

# Universidade Federal do Rio Grande do Norte

Departamento de Engenharia de Computação e Automação

DCA3602 - Processamento Digital de Sinais

Aluno: Lucas Tomaz de Moura

28 de novembro de 2024

## Projeto de Filtro FIR seletivo em frequência(parte 3):

Usando o filtro passa-baixa FIR projetado na parte 1 dessa tarefa, filtre de uma amostra de sua voz  $v(t)$ , um sinal de ruído propositadamente 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 escolhidos durante a Parte 1 dessa tarefa. 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 resposta ao impulso projetada anteriormente. 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 (a função *plotspec.m*, ou função equivalente, para obter os sinais no domínio da frequência). Também apresente a resposta ao impulso e 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 FIR que obedece os parâmetros nº 7 do arquivo disponibilizado na parte 1, no qual seus valores são:  $f_1 = 5.7KHz$ ,  $f_2 = 6KHz$ ,  $\delta_2 = 0.01$  e  $\Delta\omega = 0.05\pi$ . Com base nisso, temos os parâmetros  $N = 36$  e 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 FIR foi desenvolvido no software octave. Abaixo está o modelo de pseudocódigo descrevendo a lógica de execução do código. Nas figuras 5 e 6, estão elencados as respostas ao impulso do filtro.

### 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(4). Com base nesses resultados, posso considerar que houve uma atenuação considerável 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 pouco de ruído. Acredito que o valor da atenuação não foi de maneira completa devido aos parâmetros do filtro, levando em conta que não escolhi uma faixa de valores de frequência tão grande em relação a faixa de frequência do ruído. Tendo em vista que os parâmetro de ruído seriam 5.7K e 6K Hertz, os valores do intervalo escolhido para atenuação foram 5.0K e 6.3K Hertz.

