# Projeto de
# Filtros Digitais FIR

## PDS

**Prof. Cláudio A. Fleury**
**2013**

# Conteúdo

- Filtros FIR – *Finite Impulse Response*

- Métodos de Projeto
  - Transformada de Fourier (Janela Retangular)
  - "Janelamento" (outras Janelas)
  - Amostragem em Frequência
  - Parks-McClellan (Ótimo / Troca de Remez / Equiripple)

- Resumo

- Exercícios

# Filtros FIR

# Filtros FIR

- Filtros FIR com variação de fase linear com a frequência
  - **Transmissão sem distorção**: Ganho constante ($A$) e fase constante ($k$) em qq. freq.

$$y[n] = A.x[n-k]$$

$$Y(\Omega) = A.X(\Omega)e^{-j\Omega k}$$

$$H(\Omega) = \frac{Y(\Omega)}{X(\Omega)} = Ae^{-j\Omega k}$$

$$|H(\Omega)| = A$$

$$\phi(\Omega) = \angle H(\Omega) = -k.\Omega$$ ⬅ variação linear da fase com a frequência

- Se a **fase da Resposta em Frequência** do filtro FIR, $\Phi(\Omega)$, varia linearmente com a frequência então todas as componentes frequenciais do **sinal de entrada** propagarão à mesma velocidade no filtro → **não haverá distorção de fase** do sinal de entrada

- O atraso de grupo não é constante nos filtros IIR: $\tau = -\frac{d\phi(\Omega)}{d\Omega} \neq cte$

- O atraso de grupo é constante nos filtros FIR: $\tau = -\frac{d\phi(\Omega)}{d\Omega} = -\frac{d(-k.\Omega)}{d\Omega} = k \quad \text{(cte)}$

# Técnicas de Projeto de Filtros FIR

**Transformada de Fourier** (didática)

**Janelamento**

**Amostragem em Frequência**

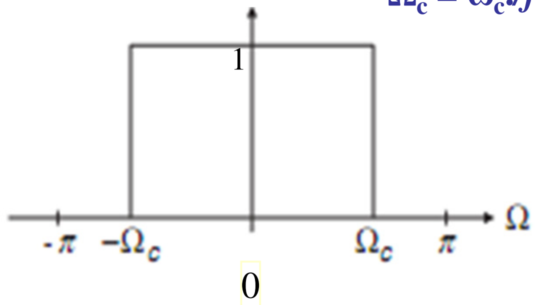
**Projeto Ótimo**

# Método da Transformada de Fourier

- **Filtro FIR, Passa Baixa, Ideal, com frequência de corte $\Omega_c$**
  - Análise da *Resposta ao Impulso* do filtro passa baixa ideal (FPBI):

Resp. em Frequência do FPBI

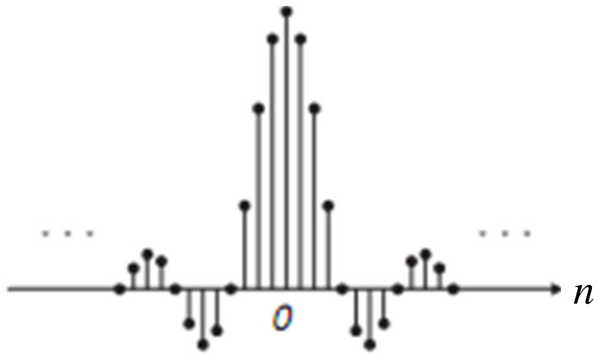
$|H_i[e^{j\Omega}]|$




$\Omega_c = \omega_c/f_s$

$F^{-1}\{\cdot\}$

$h_i[n]$




$h_i[n] \neq 0$ para $n < 0$

**SLIT não causal**

$h_i[n]$ é simétrica em torno de $n = 0$: $h_i[n] = h_i[-n]$ e existe para $-\infty < n < \infty$

Variação linear de fase

# Método da Transformada de Fourier

* Filtro FIR, Passa-Baixas, Ideal, com frequência de corte $\Omega_c$
  – A Função de Transferência do filtro pode ser obtida fazendo $e^{j\Omega} = z$:

$$H(z) = \sum_{n=-\infty}^{\infty} h[n] z^{-n}$$

$$= \cdots + h[-2]z^{2} + h[-1]z + h[0] + h[1]z^{-1} + h[2]z^{-2} + \cdots$$

que é uma soma de infinitos termos, ou seja, uma função de transferência de um **SLITD não causal**!!!

Truncamento

  – Para tornar o projeto **realizável**, trunca-se $h[n]$ com *alguns* pontos, ou seja, $2M+1^{*}$ **coeficientes <u>dominantes</u>** ($h[n]$ diminui quando $n$ aumenta):

$$H(z) = h[-M]z^{M} + \cdots + h[-1]z + h[0] + h[1]z^{-1} + \cdots + h[M]z^{-M}$$

  – Mesmo com uma quantidade finita de termos, a função de transferência ainda representa um **SLITD não causal**, pois a saída do filtro depende de amostras futuras da entrada (termos em $z$ com potência positiva)

* Para garantir a simetria da resp. ao impulso, o truncamento deve acontecer com a mesma de quantidade de amostras para o lado esquerdo e direito.

# Método da Transformada de Fourier

- **Filtro FIR, Passa-Baixas, Ideal, com frequência de corte $\Omega_c$**

  – Se a Resposta Impulsiva do filtro for atrasada em **M** amostras, então o SLITD resultante será **causal**:

Atraso

$$H(z) = h[-M] + \cdots + h[0]z^{-M} + h[1]z^{-M-1} + \cdots + h[M]z^{-2M}$$

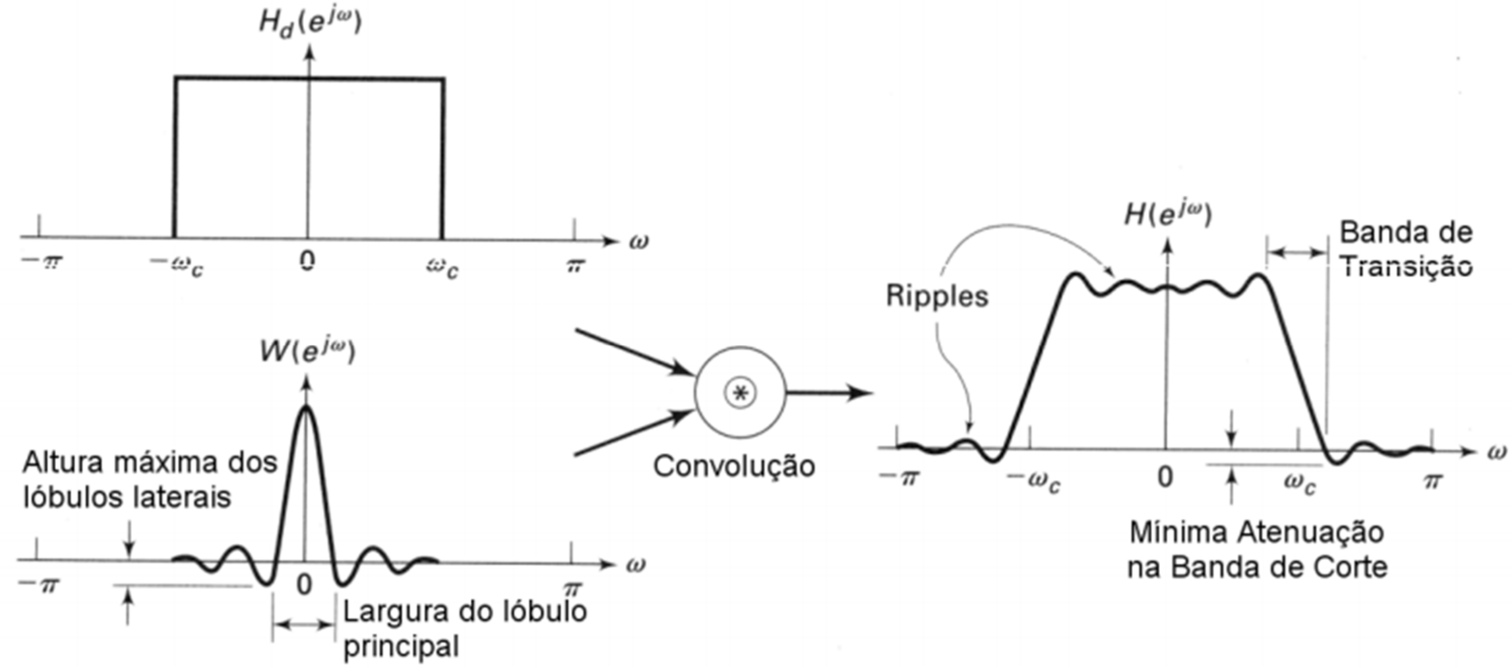
$$H(z) = b_0 + b_1 z^{-1} + \cdots + b_{2M} z^{-2M}$$

$$\text{onde :} \quad b_n = h[n-M], \quad \text{para} \;\; n = 0,1,\ldots,2M$$

$$b_0 = h[-M]; \quad b_1 = h[1-M]; \ldots \; b_M = h[0]; \ldots \; b_{2M} = h[M]$$

# Método da Transformada de Fourier

- **Filtro FIR, Passa-Baixas, Ideal, com frequência de corte $\Omega_c$**




- Características da RF da Janela Retangular de comprimento 2M+1
  - lóbulo principal tem largura proporcional a 1/M, assim como a banda de transição
  - a diferença entre os picos dos lóbulos principal e secundário será a atenuação mínima na banda de bloqueio

# Método da Transformada de Fourier

Tabela – **Filtros FIR** com **fase linear** e freq. de corte $\Omega_c$ (ou $\Omega_H$ e $\Omega_L$)

| Tipo do Filtro | Resposta ao Impulso Truncada do Filtro Ideal (filtro FIR não causal) |
|---|---|
| Passa-baixa | $\hat{h}[n] = \begin{cases} \Omega_c/\pi, & \text{p/} \;\; n = 0 \\ sen(\Omega_c n)/(\pi n), & \text{p/} \; -M \le n \le M \;\text{ e }\; n \ne 0 \end{cases}$ |
| Passa-alta | $\hat{h}[n] = \begin{cases} (\pi - \Omega_c)/\pi, & \text{p/} \;\; n = 0 \\ -sen(\Omega_c n)/(\pi n), & \text{p/} \; -M \le n \le M \;\text{ e }\; n \ne 0 \end{cases}$ |
| Passa-faixa | $\hat{h}[n] = \begin{cases} (\Omega_H - \Omega_L)/\pi, & \text{p/} \;\; n = 0 \\ [sen(\Omega_H n) - sen(\Omega_L n)]/(\pi n), & \text{p/} \; -M \le n \le M \;\text{ e }\; n \ne 0 \end{cases}$ |
| Rejeita-faixa | $\hat{h}[n] = \begin{cases} (\pi - \Omega_H + \Omega_L)/\pi, & \text{p/} \;\; n = 0 \\ [sen(\Omega_L n) - sen(\Omega_H n)]/(\pi n), & \text{p/} \; -M \le n \le M \;\text{ e }\; n \ne 0 \end{cases}$ |

Resp. ao Impulso do **filtro FIR causal**: $h[n] = \hat{h}[n - M]$

Função de transferência: $H(z) = b_0 + b_1 z^{-1} + \cdots + b_{2M} z^{-2M}, \quad b_n = \hat{h}[n - M]$

# Método da Transformada de Fourier

- Exemplo 2

Projete um FPB FIR de comprimento 3 (3-tap = 2ª ordem) com frequência de corte em 800 Hz para sinais amostrados a $f_s$ = 8 kHz, usando o Mét. da Transf. de Fourier

*i.* Frequência de corte digital normalizada: $\Omega_c = \omega_c/f_s = 2\pi f_c/f_s = 0{,}2\pi$ rad/amostra

