06: Código para Filtro de Esquecimento com *Alpha* igual a 0.50

Comentário: Com a aplicação do filtro de esquecimento utilizando um *alpha* igual a 0.50, já não é possível perceber diferença alguma para o áudio original. O mesmo é perceptível na visualização dos gráficos de espectro de frequência e amplitude que não possui praticamente nenhuma alteração para o original. Esse fato pode ser explicado por se tratar de um pequeno fator aplicado no esquecimento, que não produz mudanças significativas no áudio, ou pelo menos nesse sinal de áudio.

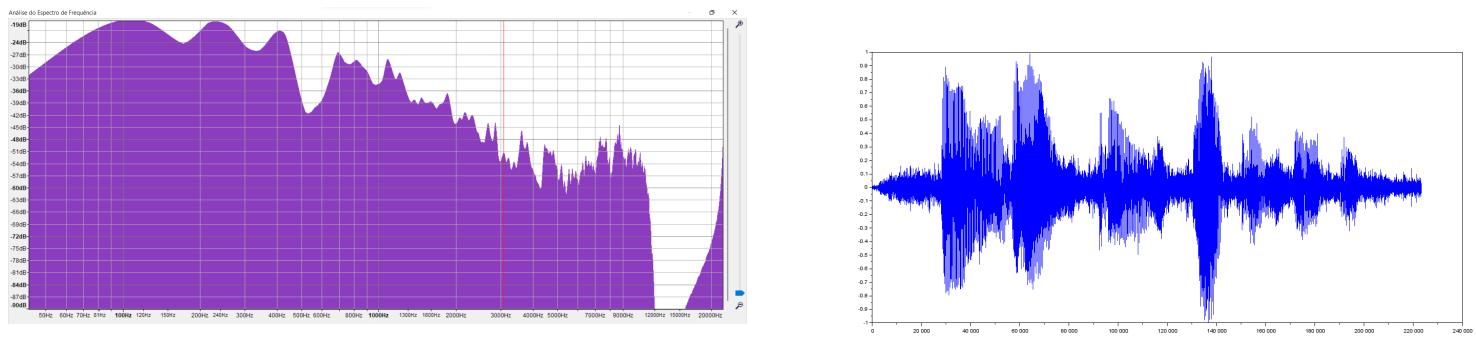


Figura 07: Análise do Espectro da Frequência e Amplitude do Filtro de Esquecimento com  $\text{Alpha}$  igual a -0.98

```

1 //-.Matheus.Lenzi.dos.Santos.-.19100420
2
3 //-.Indicando-o-caminho-onde-está-salvo-o áudio-original-no-formato-.wav
4 wave = "C:\Users\Matheus\Desktop\Projeto.Processamento.de.Voz\audio.wav";
5 wavread(wave,"Size")
6 wavread(wave,"Info")
7 //-.X -> Amplitude-do-som-entre-[-1,1],-com-uma-linha-por-canal-de-gravação
8 //-.Fs -> Frequência-de-amostragem-em-Hz
9 //-.bits -> Número-de-bits-em-X
10 [X,Fs,bits] = wavread(wave)
11
12 //-.Recuperando-o-tamanho-do-áudio-original-no-formato-.wav
13 wave_size = size(X) (2)
14
15 //-.Recuperando-apenas-um-canal-de-gravação-(Mono)-áudio-original
16 x = X(1,:);
17
18 //-.Definindo-os-valores-iniciais-dos-vetores-indexados
19 y = zeros(wave_size,1);
20 y(1) = 0;
21
22 //-.Definindo-o-valor-de-alpha-do-filtro-de-esquecimento
23 alpha = -0.98
24
25 //-.Implementando-o-filtro-de-esquecimento-(y[n] = alpha * y[n-1] + x[n])
26 for n = 2:wave_size
27 ... y(n) = alpha * y(n-1) + x(n)
28 end
29
30 //-.Normalizando-os-valores-de-y
31 y_norm = y/(max(abs(y)))
32
33 //-.Plotando-os-valores-de-y-normalizados
34 plot(y_norm)
35
36 //-.Escrevendo-os-valores-de-y-no-formato-.wav
37 wavwrite(y_norm, Fs, bits, "C:\Users\Matheus\Desktop\Projeto.Processamento.de.Voz\audio_filtro_esquecimento_-098.wav")
38

```

Figura 08: Código para Filtro de Esquecimento com  $\text{Alpha}$  igual a -0.98

Comentário: Com a aplicação do filtro de esquecimento utilizando um  $\text{alpha}$  igual a -0.98, a principal diferença que foi possível perceber em relação ao original foi uma amplificação do ruído do microfone, ou seja, no intervalo entre as proeminências das falas dos números de 1 a 5 o valor do sinal é amplificado. É possível perceber também pelo gráfico de amplitude tal comportamento comentado, e para o gráfico do espectro da frequência ocorre o surgimento em relação ao original de um pico nos valores altos de frequência.

