

Universidade Federal do Rio Grande do Norte

Departamento de Engenharia de Computação e Automação

DCA3602 - Processamento Digital de Sinais

Aluno: Lucas Tomaz de Moura

23 de dezembro de 2024

Projeto de Filtro IIR seletivo em frequência:

Usando o MATLAB (ou equivalente), especifique e projete um sistema IIR, $H(z)$, que filtre de uma amostra de seu sinal de voz $v(t)$, com duração de 5 segundos, um sinal de ruído propositalmente adicionado à $v(t)$, e representado por $r(t) = a_1 \cos(2\pi f_1 t) + a_2 \cos(2\pi f_2 t)$, em que f_n (Hz) representa os valores de frequência do ruído. O filtro deve garantir uma atenuação mínima do ruído igual a $A_r = 20 \log_{10} \delta_2$ na faixa de rejeição. Utilize as funções do matlab `audiorecorder`, `recordblocking` e `getaudiodata`, ou equivalente, para capturar uma amostra de sinal de voz. Além disso, utilize a função `filter`, ou equivalente, para filtrar o ruído do sinal de áudio contaminado, utilizando a função $H(z)$ projetada. Mostre os gráficos do sinal de áudio $v[n]$, do sinal corrompido pelo ruído $z[n] = v[n] + r[n]$, e do sinal filtrado $y[n]$, em ambos os domínios discretos: tempo e frequência (utilize as funções `fft` e `fftshift`, ou o algoritmo de aproximação da TFTD desenvolvido anteriormente, ou equivalentes, para obter os sinais no domínio da frequência). Também apresente a resposta em frequência do filtro projetado (definida em termos da resposta em magnitude e resposta de fase, obtidas por sua vez, por meio da função `freqz`, ou equivalente). Por fim, salve o sinal de voz original (antes da adição do ruído) e o sinal de voz filtrado ($y[n]$) em arquivos de áudio WAV utilizando a função `audiowrite` do matlab, ou equivalente.

Parâmetros do filtro:

Com base no enunciado, foi implementado um filtro do tipo IIR butterworth passa-baixa que obedece os parâmetros n° 7 do arquivo disponibilizado, no qual seus valores são: $f_1 = 5.7KHz$, $f_2 = 6KHz$ e $\delta_2 = 0.1$. Com base nisso, a frequência de amostragem escolhida foi de 44100 Hz, tendo em vista que foi a frequência de amostragem retornada pela função de voz do software octave.

Implementação dos parâmetros:

Todo o projeto do filtro IIR foi desenvolvido no software octave. Abaixo está o modelo de pseudocódigo descrevendo a lógica de execução do código. Na figura 6, está elencado a respostas em frequência e fase do filtro IIR projetado.

Resultados

Com base no algoritmo apresentando, foi possível obter o sinal de voz (figura 1), o sinal de voz adicionado com um ruído nas frequências especificadas nos parâmetros (figura 2 e 3) e o sinal de voz após ser filtrado (5). Com base nesses resultados, posso considerar que houve uma atenuação totalmente satisfatória do ruído. Fazendo uma comparação visual entre os espectros do sinal com ruído e o sinal original, o sinal filtrado conseguiu cumprir com o seu objetivo. Apesar de ter garantido um valor satisfatório de atenuação, o sinal de voz filtrado ainda apresenta um som abafado. Acredito que esse fator possa ser tratado utilizando outras técnicas de processamento.

