Projeto de Filtros FIR pelo método de Janelas

Table of Contents

Autor: Ignacio Sánchez-Gendriz

Data: 2022-06-06

```
close all
clearvars
clc
```

Resposta a impulso h_{lp}, de um filtro passa baixas ideal

Em aulas anteriores (ver tema de TFTD) vimos como obter a TFTD inversa do pulso rectangular. Podemos usar este conhecimento para obter a resposta ao impulso $h_{\mathrm{lp}}[n]$ de um filtro passa-baixas ideal partindo da resposta em frequncia $H_{\mathrm{lp}}(e^{j\omega})$

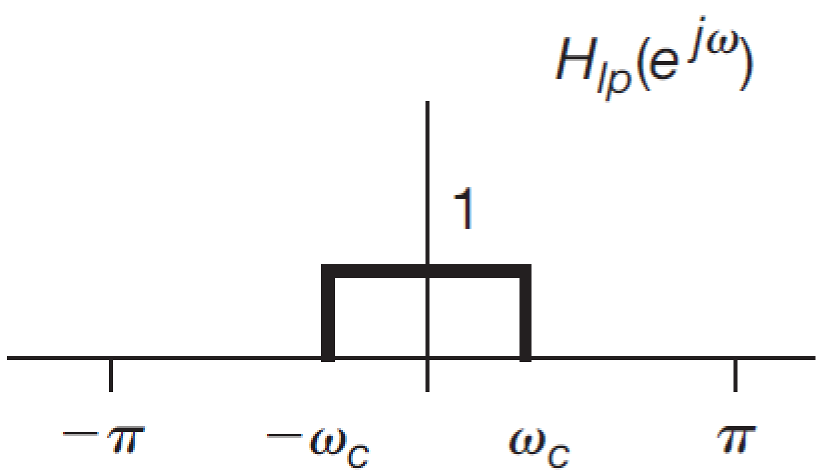


Figura 1. Resposta em frequencia de um filtro passa-baixas ideal

$$h_{lp}[n] = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega n} d\omega = \frac{1}{2j\pi n} e^{j\omega n}\Big]_{-\omega_c}^{\omega_c}$$

$$h_{\text{lp}}[n] = \frac{\omega_c}{\pi} \frac{\sin(\omega_c n)}{\omega_c n},$$

$$\omega_c = \frac{\pi \cdot f_c}{f_s/2}$$

A seguir usamos a fórmula anterior para representar $h_{\text{lp}}[n]$. Lembramos que o filtro ideal tem $h_{\text{lp}}[n]$ com duração infinita, por isso a resposta ao impulso se extende para ambos os lados (além dos pontos ilustrados na figura).

1. Obtendo a resposta ao impulso para a frequencia de corte desejada

```
fs = 200; % sample frequency in Hz
fc = 25; % cutt-off frequency in Hz
wc = pi*fc/(fs/2); % for fs = 200, wc = pi/4;

% Indices para amostras da resposta ao impulso
N1 = 110;
n1 = -N1:N1;
NSamples = length(n1);
hlp = wc/pi*(sin(wc*n1))./(wc*n1);
hlp(n1==0) = wc/pi;

fig = figure('Position',[100 100 400 200],'color','w');
stem(n1,hlp,'Marker','.');
title('Ideal Impulse response h_{lp}');
xlabel('Samples');
ylabel('Amplitude');
xlim([n1(1) n1(end)])
```

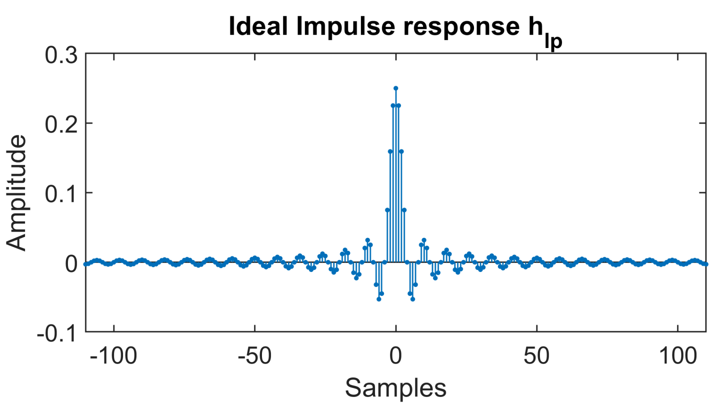


Figura 2. Representação da resposta ao impulso do filtro passa-baixas ideal

Como $h_{\text{lp}}[n]$ é infinita, teremos que usar alguma técnica para obter uma versão finita da resposta ao impulso. Veremos a seguir os efeitos de usar somente uma parte de $h_{\text{lp}}[n]$.