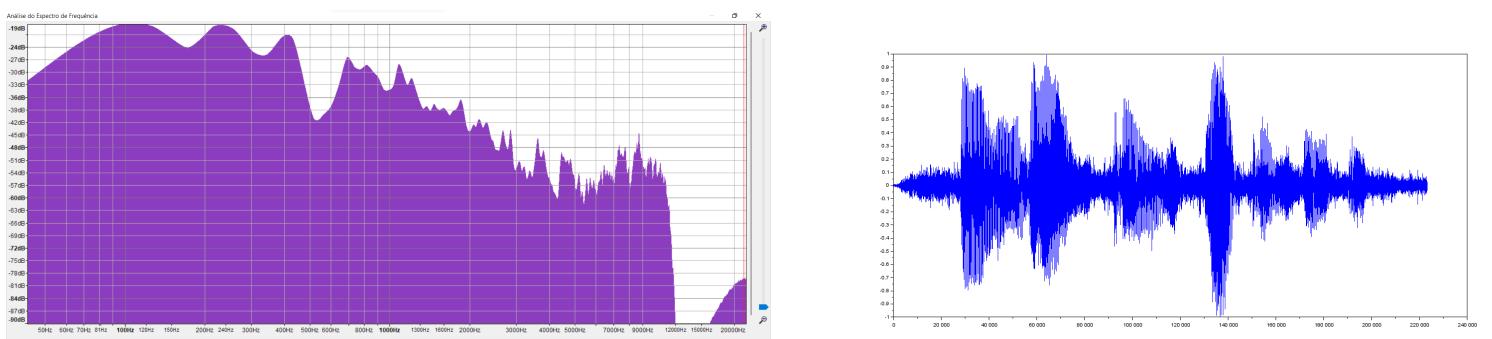


Figura 09: Análise do Espectro da Frequência e Amplitude do Filtro de Esquecimento com  $\text{Alpha}$  igual a -0.50

```

1 //-.Matheus-Lenzi-dos-Santos--19100420
2
3 //-.Indicando-o-caminho-onde-está-salvo-o-áudio-original-no-formato-.wav
4 wave = "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio.wav";
5 wavread(wave,"Size")
6 wavread(wave,"Info")
7 //-.X-->-Amplitude-do-som-entre-[-1,.1]-,com-uma-linha-por-canal-de-gravação
8 //-.Fs-->-Frequência-de-amostragem-em-Hz
9 //-.bits-->-Número-de-bits-em-X
10 [X,Fs,bits] = wavread(wave)
11
12 //-.Recuperando-o-tamanho-do-áudio-original-no-formato-.wav
13 wave_size = size(X) (2)
14
15 //-.Recuperando-apenas-um-canal-de-gravação-(Mono)-áudio-original
16 x = X(1,:);
17
18 //-.Definindo-os-valores-iniciais-dos-vetores-indexados
19 y = zeros(wave_size,1);
20 y(1) = 0;
21
22 //-.Definindo-o-valor-de-alpha-do-filtro-de-esquecimento
23 alpha = -0.50
24
25 //-.Implementando-o-filtro-de-esquecimento-(y[n]=-alpha.^k.y[n-1].^k.x[n])
26 for n = 2:wave_size
27 ... y(n) = alpha ^ k * y(n-1) + x(n)
28 end
29
30 //-.Normalizando-os-valores-de-y
31 y_norm = y/(max(abs(y)))
32
33 //-.Plotando-os-valores-de-y-normalizados
34 plot(y_norm)
35
36 //-.Escrevendo-os-valores-de-y-no-formato-.wav
37 wavwrite(y_norm, Fs, bits, "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio_filtro_esquecimento_-050.wav")
38

```

Figura 10: Código para Filtro de Esquecimento com  $\text{Alpha}$  igual a -0.50

Comentário: Com a aplicação do filtro de esquecimento utilizando um  $\text{alpha}$  igual a -0.50, como no filtro de esquecimento anterior de -0.98, foi possível perceber uma amplificação no ruído presente no sinal. Ainda, uma percepção um pouco subjetiva, foi uma estridência na pronúncia do número 3, que pela análise do gráfico da amplitude do sinal também parece estar presente. Na análise do gráfico do espectro de frequências é possível observar também um surgimento de pico nas altas frequências, menor que para o  $\text{alpha}$  de -0.98, mas presente.

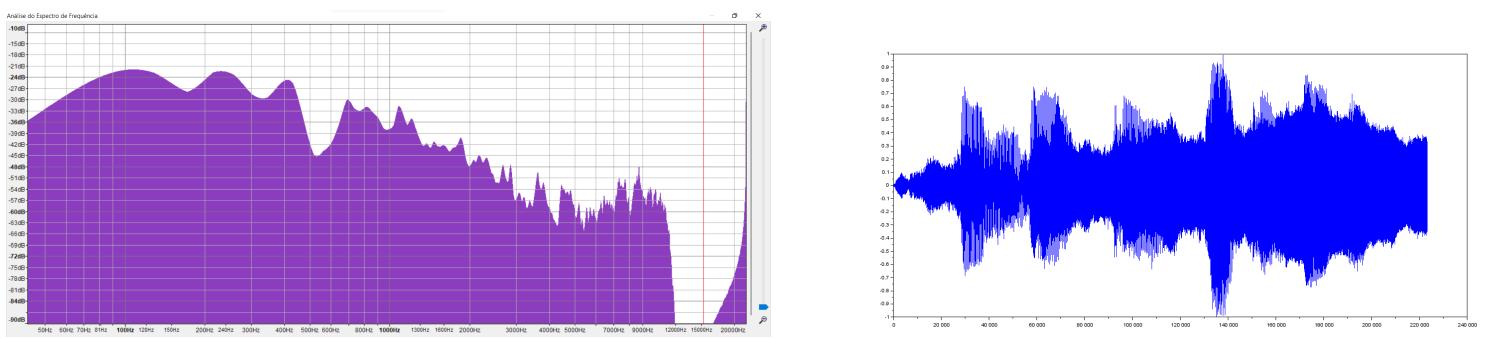


Figura 11: Análise do Espectro da Frequência e Amplitude do Filtro de Esquecimento com  $\alpha$  igual a -1.00

```

1 // .Matheus-Lenzi-dos-Santos.-.19100420
2
3 // .-Indicando-o-caminho-onde-está-salvo-o-áudio-original-no-formato-.wav
4 wave = "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio.wav";
5 wavread(wave,"Size")
6 wavread(wave,"Info")
7 // .X -> Amplitude do som entre [-1,1], com uma linha por canal de gravação
8 // .Fs -> Frequência de amostragem em Hz
9 // .bits -> Número de bits em X
10 [X,Fs,bits] = wavread(wave)
11
12 // .Recuperando o tamanho do áudio original no formato .wav
13 wave_size = size(X) (2)
14
15 // .Recuperando apenas um canal de gravação (Mono) - áudio original
16 x = X(1,:);
17
18 // .Definindo os valores iniciais dos vetores indexados
19 y = zeros(wave_size,1);
20 y(1) = 0;
21
22 // .Definindo o valor de alpha do filtro de esquecimento
23 alpha = -1
24
25 // .Implementando o filtro de esquecimento - y[n] = alpha * y[n-1] + x[n]
26 for n = 2:wave_size
27 ... y(n) = alpha * y(n-1) + x(n)
28 end
29
30 // .Normalizando os valores de y
31 y_norm = y/(max(abs(y)))
32
33 // .Plotando os valores de y normalizados
34 plot(y_norm)
35
36 // .Escrevendo os valores de y no formato .wav
37 wavwrite(y_norm, Fs, bits, "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio_filtro_esquecimento_1.wav")
38

```

Figura 12: Código para Filtro de Esquecimento com  $\alpha$  igual a -1.00

Comentário: Com a aplicação do filtro de esquecimento utilizando um  $\alpha$  igual a -1.00, a mudança perceptível para o áudio original foi uma constância no sinal do ruído, que apesar de parecer mais alto, era menos prejudicial que nos outros casos. Apesar de a deformação no gráfico da amplitude do sinal ser grande em relação ao original, o que vai ao encontro da percepção auditiva, o resultado do processamento do áudio não é ruim. O gráfico do espectro da frequência se assemelha muito ao do filtro com  $\alpha$  igual a -0.98, o que faz sentido.

