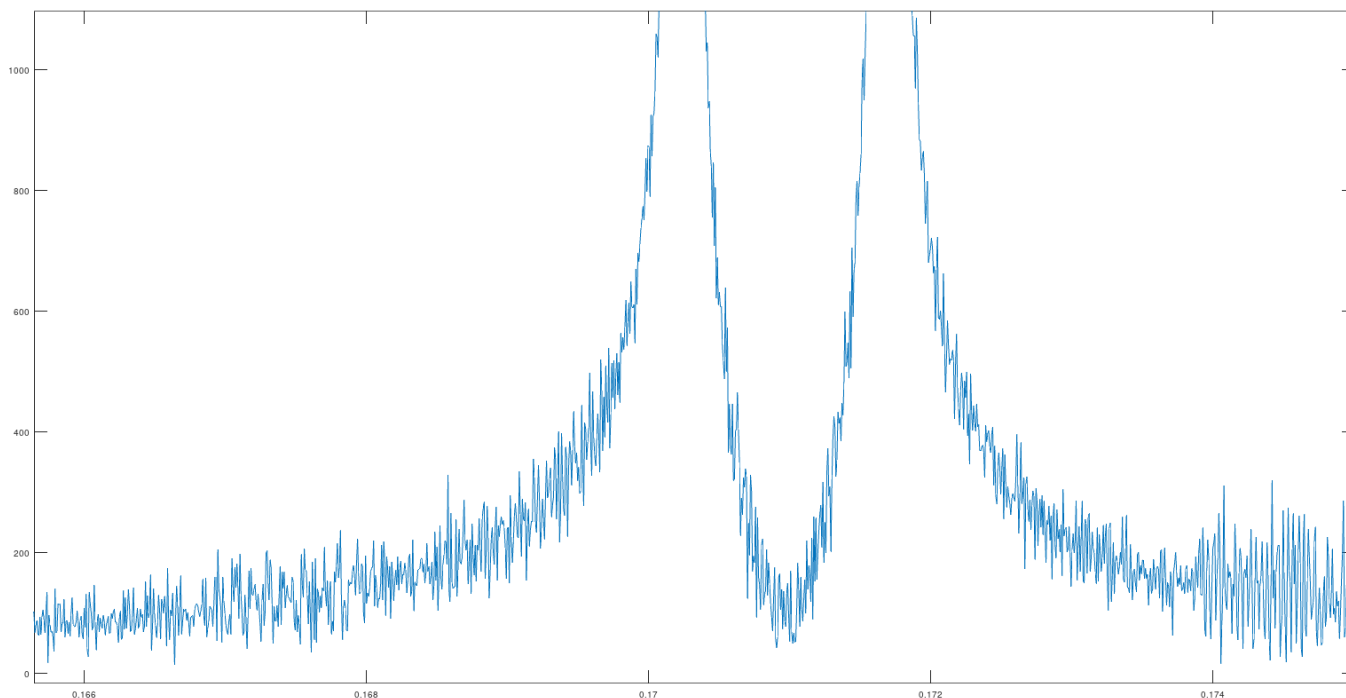


TS P12

Transformada de Fourier da Sequência $x(k)$

Transformada de Fourier, ampliado na frequência da campainha.



Filtro para rejeição da gama de frequência identificada

$$\omega_A = 0.16 \text{ e } \omega_B = 0.18$$

$$\omega_0 = \frac{(0.16 + 0.18)}{2} \text{ e } \omega_l = 0.18 - 0.16$$

$$h(k) = \delta(k) - 2 \cos(\omega_0 k) \frac{\sin(\frac{\omega_l}{2} k)}{\pi k}$$