---

**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:  $sinal \leftarrow []$                                 ▷ Inicializa o vetor de sinal final
4:  $h\_valores \leftarrow []$                             ▷ Inicializa a lista de valores de filtro  $h$ 
5:  $caminho \leftarrow \text{'Teste\_de\_voz.wav'}$             ▷ Caminho do arquivo de áudio
6:  $[Y\_voz, Fs\_voz] \leftarrow \text{audioread}(caminho)$     ▷ Lê o áudio
7:  $Ts \leftarrow 1/Fs\_voz$                             ▷ Período de amostragem
8:  $Y\_voz \leftarrow Y\_voz(:, 1)$                     ▷ Seleciona o canal mono (se necessário)
9:  $N \leftarrow \text{length}(Y\_voz)$                     ▷ Determina o tamanho do vetor de áudio
10:  $freq\_ruído_1 \leftarrow 5700$                     ▷ Primeira frequência de ruído em Hz
11:  $freq\_ruído_2 \leftarrow 6000$                     ▷ Segunda frequência de ruído em Hz
12:
13: Para  $n \leftarrow 0$  a  $N - 1$  Faça
14:   Se  $n = 18$  Então
15:      $h \leftarrow 1 + 0.205 - 0.315$ 
16:   Senão
17:      $freq\_1 \leftarrow \frac{\sin(\frac{41}{200}\pi(n-18))}{\pi(n-18)}$ 