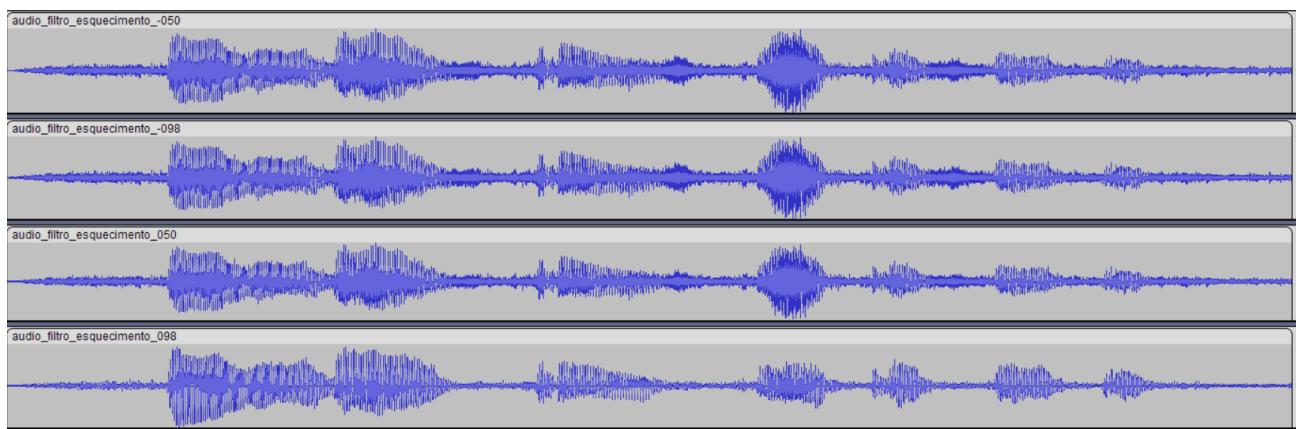
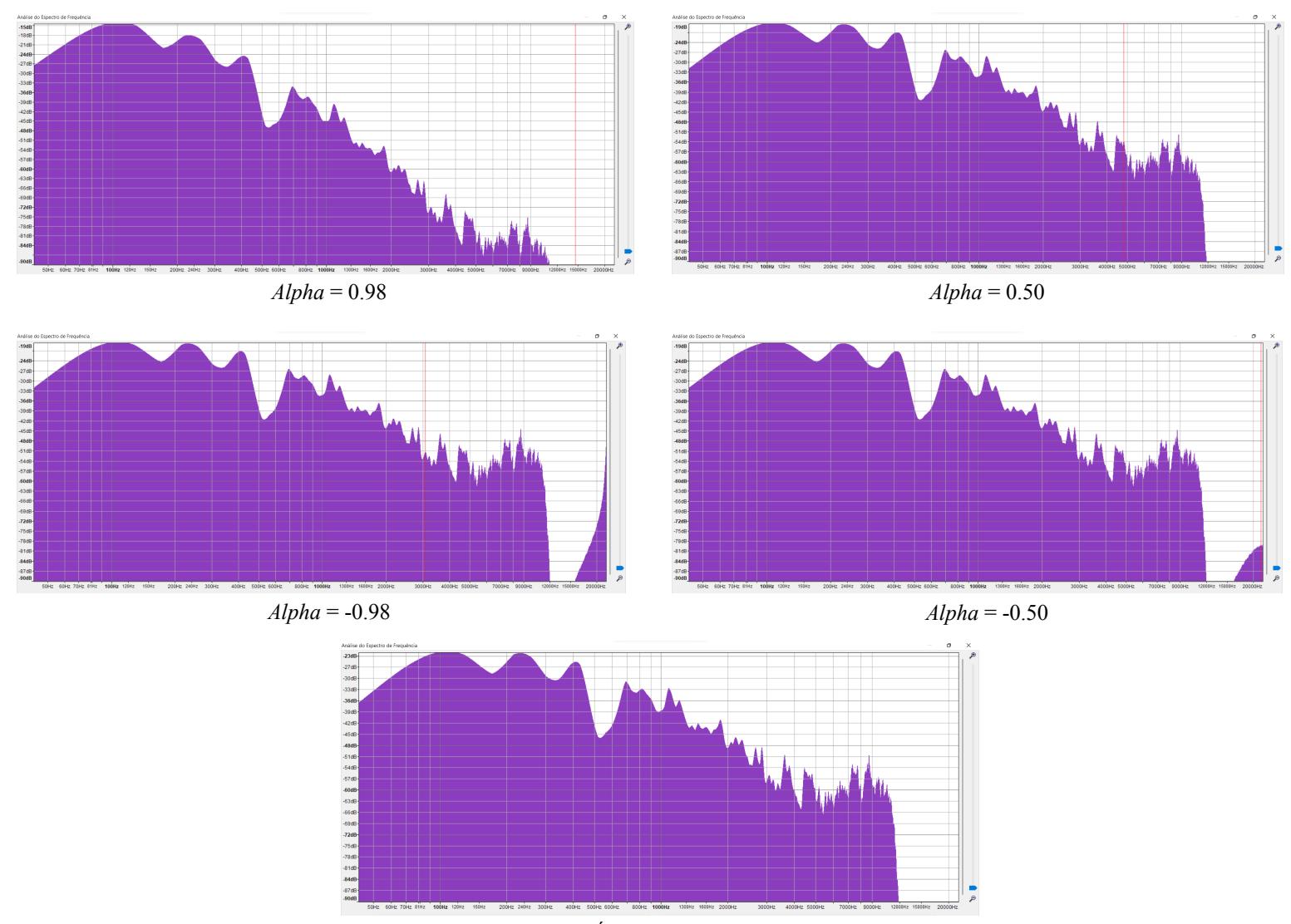


Figura 13: Análise da Amplitude do Filtro de Esquecimento com  $Alphas$  iguais a 0.98, 0.50, -0.98 e -0.50



Áudio Original

Figura 14: Análise do Espectro da Frequência do Filtro de Esquecimento com  $Alphas$  iguais a 0.98, 0.50, -0.98 e -0.50

Comentário: A sobreposição dos gráficos da amplitude do sinal e a comparação direta dos gráficos do espectro da frequência corroboram com as análises anteriormente descritas.

