Processamento Digital de Sinais

MÓDULO 7
Filtros Digitais

Gustavo Luís F. Vicente

Filtros Digitais

Introdução

- Servem para:
    - Separação de sinais
    - Restauração de sinais
- Pode ser feito com filtros analógicos
- Qual o melhor?
    - Analógicos são rápidos, baratos e com alta faixa dinâmica
    - Digitais possuem alta performance
        - Filtro PB de 1kHz com ganho de 1 ± 0,0002 de 0Hz a 1000Hz e ganho de 0,0002 acima de 1001Hz !!!

Filtros Digitais

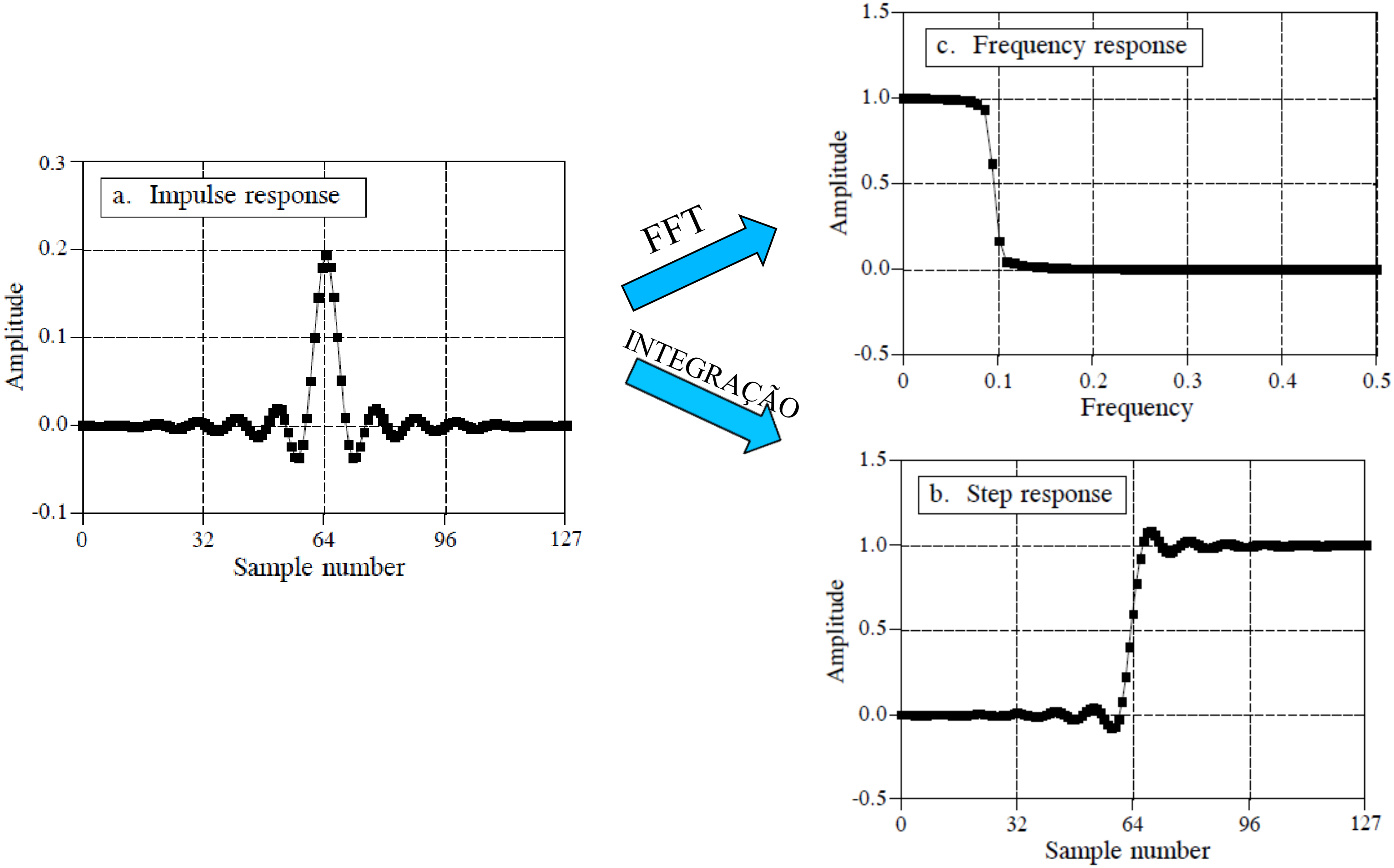
Introdução

- Implementação
  - Por convolução, através da *resposta ao impulso*
    - Utiliza pontos de entrada – FIR
    - mais precisos
    - Mais lentos
  - Por recursão
    - Utiliza pontos de entrada e de saída – IIR
    - Menos precisos
    - Mais rápidos

Filtros Digitais

Introdução

- Representações de uma resposta de um sistema

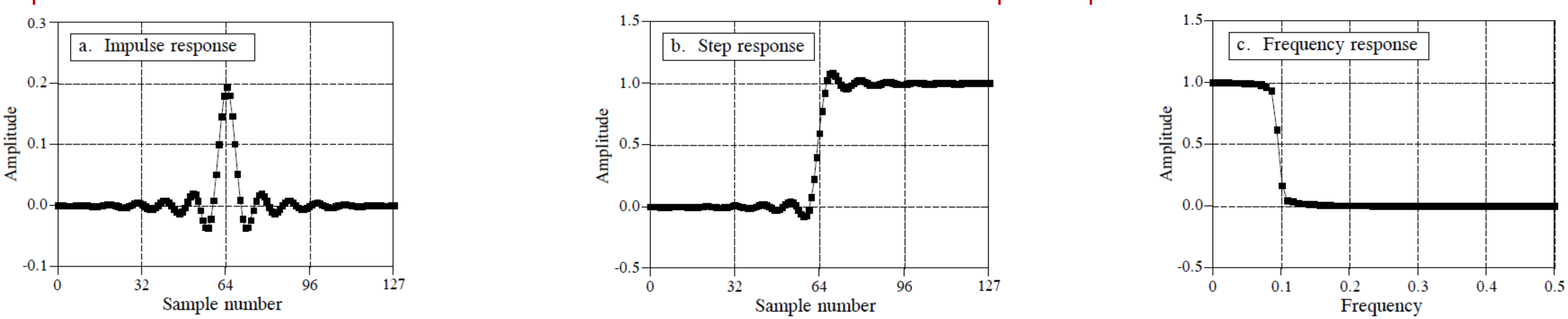


Filtros Digitais

Introdução

- Representações de uma resposta de um sistema
    - Resposta ao impulso (*Impulse Response*)
    - Resposta ao degrau (*Step Response*)
    - Resposta em frequência (*Frequency Response*)
- Três representações da mesma resposta
    - Três formas de ver o mesmo “fenômeno”

Domínio do Tempo **Domínio da Frequência**

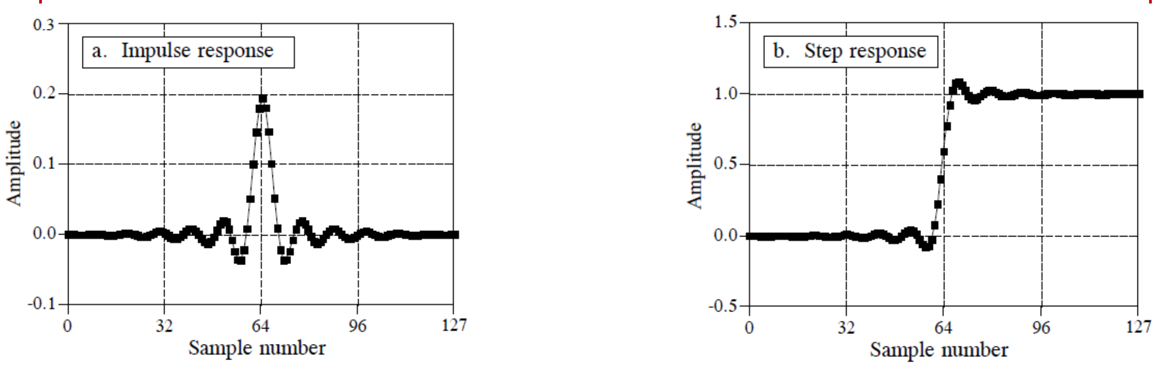


Filtros Digitais

Introdução

- Representações da resposta no Domíniio do Tempo:
  - Resposta ao impulso (*Impulse Response*)
  - Resposta ao degrau (*Step Response*)
    - mais fácil de interpretar
    - utilizada para avaliar um filtro

Domínio do Tempo

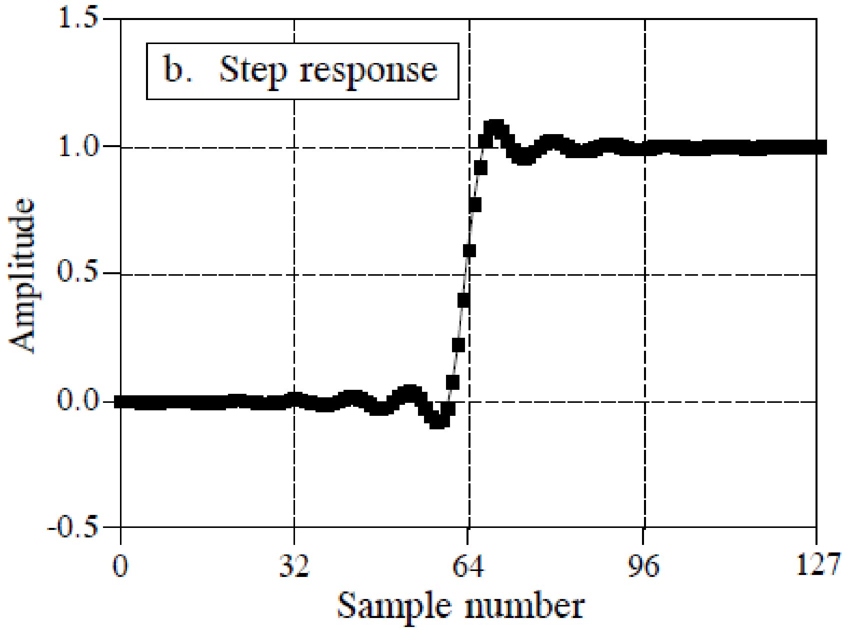


Filtros Digitais

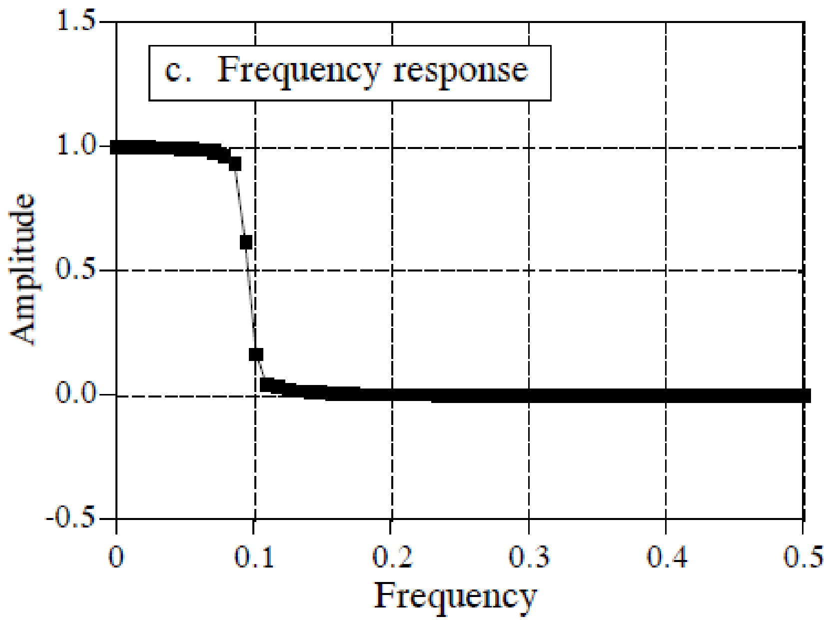
Introdução

- Representações da resposta de um filtro:
    - Resposta ao degrau
    - Resposta em frequência