18:      $freq\_2 \leftarrow \frac{\sin(\frac{63}{200}\pi(n-18))}{\pi(n-18)}$ 
19:      $h \leftarrow freq\_1 - freq\_2$ 
20:      $ruído \leftarrow a_1 \cdot \cos(2\pi \cdot freq\_ruído_1 \cdot n \cdot Ts) + a_2 \cdot \cos(2\pi \cdot freq\_ruído_2 \cdot n \cdot Ts)$ 
21:      $\text{disp}(ruído)$                                 ▷ Exibe o valor do ruído
22:      $\text{disp}(h)$                                     ▷ Exibe o valor do filtro  $h$ 
23:   end Se
24:    $h\_valores \leftarrow h\_valores \cup [h]$           ▷ Adiciona o valor de  $h$  à lista
25:    $sinal \leftarrow sinal \cup [ruído]$               ▷ Adiciona o ruído ao vetor de sinal
26: end Para
27:  $sinal\_total \leftarrow Y\_voz + sinal'$             ▷ Soma o sinal de voz com o ruído gerado
28:  $sinal\_filtrado \leftarrow \text{filter}(h\_valores', 1, sinal\_total)$   ▷ Filtra o sinal com o filtro  $h$ 
29:  $\text{audiowrite}(\text{'voz\_filtrada.wav'}, sinal\_filtrado, Fs\_voz)$   ▷ Escreve o sinal filtrado em um
    arquivo
30:  $\text{sound}(sinal\_filtrado, Fs\_voz)$                 ▷ Reproduz o sinal filtrado