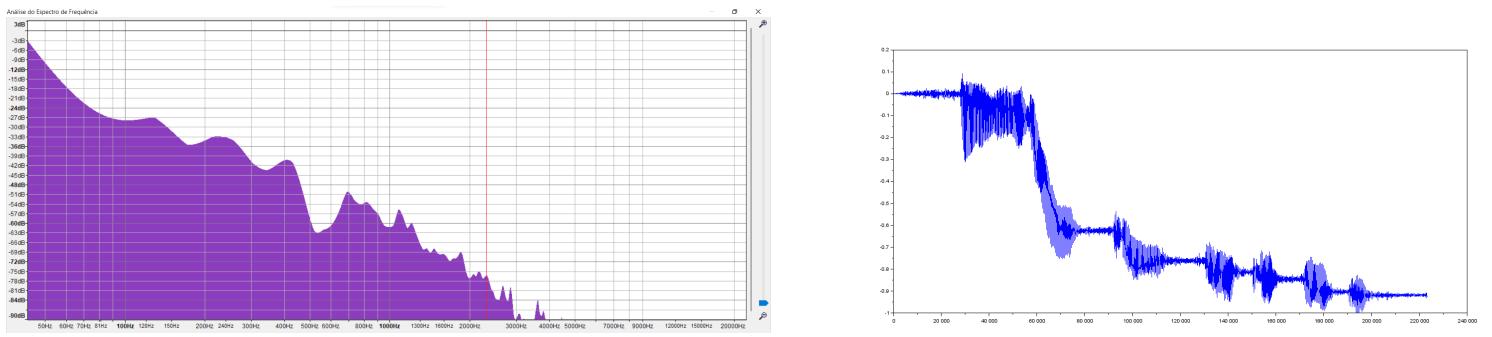


Figura 15: Análise do Espectro da Frequência e Amplitude do Filtro de Média Móvel com  $M$  igual a 50

```

1 // - Matheus-Lenzi-dos-Santos--19100420
2
3 // - Indicando-o-caminho-onde-está-salvo-o áudio-original-no-formato-.wav
4 wave = "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio.wav";
5 wavread(wave,"Size")
6 wavread(wave,"Info")
7 // - X--> Amplitude-do-som-entre-[-1,1],-com-uma-linha-por-canal-de-gravação
8 // - Fs--> Frequência-de-amostragem-em-Hz
9 // - bits--> Número-de-bits-em-X
10 [X,Fs,bits] = wavread(wave)
11
12 // - Recuperando-o-tamanho-do áudio-original-no-formato-.wav
13 wave_size = size(X)(2)
14
15 // - Recuperando-apenas-um-canal-de-gravação-(Mono)-áudio-original
16 x = X(:, :);
17
18 // - Definindo-os-valores-iniciais-dos-vetores-indexados
19 y = zeros(wave_size,1);
20 y(1) = 0;
21
22 // - Definindo-o-valor-de-alpha-do-filtro-de-esquecimento
23 m = 50
24
25 // - Implementando-o-filtro-de-média-móvel-(y(n)=y(n-1)+(x(n)-x(n-m))/m)
26 for n = 2:wave_size
27 if n > wave_size
28 y(n) = y(n-1) + ((x(n) - x(n-m))/m)
29 else
30 y(n) = y(n-1) + (x(n)/m)
31 end
32 end
33
34 // - Normalizando-os-valores-de-y
35 y_norm = y/(max(abs(y)))
36
37 // - Plotando-os-valores-de-y-normalizados
38 plot(y_norm)
39
40 // - Escrevendo-os-valores-de-y-no-formato-.wav
41 wavwrite(y_norm, Fs, bits, "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio_filtro_media_m")

```

Figura 16: Código para Filtro de Média Móvel com  $M$  igual a 50

Comentário: Com a aplicação do filtro de média móvel utilizando um  $M$  igual a 50, o áudio processado parece bem mais abafado que o áudio original, o que implica em uma diminuição drástica do ruído também. No gráfico do espectro da frequência é possível perceber uma atenuação acentuada das altas freqüências, como um filtro passa-baixas, o que de fato faz sentido na percepção da redução dos ruídos. Apesar de a janela  $M$  ser pequena no filtro de média móvel, o áudio resultante é bem diferente do original. Em um caso em que seria preciso garantir uma comunicação sem ruído parece ser uma opção viável.