*ii.* Da tabela de **Tipos de Filtros FIR**, temos para FPB: $h[0] = \Omega_c / \pi = 0{,}2$

e $\hat{h}[n] = sen(n.\Omega_c)/(\pi.n) = sen(0{,}2\pi.n)/(\pi.n)$ p/ $n \neq 0$ $\rightarrow \hat{h}[1] = \hat{h}[-1] = 0{,}1871$

*iii.* Atrasando essa Resposta Impulsiva de $M = 1$ amostra (filtro causal):

$b_0 = \hat{h}[0\text{-}1] = \hat{h}[-1] = 0{,}1871$; $b_1 = \hat{h}[1\text{-}1] = \hat{h}[0] = 0{,}2$; $b_2 = \hat{h}[2\text{-}1] = \hat{h}[1] = 0{,}1871$

Função de Transferência: $H(z) = 0{,}1871 + 0{,}2z^{-1} + 0{,}1871z^{-2}$

Resposta em Frequência: $H(e^{j\Omega}) = 0{,}1871 + 0{,}2e^{-j\Omega} + 0{,}1871e^{-j2\Omega}$

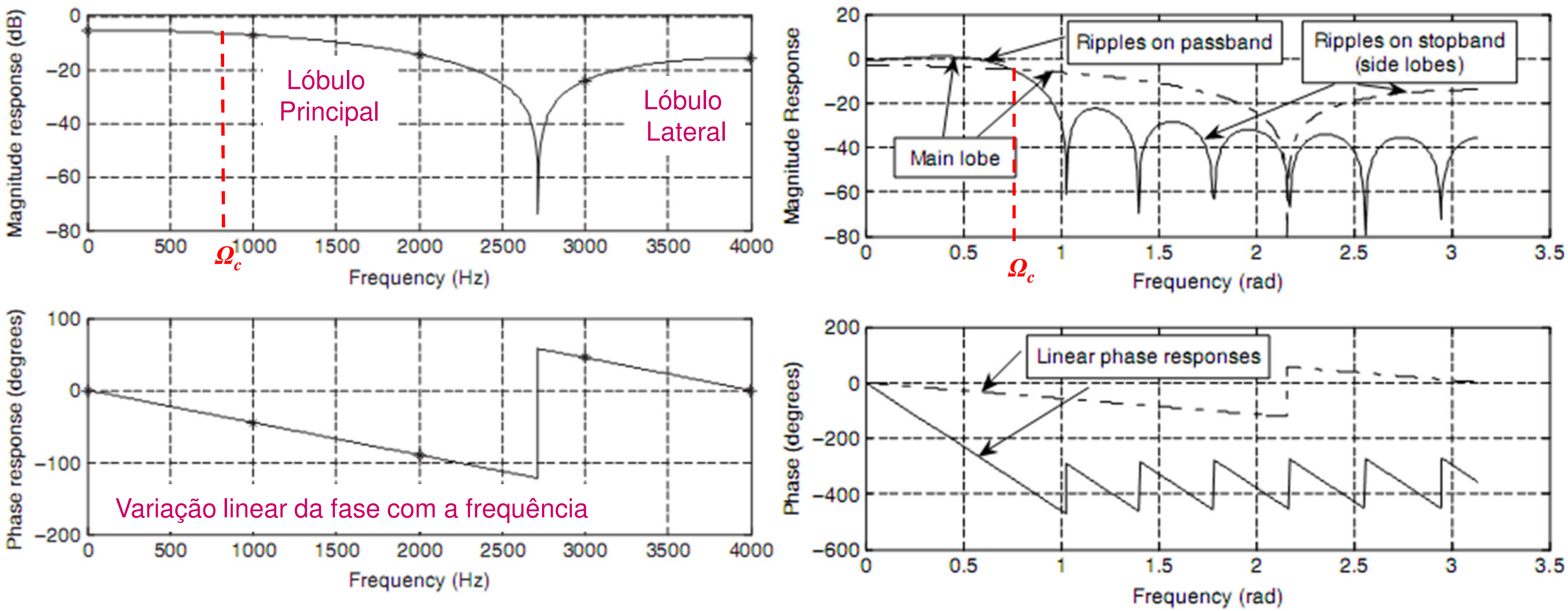
Equação de Diferenças: $y[n] = 0{,}1871\, x[n] + 0{,}2\, x[n-1] + 0{,}1871\, x[n-2]$

Devido à simetria de $h[n]$, o filtro FIR projetado tem Resposta em Freqüência com fase variando linearmente com a freqüência $\Omega$ $\rightarrow$ **sem distorção de fase**

# Método da Transformada de Fourier

- Exemplo 2 (continuação)
  FPB FIR com frequência de corte em 800 Hz para sinais amostrados a 8 kHz
  Resposta em Frequência para M=1: $H(e^{j\Omega}) = 0{,}1871 + 0{,}2e^{-j\Omega} + 0{,}1871e^{-j2\Omega}$

**$M$ = 1, ordem $N$ = 2** **$M$ = 8, ordem $N$ = 16**



# Método da Transformada de Fourier

- Exemplo 2 (continuação)

Considerações Finais:

1. As oscilações na *Banda Passante* (lóbulo principal) e na *Banda de Rejeição* (lóbulos laterais) da magnitude da resposta em frequência constituem o efeito **Gibbs**. O comportamento oscilatório do *Efeito Gibbs* aparece devido ao **truncamento abrupto** da Resposta Impulsiva Infinita (**janela Retangular**).

2. Uma quantidade maior de coeficientes (ordem maior do filtro) produz característica de decaimento (**roll-off**) mais brusca na banda de transição, mas aumenta o atraso da saída e a complexidade computacional na etapa de implementação do filtro.

3. A fase da Resposta em Frequência é linear na Banda Passante, ou seja, todas as componentes frequenciais do sinal de entrada dentro da Banda Passante estarão sujeitas à mesma quantidade de atraso na saída do filtro. Isto é um requisito para filtragem de sinais sensíveis a distorções de fase (ordem ímpar e coeficientes simétricos/anti-simétricos em torno do coeficiente central).

# Método da Transformada de Fourier

- Exemplo 3

Projete um **FPF** FIR de comprimento 5, freq. de corte inferior em 2,0 kHz e superior em 2,4 kHz, para sinais amostrados a 8,0 kHz, usando o Método da Transf. de Fourier

*i.* Frequências de corte normalizadas: $\Omega_L = 2\pi . f_L / f_s = 0{,}5\pi$ e $\Omega_H = 0{,}6\pi$ (rad/amostra)

*ii.* Da tabela de **Tipos de Filtros FIR**, temos para FPF: $h[0] = (\Omega_H - \Omega_L) / \pi = 0{,}1$

e $\hat{h}[n] = (\text{sen}(n.\Omega_H) - \text{sen}(n.\Omega_L)) / (\pi.n)$ para $n \neq 0$ ➔ $\hat{h}[1] = -0{,}015$ e $\hat{h}[2] = -0{,}093$

Pela propriedade da simetria da Resposta Impulsiva: ➔ $\hat{h}[-1] = -0{,}015$ e $\hat{h}[-2] = -0{,}093$

*iii.* Atrasando a Resposta Impulsiva de $M = 2$ amostras (filtro causal):

$b_0 = b_4 = \hat{h}[0\text{-}2] = \hat{h}[-2] = -0{,}093$; $b_1 = b_3 = \hat{h}[1\text{-}2] = \hat{h}[-1] = -0{,}015$; $b_2 = \hat{h}[2\text{-}2] = \hat{h}[0] = 0{,}1$

Função de Transferência: $H(z) = -0{,}093 - 0{,}015\, z^{-1} + 0{,}1 z^{-2} - 0{,}015\, z^{-3} - 0{,}093 z^{-4}$

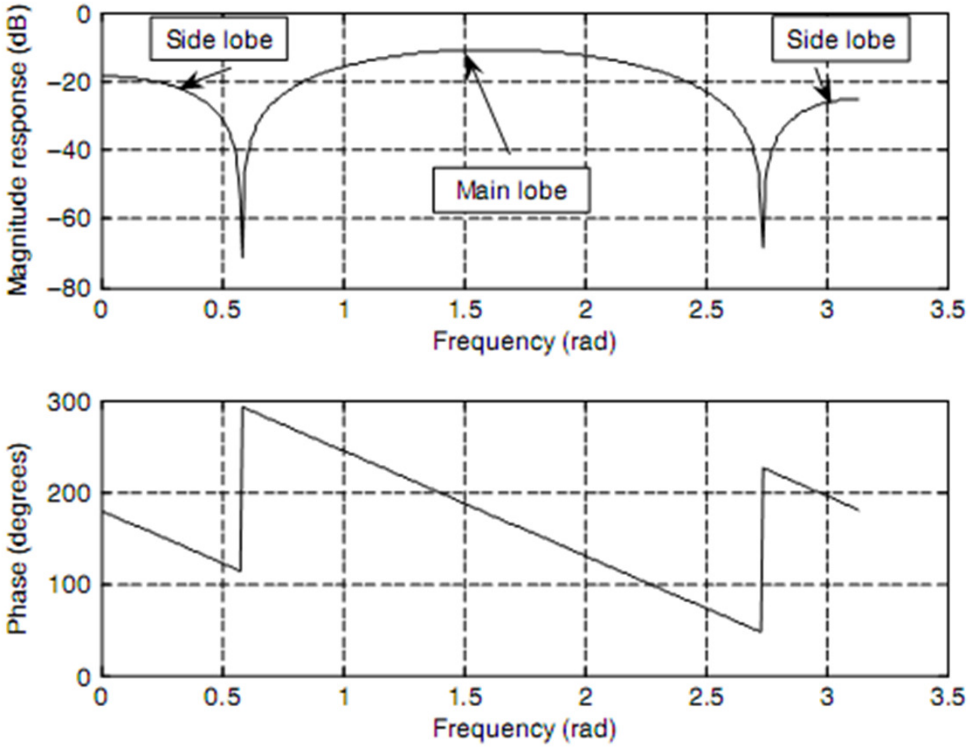
A magnitude da Resp. em Frequência (dB): $|H(e^{j\Omega})|_{dB} = 20.\log_{10} |H(e^{j\Omega})|$

# Método da Transformada de Fourier

- Exemplo 3 (continuação)

**FPF** FIR de **ordem 4** com frequência de corte inferior em 2,0 kHz, e superior em 2,4 kHz para sinais amostrados a 8,0 kHz

Resposta em Frequência: magnitude e fase




```
[hz, w] = freqz([−0.09355 −0.015580.1 −0.01558 −0.09355], [1], 512);
phi= 180*unwrap(angle(hz))/pi;
subplot(2,1,1), plot(w,20*log10(abs(hz))),grid;
xlabel('Frequency (radians)');
ylabel('Magnitude Response (dB)')
subplot(2,1,2), plot(w, phi);grid;
xlabel('Frequency (radians)');
ylabel('Phase (degrees)');
```

Exemplo

# Método da Transf. de Fourier

– **Código Matlab**

```
% Filtro FIR PB/PA/PF/RF - Transf. de Fourier (janela retangular / efeito Gibbs)
clc,  disp('>>>>>>> Projeto de Filtro FIR <<<<<<<<')
tipo = input('Tipo do filtro (1-PB, 2-PA, 3-PF, 4-RF): ');
C = input('Comprimento do filtro (qtde ímpar): ');
fs = input('Freq. de Amostragem, Fs (Hz): ');                % frequência de amostragem
if tipo == 1 || tipo == 2
    fc = input('Freq. de Corte,      Fc (Hz): ');            % frequência de corte
    wc = 2*pi*fc/fs;                                         % freq. normalizada (rad/amostra)
else
    fc1= input('Freq. de Corte Inferior, Fci (Hz): ');       % frequência de corte inferior
    fc2= input('Freq. de Corte Superior, Fcs (Hz): ');       % frequência de corte superior
    wc1= 2*pi*fc1/fs; wc2= 2*pi*fc2/fs;                      % freq.s normalizadas (rad/amostra)
end
N = C-1; M = N/2;                                            % ordem do filtro e atraso (amostras)
h = zeros(1,C);                                              % resposta ao impulso, h[n], causal
for n = -M:M
    switch tipo
        case 1                                               % passa baixas
            if n == 0
                h(n+M+1) = wc/pi;
            else
                h(n+M+1) = sin(wc*n)/(pi*n);
            end
        case 2                                               % passa altas
            if n == 0
                h(n+M+1) = (pi-wc)/pi;
            else
                h(n+M+1) = -sin(wc*n)/(pi*n);
            end
        case 3                                               % passa faixa
            if n == 0
                h(n+M+1) = (wc2-wc1)/pi;
            else
                h(n+M+1) = (sin(wc2*n)-sin(wc1*n))/(pi*n);
            end
        case 4                                               % rejeita faixa
            if n == 0
                h(n+M+1) = ((pi - wc2) - wc1)/pi;
            else
                h(n+M+1) = (sin(wc1*n)-sin(wc2*n))/(pi*n);
            end
    end
end
disp(['Coeficientes B´s: ' num2str(h)]), h                   % coeficientes do filtro
freqz(h,1)                                                   % resposta em frequência do filtro
```