Domínio do Tempo



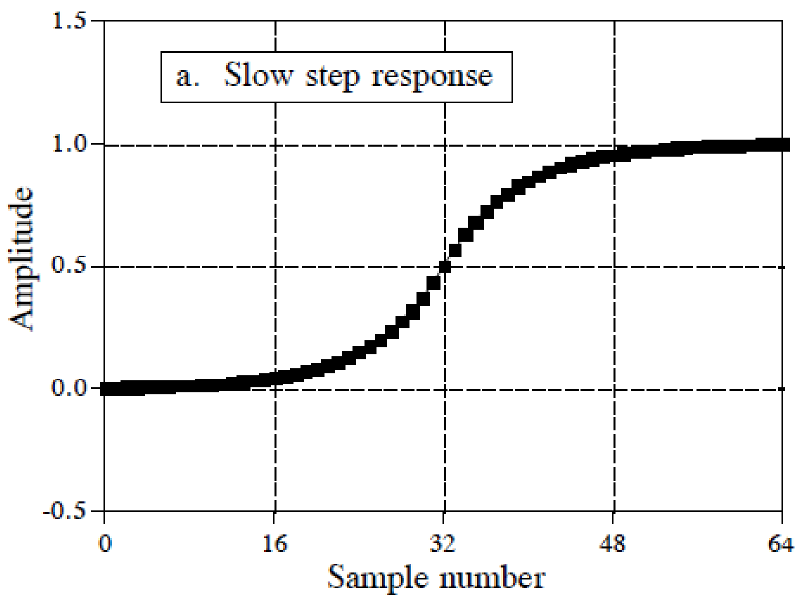
Domínio da Frequência



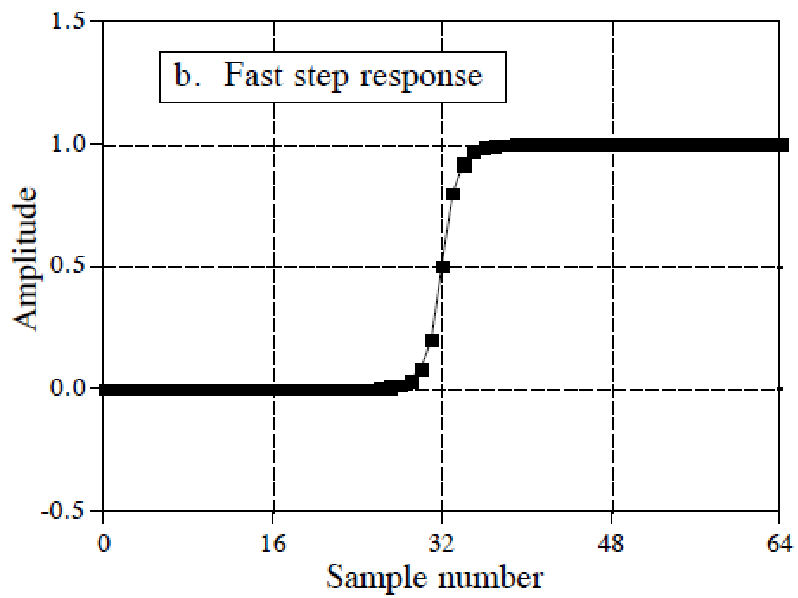
Filtros Digitais - Introdução

- Parâmetros no Domínio do Tempo (resposta ao degrau)
  - Tempo de resposta ou velocidade de transição (*risetime*)
    - Tempo entre 10% e 90% do sinal

RUIM



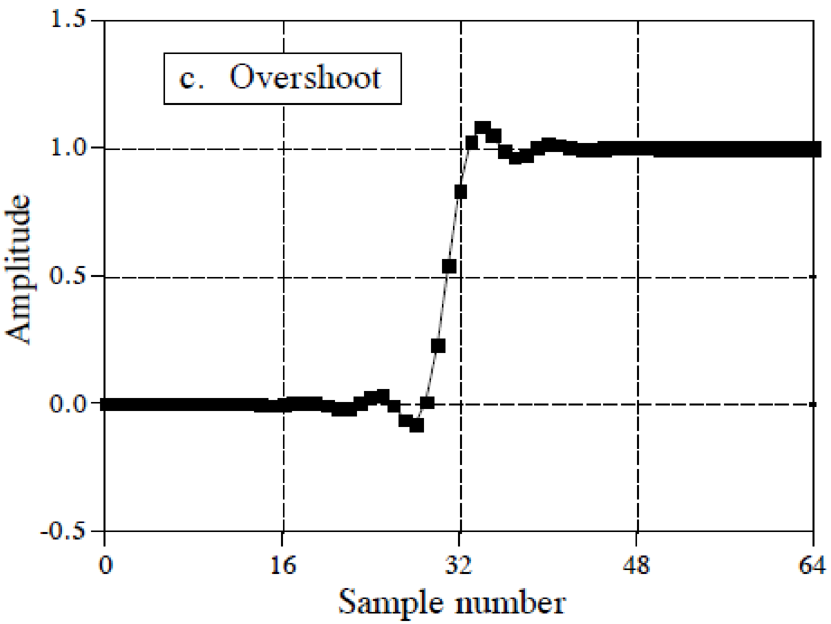
BOM



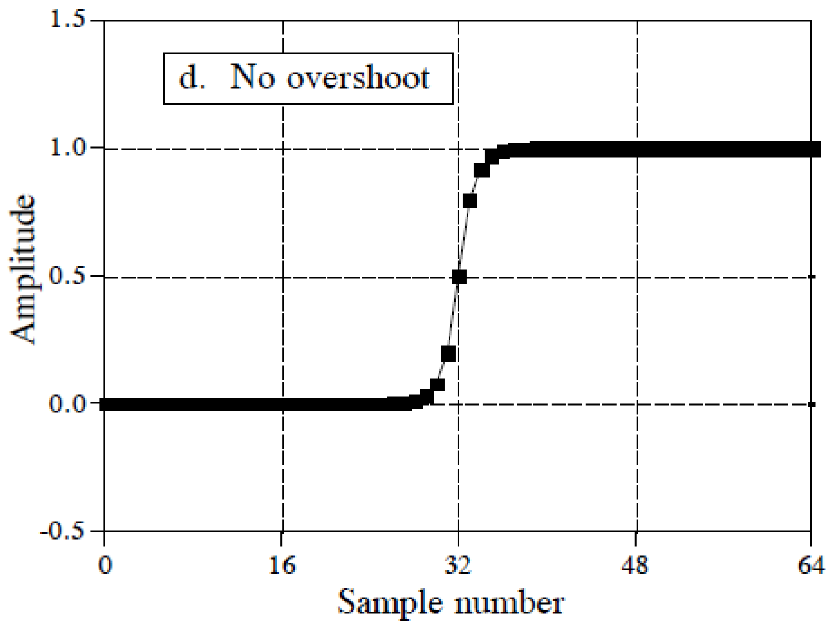
Filtros Digitais - Introdução

- Parâmetros no Domínio do Tempo (resposta ao degrau)
  - Overshoot
    - oscilação no início do degrau
    - Indesejada; deve ser reduzida ao máximo

RUIM

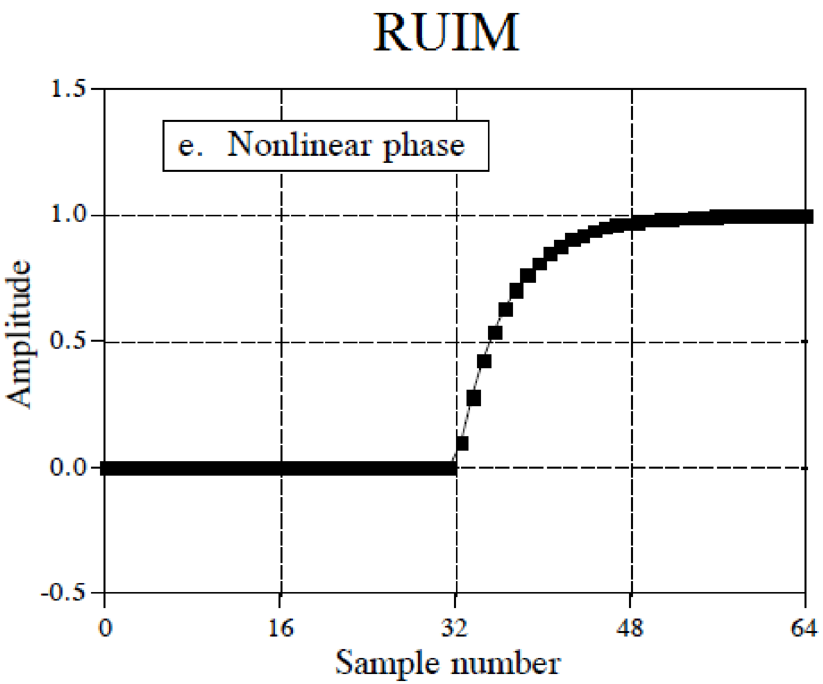


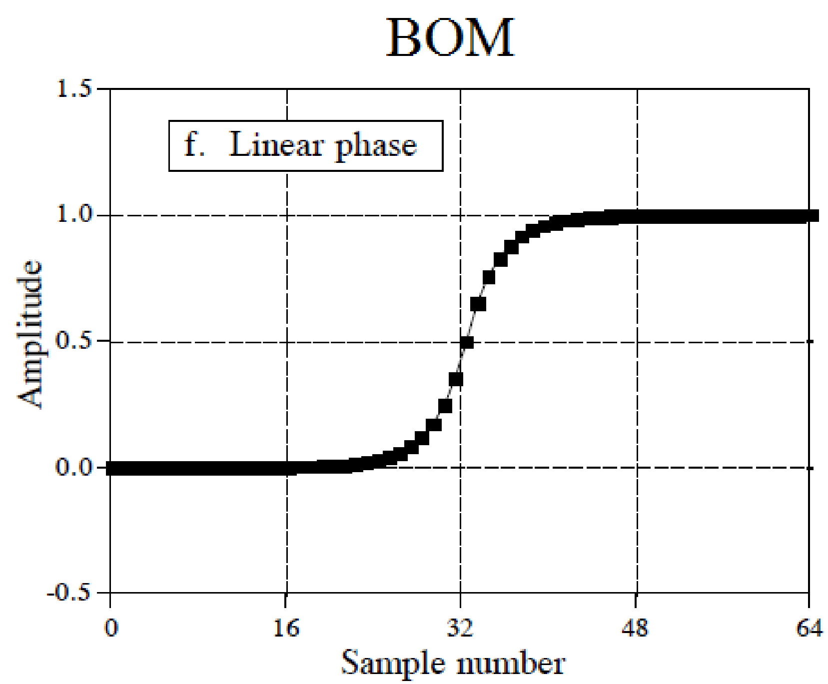
BOM



Filtros Digitais - Introdução

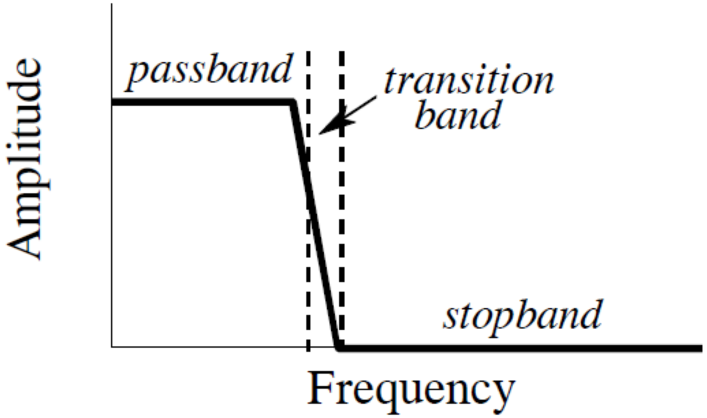
- Parâmetros no Domínio do Tempo (resposta ao degrau)
    - Simetria de Fase ou Linearidade de Fase (*Linear Phase*)
        - Deve haver simetria na resposta ao degrau