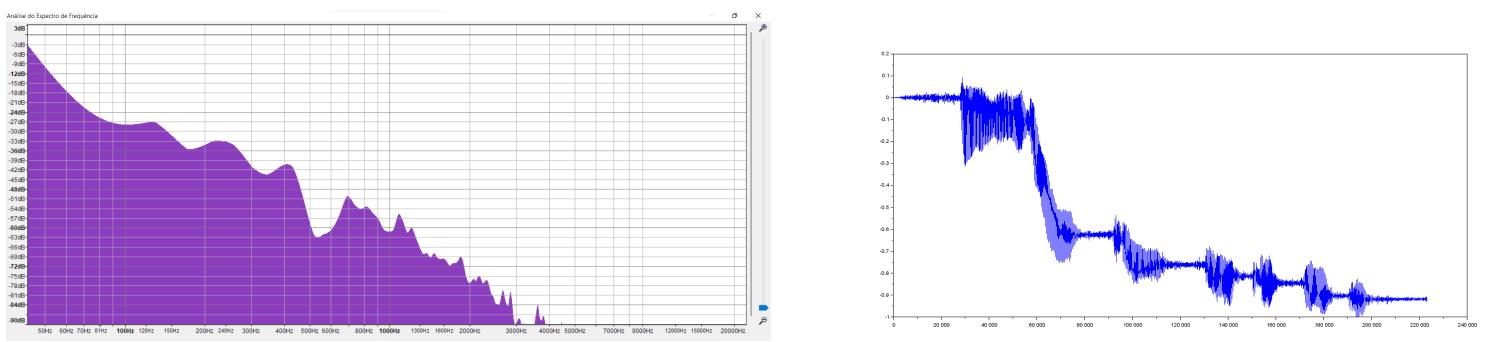


Figura 17: Análise do Espectro da Frequência e Amplitude do Filtro de Média Móvel com  $M$  igual a 100

```

3 // Indicando o caminho onde está salvo o áudio original no formato .wav
4 wave = "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio.wav";
5 wavread(wave,"Size")
6 wavread(wave,"Info")
7 // X -> Amplitude do som entre [-1,1], com uma linha por canal de gravação
8 // Fs -> Frequência de amostragem em Hz
9 // bits -> Número de bits em X
10 [X,Fs,bits] = wavread(wave)
11
12 // Recuperando o tamanho do áudio original no formato .wav
13 wave_size = size(X) (2)
14
15 // Recuperando apenas um canal de gravação (Mono) áudio original
16 x = X(:, :);
17
18 // Definindo os valores iniciais dos vetores indexados
19 y = zeros(wave_size,1);
20 y(1) = 0;
21
22 // Definindo o valor de alpha do filtro de esquecimento
23 m = 100
24
25 // Implementando o filtro de média móvel (y(n) = y(n-1) + ((x(n) - x(n-m))/m))
26 for n = 2:wave_size
27     if n > wave_size
28         y(n) = y(n-1) + ((x(n) - x(n-m))/m)
29     else
30         y(n) = y(n-1) + (x(n)/m)
31     end
32 end
33
34 // Normalizando os valores de y
35 y_norm = y/(max(abs(y)))
36
37 // Plotando os valores de y normalizados
38 plot(y_norm)
39
40 // Escrevendo os valores de y no formato .wav
41 wavwrite(y_norm, Fs, bits, "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio_filtro_media_movel_100.wav")
42

```

Figura 18: Código para Filtro de Média Móvel com  $M$  igual a 100

Comentário: Com a aplicação do filtro de média móvel utilizando um  $M$  igual a 100, o áudio processado ficou tão abafado que após a pronúncia do número 2 já não foi possível escutar mais nada. Apesar disso, a alteração no gráfico do espectro de frequência não foi tão substancial para o filtro com  $M$  igual a 50.

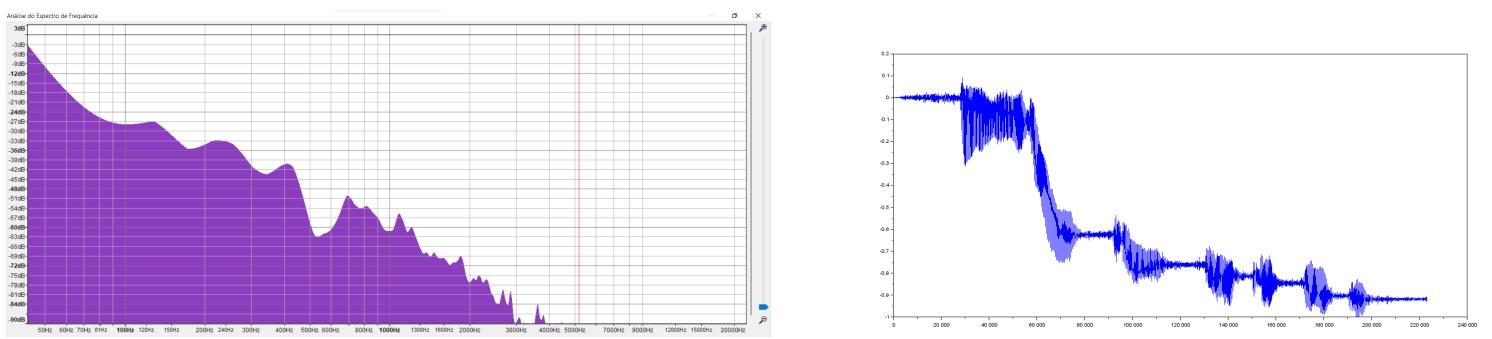


Figura 19: Análise do Espectro da Frequência e Amplitude do Filtro de Média Móvel com  $M$  igual a 1000

```

1 // Indicando o caminho onde está salvo o áudio original no formato .wav
2 wave = "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio.wav";
3 wavread(wave,"Size")
4 wavread(wave,"Info")
5 // X -> Amplitude do som entre [-1,1], com uma linha por canal de gravação
6 // Fs -> Frequência de amostragem em Hz
7 // bits -> Número de bits em X
8 [X,Fs,bits] = wavread(wave)
9
10 // Recuperando o tamanho do áudio original no formato .wav
11 wave_size = size(X) (2)
12
13 // Recuperando apenas um canal de gravação (Mono) áudio original
14 x = X(:,1);
15
16 // Definindo os valores iniciais dos vetores indexados
17 y = zeros(wave_size,1);
18 y(1) = 0;
19
20 // Definindo o valor de alpha do filtro de esquecimento
21 m = 1000;
22
23 // Implementando o filtro de média móvel (y(n) = y(n-1) + ((x(n) - x(n-m))/m))
24 for n = 2:wave_size
25     if n > wave_size
26         y(n) = y(n-1) + ((x(n) - x(n-m))/m);
27     else
28         y(n) = y(n-1) + ((x(n) - x(n-m))/m);
29     end
30 end
31
32 // Normalizando os valores de y
33 y_norm = y/(max(abs(y)));
34
35 // Plotando os valores de y normalizados
36 plot(y_norm)
37
38 // Escrevendo os valores de y no formato .wav
39 wavwrite(y_norm, Fs, bits, "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio_filtro_media_movel_1000.wav");
40
41 Linha 41, coluna 117.

```

Figura 20: Código para Filtro de Média Móvel com  $M$  igual a 1000

Comentário: Com a aplicação do filtro de média móvel utilizando um  $M$  igual a 1000, o áudio processado ficou mais abafado ainda que anteriormente, e após a pronúncia do número 1 já não foi possível escutar mais nada. Apesar disso, a alteração no gráfico do espectro de frequência não foi tão substancial para o filtro com  $M$  igual a 50 e 100, seguindo o comportamento de passa-baixas.