Exemplo

# Método da Transf. de Fourier

– Código Python

```
# -*- coding: utf-8 -*-
""" Projeto de Filtro FIR - Método da Transformada de Fourier (jan. retangular, efeito Gibbs) """
from matplotlib.pylab import plot, grid
from numpy import pi, sin, zeros, arange
from scipy.signal import freqz
print '>>>>>>> Projeto de Filtro FIR <<<<<<<<'
tipo = raw_input('Tipo do filtro (1-PB, 2-PA, 3-PF, 4-RF): ')
C = int(raw_input(u'Comprimento do filtro (qtde ímpar): '))
fs = int(raw_input('Freq. de Amostragem, Fs (Hz): '))        # frequência de amostragem
if tipo == '1' or tipo == '2':
    fc = input('Freq. de Corte,      Fc (Hz): ')             # frequência de corte
    wc = 2*pi*fc/fs                                          # freq. normalizada (rad/amostra)
else:
    fc1= int(raw_input('Freq. de Corte Inf., Fci (Hz): '))   # frequência de corte inferior
    fc2= int(raw_input('Freq. de Corte Sup., Fcs (Hz): '))   # frequência de corte superior
    wc1= 2.*pi*fc1/fs; wc2= 2.*pi*fc2/fs                     # freq.s normalizadas (rad/amostra)
N = C-1                                                      # ordem do filtro
M = N/2                                                      # qtde de amostras para atraso (causal)
h = zeros((C))                                               # resposta ao impulso, h[n], causal
for n in arange(-M,M+1):
    if tipo == '1':                                          # passa baixas
        if n == 0:
            h[n+M] = wc/pi
        else:
            h[n+M] = sin(wc*n)/(pi*n)
    elif tipo == '2':                                        # passa altas
        if n == 0:
            h[n+M] = (pi-wc)/pi
        else:
            h[n+M] = -sin(wc*n)/(pi*n)
    elif tipo == '3':                                         # passa faixa
        if n == 0:
            h[n+M] = (wc2-wc1)/pi
        else:
            h[n+M] = (sin(wc2*n)-sin(wc1*n))/(pi*n)
    elif tipo == '4':                                        # rejeita faixa
        if n == 0:
            h[n+M] = ((pi - wc2) - wc1)/pi
        else:
            h[n+M] = (sin(wc1*n)-sin(wc2*n))/(pi*n)
print "Coeficientes B's: ", h                               # coef.s do filtro
w,H = freqz(h,1); plot(w,abs(H)); grid('on')
```

# Método da Janela

- **Objetivo**: diminuir as oscilações do **efeito Gibbs** presentes na banda passante e na banda de rejeição da Resposta em Frequência dos filtros FIR oriundos do **truncamento abrupto** (feito pela janela retangular) da Resposta Impulsiva Infinita do FPB Ideal
- **Função Janela** $w[n]$ deve diminuir gradativamente os coeficientes em ambos lados, dentro da faixa $-M \leq n \leq M$.

$$h_w[n] = h[n].w[n]$$

- Etapas do Método da Janela
  1. Obter os coeficientes do filtro pelo Método da Transformada de Fourier
  2. Multiplicar os 2***M***+1 coeficientes do filtro pela **Função Janela** escolhida
  3. Atrasar o resultado da multiplicação por ***M*** amostras (filtro de ordem 2***M***), gerando a resposta impulsiva "janelada" do filtro causal

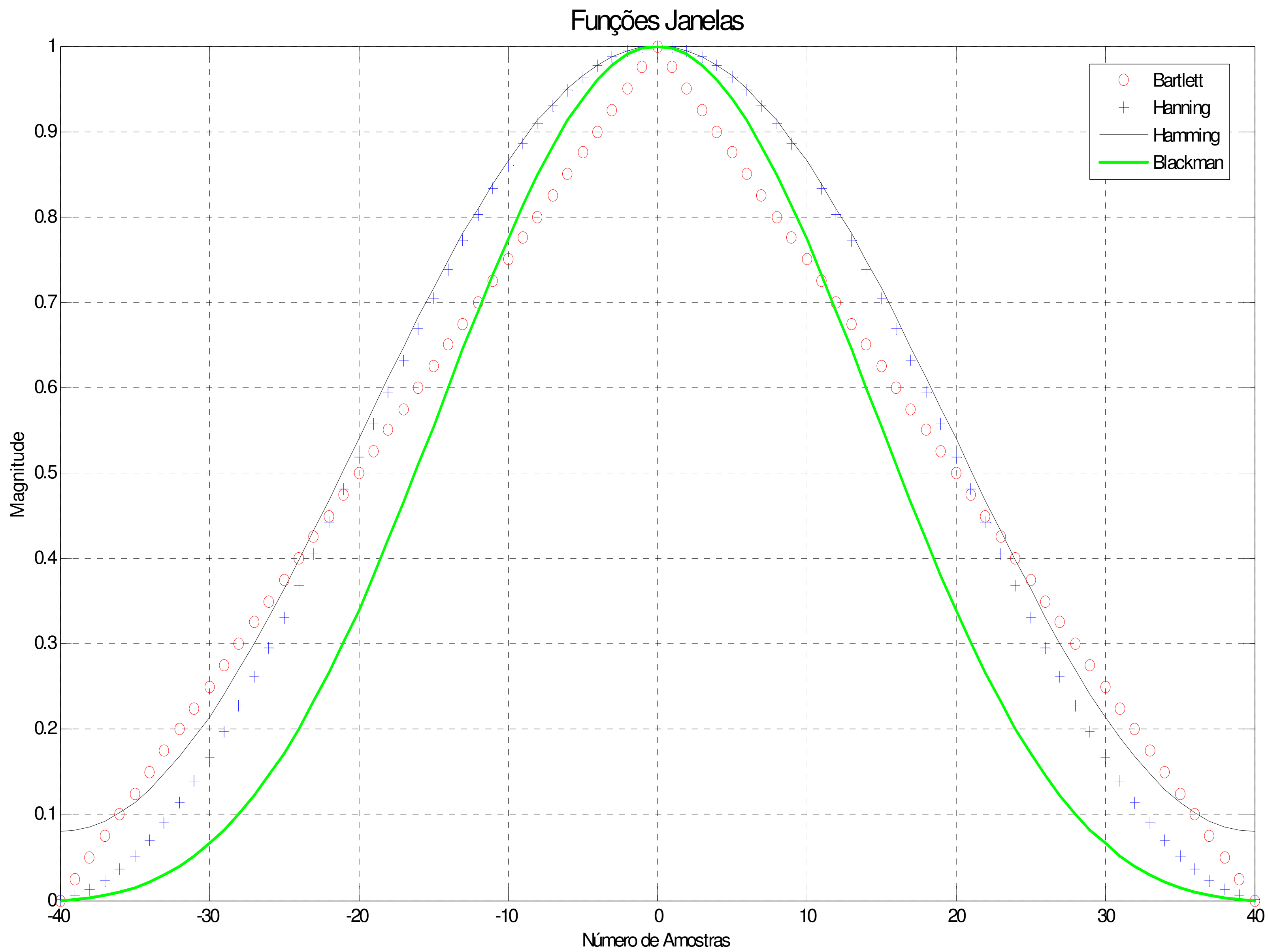
# Método da Janela

- Janelas usuais:

$$-M \leq n \leq M$$

| Janela | Definição |
|---|---|
| Retangular | $w[n] = 1$ |
| Bartlett (triangular) | $w[n] = 1 - \dfrac{\lvert n \rvert}{M}$ |
| Hanning | $w[n] = 0{,}5 + 0{,}5\cos\left(\dfrac{n\pi}{M}\right)$ |
| **Hamming** | $w[n] = 0{,}54 + 0{,}46\cos\left(\dfrac{n\pi}{M}\right)$ |
| Blackman | $w[n] = 0{,}42 + 0{,}5\cos\left(\dfrac{n\pi}{M}\right) + 0{,}08\cos\left(\dfrac{2n\pi}{M}\right)$ |

# Método da Janela



# Método da Janela

- Características das janelas:

Estimativa da Ordem do Filtro FIR para cada tipo de janela

$$\theta = \pi / M$$

| Janela | Definição $w[n]$ | Compr. da Janela, $C$ | Ripple na B. Passante (dB) | Atenuação na B. Bloq. (dB) |
|---|---|---|---|---|
| Retangular | $w[n] = 1$ | $C = 0{,}9/\Delta f$ | 0,7416 | 21 |
| Bartlett | $w[n] = 1 - \frac{\lvert n \rvert}{M}$ | $C = 3{,}05/\Delta f$ | 0,4237 | 25 |
| Hanning | $w[n] = 0{,}5 + 0{,}5\cos(\theta.n)$ | $C = 3{,}1/\Delta f$ | 0,0546 | 44 |
| **Hamming** | $w[n] = 0{,}54 + 0{,}46\cos(\theta.n)$ | $C = 3{,}3/\Delta f$ | 0,0194 | 53 |
| Blackman | $w[n] = 0{,}42 + 0{,}5\cos(\theta.n) + 0{,}08\cos(2\theta.n)$ | $C = 5{,}5/\Delta f$ | 0,0017 | 74 |

Largura normalizada da Banda de Transição : $\Delta f = |f_{pass} - f_{stop}| / f_s$

# Método da Janela

```
>>> h = [0.1514 0.1871 0.2 0.1871 0.1514]
>>> (0.54+0.46*np.cos((-2:2)*pi/2)) * h
  0.0121   0.1010   0.2000   0.1010   0.0121
```

```
>> h = [0.1514 0.1871 0.2 0.1871 0.1514]
>> (0.54+0.46*cos((-2:2)*pi/2)) .* h
ans =
   0.0121   0.1010   0.2000   0.1010   0.0121
```

- Exemplo 4

Projete um FPB FIR de ordem 4 com frequência de corte em 800 Hz para sinais amostrados a 8 kHz, usando a **Função Janela de Hamming**

*i.* Frequência de corte digital normalizada: $\Omega_c = \omega_c/f_s = 2\pi\, f_c / f_s = 0{,}2\pi$ rad/amostra

*ii.* Da tabela de Tipos de Filtros (slide 19) e do Exemplo 2 (slide 20), temos:

$h[0] = \Omega_c / \pi = 0{,}2$ e $h[n] = \mathrm{sen}(n.\Omega_c) / (\pi.n) = \mathrm{sen}(0{,}2\pi.\mathrm{n}) / (\pi.n)$ para $n \neq 0$

$h[1] = h[-1] = 0{,}1871$ e $h[2] = h[-2] = 0{,}1514$

*iii.* Multiplicandos por Hamming: $h[0] = 0{,}2$; $h[-1] = h[1] = 0{,}101$ e $h[-2] = h[2] = 0{,}0121$

*iv.* Atrasando a Resposta Impulsiva de M=2 amostras (filtro causal):

$b_0 = 0{,}0121$; $b_1 = 0{,}1010$; $b_2 = 0{,}2$; $b_3 = 0{,}1010$; $b_4 = 0{,}0121$