$$h_w[n] = h_{\text{lp}}[n] \cdot w[n]$$

$$\begin{cases} w[n] = 1 & -\dfrac{M-1}{2} \leq n \leq \dfrac{M-1}{2} \\ w[n] = 0 & \text{nos casos restantes} \end{cases}$$

2. Usando um segmento finito da resposta ao impulso: janela rectangular

Usar uma jenela rectangular é multiplicar por 1. Simplesmente selecionamos parte dos coeficientes desejados, a partir de $h_{\text{lp}}[n]$.

Definimos como M o número de coefficentes do filtro.

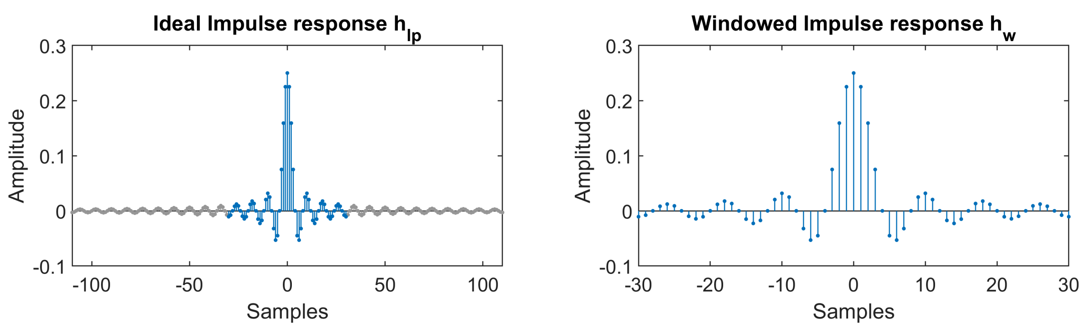
Lembramos que para os filtros FIR os coeficientes do filtro são a mesma coisa que os valores da resposta ao impulso.

```
M = 61;
M1 = -(M-1)/2;
M2 = (M-1)/2;
Idx = (n1 >= M1) & (n1 <= M2);

% Rectangular window function
w = ones(1,M);
hw = hlp(Idx).*w;

fig = figure('Position',[100 100 800 200],'color','w');
subplot(1,2,1); hold on
stem(n1,hlp,'Marker','.','Color',[.6 .6 .6]);
stem(n1(Idx),hw,'Marker','.');
title('Ideal Impulse response h_{lp}');
xlabel('Samples');
ylabel('Amplitude');
box on;
xlim([n1(1) n1(end)])

subplot(1,2,2)
% stem(n1(Idx),circshift(hlp(Idx),M2),'Marker','.');
stem(n1(Idx),hw,'Marker','.');
title('Windowed Impulse response h_{w}');
xlabel('Samples');
ylabel('Amplitude');
```



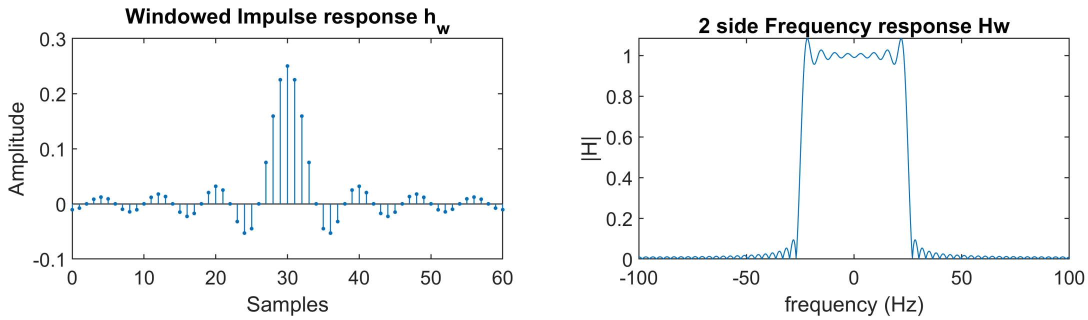
Frequency response

Um filtro obtido a partir da figura anterior é um filtro não causal. Para que o filtro seja causal a resposta ao impulso somente poderá ter valores de zero para $n \geq 0$.