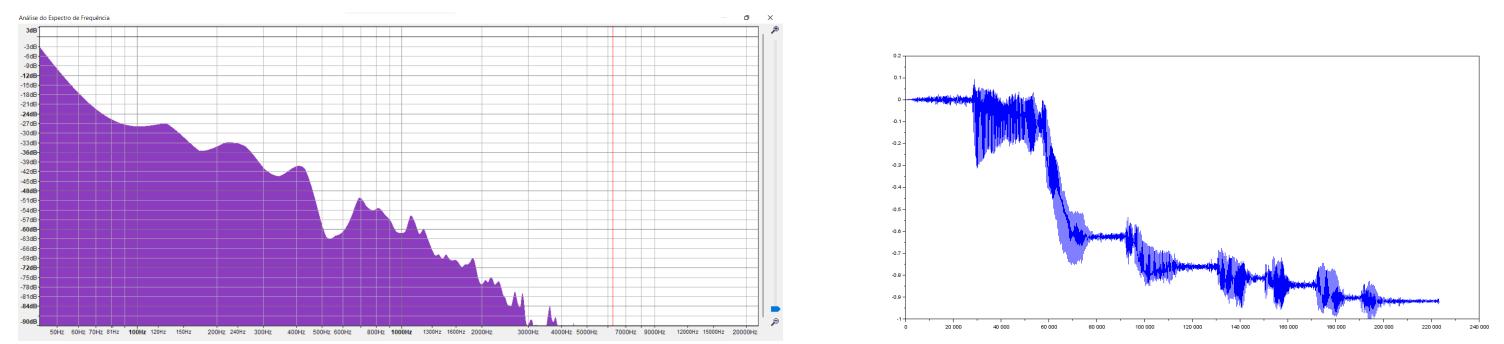


Figura 21: Análise do Espectro da Frequência e Amplitude do Filtro de Média Móvel com  $M$  igual a 10000

```

3 // - Indicando o caminho onde está salvo o áudio original no formato .wav
4 wave = "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio.wav";
5 wavread(wave,"Size")
6 wavread(wave,"Info")
7 // -> Amplitude do som entre [-1, 1], com uma linha por canal de gravação
8 // -> Freqüência de amostragem em Hz
9 // -> Número de bits em X
10 [X,Fs,bits] = wavread(wave)
11
12 // - Recuperando o tamanho do áudio original no formato .wav
13 wave_size = size(X) (2)
14
15 // - Recuperando apenas um canal de gravação (Mono) áudio original
16 x = X(:, :);
17
18 // - Definindo os valores iniciais dos vetores indexados
19 y = zeros(wave_size, 1);
20 y(1) = 0;
21
22 // - Definindo o valor de alpha do filtro de esquecimento
23 m = 10000
24
25 // - Implementando o filtro de média móvel (y(n) = y(n-1) + ((x(n) - x(n-m))/m))
26 for n = 2:wave_size
27     if n > wave_size
28         y(n) = y(n-1) + ((x(n) - x(n-m))/m)
29     else
30         y(n) = y(n-1) + (x(n)/m)
31     end
32 end
33
34 // - Normalizando os valores de y
35 y_norm = y/(max(abs(y)))
36
37 // - Plotando os valores de y normalizados
38 plot(y_norm)
39
40 // - Escrevendo os valores de y no formato .wav
41 wavwrite(y_norm, Fs, bits, "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio_filtro_media_movel_10000.wav")
42
Linha 41, coluna 118.

```

Figura 22: Código para Filtro de Média Móvel com  $M$  igual a 10000

Comentário: Com a aplicação do filtro de média móvel utilizando um  $M$  igual a 10000, não foi possível notar muita diferença para o áudio processado anteriormente com  $M$  igual a 1000, de uma certa maneira parece que satura a capacidade do filtro de média móvel sob o áudio. Para o gráfico do espectro de frequências não foi possível observar novamente nenhuma alteração substancial para os exemplos anteriores.

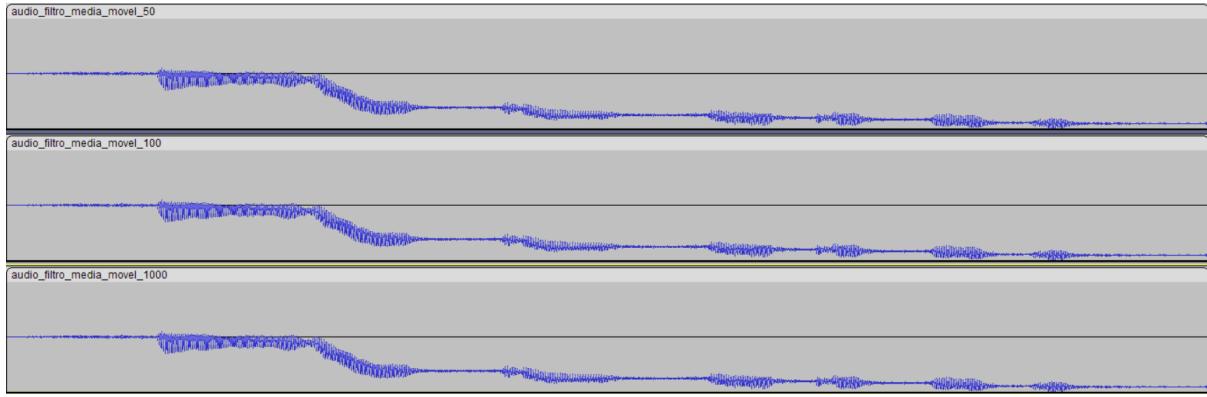


Figura 23: Análise da Amplitude do Filtro de Média Móvel com  $M_s$  igual a 50, 100 e 1000