Função de Transferência: $H(z) = 0{,}0121 + 0{,}101z^{-1} + 0{,}2z^{-2} + 0{,}101z^{-3} + 0{,}0121z^{-4}$

Resposta em Frequência: $H(e^{j\Omega}) = 0{,}0121 + 0{,}101e^{-j\Omega} + 0{,}2e^{-j2\Omega} + 0{,}101e^{-j3\Omega} + 0{,}0121e^{-j4\Omega}$

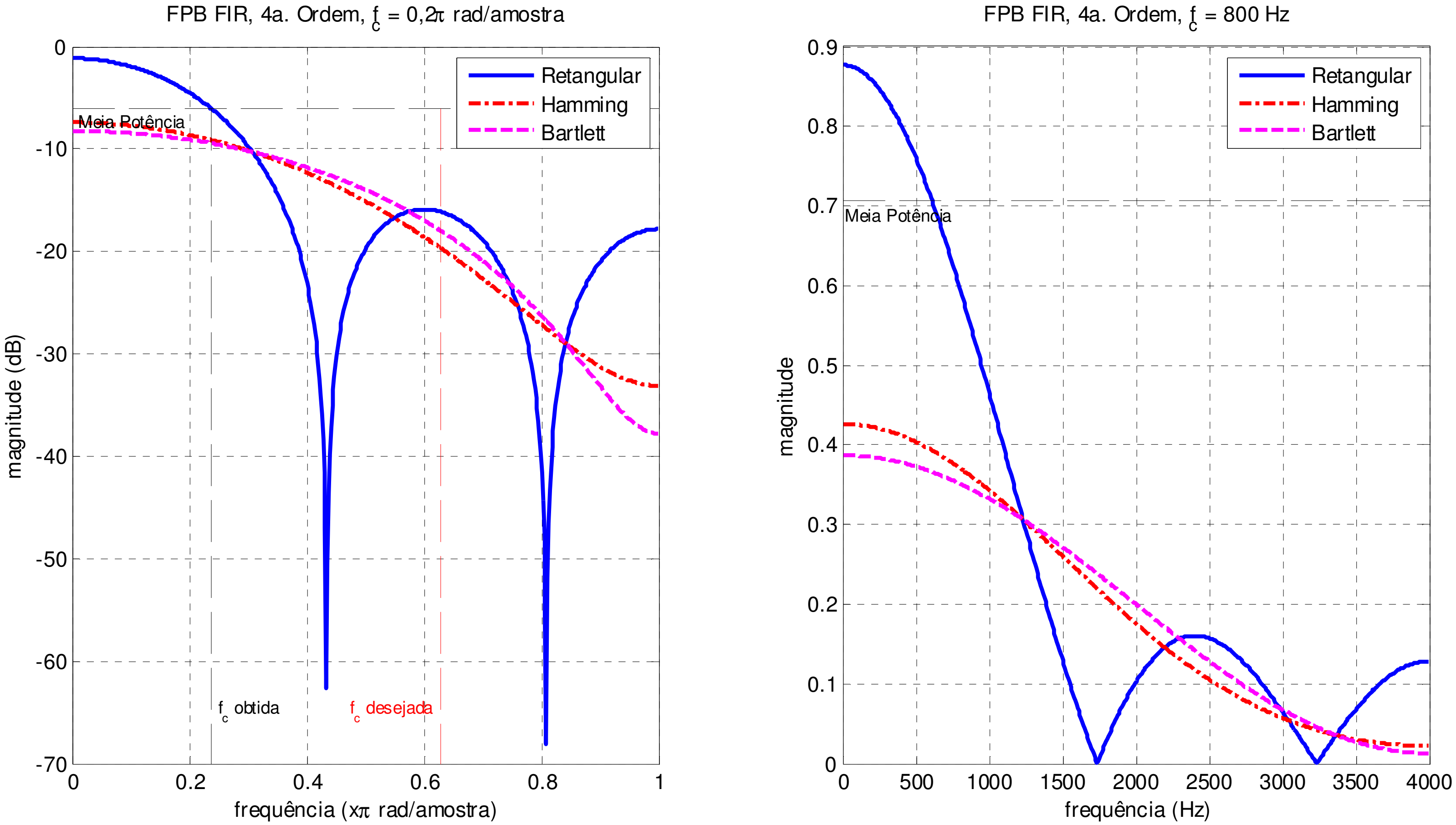
Equação de Diferenças: $y[n] = 0{,}0121x[n]+0{,}101x[n-1]+0{,}2x[n-2]+0{,}101x[n-3]+0{,}0121\, x[n-4]$

A aplicação da janela muda as características da Resposta em Freqüência do filtro FIR: a **largura do lóbulo principal** e a **atenuação dos lóbulos laterais** aumentam

# Método da Janela

- Exemplo 4 (continuação): FPB FIR de compr. 5 (5-tap ou ordem 4) com $f_c$ em 800 Hz para sinais amostrados a 8 kHz, usando a função janela de Hamming.

## Magnitude da Resposta em Frequência



# Método da Janela

- Exemplo 4: Script Python

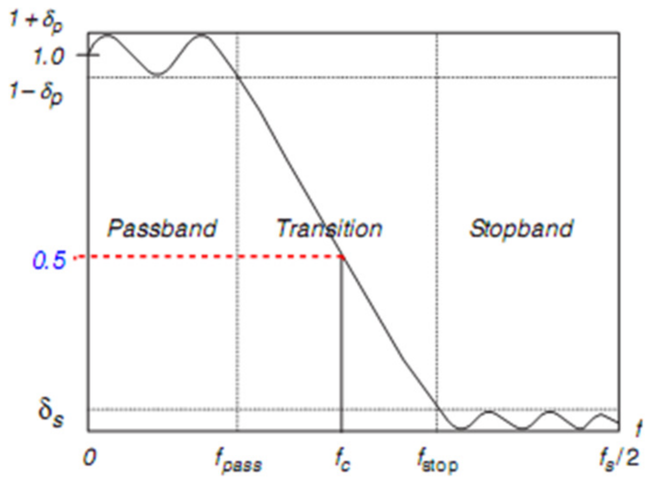
```
# -*- coding: utf-8 -*- Prof. Cláudio - Nov/2014
""" FPB FIR de compr. 5, fc = 800 Hz, fs = 8 kHz, usando a função janela de Hamming."""
from scipy.signal import firwin, freqz
from matplotlib.pylab import plot,subplot,xlabel,ylabel,grid,title,legend,ylim,text
from matplotlib.mlab import find
from numpy import pi, log10, abs, sqrt
fs = 8000; fc = 800        # freq. de amostragem e de corte do FPB FIR (Hz)
Wc = 2*pi*fc/fs            # freq. de corte normalizada (rad/amostra)
N = 4                      # ordem do FPB FIR de fase linear
b1 = firwin(N,Wc/pi,window='boxcar');   w,H1 = freqz(b1,1)
b2 = firwin(N,Wc/pi,window='hamming');  w,H2 = freqz(b2,1)
b3 = firwin(N,Wc/pi,window='bartlett'); w,H3 = freqz(b3,1)
# traçado das curvas de Magnitude da Resp. em Freq. dos Filtros PB FIR (janelas)
subplot(1,2,1);  plot(w/pi,20*log10(abs(H1)),linewidth=2), grid('on')
title('FPB FIR, 4a. Ordem, $f_c = 0,2\pi$ rad/amostra'); ylim1 = ylim()
xlabel(u'frequência (x $\pi$ rad/amostra)'); ylabel('magnitude (dB)')
plot(w/pi,20*log10(abs(H2)),'r-.',linewidth=2)
plot(w/pi,20*log10(abs(H3)),'m--',linewidth=2)
legend(('Retangular','Hamming','Bartlett')); plot([0,1],[-6,-6], 'k--')
text(0.01,-7.5,u'Meia Potência',fontsize=8);         # meia potência
i = find(20*log10(abs(H1)) <= -6.0206)
plot([w[i[0]]/pi,w[i[0]]/pi],[-6.02,ylim1[0]],'k--') # freq. de corte do filtro projetado
plot([Wc,Wc],[-6.02,ylim1[0]],'r--')                 # freq. de corte original
text(w[i[0]]/pi+0.01,ylim1[0]+5,'$f_c$ obtida',fontsize=8);
text(Wc+0.02,ylim1[0]+5,'$f_c$ desejada',fontsize=8,color=[1, 0, 0])
subplot(1,2,2);  f = w/pi*fs/2                       # Hz
plot(f,abs(H1),linewidth=2), grid('on'); ylabel('magnitude')
title('FPB FIR, 4a. Ordem, $f_c$ = 800 Hz'); xlabel(u'frequência (Hz)')
plot(f,abs(H2),'r-.',linewidth=2); plot(f,abs(H3),'m--',linewidth=2)
legend(('Retangular','Hamming','Bartlett'))
plot([0, fs/2],[1./sqrt(2), 1./sqrt(2)],'k--')       # linha de meia potência
text(10,1./sqrt(2)-0.02,u'Meia Potência',fontsize=8)
plot([0, fs/2],[1/sqrt(2), 1/sqrt(2)], 'k--')          # linha de meia potência
text(10,1/sqrt(2)-0.02,'Meia Potência',fontsize=8)
```

# Método da Janela

- Exemplo 5

  Estime a ordem de um FPB FIR projetado com o Método da Janela e que atenda às seguintes especificações:

  – Ripple na banda passante: $\delta_p \; dB = 20 \cdot \log_{10}(1+\delta_p)$
  – Atenuação na b. de rejeição: $\delta_s \; dB = -20\log_{10}(\delta_s)$
  – Frequência de corte: $f_c = (f_{pass} + f_{stop})/2$.




Banda Passante: 0 – 1850 Hz; Banda de Bloqueio: 2150 – 4000 Hz;
Ripple na Banda Passante < 1,0 dB; Taxa de Amostragem: 8000 Hz.
Atenuação mínima na banda de bloqueio de 20 dB

Banda de Transição Normalizada: $\Delta f = |f_{pass} - f_{stop}|/f_s = |2150\text{-}1850|/8000 = 0{,}0375$

A **Janela Retangular** apresenta 0,74 dB de ripple na Banda Passante e atenuação de 21 dB na Banda de Bloqueio, atendendo às especificações do filtro.

Assim a ordem do filtro FIR é dada por: $\boldsymbol{N} = \boldsymbol{C} - 1 = 0{,}9/\Delta f - 1 = 0{,}9/0{,}0375 - 1 = 23 \rightarrow \boldsymbol{N} = 24$

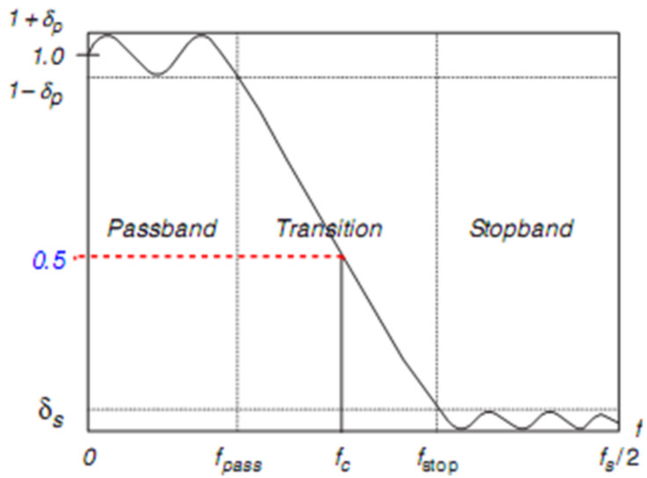
**Com ordem resultante ímpar deve-se escolher a ordem par superior (p/ gerar simetria de $h[n]$).**

# Método da Janela

- Exercício

  Estime a ordem de um FPB FIR projetado com o Método da Janela e que atenda às seguintes especificações:

  – Ripple na banda passante: $\delta_p \; dB = 20 \cdot \log_{10}(1+\delta_p)$
  – Atenuação na b. de rejeição: $\delta_s \; dB = -20\log_{10}(\delta_s)$
  – Frequência de corte: $f_c = (f_{pass} + f_{stop})/2$.