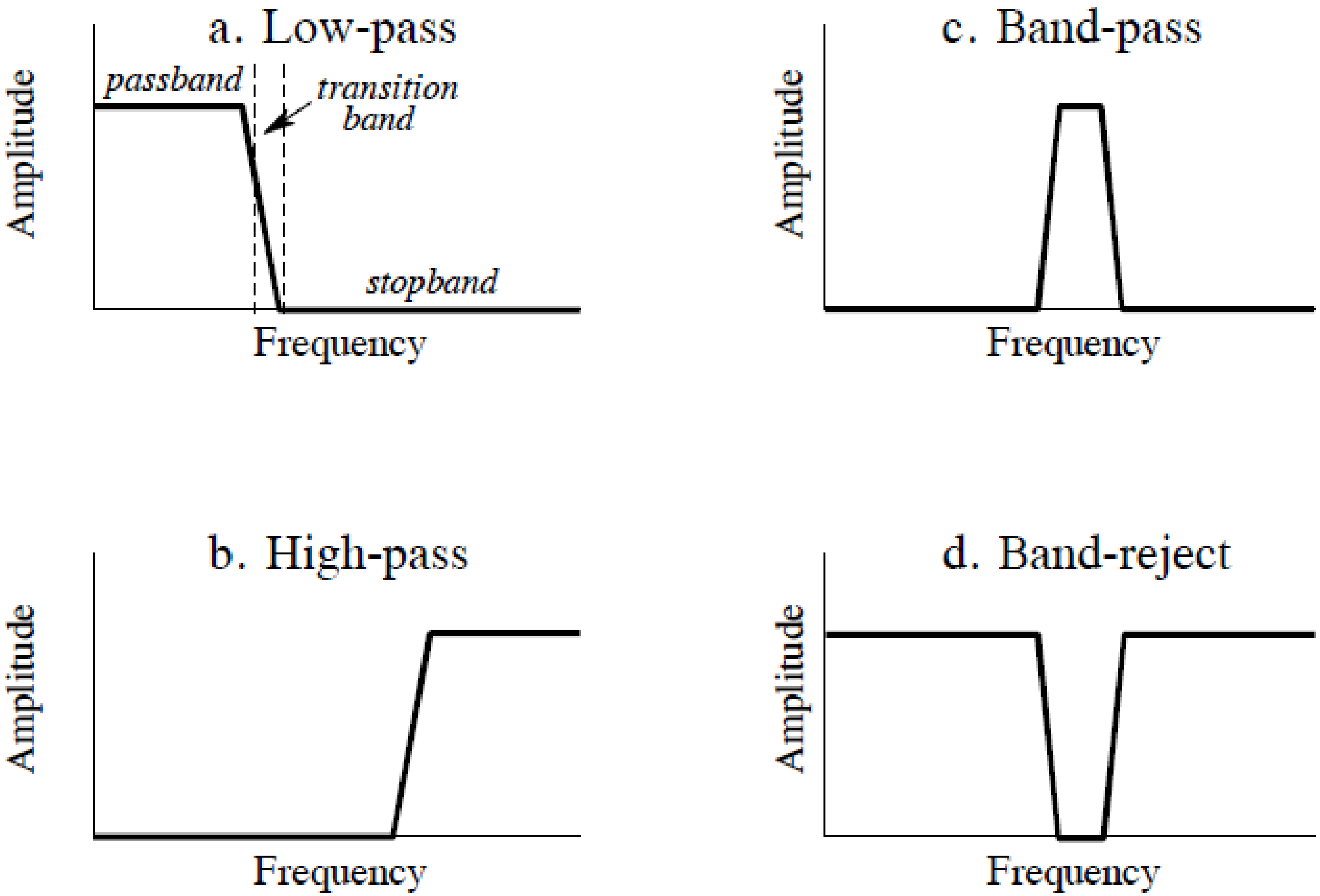
Filtros Digitais - Introdução

- Parâmetros no Domínio da Frequência
- Regiões de uma resposta em frequência
  - *Pass band* (banda passante): faixa de frequências do sinal que o filtro deixa passar
  - *Stop band* (banda de bloqueio): faixa de frequências que o filtro não deixa passar
  - *Transition band* (faixa de transição): faixa de frequências entre a *pass band* e a *stop band*



Filtros Digitais - Introdução

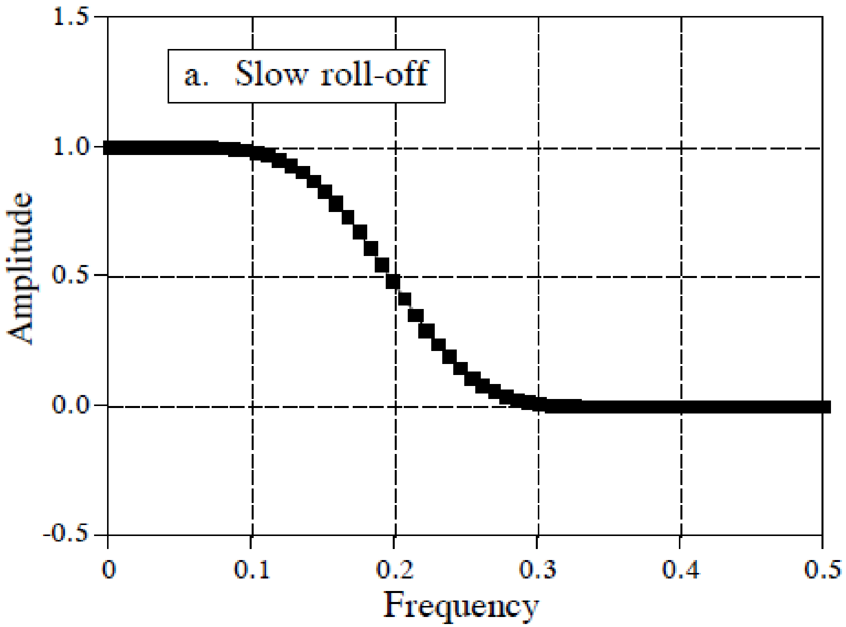
- Parâmetros no Domínio da Frequência
- Respostas básicas de filtros



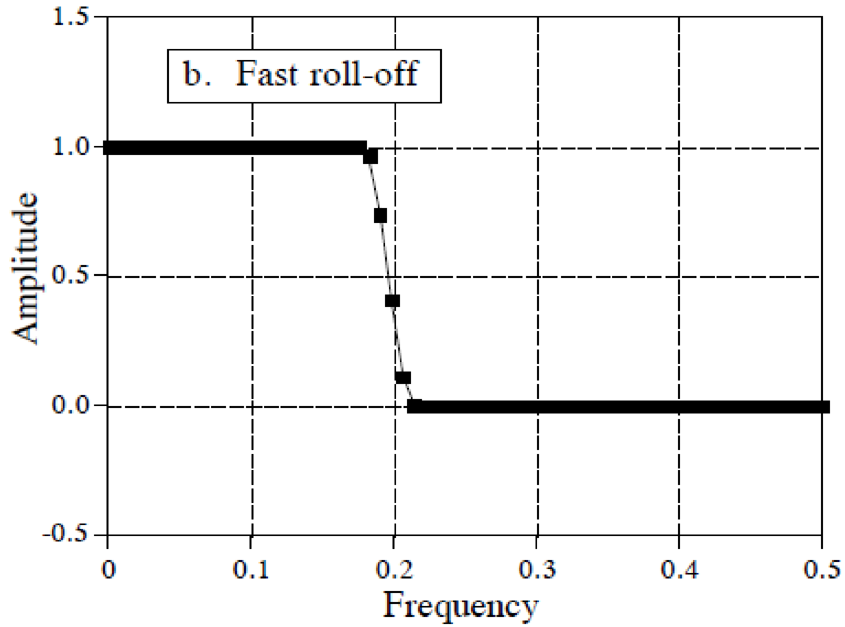
Filtros Digitais - Introdução

- Parâmetros no Domínio da Frequência
  - *roll-off*
    - Largura da faixa de transição (*transition band*)

