

Filtro FIR por janela de Kaiser

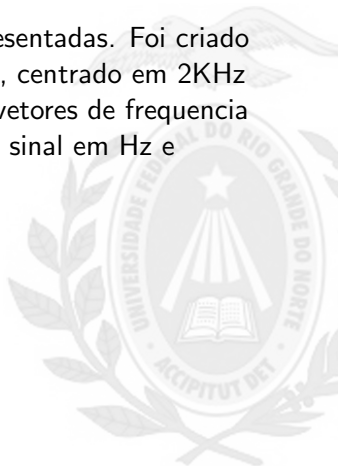
23 de Outubro de 2019



Especificações e parâmetros

O filtro é modelado seguindo as diretivas apresentadas. Foi criado um ruído branco, filtrado com um passa faixa, centrado em 2KHz limitado em faixa por 300Hz. Foram criados vetores de frequência e de tempo do para colocar a amostragem do sinal em Hz e segundos respectivamente.

```
breaklines
1 [y,Fs]=audioread('pds_fir.mp3');
2 s=length(y);
3 %frequencias normalizadas (f)
4 fs1=1700/(Fs/2);
5 fp1=1850/(Fs/2);
6 fp2=2150/(Fs/2);
7 fs2=2300/(Fs/2);
8 f0=Fs/s;
9 f1=f0:f0:Fs/2;
10 f2=-Fs/2:f0:-f0;
11 f=[f2,f1];
12 t0=1/Fs;
13 t= t0:t0:s/Fs;
```



Codificação do Ruído

```
breaklines
1 white = randn(s,1);
2 white = max(y)*white/max(white);
3 b = firpm(1000,[0 fs1 fp1 fp2 fs2 1],[0 0 1 1 0 0]);
4 colornoise = filter(b,1,white);
5
6 sujo = colornoise+y;
```

Desse modo o filtro cria uma limitação em banda de 300Hz centrado em 2kHz 150Hz para cada lado e com pequena região de transição

$$fs1 = 1700Hz \quad fp1 = 1850Hz \quad fp2 = 2150Hz \quad fs2 = 2300Hz \quad (1)$$

Audio

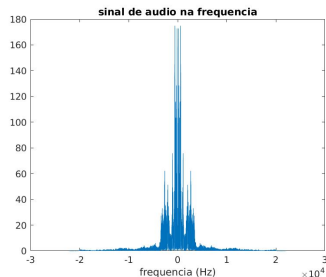
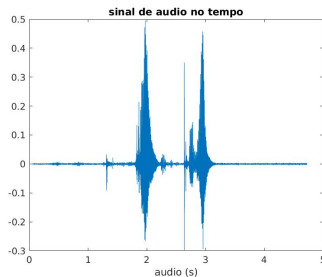


Figura 1: Sinal de audio no tempo e na frequencia.

Ruído limitado em banda

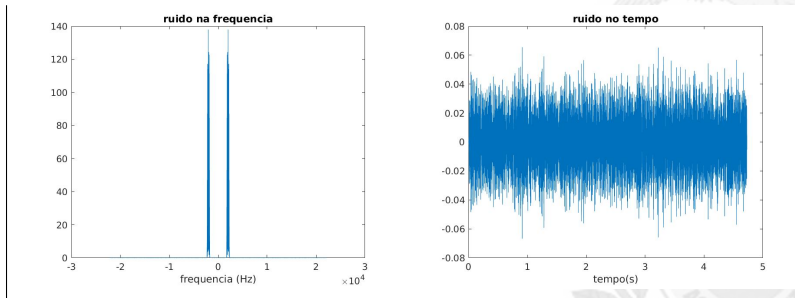


Figura 2: ruído na frequência e no tempo

A frequência F_s é de 44100 para produzir o ruído é normalizado em $F_s/2$

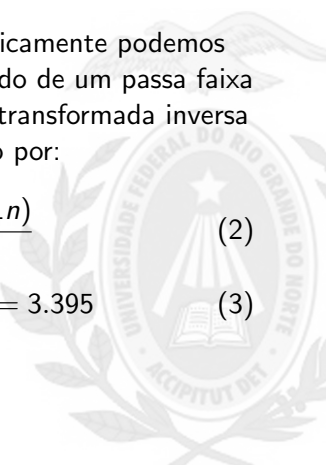


Definição do Filtro

O filtro é por definição um rejeita faixa, analiticamente podemos pensar na frequência como 1(ativador) subtraído de um passa faixa (OPPENHEIM; SCHAFER, 2010). fazendo a transformada inversa de fourrier temos que nosso filtro será definido por:

$$\delta(n) = \frac{\sin(wc2n)}{\pi n} + \frac{\sin(wc1n)}{\pi n} \quad (2)$$

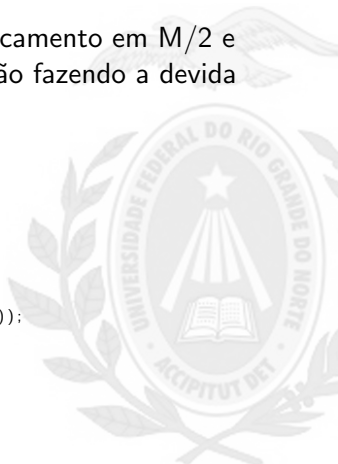
$$\delta = \epsilon = 0.01 - > Ar = 40DB, \beta = 3.395 \quad (3)$$



Definição do Filtro

Seguindo a definição anterior temos um deslocamento em $M/2$ e foi utilizado um sinc para evitar indeterminação fazendo a devida compensação

```
breaklines
1 M = ceil(32/(2.285*pi*100/(Fs/2)));
2 M=M+mod(M,2);
3 w = kaiser(M+1,3.329);
4 n=[1:M+1];
5
6 fc1=(fp1+fs1)/2;
7 fc2=(fp2+fs2)/2;
8
9 hd = ((sinc((fc1)*(n-M/2))*fc1)-(sinc(fc2*(n-M/2))*fc2));
10
11 hd(M/2)=hd(M/2)+1;
12
13 h=(hd'.*w);
14
15 out = filter(h,1,sujo);
```



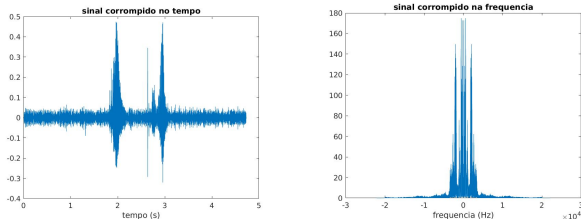


Figura 3: Sinal corrompido no tempo e na frequencia

Gráficos

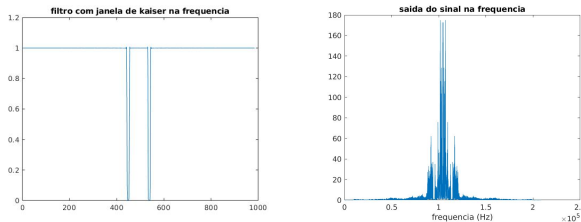


Figura 4: Filtro e saída do sinal na frequência.

Gráficos

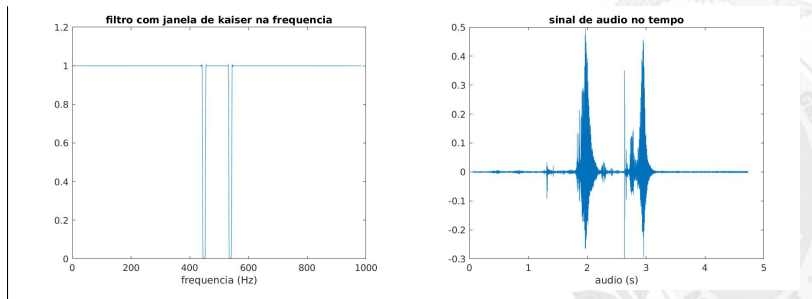


Figura 5: Comparação no tempo.

Gráficos

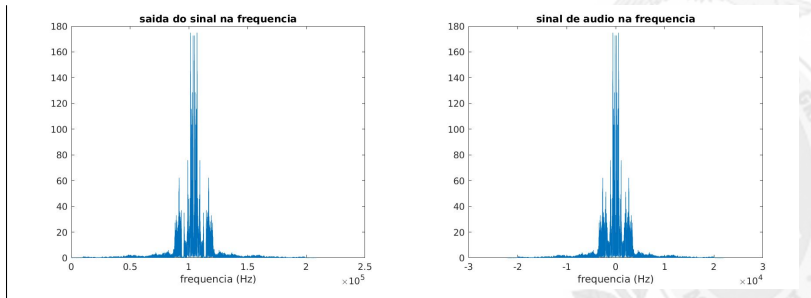


Figura 6: Comparação na frequencia.

Referências

 OPPENHEIM, A. V.; SCHAFER, R. W. *Discrete-time Signal Processing*. 2nd. ed. [S.l.]: Pearson Higher Education, 2010.