Banda Passante: 0 – 1850 Hz; Banda de Bloqueio: 2150 – 4000 Hz;
Ripple na Banda Passante < 0,1 dB; Taxa de Amostragem: 8000 Hz.
Atenuação mínima na banda de bloqueio de 40 dB

Banda de Transição Normalizada: $\Delta f = |f_{pass} - f_{stop}|/f_s = |2150\text{-}1850|/8000 = 0{,}0375$

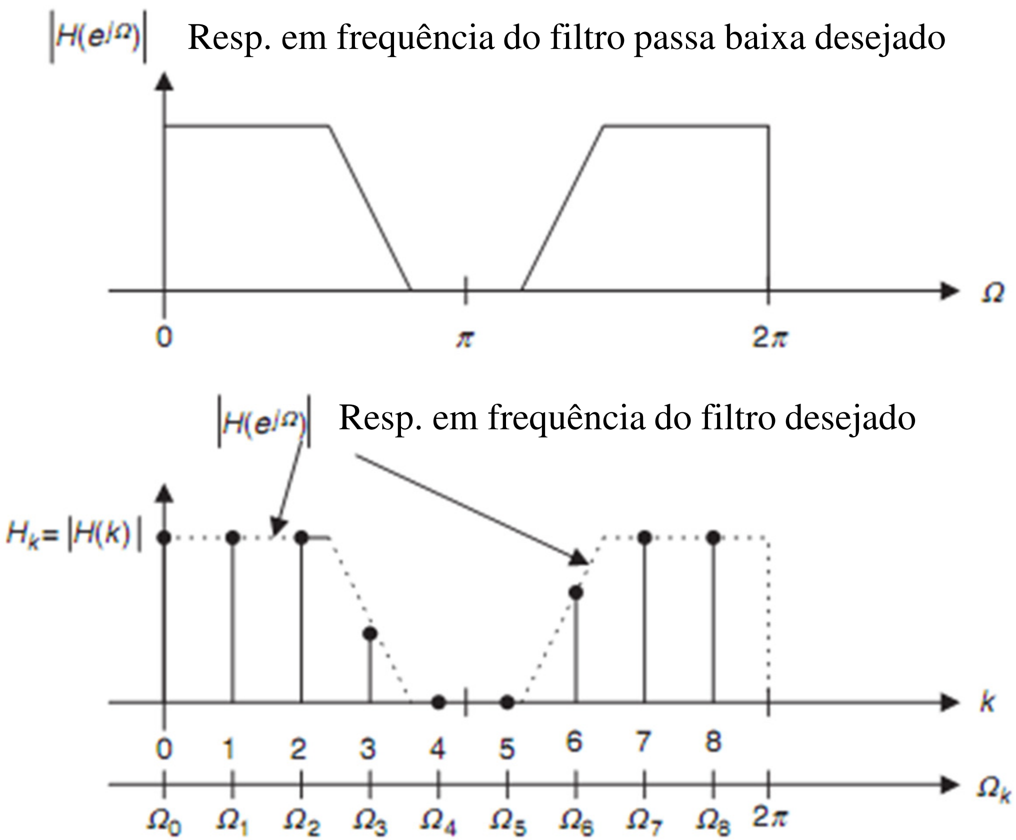
A **Janela Retangular** apresenta 0,74 dB de ripple na Banda Passante e atenuação de 21 dB na Banda de Bloqueio, atendendo às especificações do filtro.

Assim a ordem do filtro FIR é dada por: $N = C - 1 = 0{,}9/\Delta f - 1 = 0{,}9/0{,}0375 - 1 = 23 \rightarrow N = 24$

**Com ordem resultante ímpar deve-se escolher a ordem par superior (p/ gerar simetria de $h[n]$).**

# Método da Amostragem em Frequência - MAF

- Maior flexibilidade de projeto
- Os coeficientes do filtro FIR são calculados a partir da magnitude das amostras da Resposta em Frequência desejada
- $h[n]$ é a resposta impulsiva causal que se aproxima do filtro FIR desejado, e $H[k]$ é a sua DFT
- $H[k]$ pode ser obtida da amostragem da resposta em frequência do filtro desejado $H(e^{j\Omega})$, em frequências igualmente espaçadas



# Definição

- A Resposta ao Impulso do filtro FIR desejado pode ser calculada a partir da DFT Inversa (IDFT):

$$h[n] = \frac{1}{N}\sum_{k=0}^{N-1} H[k].W_N^{-kn}, \quad \text{para} \;\; n = 0,1,\ldots,N-1$$

$$\text{onde: } \; W_N = e^{-j2\pi/N} = \cos\frac{2\pi}{N} - j\,\text{sen}\,\frac{2\pi}{N}$$

- Assumindo que o filtro FIR deve ter fase linear (simetria ou anti-simetria) e o comprimento do filtro seja ímpar ($C = 2M+1$), então

$$h[n] = \frac{1}{2M+1}\left\{H[0] + 2\sum_{k=1}^{M} H[k]\cos\left(\frac{2k\pi(n-M)}{2M+1}\right)\right\}$$

$$\text{para} \;\; n = 0,1,\ldots,M \quad \text{onde: } M = (N-1)/2$$

- $H[k]$ é a magnitude da Resposta em Frequência do filtro desejado, amostrada nas frequências $\Omega_k = 2k\pi/(2M+1)$

MAF

# Etapas do Método

1. Especifique a magnitude da resposta em frequência ***H*[*k*]** do filtro desejado de **<u>comprimento</u>** **2*M*+1** (ímpar), em pontos igualmente espaçados na frequência digital normalizada de 0 a 2π.

2. Calcule os coeficientes do filtro FIR a partir de:

$$h[n] = \frac{1}{2M+1}\left\{H[0] + 2\sum_{k=1}^{M} H[k].\cos\left(\frac{2k\pi(n-M)}{2M+1}\right)\right\}$$

$$\text{para } \; n = 0,1,\ldots,M$$

3. Use a propriedade da simetria (fase linear) para determinar os demais coeficientes:

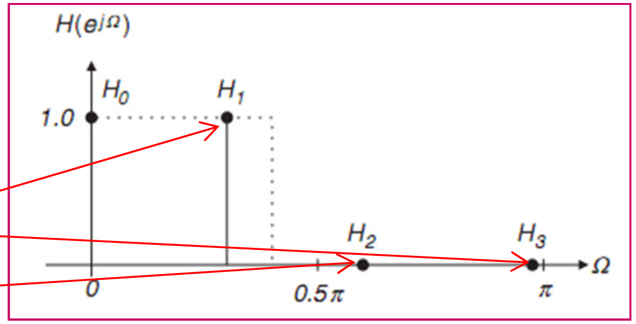
$$h[n] = h[2M - n]$$

$$\text{para}: \quad n = M+1, \ldots, 2M$$

MAF

# Exemplo 6

Projete um FPB FIR de fase linear e compr. $C$=7, com $\Omega_c = 0{,}3\pi$ rad/amostra, **usando MAF**




Como: $C = 2.M + 1 = 7 \rightarrow M = 3$, que é o número ímpar de amostras em freq.s positivas

$\Omega_k = k.2\pi / 7$ para $k = 0, 1, 2, 3 \rightarrow \Omega_k = \{ 0, 2\pi/7, 4\pi/7, 6\pi/7 \}$

Do gráfico temos: $H[k] = \{ \underline{1}, 1, 0, 0 \}$

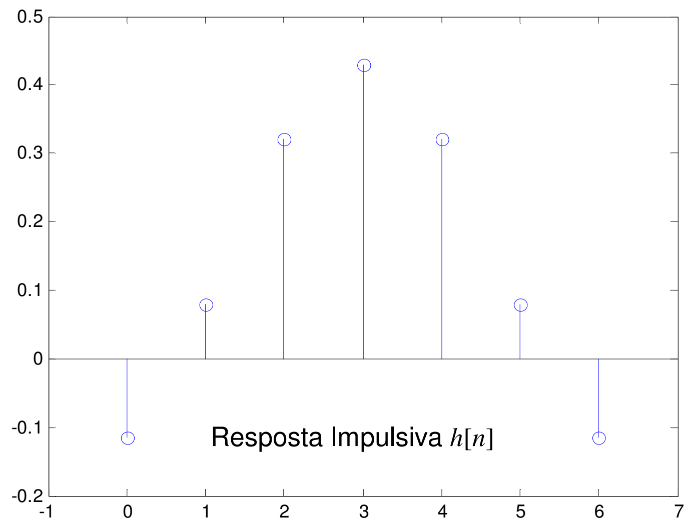
Para um período completo ($2\pi$): $H[k] = \{ \underline{1}, 1, 0, 0, 0, 0, 1 \}$

Usando a equação dos coeficientes (slide anterior):

$$h[n] = \frac{1}{7}\left\{ H[0] + 2\sum_{k=1}^{3} H[k].\cos\left[2k\pi(n-3)/7\right] \right\}$$

$$h[n] = \left\{ 1 + 2\cos\left[2\pi(n-3)/7\right] \right\}/7$$

```
>>> from numpy import arange, cos, pi
>>> n = arange(0,7)
>>> h = (1+2*cos(2*pi*(n-3)/7.))/7.
>>> for b in h: print "%.4f" % b,
-0.1146 0.0793 0.3210 0.4286 0.3210 0.0793 -0.1146
```




Equação de Diferenças:

$y[n] = h[0].x[n] + h[1].x[n-1] + h[2].x[n-2] + \ldots + h[6].x[n-6]$

MAF

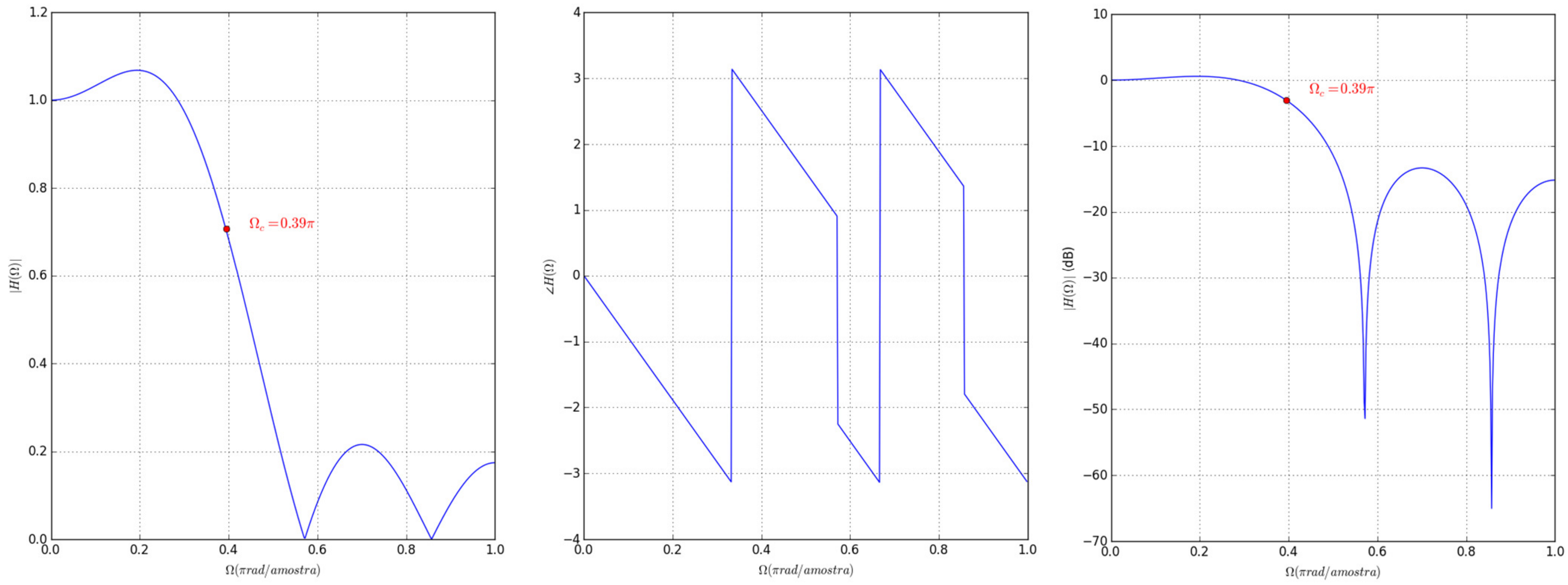
# Exemplo 6

Projete um FPB FIR de fase linear e compr. $C$=7, com $\Omega_c$=0,3π rad/amostra, usando **MAF**