Felizmente, quando usamos $h_w[n]$ os indices serão considerados como valores não negativos

```
fig = figure('Position',[100 100 800 200],'color','w');

% Plot in time domain
subplot(1,2,1);
nw = 0:M-1;
stem(nw,hw,'Marker','.');
title('Windowed Impulse response h_{w}');
xlabel('Samples');
ylabel('Amplitude');
% Frequncy responce
subplot(1,2,2);
Hw = fft(hw,1000);
Hw_2side = fftshift(Hw);
f_2side = linspace(-fs/2,fs/2,length(Hw));
plot(f_2side,abs(Hw_2side))
title('2 side Frequency response Hw');
xlabel('frequency (Hz)');
ylabel('|H|');
```



Plotando a resposta em frequencia

Note no código a seguir que usamos uma função do Matlab/Octave (***'freqz'***) para obter a resposta em frequencia. Neste caso estamos somente usando a metade do espectro da resposta ao impulso, que poderia ter sido obtido usando a fft da resposta ao impulso.

```
% Obtain the frequency response - rect window
[Hw,f] = freqz(hw,1,500,fs);

% Index for truncated impulse response
%n = 0:(M-1);

fig = figure('Position',[100 100 1200 300],'color','w');
subplot(1,3,1)
stem(nw,hw,'Marker','.');
title('Windowed Impulse response h');
xlabel('Samples');
```

```
ylabel('Amplitude');
% Lets find the frequency response

subplot(1,3,2);
plot(f,abs(Hw));
title('Frequency response H');
xlabel('frequency (Hz)');
ylabel('Magnitude Linear');

subplot(1,3,3);
HdB = 20*log10(abs(Hw));
plot(f,HdB);
title('Frequency response H');
xlabel('frequency (Hz)');
ylabel('Magnitude dB');
```

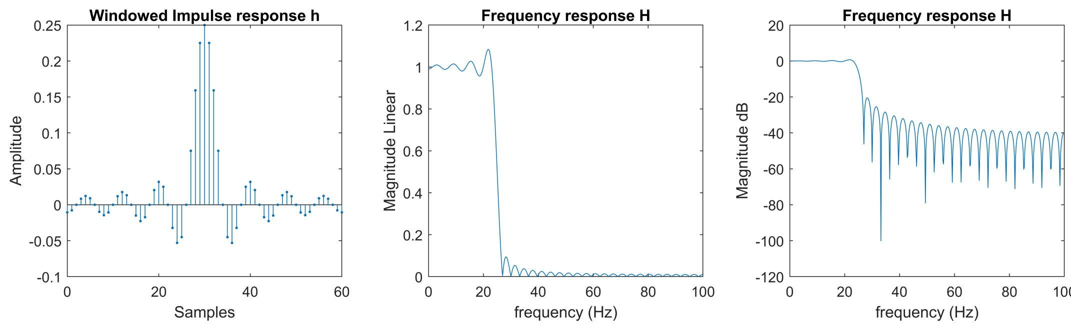


Figura 1. Características do filtro FIR projetado usando janela retangular

Temos projetado nosso primeiro filtro FIR, que bacana!

Vamos usar este filtro para limpar um sinal conhecido, bora!

Usando nosso primer filtro FIR para o sinal de respiração

```
MafFile = matfile('Respiration_Downsampled.mat');
x = MafFile.x;
NSamples = length(x);
t = (0:NSamples-1)/fs;

x1 = filter(hw,1,x);

clf
subplot(1,2,1)
plot(t,x);
title('Raw signal');
xlabel('time (s)');
ylabel('Amplitude');
ylim([-0.6 0.6])

subplot(1,2,2)
plot(t,x1)
title('Low-pass filtered signal');
xlabel('time (s)');
ylabel('Amplitude');
```

```
ylim([-0.6 0.6])
```