$$\log h_0 = 1 - \frac{\omega_l}{\pi}$$

Implementação

```

[x,fs]=audioread(["sinal_4491.wav"]); % importa o registo sonoro
who % lista as variaveis existentes no workspace

N= length(x) % comprimento do registo x(k)
Xw= fft(x); % transformada de Fourier X(w) -- w: [ 0 , 2*pi]
w= (0:N-1)*2*pi/N;
plot(w,abs(Xw)) % representa p-ao grafica de | X(w) |

N1=15000; % este valor pode ter de ser ajustado
k= -N1:N1; % intervalo de tempom da resp.impulsional

W0 = (0.16 + 0.18)/2
Wl = 0.18 - 0.16
h = (k==0) -2*cos( W0.*k ).*( ( sin( (Wl/2).*k ) )./(pi.*k) );

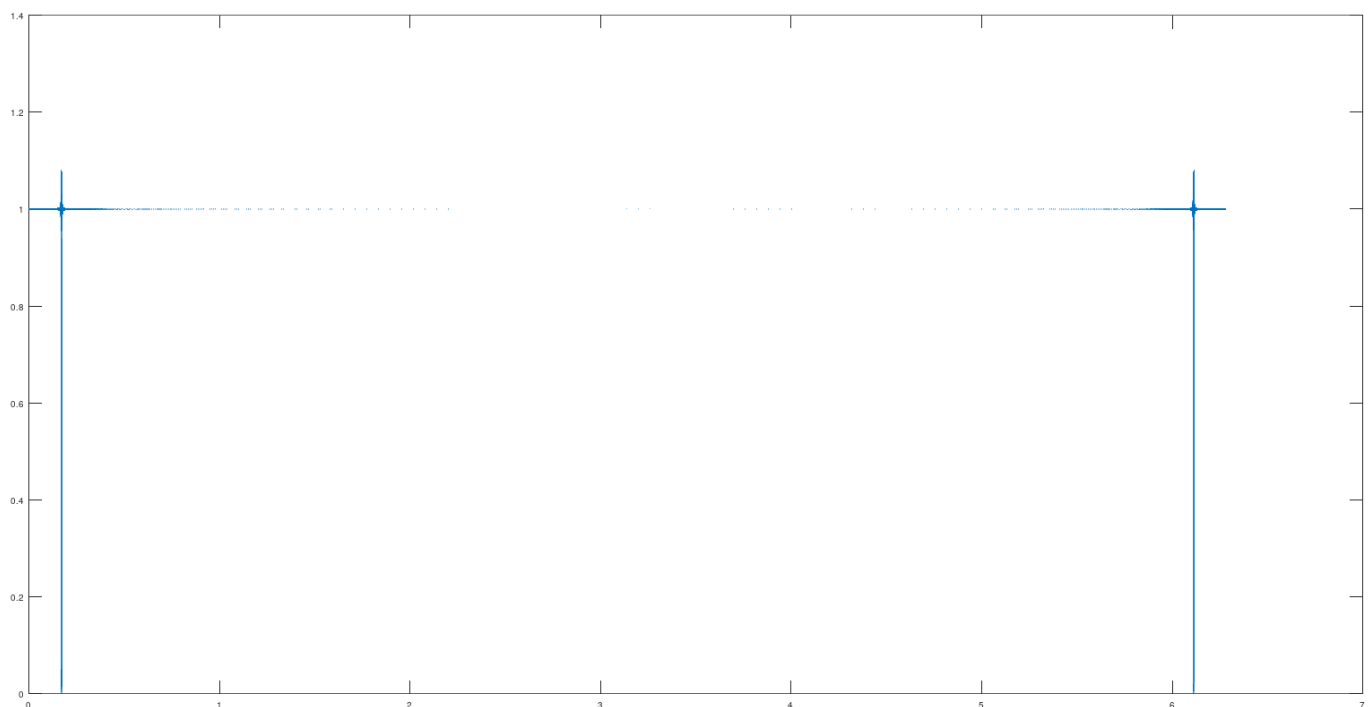
h( find(k==0) )= 1 - (Wl/pi) ; % o valor que calculou
y=conv( h , x );