Resposta em Frequência:

```
>>> from scipy.signal import freqz
>>> from pylab import plot, subplot
>>> b = h                    # coeficientes B's
>>> W,H = freqz(b,1)         # resp. em frequência
>>> subplot(2,1,1)
>>> plot(W,abs(H))           # magnitude da resp.freq
>>> subplot(2,1,2)
>>> plot(W,angle(H))         # fase da resp. em frequência
```

Filtro FIR Passa Baixa - 6a. Ordem



MAF

# Exemplo 6

Projete um **FPB** FIR de fase linear e comprimento 7 e com $\Omega_c = 0{,}3\pi$ rad/amostra, usando a **IDFT**

Como $C = 2.M+1 = 7$ ➔ $M = 3$, que é o número de amostras em frequência positiva

Considerando amostras distribuídas nas freq.s de 0 a $2\pi$:

$\Omega_k = \{\ 0, 2\pi/7, 4\pi/7, 6\pi/7, 8\pi/7, 10\pi/7, 12\pi/7\ \}$

$\Omega_k = \{\ 0, 0.29\pi, 0.57\pi, 0.86\pi, 1.14\pi, 1.43\pi, 1.71\pi\ \}$

$H_k = \{\ 1, 1, 0, 0, 0, 0, 1\ \}$ % função periódica em $2\pi$

Usando a ***ifft***( ) para calcular os coeficientes, $h[n]$:

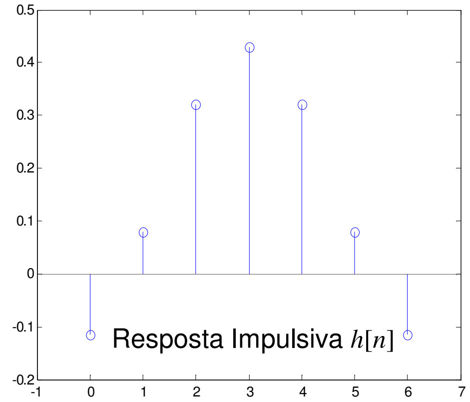
```
>> h = ifft(H)
```

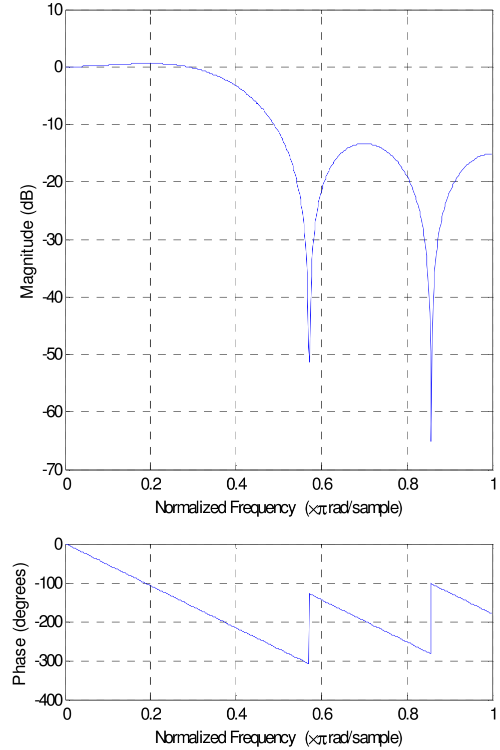
0.4286 0.3210 0.0793 -0.1146 -0.1146 0.0793 0.3210

Deslocando os coeficientes para se obter $h[n]$ simétrico:

```
>> h = [ h(1:M+1) h(M+2:N) ]
```

| | | | | | | |
|---|---|---|---|---|---|---|
| -0.1146 | 0.0793 | 0.3210 | 0.4286 | 0.3210 | 0.0793 | -0.1146 |
| $b_0$ | $b_1$ | $b_2$ | $b_3$ | $b_4$ | $b_5$ | $b_6$ |



MAF

# Exemplo 7 – *Script* matlab

Projete um FPF FIR de fase linear e compr. $C$=25 com freq.s de corte inferior e superior em 1 kHz e 3 kHz para sinais amostrados em 8 kHz, usando o MAF.

Como: C = 2M+1 = 25 → M = 12 (qtde de amostras em frequência positiva)

$\Omega_k = 2k.\pi / 25$ para $k = 0, 1, ..., 12.$

Amostras da R.F. nas frequências $\Omega_k$ : $H_k(\Omega_k) = \{ 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0 \}$

Cálculo dos coeficientes do filtro desejado (código Matlab):

```
% Mét. da Amostragem em Freq. - Compar. filtros com  resp. em freq. aguda e suave
fs = 8000;                              % frequencia de amostragem (Hz)
H1 = [0 0 0 0 1 1 1 1 1 0 0 0 0];      % valores das amostras da RF desejada
B1 = firfs(25,H1);                      % projeta o filtro c/ 'frequency sampling'
[h1,f] = freqz(B1,1,512,fs);            % magnitude da resp. em frequencia
H2 = [0 0 0 0 1/2 1 1 1 1/2 0 0 0 0];    % valores da RF desejada mais suaves
B2 = firfs(25,H2);                      % projeta o filtro c/ ‘freq. sampling’
[h2,f] = freqz(B2,1,512,fs);            % magnitude da resp. em frequencia
f1 = 180*unwrap(angle(h1)')/pi;  f2 = 180*unwrap(angle(h2)')/pi; % desempacota fase
subplot(2,1,1); plot(f,20*log10(abs(h1)),'-.',f,20*log10(abs(h2))); grid;
axis([0 fs/2 -100 10]); xlabel('Frequência (Hz)'); ylabel('Magnitude (dB)');
subplot(2,1,2); plot(f,f1,'-.',f,f2); grid;
xlabel('Frequência (Hz)'); ylabel('Fase (graus)');
```

MAF

# Exemplo 7 – **Função matlab**

função auxiliar: **firfs(C,Hk)** – FIR filter frequency sampling design

```
function B = firfs(C,Hk)
% B = firfs(C,Hk)
%
% FIR filter design using the frequency sampling method.
% Input parameters:
%   C:  the number of filter coefficients (C must Bessel odd).
%   Hk: sampled frequency response, for k = 0,1,2,...,M=(C-1)/2.
% Output:
%   B: linear phase FIR filter coefficients.

M = (C - 1)/2;
for n = 1:C
    B(n)  = (1/C)*(Hk(1)+2*sum(Hk(2:M+1).*cos(2*pi*([1:M])*(n-1-M)/C)));
end
```

MAF

# Exemplo 7 – *Script* Python

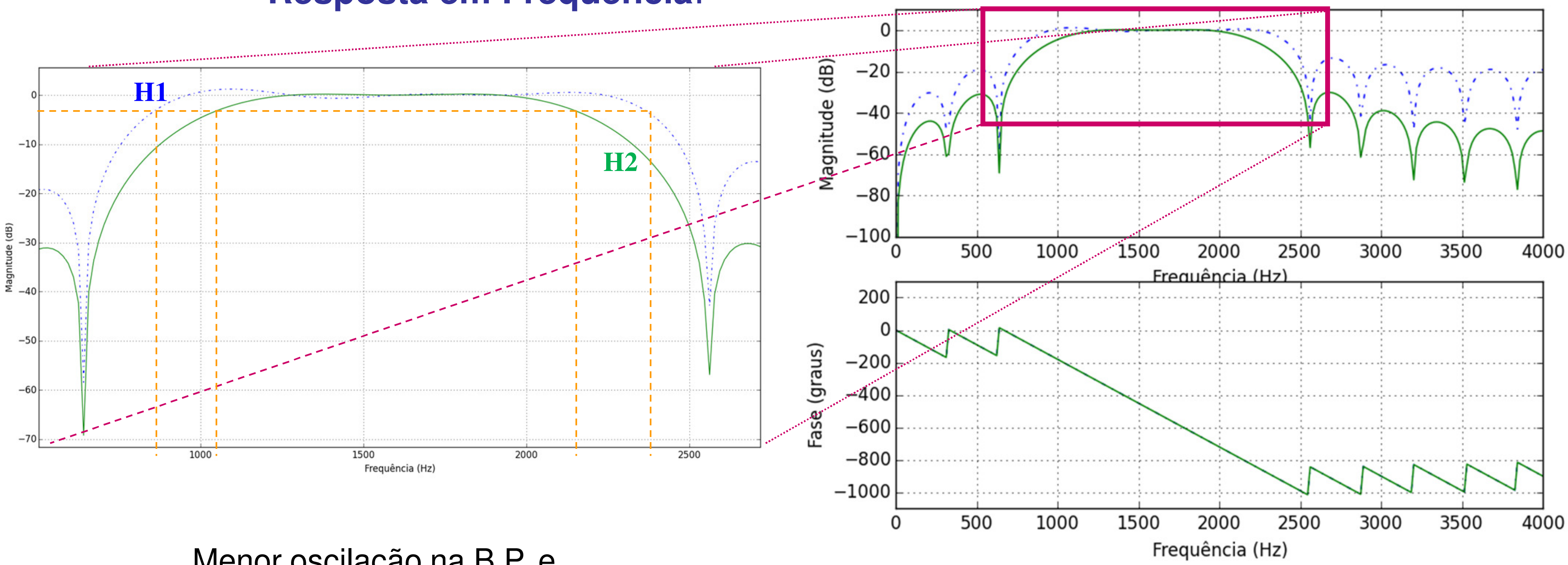
```
from numpy import arange, array, abs, angle, exp, pi, zeros, cos, log10, unwrap
from matplotlib.pylab import plot, subplot, xlabel, ylabel, axis, grid
from scipy.signal import freqz

def firfs(C,Hk):
# Projeta filtro FIR usando o método da amostragem em frequência.
#   C:  quantidade de coeficientes do filtro (C deve ser ímpar).
#   Hk: resposta em frequência (RF) amostrada, para k = 0,1,2,...,M = (C-1)/2.
# Saída: vetor B com coeficientes do filtro FIR de var. de fase linear c/ a freq.
  M = (C - 1)/2                                # metade dos coeficientes
  B = zeros(C)*1j                              # vetor dos 2M+1 coeficientes (complexos)
  for n in arange(0,C):
    B[n] = (1/float(C))*(Hk[0]+2*sum(Hk[1:M+1]*cos(2*pi*arange(0,M)*(n-M)/C)))
  return B

fs = 8000                                      # frequência de amostragem (Hz)
H1 = array([0,0,0,0,1,1,1,1,1,0,0,0,0])        # valores das amostras da RF desejada
B1 = firfs(25,H1)                              # projeta o filtro c/ freq. de amostr. fs
w, h1 = freqz(B1,1,512,fs)                     # magnitude da resp. em frequência
f1 = 180.*unwrap(angle(h1))/pi                 # fase da resp. em frequência (em graus)
H2 = array([0,0,0,0,0.5,1,1,1,0.5,0,0,0,0])# valores mais suaves da RF desejada
B2 = firfs(25,H2)                              # projeta o filtro c/ freq. de amostr. fs
w, h2 = freqz(B2,1,512,fs)                     # magnitude da resp. em frequencia
f2 = 180.*unwrap(angle(h2))/pi                 # unwrap(): desdobra a fase
f = w/pi*fs/2.                                 # frequência (em Hz)

subplot(2,1,1); plot(f,20.*log10(abs(h1)),'-.',f,20.*log10(abs(h2))); grid('on')
axis(array([0., fs/2., -100., 10.])); xlabel(u'Frequência (Hz)');
ylabel('Magnitude (dB)'); subplot(2,1,2); plot(f,f1,'-.',f,f2); grid('on')
axis(array([0., fs/2., -1100., 300.])); xlabel(u'Freq(Hz)'); ylabel('Fase (graus)')
```

MAF

# Exemplo 7 - Resultados

FPF FIR de fase linear e compr. *C*=25 com freq.s de corte inferior e superior em 1 k e 3 kHz para sinais amostrados em 8 kHz, usando o método MAF.