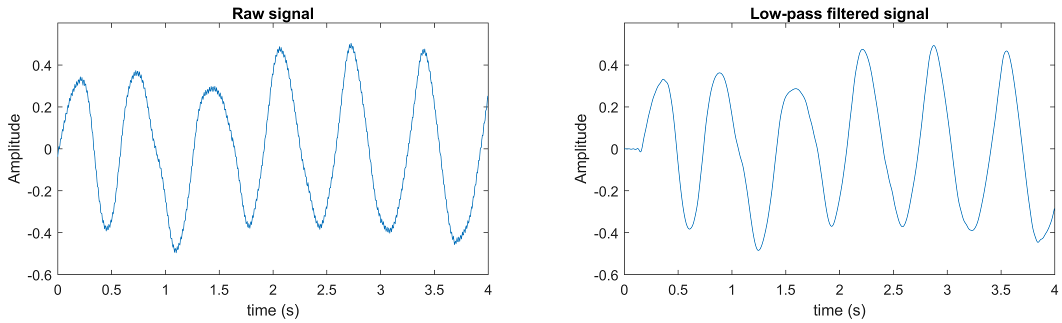


Figura 2. Exemplo de uso do filtro FIR, projetado usando janela retangular

Melhorando a janela (window) para obter uma melhor atuação no domínio frequencia

Resulta importante relembrar os passos que seguimos para projetar nosso primer filtro FIR (é dizer para obter os coeficientes do filtro que permitiram obter uma resposta em frequencia desejada).

1. Obter a resposta ao impulso do filtro passa baixas ideal: $h_{\mathrm{lp}}[n]$
2. Multiplicamos por uma função de janela $\omega[n]$ (rectangular) para obter uma resposta ao impulso de duração finita: $h[n]$; para ver a resposta em frequencia $H[k]$ poderiamos calcular a TFD de $h[n]$
3. Finalmente, poderiamos usar os coeficientes do filtro FIR (os mesmos valores de $h[n]$) para implementar a filtragem.

Um dos conceitos que tratamos ao longo do curso é que uma convolução no domínio tempo dos sinais $h_{\mathrm{lp}}[n]$ e $\omega[n]$ resulta numa multiplicação dos sinas $H_{\mathrm{lp}}[k]$ e $W[k]$ (sendo $H_{\mathrm{lp}}[k]$ e $W[k]$ as TFD de $h_{\mathrm{lp}}[n]$ e $\omega[n]$).

$$h_{\mathrm{lp}}[n] * \omega[n] \longleftrightarrow H_{\mathrm{lp}}[k] \cdot W[k]$$

Nota: Lembrar que para que a convolução linear possa ser relacionada com a TFD, temos que garantir que os sinais sejam completados com zeros antes de calcular a TFD, de modo que ambos os sinais tenham duração $m = N_1 + N_2 - 1$. Como não implementaremos a fltragem desde o dom. frequencia esta nota é simplesmente um comentario para rembrar do conceito. Consideramos a seguir que o requisito será cumprido no caso.

Também é verdadeiro:

$$h_{\mathrm{lp}}[n] \cdot \omega[n] \longleftrightarrow H_{\mathrm{lp}}[k] * W[k]$$

Sabemos então que a resposta em frequencia $H[k]$ poderá ser obtida da expressão:

$$H[k] \; = \; H_{\mathrm{lp}}[k] * W[k]$$

Como na obtenção do filtro usamos uma janela rectangular, é claro qua a respostam em frequencia estará influenciada pela forma de $W[k]$ (a TFD da janela $\omega[n]$).

O uso de uma janela rectangular implica que $W[k]$ tenha lóbulos secundários, devido a discontinuaidade inherente a $\omega[n]$. Estes lóbulos terão um efeito não desejado na resposta em frequencia do filtro, como pode ser visualizado nas figuras anteriores (oscilações na banda de passo e de rejeição, conhecido como fenómeno de Gibs [2]).

Usando um truncamento menos abrupto de $h_{\mathrm{lp}}[n]$, é dizer usando outras janelas em lugar da retangular, poderemos regular melhor a resposta em frequencia. A seguir ilustramos outros 2 tipos de janelas que podem ser usadas em lugar da retangular, estas são as janelas (ver [1], seção 5.3):

- Chebyshev
- Kaiser

nota: Para maiores detalhes referente a diferentes tipos de janelas, consultar as referências recomendadas.

```
% Chebyshev window
r = 80; % Controls the dBs below the mainlobe magnitude.
w1 = chebwin(M,r)'; % transpose for obtain a row vector

% Kaiser window
beta = 0.1102*(r - 8.7);
w2 = kaiser(M,beta)'; % transpose for obtain a row vector

% Obtain the truncated impulse response - Chebyshev window
h1 = hlp(Idx).*w1;
[H1,f] = freqz(h1,1,500,fs);

% Obtain the truncated impulse response - Kaiser window
h2 = hlp(Idx).*w2;
[H2,f] = freqz(h2,1,500,fs);
```