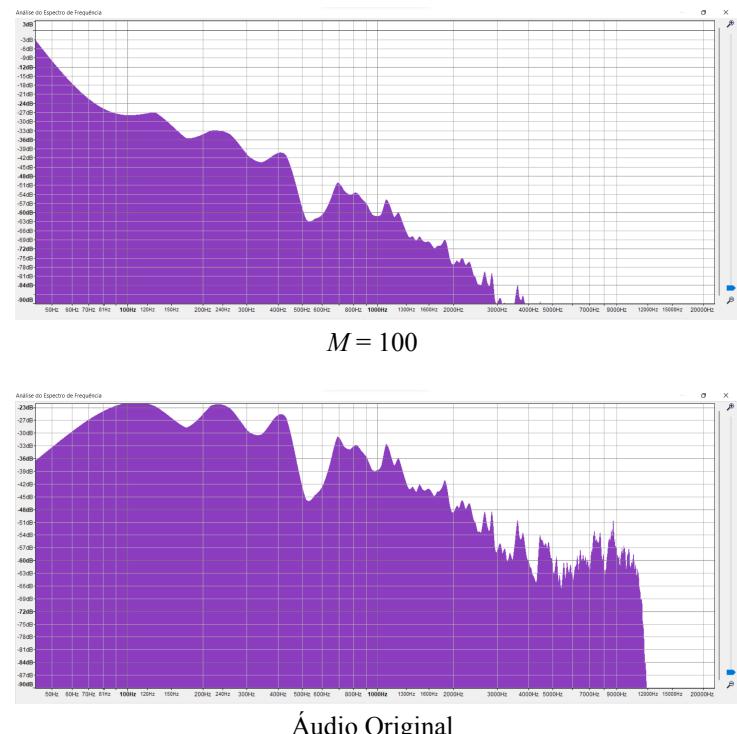
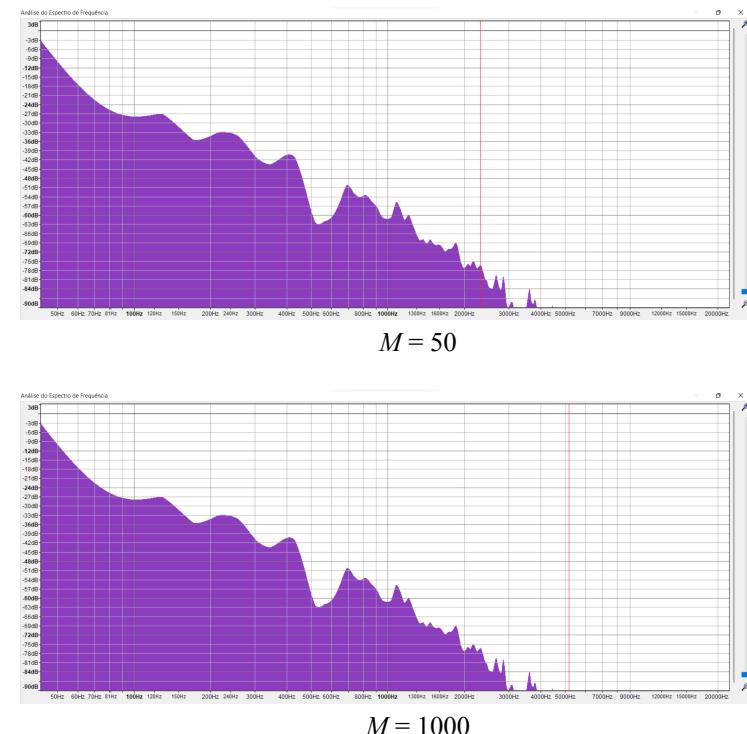


Figura 24: Análise do Espectro da Frequência do Filtro de Esquecimento com  $M_s$  iguais a 50, 100 e 1000

Comentário: Os gráficos do espectro em frequência do sinal como descrito nas análises acima não parecer ter muita diferença de comportamento em cada um dos casos, agindo sempre como um filtro passa-baixas que é capaz de reduzir o ruído, mas parece ter um limiar para atenuação do sinal. Após um certo valor de  $M$  ocorre perda de informação na comunicação visto que o áudio processado fica tão abafado que já não é mais possível ouvi-lo.

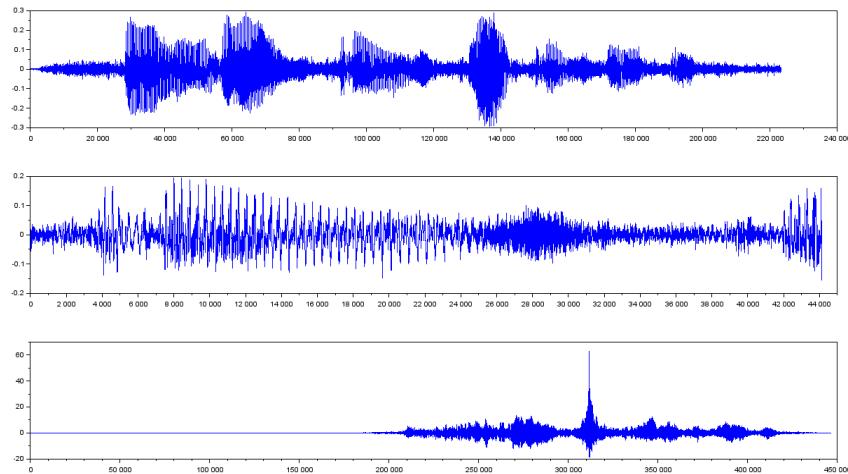


Figura 25: Análise da Amplitude do Áudio Original, Áudio Cortado entre 2 e 3 e a Correlação deles

```

//.Matheus Lenzi dos Santos - 19100420
1 // Indicando o caminho onde está salvo o áudio original no formato .wav
2 wave = "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio.wav";
3 wavread(wave,"Size")
4 wavread(wave,"Info")
5 //X1--> Amplitude do som entre [-1, 1], com uma linha por canal de gravação
6 //Fs --> Frequência de amostragem em Hz
7 //bits --> Número de bits em X
8 [X1,Fs,bits] = wavread(wave)
9
10 // Recuperando o tamanho do áudio original no formato .wav
11 wave_size = wavread(wave,"Size");
12
13 // Recuperando apenas um canal de gravação (Mono) áudio original
14 x = X1(:,1);
15
16 // Plotando o áudio original
17 subplot(3,1,1)
18 plot(x)
19
20 // Indicando o caminho onde está salvo o áudio cortado no formato .wav
21 wave2 = "C:\Users\Matheus\Desktop\Projeto Processamento de Voz\audio_cortado.wav";
22 wavread(wave2,"Size")
23 wavread(wave2,"Info")
24 //X2 --> Amplitude do som entre [-1, 1], com uma linha por canal de gravação
25 //Fs2 --> Frequência de amostragem em Hz
26 //bits2 --> Número de bits em X
27 [X2,Fs2,bits2] = wavread(wave2);
28
29 // Recuperando o tamanho do áudio cortado no formato .wav
30 wave_size2 = wavread(wave2,"Size");
31
32 // Recuperando apenas um canal de gravação (Mono) áudio cortado
33 x2 = X2(:,1,:);
34
35 // Plotando o áudio cortado
36 subplot(3,1,2)
37 plot(x2)
38
39 // Aplicando a correlação
40 y = xcorr(x,x2);
41
42 // Normalizando os valores de y
43 ynorm = y/max(y);
44
45 // Plotando o áudio original
46 subplot(3,1,3)
47 plot(ynorm)
48
49
50