**Resposta em Frequência**:




Menor oscilação na B.P. e
maior atenuação na B.R.
para RF mais suave, H2 (verde)
Atenção: freq.s de corte são alteradas

# Considerações sobre o MAF

- As **oscilações** resultantes do efeito Gibbs nas bandas passante e de rejeição podem ser **reduzidas** com o aumento da largura do lóbulo principal da RF (implica em banda de transição maior)

- As **oscilações** na banda passante podem ser **reduzidas** se as especificações da magnitude da RF implicarem em um decaimento **mais suave** na banda de transição

- Como os valores da magnitude da RF, $H_k$ para $k$ = 0,1,...,$M$ são especificadas de acordo com as necessidades do projeto, o MAF[1] é considerado o método **mais flexível** para Projeto de Filtros FIR

- Para mais detalhes sobre esse método, veja o livro-texto (Lathi, p.779, M9.3)

[1] Método da Amostragem em Frequência

# Método de Projeto Ótimo

- **Algoritmo de Parks-McClellan** (**minmax**)
  - É o método de projeto otimizado mais conhecido: eficiente e flexível
  - James H. McClellan e Thomas W. Parks publicaram[1] o algoritmo em 1972, baseado no critério de alternância de mínimos e máximos dos polinômios de Chebyshev e usando o **algoritmo de trocas de Remez** na otimização iterativa:

    minimização do erro máximo da aproximação do **polinômio de Chebyshev** à magnitude da Resposta em Frequência (RF) do filtro desejado

    - Dado a RF desejada, $H_d(e^{j\Omega})$, o erro de aproximação ponderado $E_p(e^{j\Omega})$ será:

$$E_p(e^{j\Omega}) = W(e^{j\Omega}).|H(e^{j\Omega}) - H_d(e^{j\Omega})|$$

onde: $H(e^{j\Omega})$ é a RF do filtro FIR de fase linear projetado,
$W(e^{j\Omega})$ é a função peso usada para enfatizar certas bandas de frequências durante o processo de otimização:

$$\mathbf{min(max(}E_p(e^{j\Omega})\mathbf{))}\quad \text{para um dado conjunto de coeficientes}$$

[1] J. H. McClellan and T. W. Parks, "A unified approach to the design of optimum FIR linear phase digital filters", IEEE Trans. Circuit Theory, vol. CT-20, pp. 697-701, 1973.
J. H. McClellan, T. W. Parks and L. R. Rabiner, "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters", IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 506-525, 1973.

# Método de Projeto Ótimo

Mais usado!!!

– Mét. de Parks-McClellan é uma variação do algoritmo de trocas de Remez

– Usa o algoritmo de **trocas de Remez** e a aproximação de **Chebyshev** para produzir filtro FIR com $|H(e^{j\Omega})|_{\text{projetada}}$ mais próxima possível da $|H(e^{j\Omega})|_{\text{desejada}}$

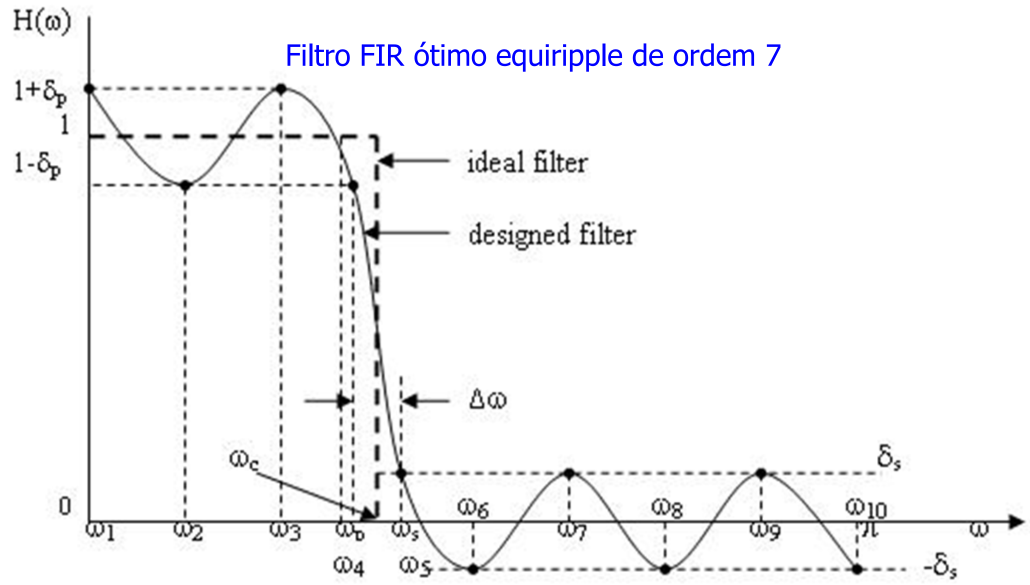
– **MinMax** – algoritmo minimiza o erro máximo entre $|H(e^{j\Omega})|_{\text{des}}$ e $|H(e^{j\Omega})|_{\text{proj}}$

de $-\delta_p$ a $+\delta_p$

– Filtros de fase linear com comportamento *equiripple* em suas RFs: $|H(\Omega)|$'s

– Desde a década de 60 sabe-se que os melhores filtros (ótimos) tem RF's do tipo *equiripple*

– Método essencialmente computacional:
no *Matlab* `firpm()` e
no *Python* `remez()`



# Método de Projeto Ótimo

- Etapas do projeto de filtros FIR usando o Método Ótimo:
  1. Especifique as frequências da B.P. e da B.R., o *ripple* máximo na B.P., a atenuação mínima na B.R., a ordem do filtro, e a frequência de amostragem do sistema DSP.
  2. Normalize as frequências das bordas das bandas para o limite de Nyquist (frequência de **Nyquist** ou frequência de dobra: $\boldsymbol{f_s}/2$) e especifique as magnitudes desejadas.
  3. Calcule os valores absolutos das oscilações (*ripples*) nas bandas passante e de rejeição se foram dados em dB:

$$\delta_P = 10^{\delta_{P(dB)}/20} - 1$$

$$\delta_S = 10^{-\delta_{S(dB)}/20}$$

  4. Estabeleça os pesos para as bandas passante e de rejeição: $\dfrac{\delta_P}{\delta_S} = \dfrac{W_S}{W_P}$
  5. Use o algoritmo de trocas de **Remez** para calcular os coeficientes do filtro.
  6. Se as especificações não forem atendidas, aumente a ordem do filtro e repita o passo anterior.

# Método de Projeto Ótimo

- Exemplo 8

FPB FIR de fase linear com B.P. de 0 a 800 Hz e B.R. de 1k a 4 kHz, ripple máx. na B.P. de 1 dB, e atenuação mínima na B.R. de 40 dB, p/ sinais amostrados em 8 kHz, usando o método **de Projeto Ótimo** (minmax).

Frequências normalizadas: $\Omega_P = 2\pi.800/8000 = 0{,}2\pi$; $\Omega_S = 2\pi.1000/8000 = 0{,}25\pi$

Ripples em valores absolutos: $\delta_P = 10^{1/20} - 1 = 0{,}122$; $\delta_S = 10^{-40/20} = 0{,}01$

Fatores de ponderação: $\delta_P / \delta_S = 12{,}2 \approx 12 / 1 = W_S / W_P \rightarrow W_P = 1, W_S = 12$

Usando a função **`firpmord()`** do **Matlab** para estimar a ordem do Filtro:

```
%%%%%%%%%%  Método de Projeto Ótimo – Parks-McClellan  %%%%%%%%%%%%%%%
fs = 8000;                              % frequência de amostragem
Bt  = [800 1000];                       % banda de transição
r = [0.122 0.01];                       % ripple máximo nas bandas de passagem e de rejeição
W = [1 12];                             % fatores de ponderação do erro
A = [1 0];                              % amplitude da RF nas bandas definidas em 'Bt'
[N,fo,mo,W] = firpmord( Bt, A, r, fs);  % estimação da ordem do filtro de Parks-McClellan
```

Ordem estimada:48

# Método de Projeto Ótimo

- Exemplo 8

FPB FIR de fase linear com B.P. de 0 a 800 Hz e B.R. de 1k a 4 kHz, ripple máx. na B.P. de 1 dB, e atenuação mínima na B.R. de 40 dB, p/ sinais amostrados em 8 kHz, usando o método **de Projeto Ótimo** (minmax).

Frequências normalizadas: $\Omega_P = 2\pi.800/8000 = 0,2\pi$; $\Omega_S = 2\pi.1000/8000 = 0,25\pi$

Ripples em valores absolutos: $\delta_P = 10^{1/20} - 1 = 0,122$; $\delta_S = 10^{-40/20} = 0,01$

Fatores de ponderação: $\delta_P / \delta_S = 12,2 \approx 12 / 1 = W_S / W_P \rightarrow W_P = 1, W_S = 12$

Usando o **Matlab** para projetar o filtro (cont. slide anterior):

```
%%%%%%%%%%  Método de Projeto Ótimo – Parks-McClellan  %%%%%%%%%%%%%%
%    Resultados dos cálculos realizados pela estimação da ordem (slide anterior)
N = 48                              % ordem do filtro desejado
fo  = [0 0.2 0.25 1]                % frequências normalizadas das bordas das bandas
mo = [1 1 0 0]                      % magnitudes da RF do projeto desejado
W = [1 12.2]                        % fatores de ponderação do erro
b = firpm(N,fo,mo,W);                       % coeficientes do filtro ótimo
freqz(b,1,512,fs); axis([0 fs/2 -80 5]);    % plota a resposta em frequência
```

# Método de Projeto Ótimo

- Exemplo 8

FPB FIR de fase linear com B.P. de 0 a 800 Hz e B.R. de 1k a 4 kHz, ripple máx. na B.P. de 1 dB, e atenuação mínima na B.R. de 40 dB, p/ sinais amostrados em 8 kHz, usando o método **de Projeto Ótimo** (minmax).