Plotando os resultados para as diferentes janelas

```
clf;

% Plot windows in time domain
subplot(1,3,1);hold on
l = plot(nw,w,'LineWidth',1,'Color',[.4 .4 .4]);
l1 = plot(nw,w1,'LineWidth',1);
l2 = plot(nw,w2,'LineWidth',1,'Color','b');
title('Windows Functions');
xlabel('Samples');
ylabel('Amplitude');
box on
axis([-5 65 0 1.1])

% Plot frequency response for windows
H1dB = 20*log10(abs(H1));
H2dB = 20*log10(abs(H2));

subplot(1,3,2); hold on
l = plot(f,abs(Hw),'LineWidth',1,'Color',[.4 .4 .4]);
l1 = plot(f,abs(H1),'LineWidth',1);
l2 = plot(f,abs(H2),'LineWidth',1,'Color','b');
title('Frequency response H');
xlabel('frequency (Hz)');
```

```
ylabel('Magnitude linear');
box on
lg = legend([l l1 l2],{'Rect','Cheb','Kaiser'},'Location','best');

subplot(1,3,3); hold on
l = plot(f,HdB,'LineWidth',1,'Color',[.4 .4 .4]);
l1 = plot(f,H1dB,'LineWidth',1);
l2 = plot(f,H2dB,'LineWidth',1,'Color','b');
title('Frequency response H');
xlabel('frequency (Hz)');
ylabel('Magnitude dB');
box on
ylim([-150 10])
```

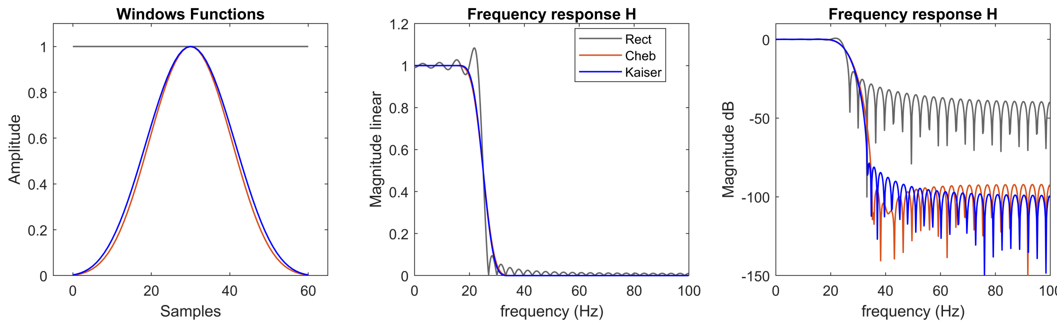


Figura 3. Diferentes tipos de Janelas e suas respectivas respostas em frequencias

Podemos observar que as janelas de Kaiser e Chebyshev diminuem a amplitude dos lóbulos secundários. A continuação iremos comparar a filtragem quando usadas a janelas de Kaiser e rectangular.

Usemos o filtro obtido usando a Janela de Kaiser vs Janela Rectangular

```
% Create simulated signal
TSim = 1;
NSamples = TSim*fs;
t = (0:NSamples-1)/fs;
f1 = 10;
f2 = 60;
x1 = 0.1*sin(2*pi*f1*t);
x2 = sin(2*pi*f2*t);
x = x1 + x2;
[Y,f] = ComputeSpectrum(x,NSamples,fs);

y1 = filter(hw,1,x);
[Y1,f] = ComputeSpectrum(y1,NSamples,fs);
y2 = filter(h2,1,x);
[Y2,f] = ComputeSpectrum(y2,NSamples,fs);


fig = figure('Position',[10 10 1200 300],'color','w');
subplot(1,3,1);
plot(f,(Y));
title('Original Signal');
xlabel('frequency (Hz)');
ylabel('Amplitude');
```

```
subplot(1,3,2);
plot(f,(Y1));
% plot(t,x1+1,'LineStyle',':')
title('FIR: Rect. Window');
xlabel('frequency (Hz)');
ylabel('Amplitude');
% ylim([-0.6 0.6])

subplot(1,3,3); %hold on
% plot(t,x1)
plot(f,(Y2))
title('LP-FIR: Kaiser. Window');
xlabel('frequency (Hz)');
ylabel('Amplitude');
% ylim([-1.5 1.5])
box on
```