Algoritmo 1: Filtro de Ruído em Áudio

```
1:  $a_1 \leftarrow 0.1$                                 ▷ Coeficiente do primeiro componente de ruído
2:  $a_2 \leftarrow 0.1$                                 ▷ Coeficiente do segundo componente de ruído
3:  $freq\_ruído\_1 \leftarrow 5700$                     ▷ Primeira frequência de ruído em Hz
4:  $freq\_ruído\_2 \leftarrow 6000$                     ▷ Segunda frequência de ruído em Hz
5:  $senal \leftarrow []$                                 ▷ Inicializa o vetor de ruído gerado
6:  $caminho \leftarrow \text{'Teste\_de\_voz.wav'}$             ▷ Caminho do arquivo de áudio
7:  $[Y\_voz, Fs\_voz] \leftarrow \text{audioread}(caminho)$     ▷ Lê o áudio do arquivo
8:  $Ts \leftarrow 1/Fs\_voz$                             ▷ Período de amostragem
9:  $Y\_voz \leftarrow Y\_voz(:, 1)$                     ▷ Seleciona o canal mono (se necessário)
10:  $N \leftarrow \text{length}(Y\_voz)$                     ▷ Tamanho do vetor de áudio
11: Para  $n \leftarrow 0$  to  $N - 1$  Faça
12:    $ruído \leftarrow a_1 \cdot \cos(2\pi \cdot freq\_ruído\_1 \cdot n \cdot Ts) + [2] a_2 \cdot \cos(2\pi \cdot freq\_ruído\_2 \cdot n \cdot Ts)$ 
13:    $senal \leftarrow senal \cup [ruído]$                 ▷ Adiciona o ruído ao vetor
14: end Para
15:  $senal\_total \leftarrow Y\_voz + \text{transpose}(senal)$     ▷ Soma o ruído ao áudio original
16:  $Wp \leftarrow 2 \cdot 2300 \cdot Ts$                 ▷ Frequência de passagem normalizada
17:  $Ws \leftarrow 2 \cdot 3000 \cdot Ts$                 ▷ Frequência de rejeição normalizada
18:  $Gp \leftarrow 0.1$                                 ▷ Atenuação máxima na banda de passagem (dB)
19:  $Gs \leftarrow 30$                                 ▷ Atenuação mínima na banda de rejeição (dB)
20:  $[N, w0] \leftarrow \text{buttord}(Wp, Ws, Gp, Gs)$         ▷ Ordem do filtro Butterworth
21:  $[B, A] \leftarrow \text{butter}(N, 1, 's')$             ▷ Coeficientes do filtro no domínio analógico
22:  $[b, a] \leftarrow \text{lp2lp\_oc}(B, A, w0 \cdot \pi/Ts)$     ▷ Transformação para frequência desejada
23:  $[bt, at] \leftarrow \text{bilinear}(b, a, Ts)$           ▷ Transformação bilinear para o domínio digital
24:  $xbutter \leftarrow \text{filter}(bt, at, senal\_total)$     ▷ Filtra o sinal com o filtro Butterworth
25:  $\text{audiowrite}(\text{'voz\_filtrada.wav'}, xbutter, Fs\_voz)$  ▷ Salva o sinal filtrado
26:  $\text{sound}(xbutter, Fs\_voz)$                         ▷ Opcional: reproduz o áudio filtrado
27:  $\text{plotspec}(xbutter, Ts)$                         ▷ Opcional: exibe o espectro do sinal filtrado
```

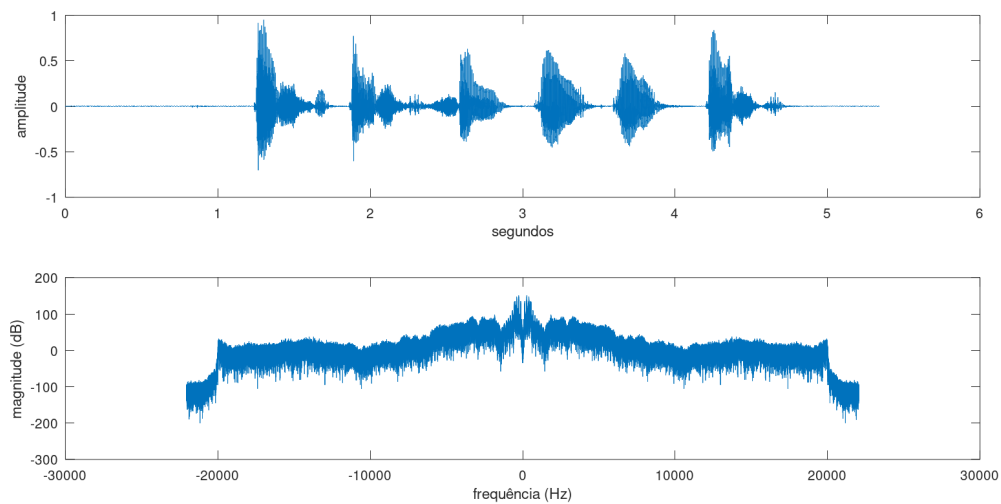


Figura 1: .Representação gráfica do sinal de voz original no domínio do tempo e frequência, respectivamente.