Frequências normalizadas: $\Omega_P = 2\pi.800/8000 = 0{,}2\pi$; $\Omega_S = 2\pi.1000/8000 = 0{,}25\pi$

Ripples em valores absolutos: $\delta_P = 10^{1/20} - 1 = 0{,}122$; $\delta_S = 10^{-40/20} = 0{,}01$

Fatores de ponderação: $\delta_P / \delta_S = 12{,}2 \approx 12 / 1 = W_S / W_P \rightarrow W_P = 1, W_S = 12$

Usando **Python** para projetar o filtro:

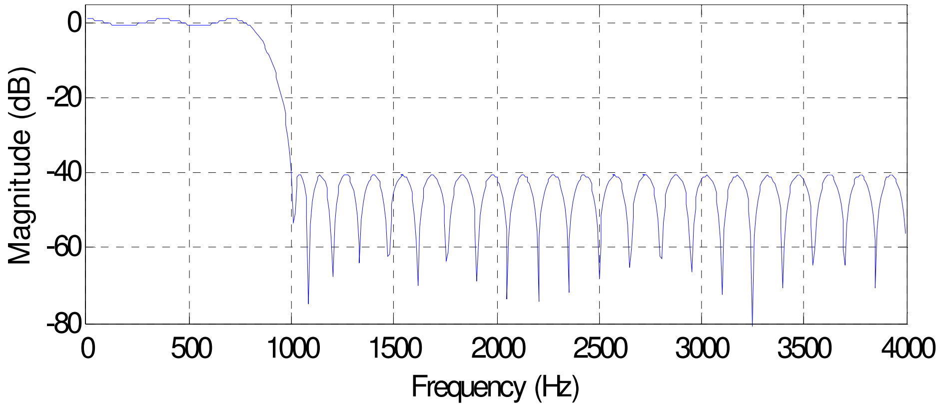
```
from scipy.signal import  remez, freqz
from matplotlib.pyplot import plot, grid
from numpy import array, pi, log10
N = 48; fo  = [0,0.2,0.25,1]/2           # ordem e frequências normalizadas das bordas das bandas
mo = [1,0]; W = [1,12.2]                 # magnitudes da RF e fatores de ponderação do erro
hn = remez(array(N), array(fo), array(mo), array(W))
w, rf = freqz(hn)
mag = 20.*log10(abs(rf))
plot(w/(2*pi)*8000, mag, 'b-')           # frequência em Hz
grid(‘on’)
```

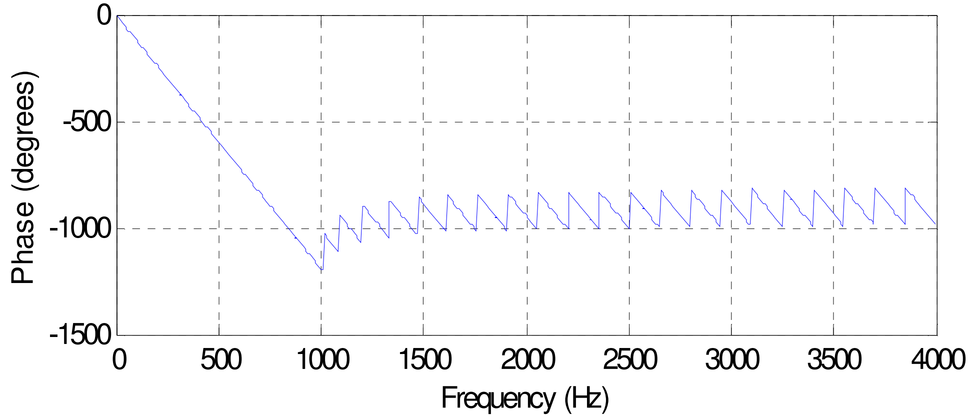
# Método de Projeto Ótimo

- Exemplo 8

FPB FIR de fase linear com B.P. de 0 a 800 Hz e B.R. de 1k a 4 kHz, ripple máx. na B.P. de 1 dB, e atenuação mínima na B.R. de 40 dB, p/ sinais amostrados em 8 kHz, usando o método de projeto ótimo (**minmax**). Ordem necessária = 53.

Resposta em Frequência:




Coeficientes do filtro projetado:

| | |
|---|---|
| $b_0 = b_{53} = -0.006075$ | $b_1 = b_{52} = -0.00197$ |
| $b_2 = b_{51} = 0.001277$ | $b_3 = b_{50} = 0.006937$ |
| $b_4 = b_{49} = 0.013488$ | $b_5 = b_{48} = 0.018457$ |
| $b_6 = b_{47} = 0.019347$ | $b_7 = b_{46} = 0.014812$ |
| $b_8 = b_{45} = 0.005568$ | $b_9 = b_{44} = -0.005438$ |
| $b_{10} = b_{43} = -0.013893$ | $b_{11} = b_{42} = -0.015887$ |
| $b_{12} = b_{41} = -0.009723$ | $b_{13} = b_{40} = 0.002789$ |
| $b_{14} = b_{39} = 0.016564$ | $b_{15} = b_{38} = 0.024947$ |
| $b_{16} = b_{37} = 0.022523$ | $b_{17} = b_{36} = 0.007886$ |
| $b_{18} = b_{35} = -0.014825$ | $b_{19} = b_{34} = -0.036522$ |
| $b_{20} = b_{33} = -0.045964$ | $b_{21} = b_{32} = -0.033866$ |
| $b_{22} = b_{31} = 0.003120$ | $b_{23} = b_{30} = 0.060244$ |
| $b_{24} = b_{29} = 0.125252$ | $b_{25} = b_{28} = 0.181826$ |
| $b_{26} = b_{27} = 0.214670$ | |

# Resumo

| | **Método da Janela** | **Método de Amostragem em Frequência** | **Método de Projeto Ótimo** |
|---|---|---|---|
| **Tipo de Filtro** | FPB, FPA, FPF, FRF<br>(não serve para seletividade frequencial arbitrária) | Qualquer tipo | Qualquer tipo |
| **Fase Linear** | Sim | Sim | Sim |
| **Especificações de ripple e atenuação** | Usadas na determina-ção da ordem do filtro e freqüência(s) de corte | Precisam ser verificadas após cada tentativa | Usadas no algoritmo;<br>Precisam ser verif. após cada tentativa |
| **Complexidade algorítmica**<br>(cálc.dos coeficientes) | **Moderada**<br>-Cálculo da seq. Impulso<br>-Ponderação c/ função janela | **Baixa**<br>Equação única | **Alta**<br>- Alg. de Parks-McClellan<br>- Alg. de trocas de Remez |
| **Ferramenta mínima de projeto** | Calculadora | Calculadora | Computador |

# Resumo

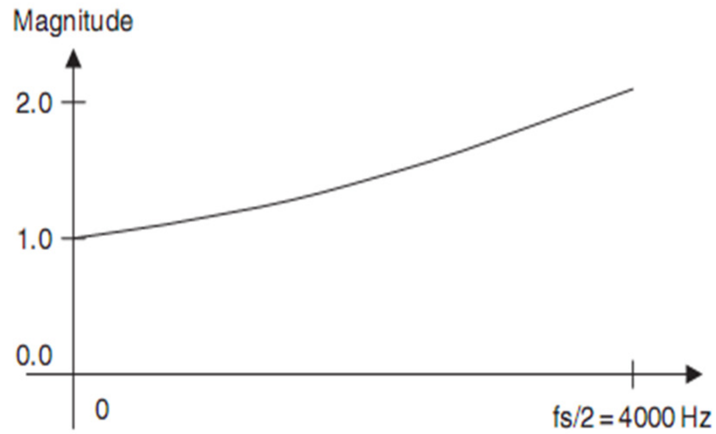
- Método da Transformada de Fourier (janela retangular) calcula coeficientes de filtros não-causais do tipo passa-baixas, passa-altas, passa-faixa e rejeita-faixa.
- Deslocamento temporal dos coeficientes produz filtros causais e introduz variação linear de fase com a freqüência. A saída de Filtros com variação de fase da RF do tipo linear, tem a mesma quantidade de atraso de tempo para todos os sinais de entrada com componentes frequenciais na banda passante do filtro.
- A janela retangular produz o efeito de Gibbs (oscilações nas bandas passante e de rejeição) na magnitude da RF devido ao truncamento abrupto da resposta impulsiva do filtro FIR (seqüência de coeficientes).
- Melhorias substanciais na magnitude da RF podem ser obtidas com outros tipos de função janela (amortecimento progressivo da magnitude na direção dos extremos da janela).
- O método da Amostragem em Freqüência é o mais flexível (permite o projeto de qualquer forma arbitrária da RF).
- O método de projeto ótimo, Algoritmo de Parks-McClellan usando o algoritmo de troca de Remez, oferece flexibilidade nas especificações do filtro.

# Exercício

– Determine o método apropriado de projeto de filtros FIR para cada uma das seguintes aplicações

1. Sistema *crossover* digital de duas bandas. O filtro deve atender:
   - Taxa de amostragem: 44,1kHz
   - Frequência de crossover: 1kHz (frequência de corte)
   - Banda de Transição: 600-1400Hz
   - FPB = B.P. de 0 a 600Hz com ripple de 0,02 dB e B.R. a partir de 1,4kHz com atenuação mínima de 50 dB
   - FPA = B.P. de 1,4 a 22,05kHz com ripple de 0,02 dB e B.R. de 0-600Hz com atenuação mínima de 50 dB
   - Não existe disponibilidade de uso do software para o projeto do filtro.

2. Equalizar um sinal de fala amostrado em 8kHz, usando um filtro FIR de fase linear baseado na RF mostrada na figura. Não existe disponibilidade de uso do software para o projeto do filtro.



# Exercício

Soluções

1. O método de projeto da janela é a primeira escolha, visto que nele a equação está em termos da frequência de corte (freq. crossover), a ordem do filtro é baseada na banda de Transição, e os tipos de filtros são padronizados: FPB e FPA. As especificações de ripple e da banda de rejeição podem ser satisfeitas pela escolha da janela de Hamming. O método de projeto ótimo também faria o trabalho com um desafio de satisfazer ganhos unitários combinados na frequência crossover (1kHz) se o algoritmo remez( ) estivesse disponível.

2. Como a magnitude da Resposta em Frequência não é a de um filtro padronizado, FPB, FPA, FPF ou FRF, e o algoritmo remez( ) não pode ser executado, a escolha seria o método de amostragem em frequência.

# Exercícios

1. Projete um FRF FIR de ordem 6 com frequência de corte inferior em 2 kHz e superior em 2,4 kHz para sinais amostrados a 8 kHz, usando a função janela de Hamming.
2. Desenvolva uma função Matlab/Python para projetar filtros FIR pelo método da janela, retornando um vetor com os coeficientes $b_i$:

```
function B = firjan(C, TipoF, Winf, Wsup, Jan)
% B = firjan(C, tipoF, Winf, Wsup, janela)
% Projeta um filtro digital FIR usando o método de Janelamento
%       Parâmetros de entrada:
%           C - qtde de coeficientes do filtro FIR (deve ser ímpar)
%           TipoF - o tipo do filtro
%                      1 para passa baixa
%                      2 para passa alta
%                      3 para passa faixa
%                      4 para rejeita faixa
%           Winf - freq. de corte inferior (rad/amostra). Nulo para filtro passa alta.
%           Wsup - freq. de corte superior (rad/amostra). Nulo para filtro passa baixa.
%           Jan - indicador da função janela
%                      1 para janela retangular
%                      2 para janela triangular
%                      3 para janela de Hanning
%                      4 para janela de Hamming
%                      5 para janela de Blackman
%       Parâmetros de saída:
%           B – coeficientes do filtro digital FIR (numerador da função de transferência)
```

# Exercícios

3. Usando a função desenvolvida no exercício anterior, projete um FPB FIR de ordem 25 com frequência de corte em 2kHz, para sinais amostrados a 8kHz. Plote a magnitude e a fase da resposta em frequência dos filtros obtidos para cada tipo de janela estudado, bem como suas respostas impulsivas.
4. Projete um FPA FIR com banda de rejeição de 0 a 1500 Hz, banda passante de 2500 a 4000 Hz, atenuação mínima na banda de rejeição de 40 dB, e ripple máximo na banda passante de 0,1 dB. Taxa de amostragem de 8000 Hz.
5. Idem ao exercício anterior, para um FPF FIR com banda de rejeição de 0 a 500 Hz e de 3500 a 4000 Hz, banda passante de 1600 a 2300 Hz, atenuação mínima na banda de rejeição de 50 dB, e ripple máximo na banda passante de 0,05 dB. Taxa de amostragem de 8000 Hz.
6. Desenvolva uma função Matlab/Python para projetar filtros FIR pelo método da amostragem em frequência, retornando um vetor com os coeficientes $b_i$.

```
function B = firfs(C, Hk)
% B = firfs(C, tipoF, Winf, Wsup, janela)
% Projeta um filtro digital FIR usando o método de Amostragem em Frequência.
%   Parâmetros de entrada:
%          C - qtde de coeficientes do filtro FIR (deve ser ímpar)
%          Hk - resposta em frequência amostrada em k = 0, 1, 2, ..., M = (C-1)/2.
%   Parâmetros de saída:
%        B - coeficientes do filtro digital FIR (numerador da f. de transferência).
```

# Exercícios

7. Projete um FPF FIR de fase linear e ordem 25 com B.P. de 1 k a 1,6 kHz e B.Rs. de 0 a 600 Hz e de 2 k a 4 kHz, ripple máx. na B.P. de 1dB, e atenuação mínima nas B.R.s de 30 dB, para sinais amostrados em 8 kHz, usando o método Ótimo.