G_amplif=0.8/max(abs(y)); % factor de amplificacao, melhora
% audicao do excerto
audiowrite(["sinal_limpo.wav"],G_amplif*y,fs); % guarda sinal

Hw=fft( h );
Nh=length(h);
wh= (0 : Nh-1 )* 2*pi/Nh;
plot(wh , abs( Hw ) )
% "limpo" que pode ser escutado com o reprodutor de audio do seu
% computador

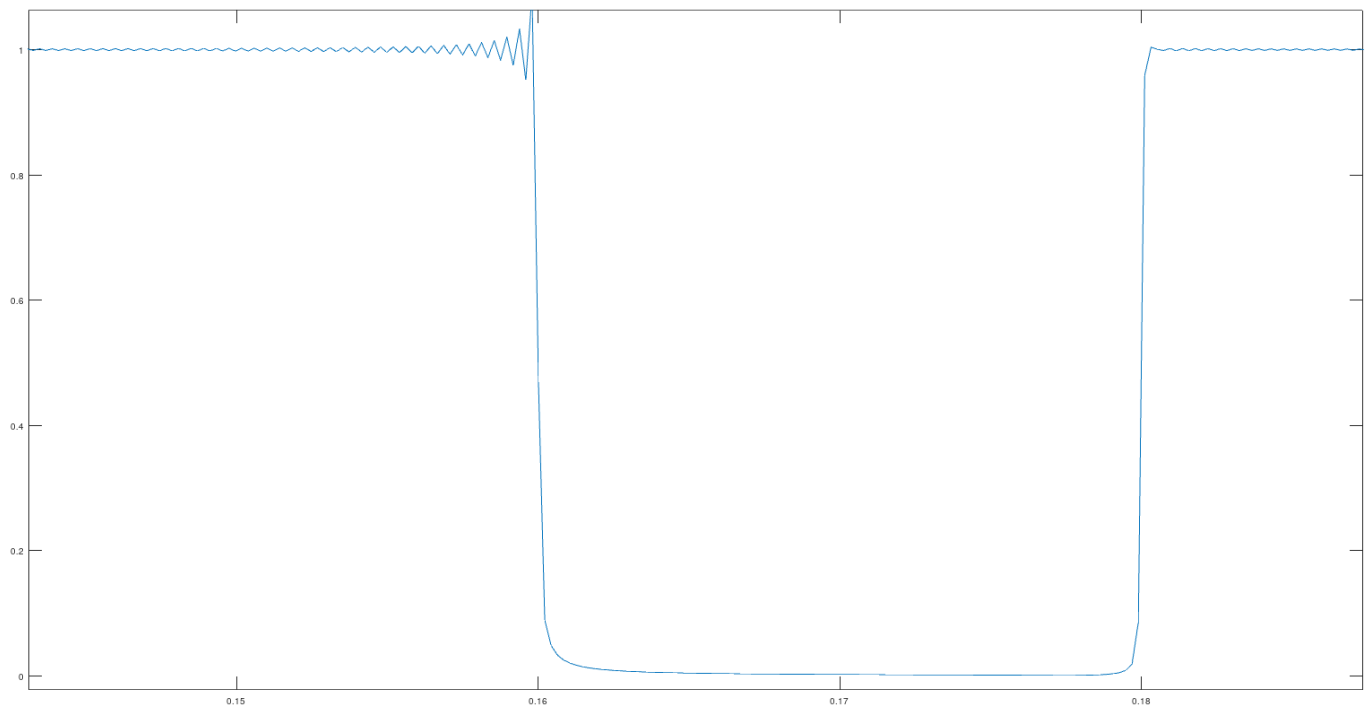
```

No ficheiro final, embora o som da campainha ainda seja perceptível esta consideravelmente mais atenuado.

Transformada de Fourier do filtro



Com zoom na zona das frequência da campainha.



O filtro não é ideal, observa-se o fenômeno de Gibbs quando passa para 0, e não é perfeitamente reto onde devia.

Para diminuir estas diferenças é necessário aumentar o *sampling rate*.