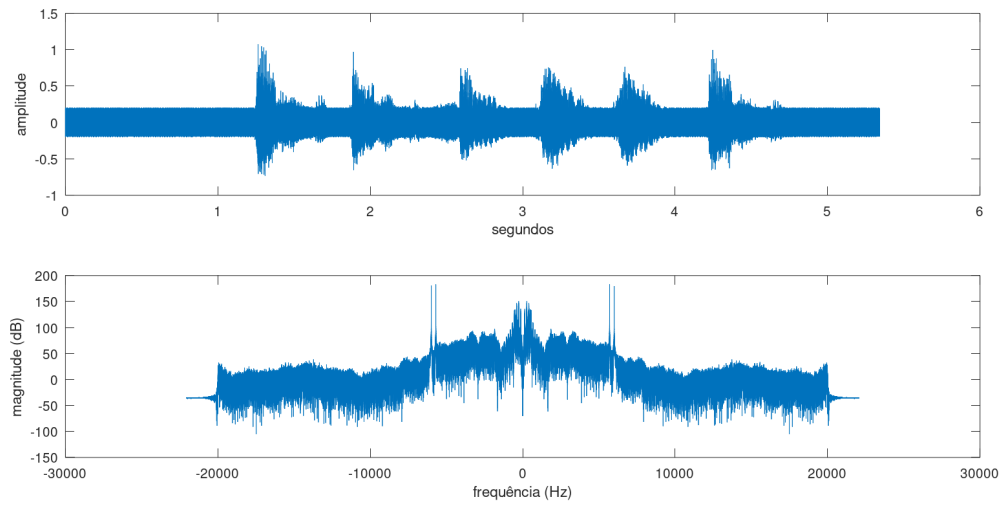


Figura 2: Representação gráfica do sinal de voz ao ser adicionado o ruído característico, no domínio do tempo e frequência, respectivamente.

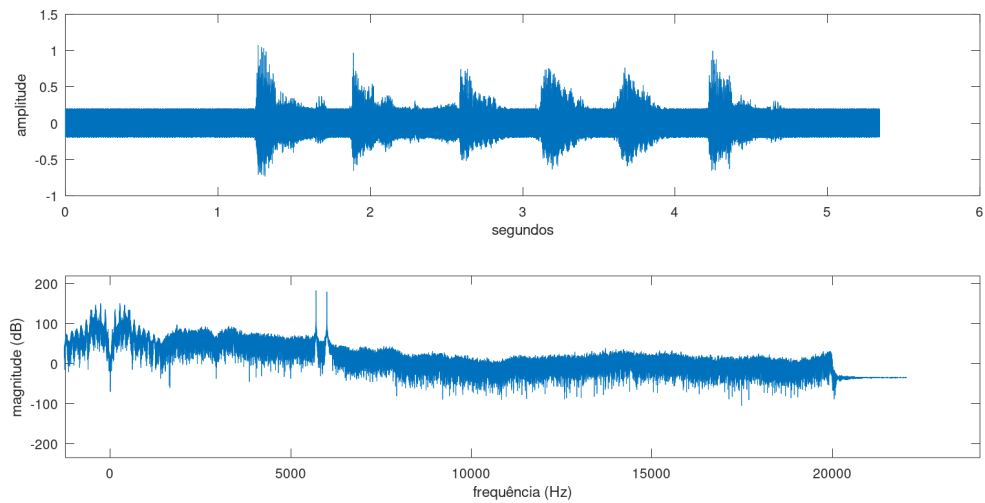


Figura 3: Representação gráfica do sinal de voz com o ruído considerando um zoom na região de frequência correspondente aos parâmetros de ruído entre 5,7K e 6K.