RUIM

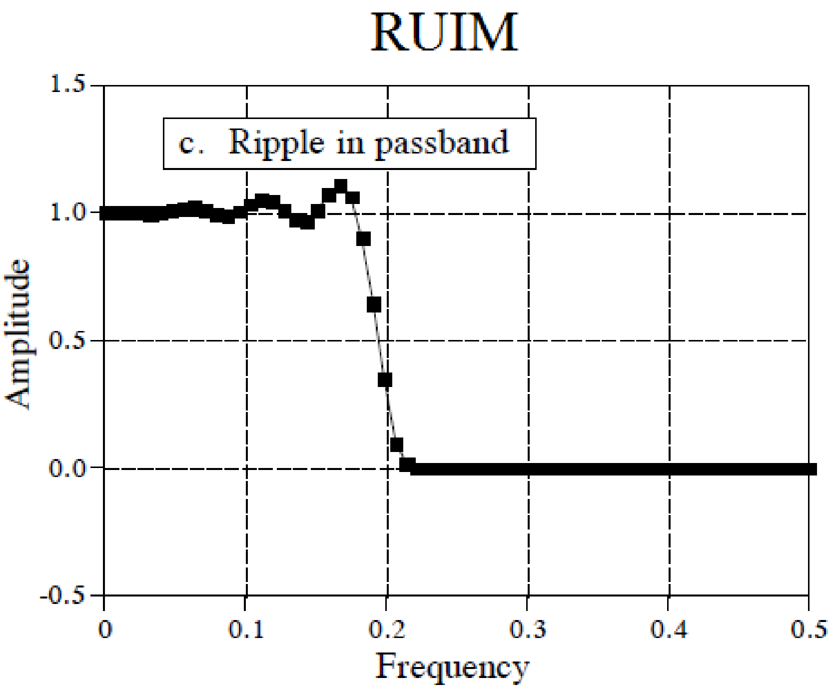


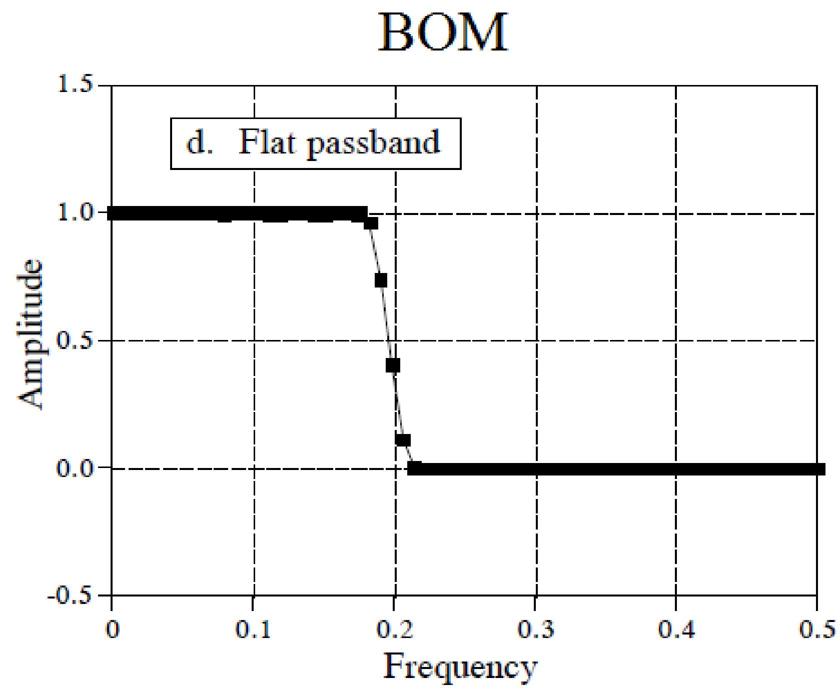
BOM



Filtros Digitais - Introdução

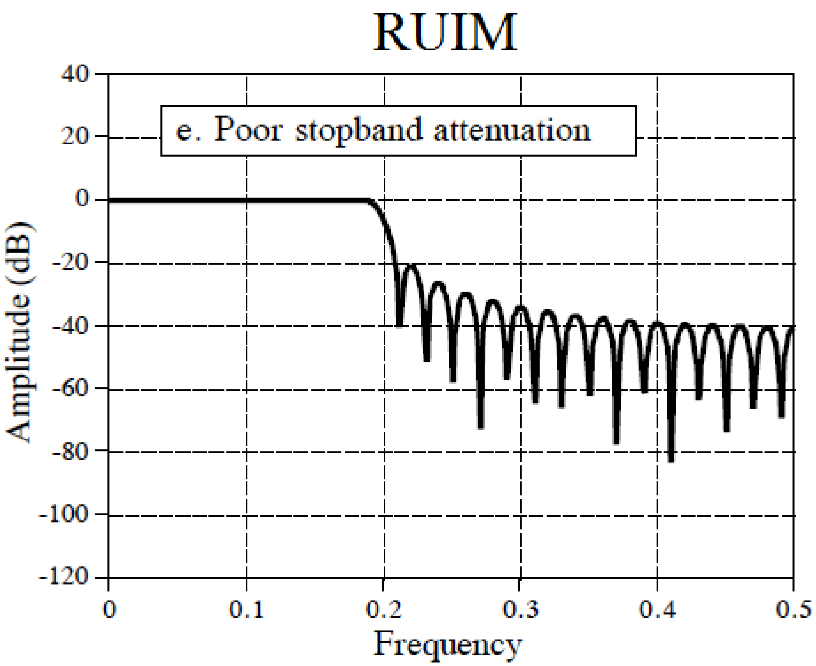
- Parâmetros no Domínio da Frequência
  - *pass band ripple*
    - Oscilação no final da banda passante

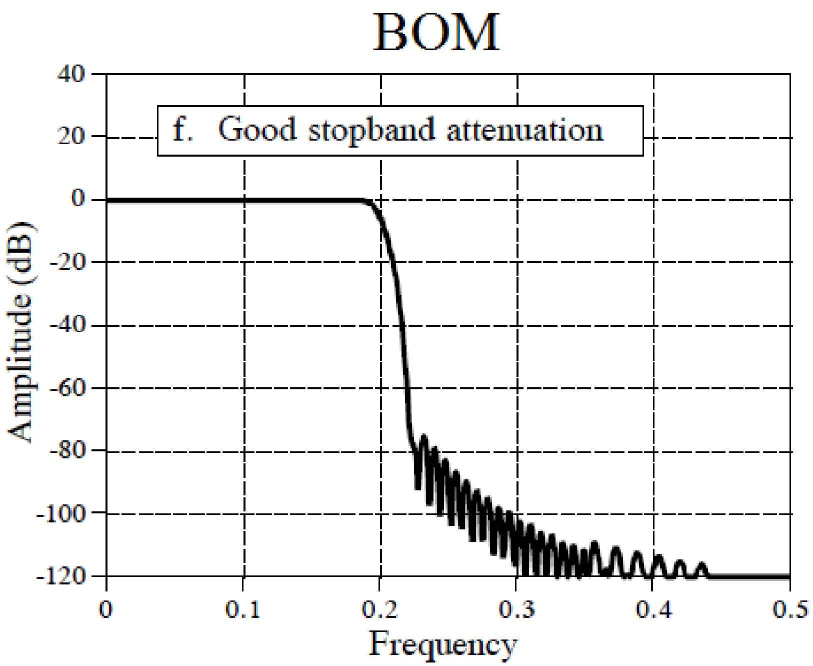




Filtros Digitais - Introdução

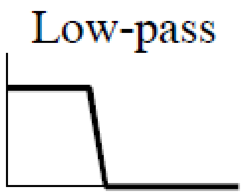
- Parâmetros no Domínio da Frequência
  - *stop band attenuation*
    - Grau de atenuação na banda de bloqueio