31:  $\text{plotspec}(sinal\_filtrado, Ts)$                 ▷ Plota o espectro do sinal filtrado
```

---

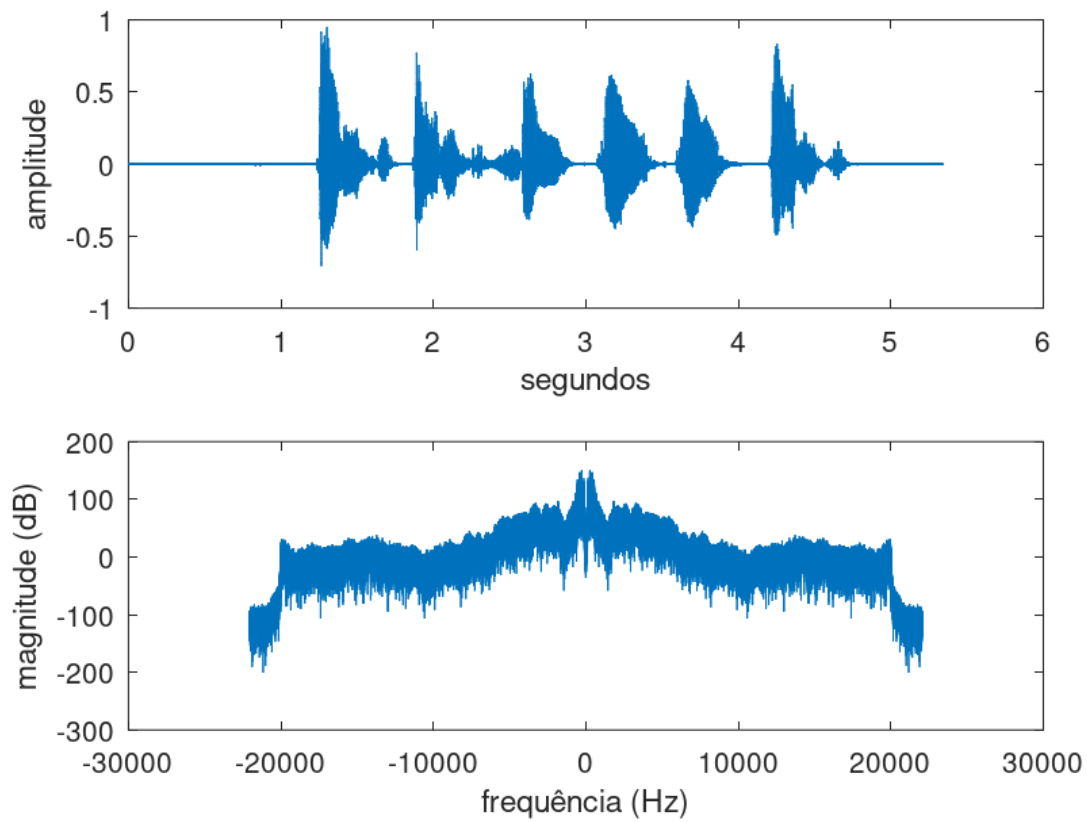


Figura 1: .Representação gráfica do sinal de voz no domínio do tempo e frequência, respectivamente.

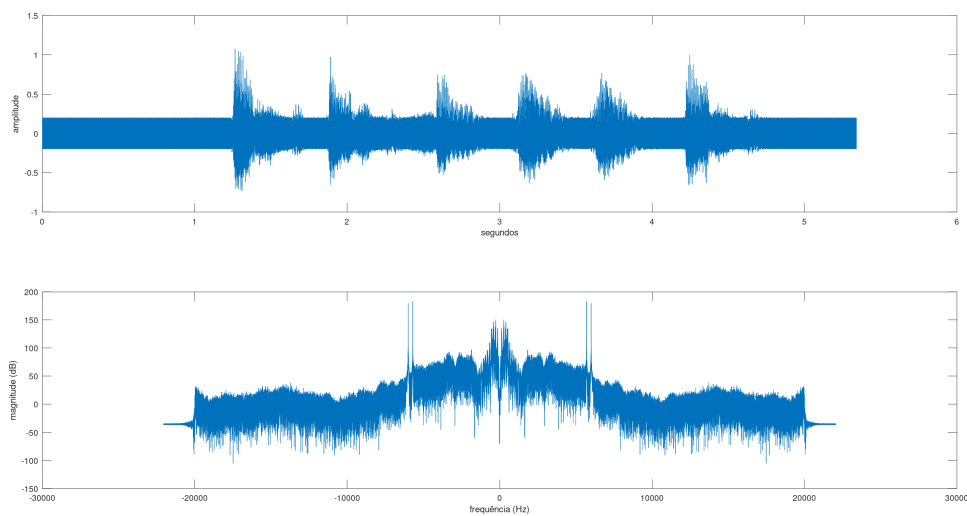


Figura 2: Representação gráfica do sinal de voz ao ser adicionado o ruído característico, no domínio do tempo e frequência, respectivamente.

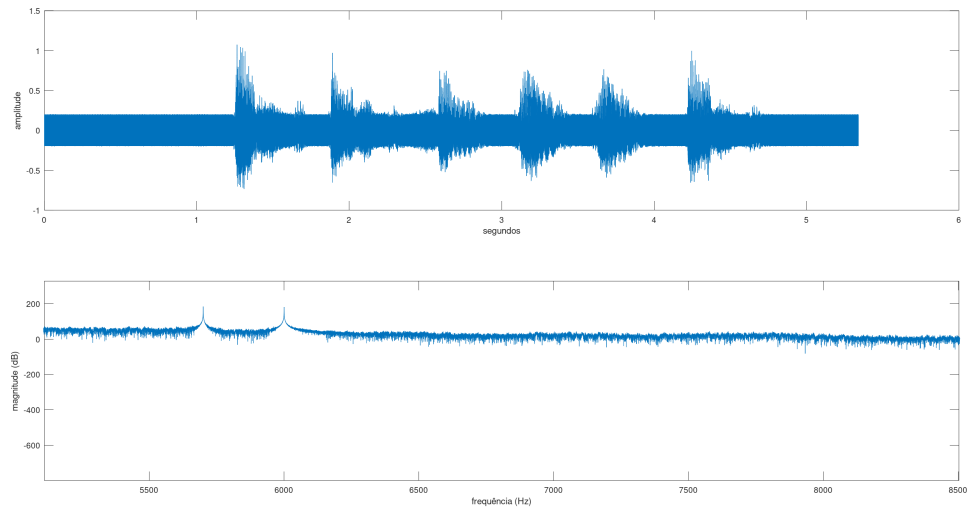


Figura 3: Representação gráfica do sinal de voz com o ruído considerando um zoom na região de frequência correspondente aos parâmetros de ruído entre 5,7K e 6K.

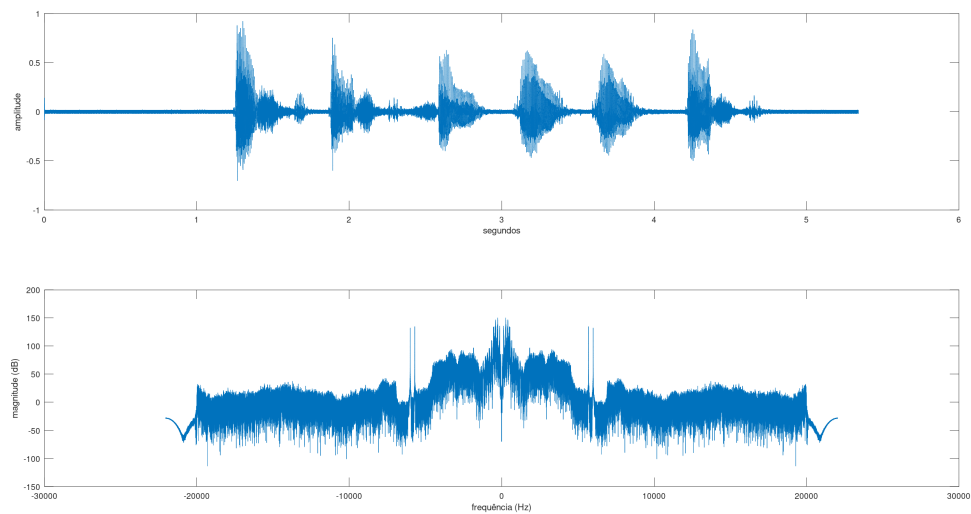


Figura 4: Representação gráfica do sinal de voz filtrado utilizando a função de transferência encontrada no filtro rejeita-faixa, demonstrado na parte 1, no domínio do tempo e frequência, respectivamente.

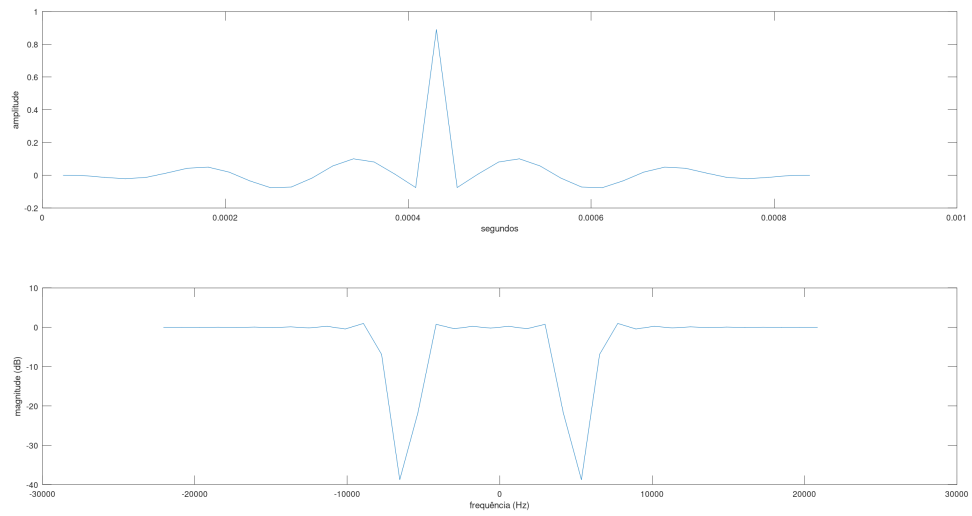


Figura 5: Resposta do impulso do filtro, no domínio do tempo e frequência.

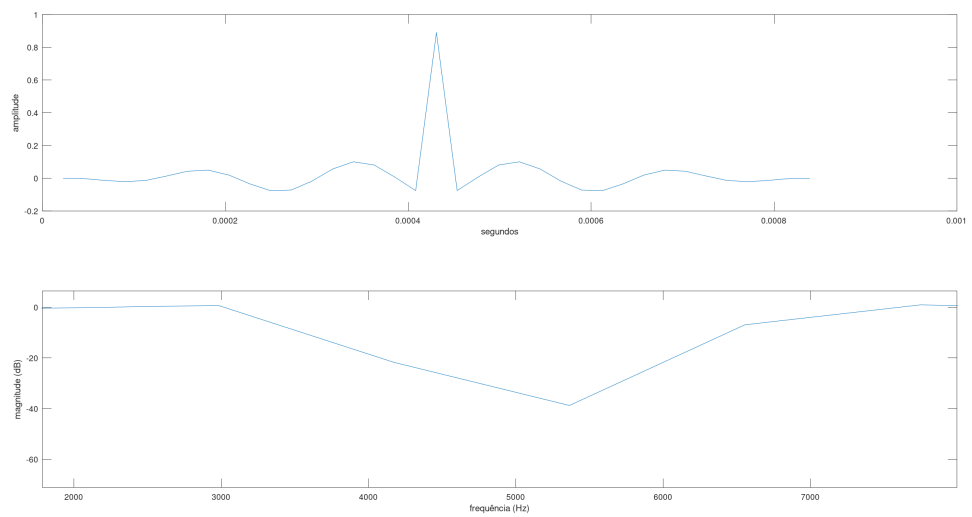


Figura 6: Resposta do impulso do filtro, no domínio do tempo e frequência, dando um zoom na região de atenuação.