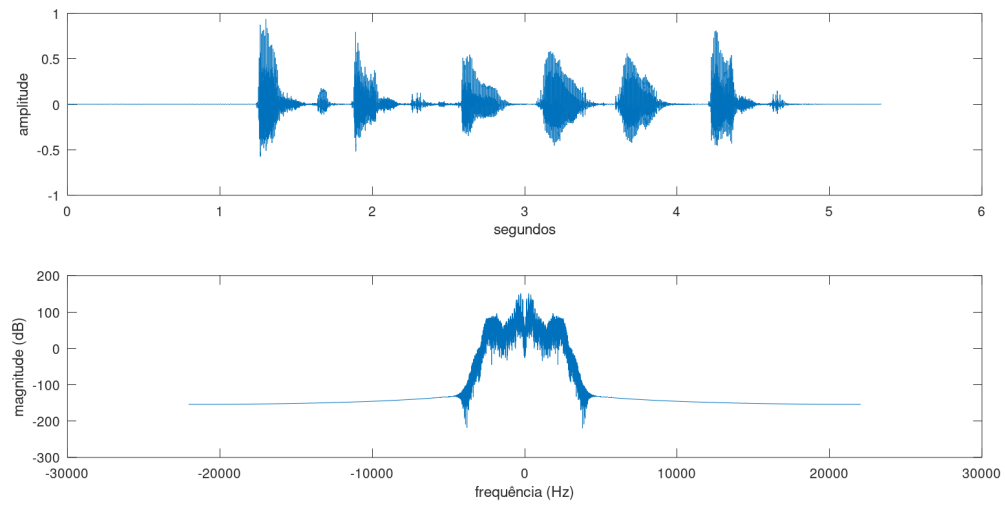


Figura 4: Representação gráfica do sinal de voz filtrado utilizando a função de transferência encontrada no filtro butterworth passa-baixa, no domínio do tempo e frequência, respectivamente.

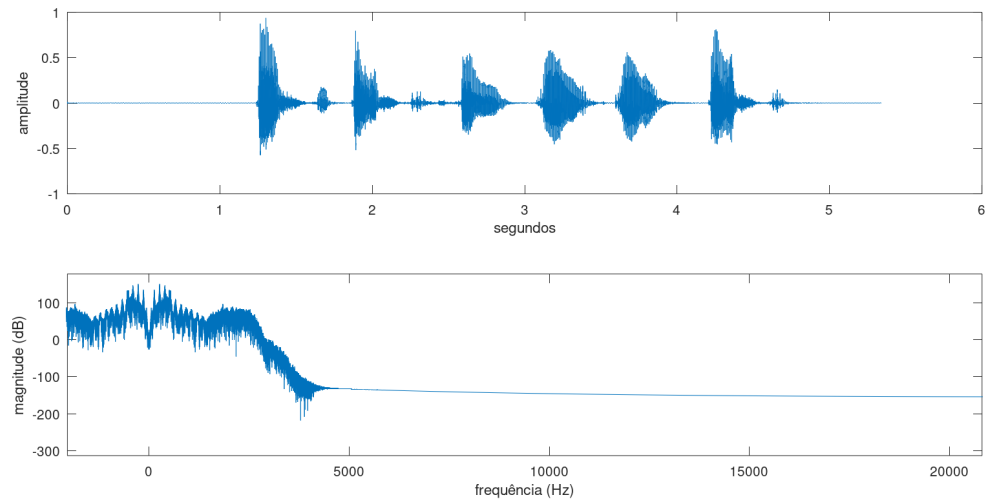


Figura 5: Representação gráfica do sinal de voz filtrado considerando um zoom na região de frequência correspondente aos parâmetros de ruído atenuados entre 5,7K e 6K.

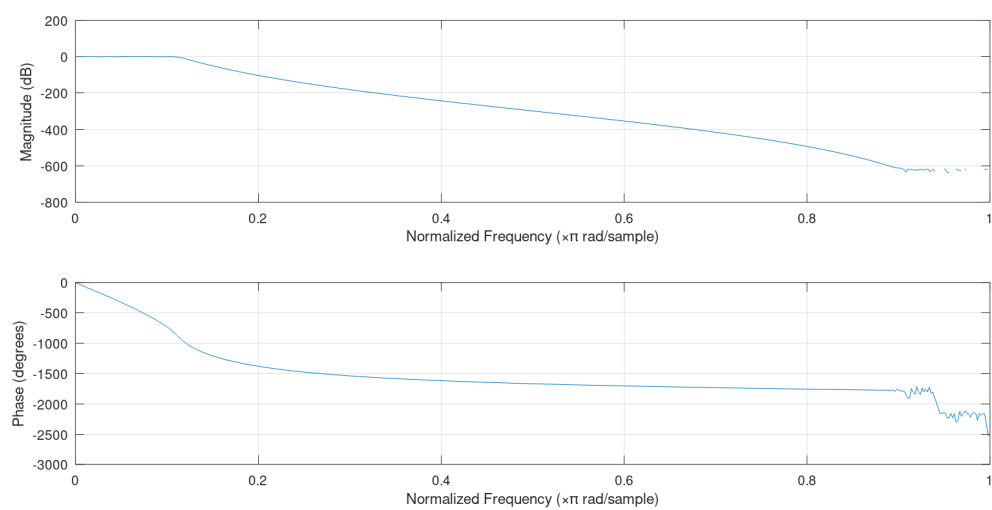


Figura 6: Resposta em frequência e fase do filtro, respectivamente.