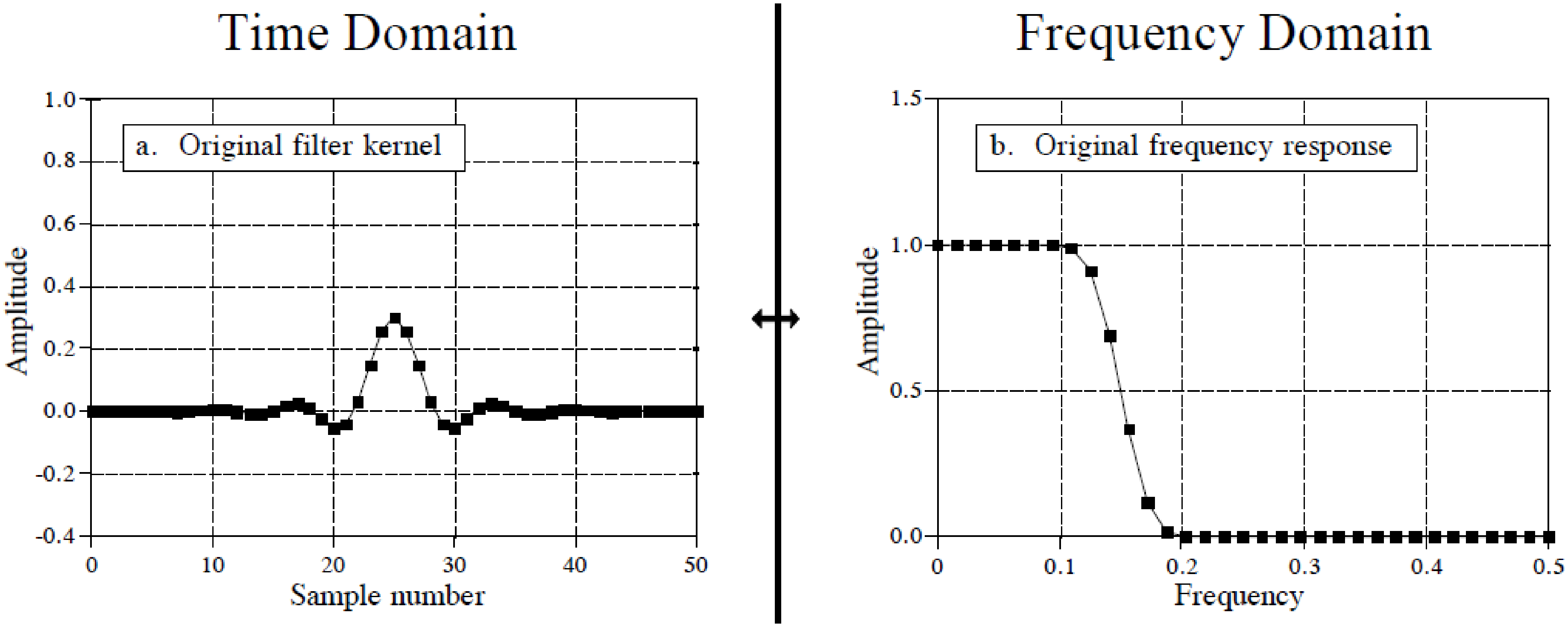




Filtros Digitais - Introdução

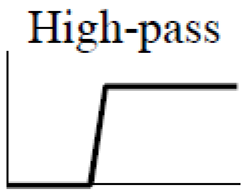
- FILTRO BASSA-BAIXAS

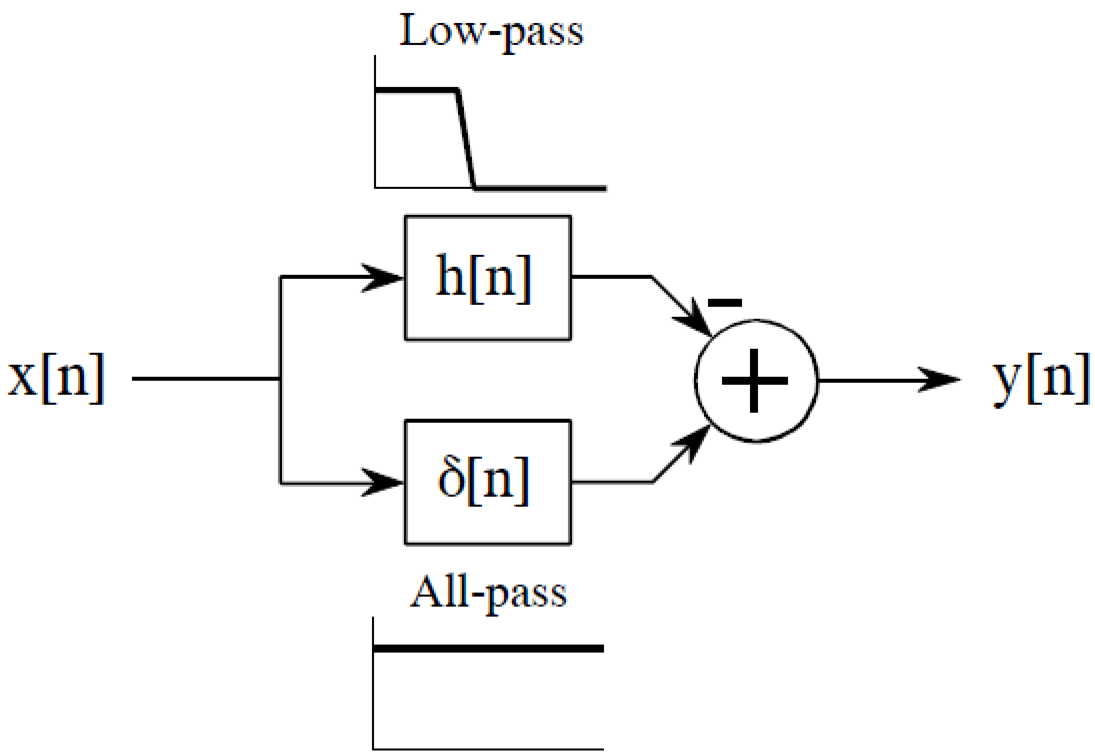


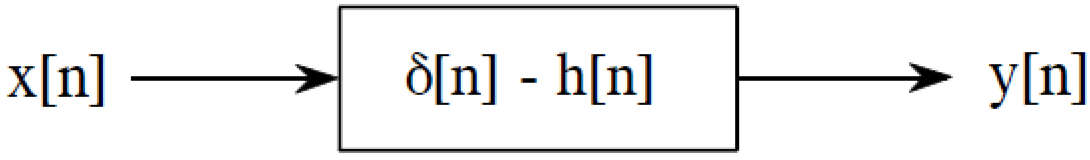


Filtros Digitais - Introdução

- FILTRO PASSA-ALTAS

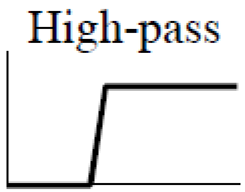


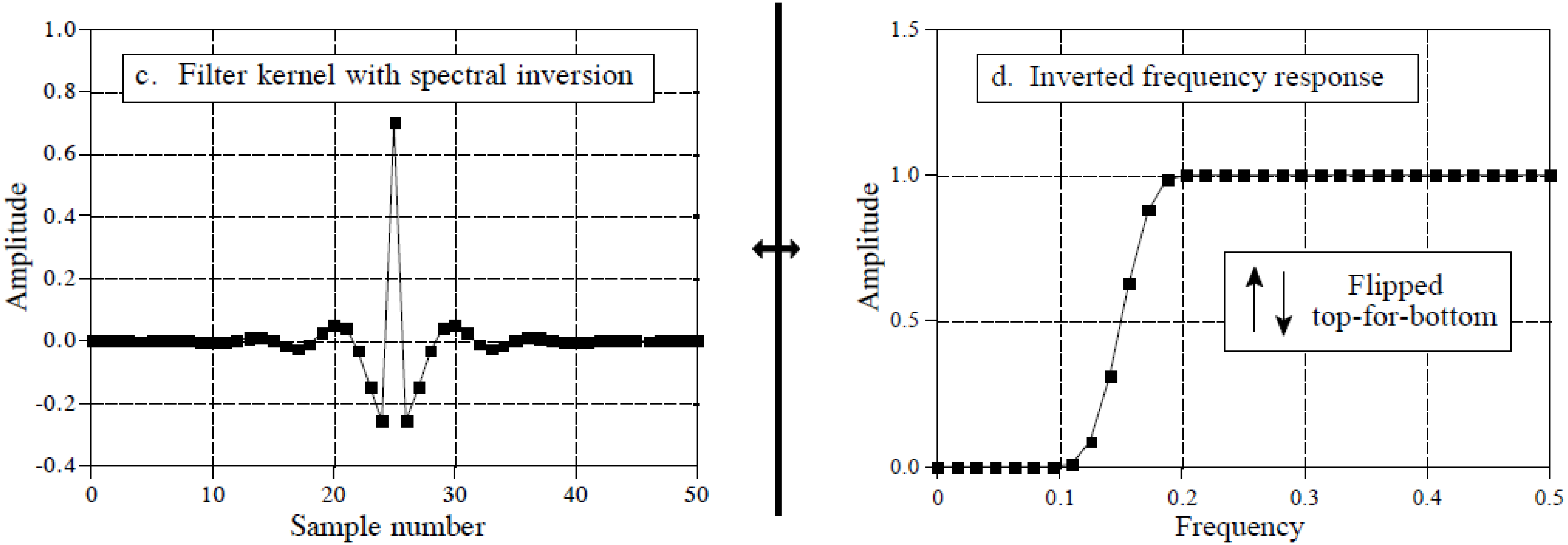




Filtros Digitais - Introdução

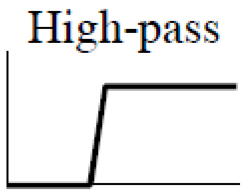
- FILTRO PASSA-ALTAS
  - por inversão de espectro
    - inverte a resposta ao impulso do filtro passa-baixa e soma 1 à amostra central

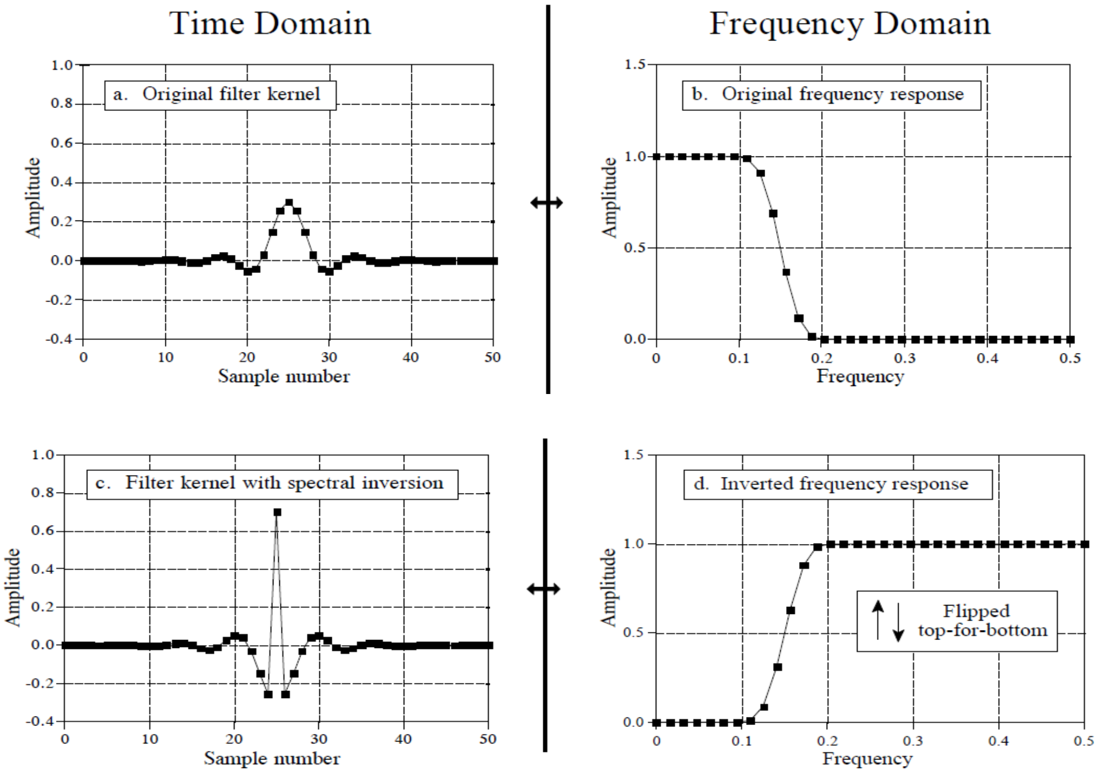




Filtros Digitais - Introdução

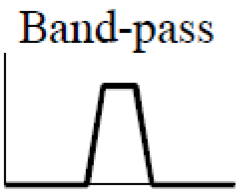
- FILTRO PASSA-ALTAS
  - por inversão de espectro
    - inverte a resposta ao impulso do filtro passa-baixa e soma 1 à amostra central

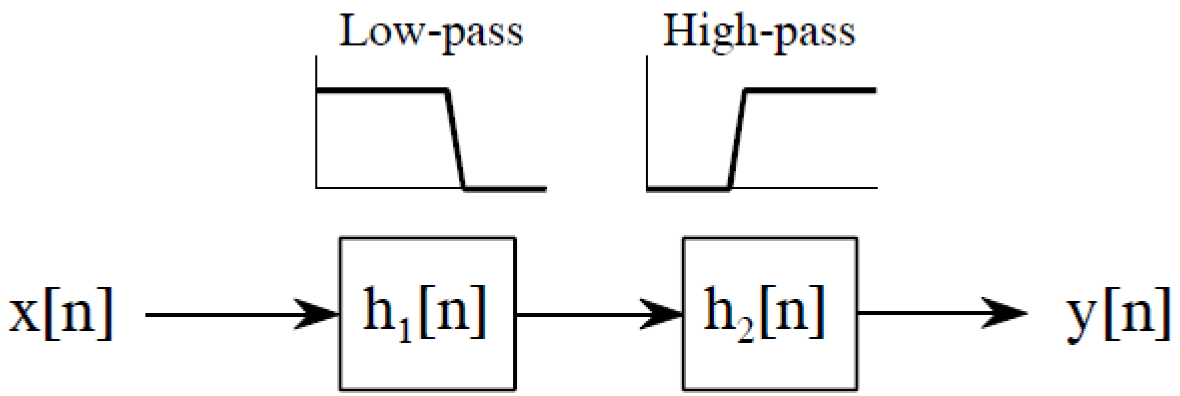


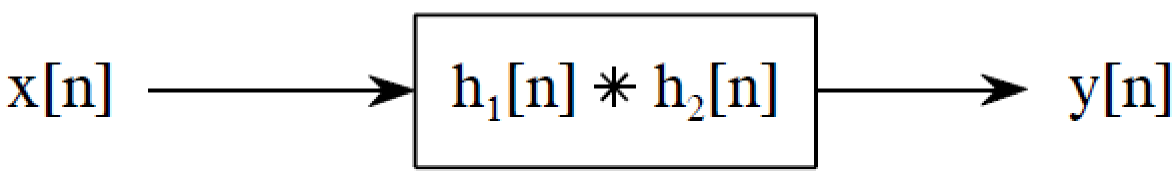


Filtros Digitais - Introdução

- FILTRO PASSA-FAIXA

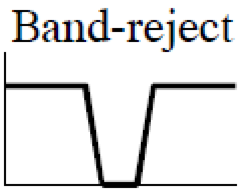


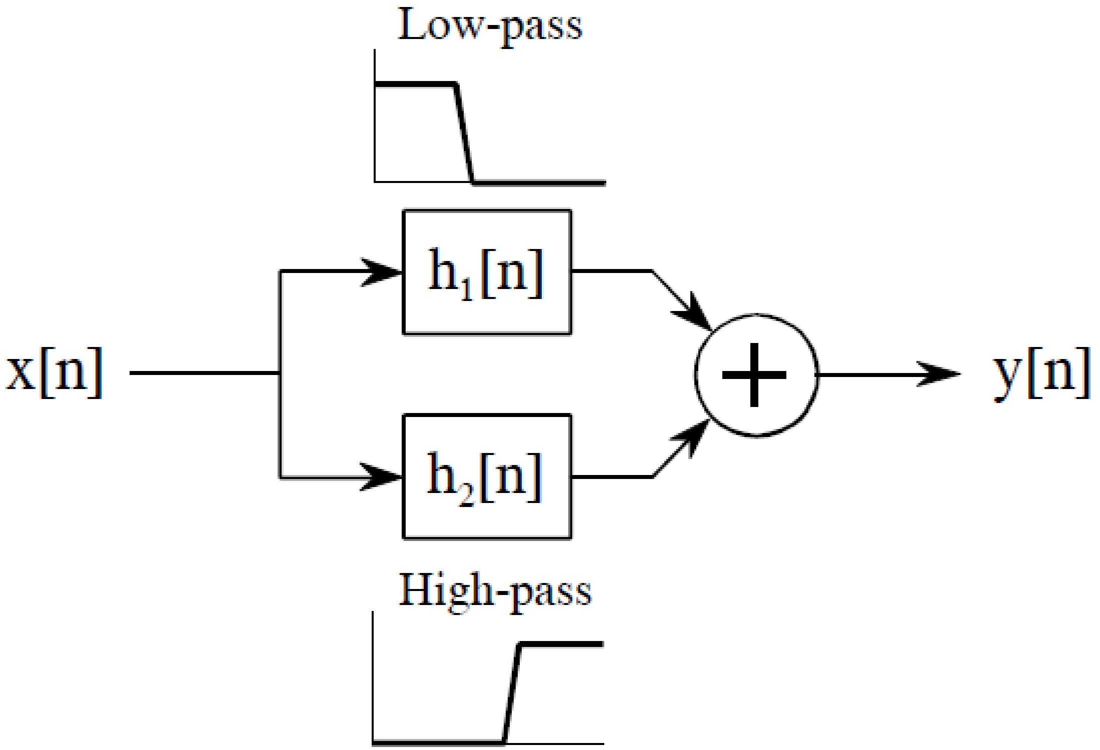


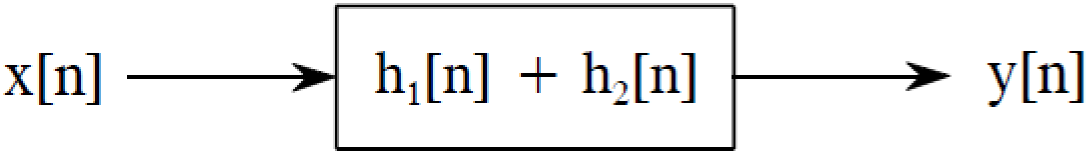


Filtros Digitais - Introdução

- FILTRO REJEITA-FAIXA







IMPLEMENTAÇÃO DE FILTROS

MOVING AVERAGE FILTER

Moving Average Filter (Filtro de Média Móvel)