```

Figura 26: Código da Análise da Correlação entre o Áudio Original e o Áudio Cortado entre 2 e 3

Comentário: O gráfico da amplitude dos sinais, aliado ao gráfico da correlação deles, é uma demonstração clara da similaridade entre eles, visto que um dos áudios é um corte do outro. É possível perceber por meio de uma análise mais minuciosa do gráfico da correlação que de fato o áudio cortado está presente dentro do áudio original e com uma precisão bem alta neste espaço temporal.

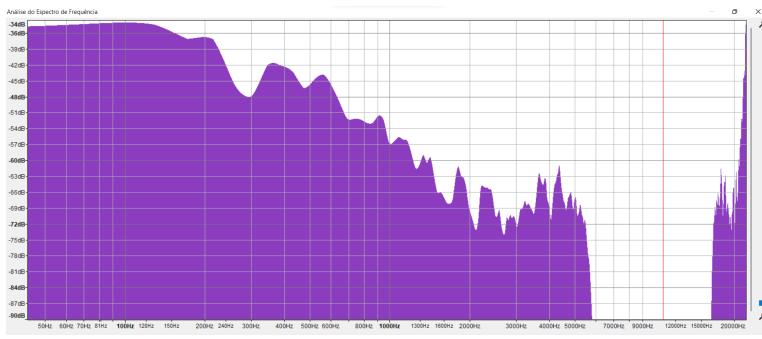


Figura 27: Análise do Espectro da Frequência do Aumento e Diminuição de Taxa pelo Fator 2

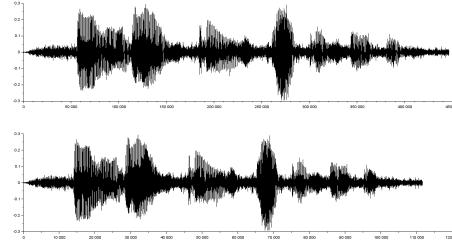


Figura 28: Análise da Amplitude do Aumento e Diminuição de Taxa pelo Fator 2

```

SciNotes - alteracao_de_taxa.sce (C:\Users\Matheus\Desktop\Projeto Processamento de Voz\alteracao_de_taxa.sce) - SciNotes
Arquivo Editar Formatar Opções Janela Executar ?
alteracao_de_taxa.sce (C:\Users\Matheus\Desktop\Projeto Processamento de Voz\alteracao_de_taxa.sce) - SciNotes
alteracao_de_taxa.sce

1 // .Matheus.Lenzini.dos.Santos. -- 19100420
2
3 // .Indicando o caminho onde está salvo o áudio original no formato .wav
4 wave = "C:\Users\Matheus\Desktop\Projeto.Processamento.de.Voz\audio.wav";
5 wavread(wave, "Size")
6 wavread(wave, "Info")
7 // X -> Amplitude do som entre [-1, 1], com uma linha por canal de gravação
8 // Fs -> Frequência de amostragem em Hz
9 // bits -> Número de bits em X
10 [X,fs,bits] = wavread(wave)
11
12 // Recuperando o tamanho do áudio original no formato .wav
13 wave_size = size(X)(2)
14
15 // Recuperando apenas um canal de gravação (Mono) áudio original
16 x = X(:,1);
17
18 // Definindo os valores iniciais dos vetores indexados
19 y_upscale = zeros(2*wave_size,1);
20 y_upscale(1) = 0;
21
22 // Implementando o aumento de taxa pelo fator 2
23 for n = 2:wave_size
24   .. y_upscale(2*n) = x(n)
25 end
26
27 // Plotando os valores de y
28 subplot(2,1,1)
29 plot3d(y_upscale)
30
31 // Escrevendo os valores de y no formato .wav
32 wavwrite(y_upscale, fs, bits, "C:\Users\Matheus\Desktop\Projeto.Processamento.de.Voz\audio_aumento_fator_2.wav")
33
34 // Definindo os valores iniciais dos vetores indexados
35 y_downscale = zeros(floor(wave_size/2),1);
36 y(1) = 0;
37
38 // Implementando a diminuição de taxa pelo fator 2
39 for n = 2:floor(wave_size/2)
40   .. y_downscale(n) = x(2*n)
41 end
42
43 // Plotando os valores de y
44 subplot(2,1,2)
45 plot3d(y_downscale)
46
47 // Escrevendo os valores de y no formato .wav
48 wavwrite(y_downscale, fs, bits, "C:\Users\Matheus\Desktop\Projeto.Processamento.de.Voz\audio_diminuicao_fator_2.wav")
49

```

Figura 29: Código Aumento e Diminuição de Taxa pelo Fator 2

Comentário: Com o aumento e diminuição de taxa pelo fator 2 a deformação no áudio original é muito significativa. O aumento da taxa prolonga a duração do áudio e por esse motivo distorce a voz no sentido de deixá-la mais grave. E a diminuição da taxa provoca uma redução na duração do áudio e dessa forma distorce a voz deixando-a mais aguda. O resultado do processamento desses áudios em algum sentido se assemelha ao fato de assistir vídeos em uma velocidade de reprodução diferente da qual foi gravado, onde também ocorre uma distorção, em menor escala devido a um processo de equalização, do áudio presente. No gráfico do espectro de amplitude do sinal é possível perceber a extensão e compressão da duração do áudio, mas o espectro de frequências parece não ter o comportamento alterado entre os casos, mas possui uma diferença substancial para o original.