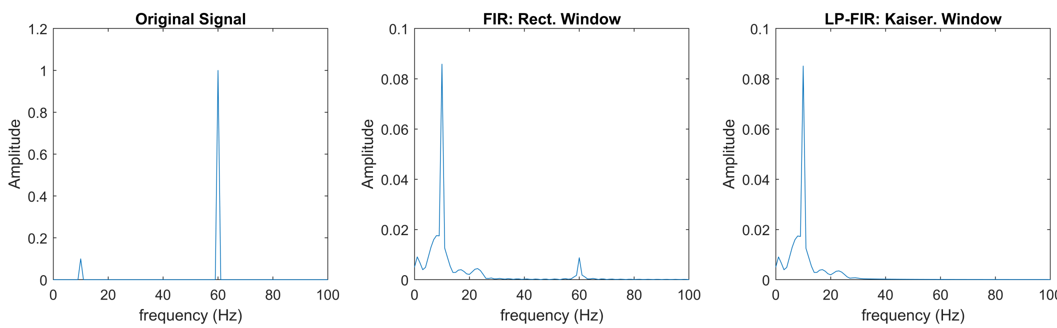


Figura 4. Exemplo de uso do filtro FIR, projetado usando janela de Kaiser

Função Matlab/Octave para projeto FIR usando Janelas

Hoje aprendimos como criar um filtro FIR passa-baixas (*'from scratch'*), para isso usamos o método de janelas. No Matlab/Octave existe uma função que permite realizar essa tarefa, como veremos a seguir.

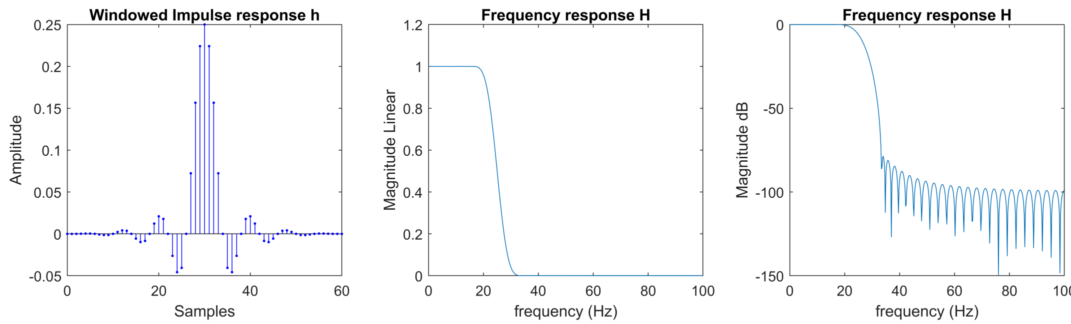
```
wn = fc/(fs/2);
h2a = fir1(60,wn,kaiser(M,beta));
[H2a,f] = freqz(h2a,1,500,fs);

fig = figure('Position',[100 100 1200 300],'color','w');
subplot(1,3,1);hold on
stem(nw,h2a,'Marker','.');
stem(nw,h2a,'Marker','.','Color','b');
box on

title('Windowed Impulse response h');
xlabel('Samples');
ylabel('Amplitude');

% Lets find the frequency response
subplot(1,3,2);
plot(f,abs(H2a));
title('Frequency response H');
xlabel('frequency (Hz)');
ylabel('Magnitude Linear');
```

```
subplot(1,3,3);
HdB = 20*log10(abs(H2a));
plot(f,HdB);
title('Frequency response H');
xlabel('frequency (Hz)');
ylabel('Magnitude dB');
```




```
rms(h2 - h2a)
```

```
ans = 2.9499e-06
```

Referencias

1. Richard G. Lyons - Understanding Digital Signal Processing-Prentice Hall (2010), cap. 5
2. OPPENHEIM, Alan V; SCHAFER, Ronald W; VIEIRA, Daniel. Processamento em tempo discreto de sinais. 3. ed. São Paulo, SP: Pearson, 2012. 665 p. ISBN: 9788581431024. Cap. 7 (pág. 316).
3. Aula TFTD, slide 11