- Fácil de ser implementado
- Ótimo para eliminar ruído, com boa resposta ao degrau
    - Bom para “suavizar” o sinal, para um mesmo tempo de transição
- Péssima resposta em frequência (filtro PB ruim)
    - Não separa frequências próximas

Moving Average Filter (Filtro de Média Móvel)

- Implementado calculando-se a média dos M últimos pontos do sinal de entrada

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

onde:

y[ ] é o sinal de saída
x[ ] é o sinal de entrada
...[i] é o i-ésimo ponto do sinal
M é o número de pontos da média

Moving Average Filter (Filtro de Média Móvel)

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

onde:

y[ ] é o sinal de saída
x[ ] é o sinal de entrada
...[i] é o i-ésimo ponto do sinal
M é o número de pontos da média

Exemplo: um filtro de 5 pontos tem a seguinte expressão:

$$y[80] = \frac{x[80] + x[81] + x[82] + x[83] + x[84]}{5}$$

Moving Average Filter (Filtro de Média Móvel)

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

onde:

y[ ] é o sinal de saída
x[ ] é o sinal de entrada
...[i] é o i-ésimo ponto do sinal
M é o número de pontos da média

Pode ser implementado também, com simetria em relação ao ponto do sinal de saída:

$$y[80] = \frac{x[78] + x[79] + x[80] + x[81] + x[82]}{5}$$

Neste caso,
M deve ser ímpar...

Moving Average Filter (Filtro de Média Móvel)

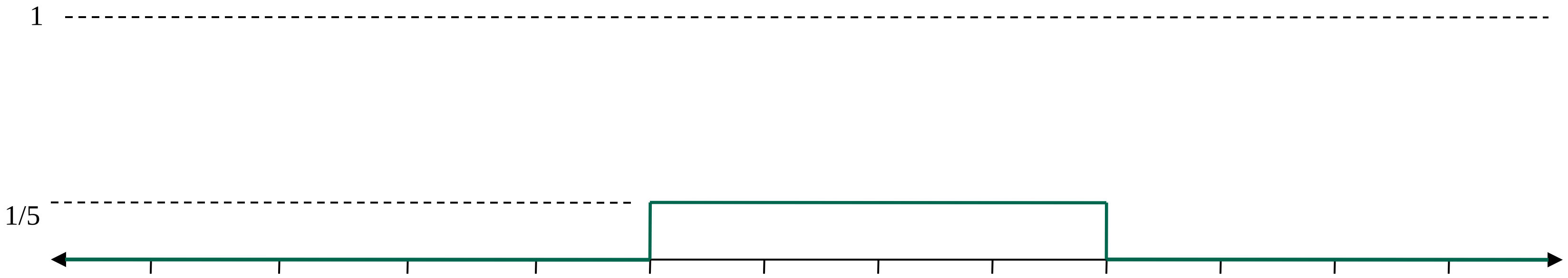
$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

Exemplo: um filtro de 5 pontos tem a seguinte expressão

$$y[80] = \frac{x[80] + x[81] + x[82] + x[83] + x[84]}{5}$$

Equivale à convolução do sinal x[ ] com o seguinte kernel:

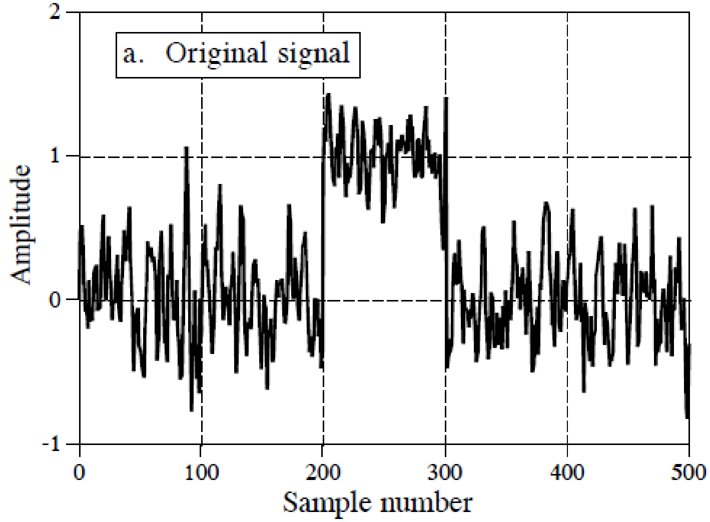
..., 0, 0, 0, 1/5, 1/5, 1/5, 1/5, 1/5, 0, 0, 0, ...

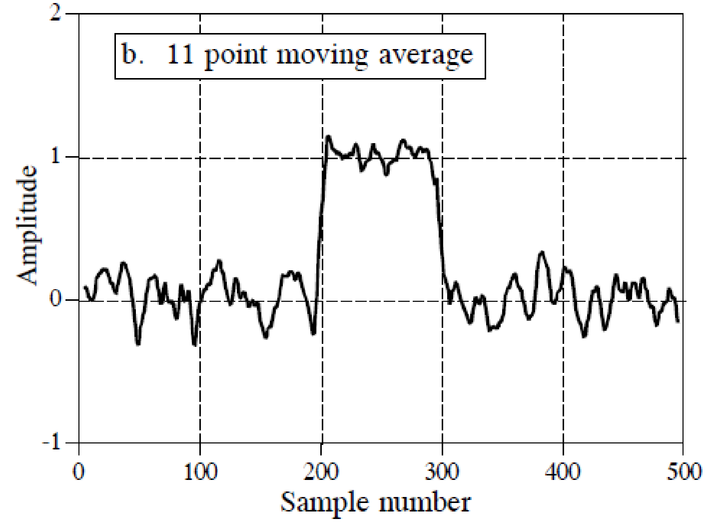


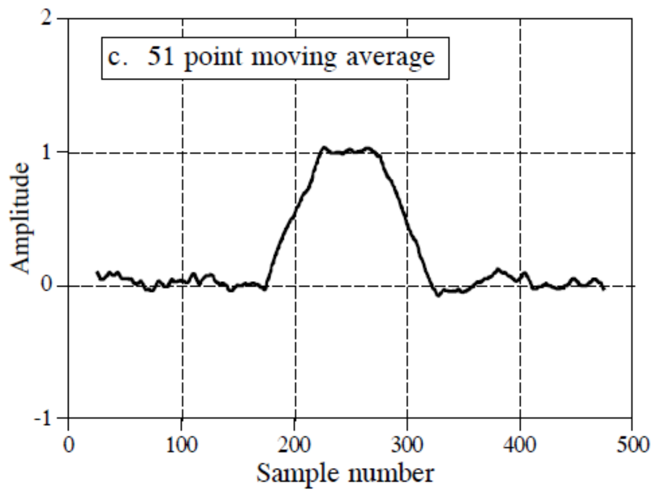
Moving Average Filter (Filtro de Média Móvel)

```
100 'MOVING AVERAGE FILTER
110 'This program filters 5000 samples with a 101 point moving
120 'average filter, resulting in 4900 samples of filtered data.
130 '
140 DIM X[4999]                          'X[ ] holds the input signal
150 DIM Y[4999]                          'Y[ ] holds the output signal
160 '
170 GOSUB XXXX                           'Mythical subroutine to load X[ ]
180 '
190 FOR I% = 50 TO 4949                  'Loop for each point in the output signal
200   Y[I%] = 0                          'Zero, so it can be used as an accumulator
210   FOR J% = -50 TO 50                 'Calculate the summation
220     Y[I%] = Y[I%] + X(I%+J%]
230   NEXT J%
240   Y[I%] = Y[I%]/101                  'Complete the average by dividing
250 NEXT I%
260 '
270 END
```

Moving Average Filter (Filtro de Média Móvel)







Moving Average Filter (Filtro de Média Móvel)

Implementação recursiva

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

Exemplo:

$$y[50] = x[47] + x[48] + x[49] + x[50] + x[51] + x[52] + x[53]$$

$$y[51] = x[48] + x[49] + x[50] + x[51] + x[52] + x[53] + x[54]$$

$$\Downarrow$$

$$y[51] = y[50] + x[54] - x[47]$$

$$\Downarrow$$

$$y[i] = y[i-1] + x[i+p] - x[i-q]$$

onde:

$p = (M-1)/2$

$q = p + 1$

Moving Average Filter (Filtro de Média Móvel)

```
100 'MOVING AVERAGE FILTER IMPLEMENTED BY RECURSION
110 'This program filters 5000 samples with a 101 point moving
120 'average filter, resulting in 4900 samples of filtered data.
130 'A double precision accumulator is used to prevent round-off drift.
140 '
150 DIM X[4999]                     'X[ ] holds the input signal
160 DIM Y[4999]                     'Y[ ] holds the output signal
170 DEFDBL  ACC                     'Define the variable ACC to be double precision
180 '
190 GOSUB XXXX                      'Mythical subroutine to load X[ ]
200 '
210 ACC = 0                         'Find Y[50] by averaging points X[0] to X[100]
220 FOR I% = 0 TO 100
230   ACC = ACC + X[I%]
240 NEXT I%
250 Y[[50] = ACC/101
260 '                               'Recursive moving average filter (Eq. 15-3)
270 FOR I% = 51 TO 4949
280   ACC = ACC + X[I%+50] - X[I%-51]
290   Y[I%] = ACC
300 NEXT I%
310 '
320 END
```

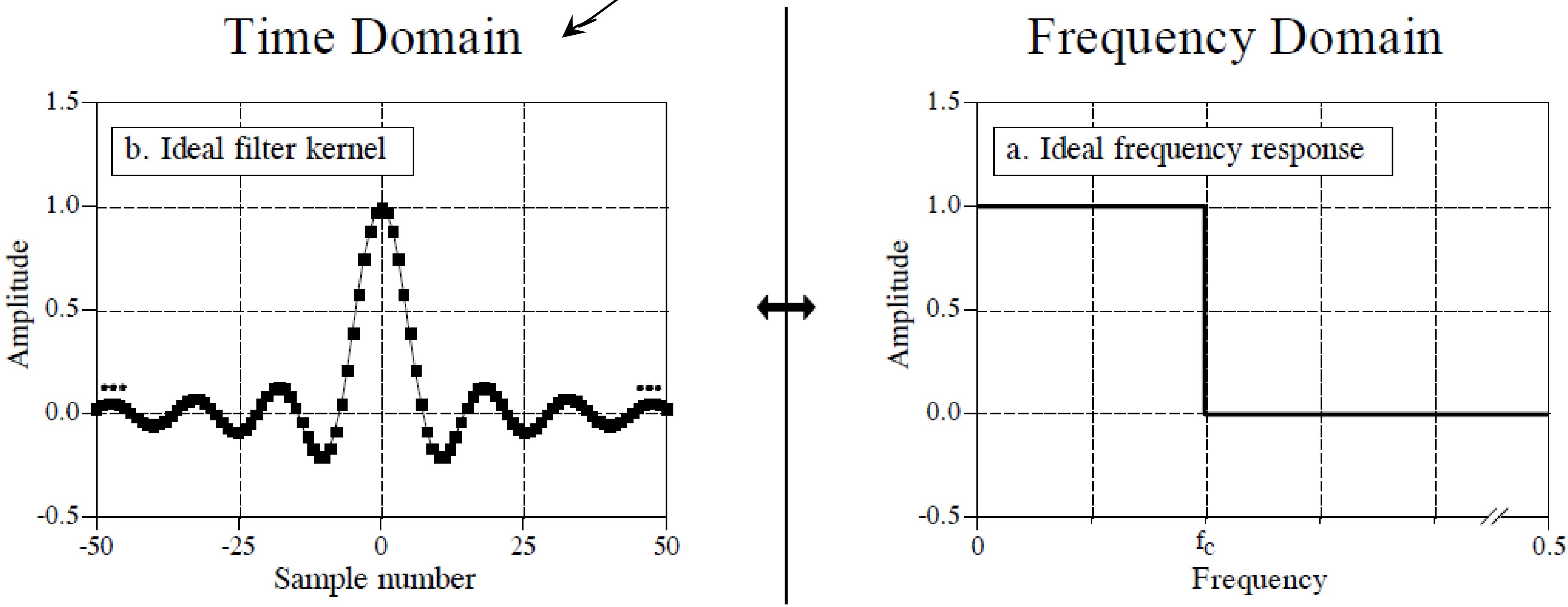
WINDOWED SINC FILTER

Windowed-Sinc Filter

Estratégia de formação:

- O filtro ideal
  - Resposta ao impulso: função sinc

$$h[i] = \frac{\sin(2\pi f_C i)}{i\pi}$$



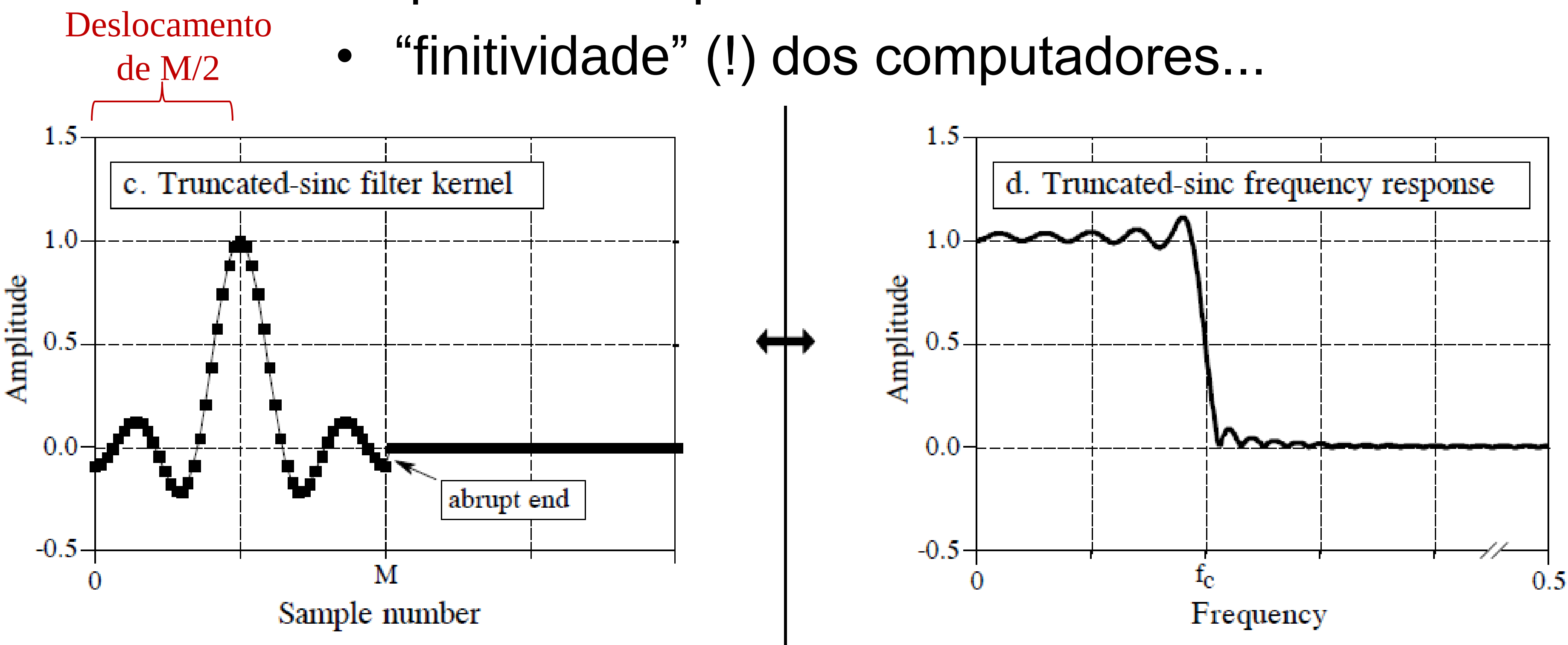
Simetria em relação à origem

Windowed-Sinc Filter

Estratégia de formação:

$$h[i] = \frac{\sin(2\pi f_c i)}{i\pi}$$

- O filtro real
  - Deslocamento de M/2 pontos, para trabalhar com valores positivos
  - Resposta ao impulso truncada
    - “finitividade” (!) dos computadores...

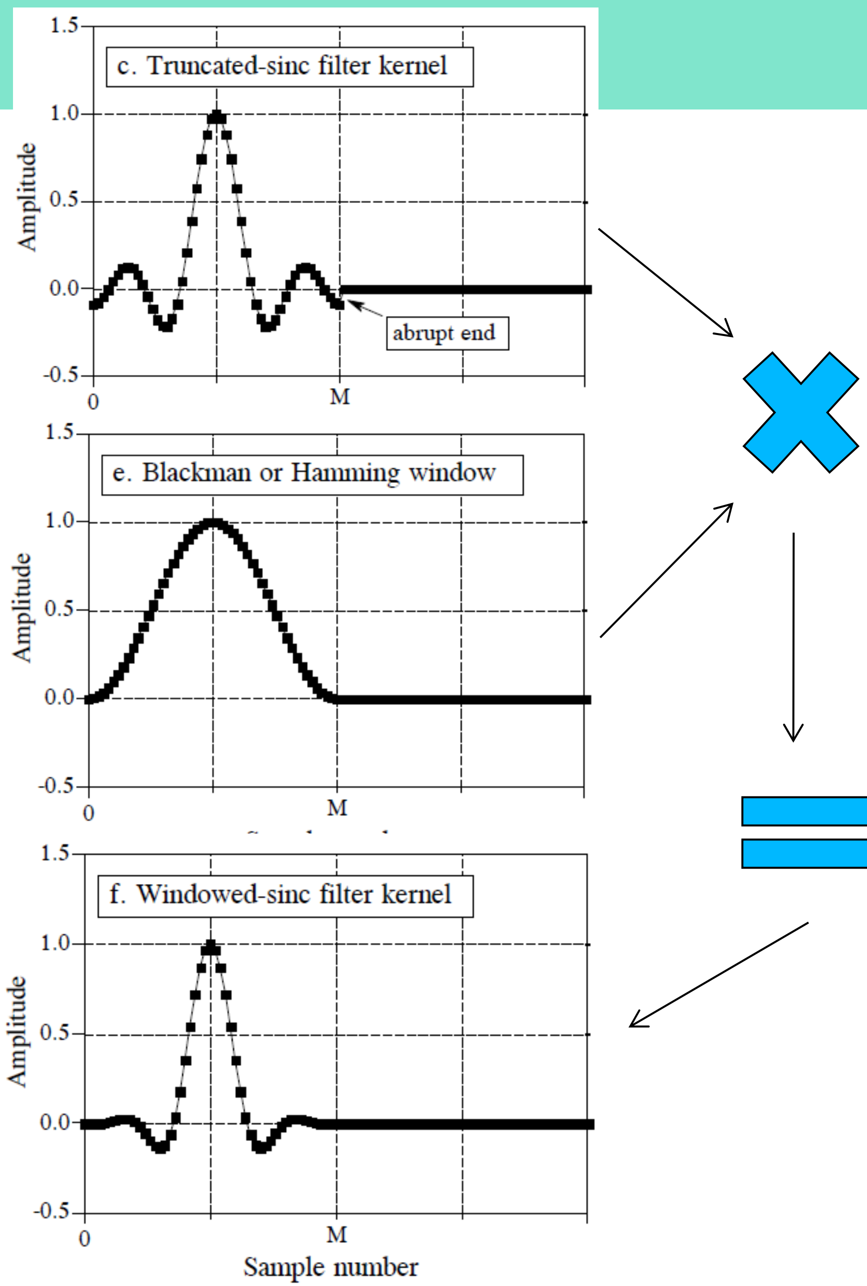


Windowed-Sinc Filter

Estratégia de formação:

- “Remediação”

“Janelamento”

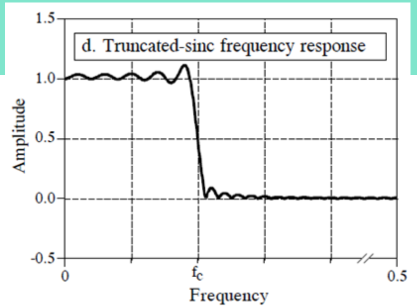


Windowed-Sinc Filter

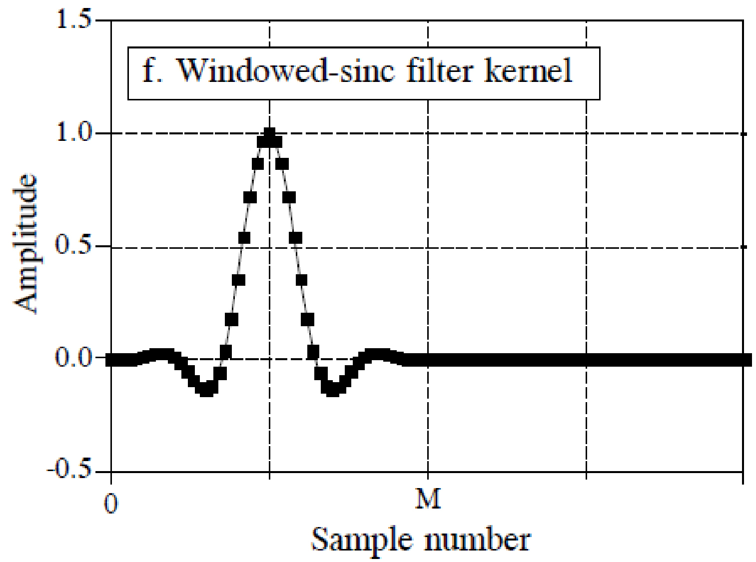
Estratégia de formação:

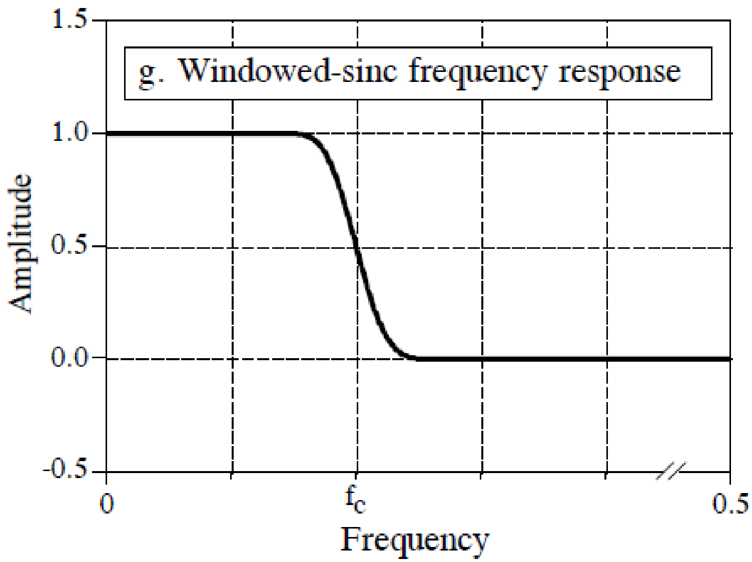
- “Remediação”

RUIM

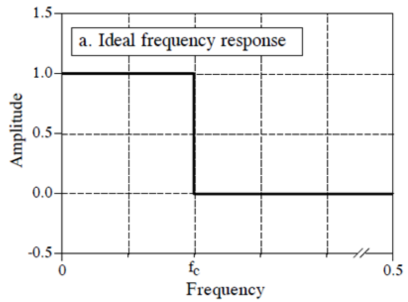


BOM





IDEAL



Windowed-Sinc Filter

Janelamentos:

- Janela Blackman

$$w[i] = 0.42 - 0.5\cos(2\pi i/M) + 0.08\cos(4\pi i/M)$$

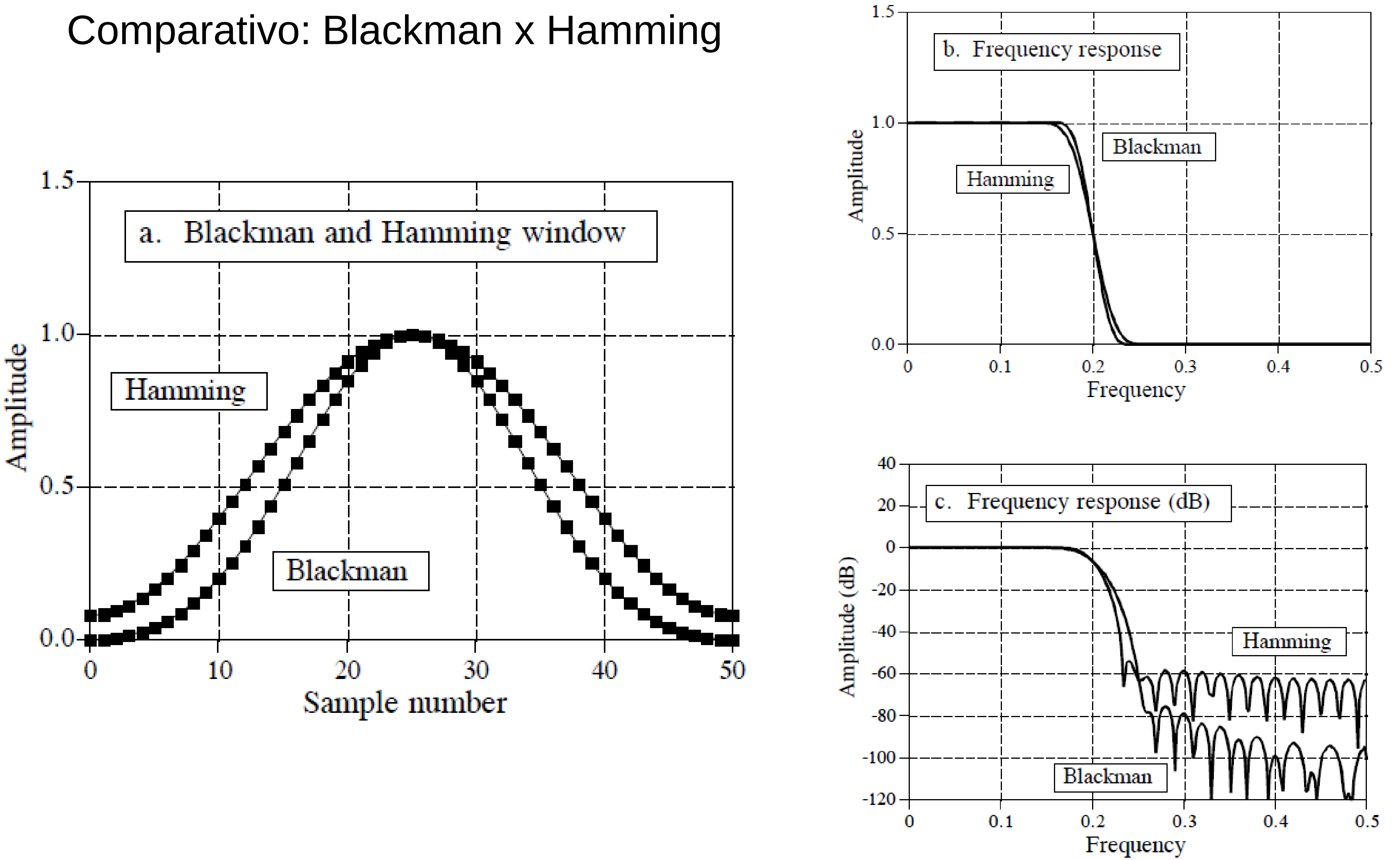
- Janela de Hamming

$$w[i] = 0.54 - 0.46\cos(2\pi i/M)$$

- Hanning, Retangular, de Bartlett, ...

Windowed-Sinc Filter

Comparativo: Blackman x Hamming



Windowed-Sinc Filter

Projeto do filtro:

- Parâmetros a definir
  - Frequência de corte (f_c)
    - Expressa como uma fração da taxa de amostragem (0 – 0,5f_s)
  - Tamanho do kernel do filtro (M)
    - Deve ser um número ímpar
    - O tamanho do kernel define aproximadamente a faixa de transição do filtro (BW):

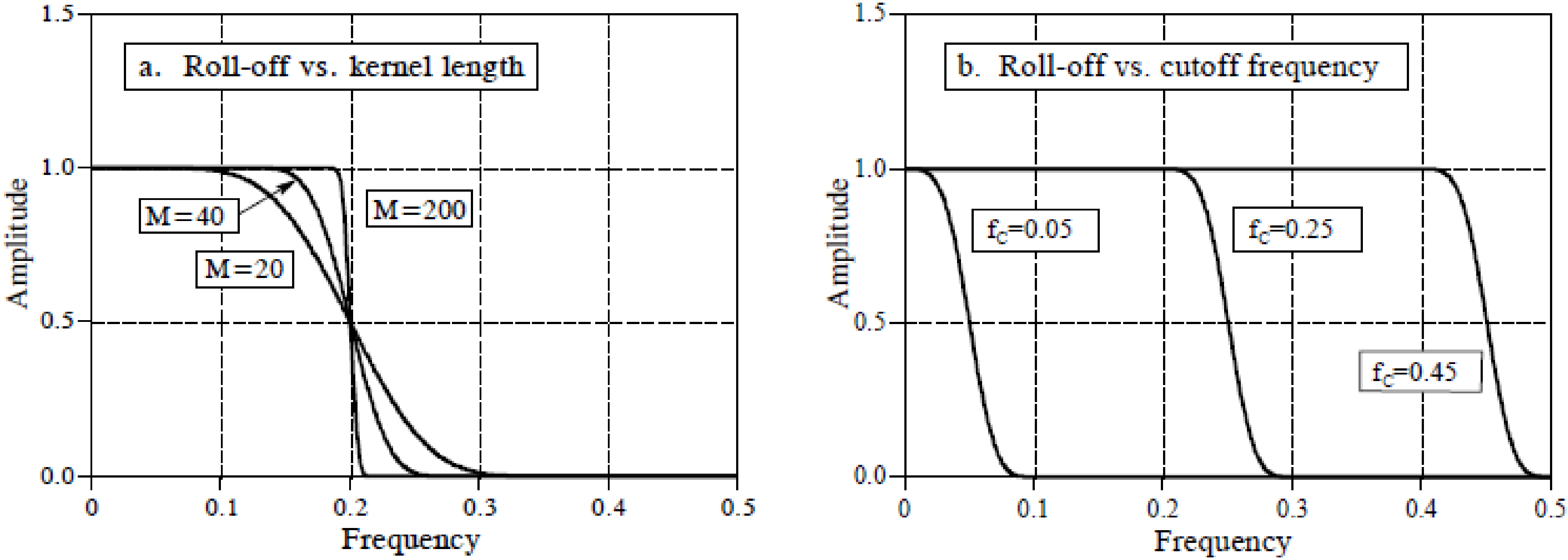
$$M \approx \frac{4}{BW}$$

    - BW também é expressa como uma fração da taxa de amostragem

Windowed-Sinc Filter

Projeto do filtro:

- Influência dos parâmetros na performance do filtro
  - M interfere no *roll-off*
  - *Roll-off* não sofre influência da frequência de corte



Windowed-Sinc Filter

Projeto do filtro:

- Expressão do filtro:

$$h[i] = \underbrace{K}_{\text{Fator de normalização}} \underbrace{\frac{\sin(2\pi f_c\,(\overbrace{i - M/2}^{\text{deslocamento}}))}{i - M/2}}_{\text{Função sinc}} \underbrace{\left[0.42 - 0.5\cos\left(\frac{2\pi i}{M}\right) + 0.08\cos\left(\frac{4\pi i}{M}\right)\right]}_{\text{Janela de Blackman}}$$

$$h[M/2] = 2\pi f_c K$$

Windowed-Sinc Filter

```
100 'LOW-PASS WINDOWED-SINC FILTER
110 'This program filters 5000 samples with a 101 point windowed-sinc filter,
120 'resulting in 4900 samples of filtered data.
130 '
140 DIM X[4999]                    'X[ ] holds the input signal
150 DIM Y[4999]                    'Y[ ] holds the output signal
160 DIM H[100]                     'H[ ] holds the filter kernel
170 '
180 PI = 3.14159265
190 FC = .14                       'Set the cutoff frequency (between 0 and 0.5)
200 M% = 100                       'Set filter length (101 points)
210 '
220 GOSUB XXXX                     'Mythical subroutine to load X[ ]
230 '
240 '                              'Calculate the low-pass filter kernel via Eq. 16-4
250 FOR I% = 0 TO 100
260   IF (I%-M%/2) = 0 THEN H[I%] = 2*PI*FC
270   IF (I%-M%/2) <> 0  THEN H[I%] = SIN(2*PI*FC * (I%-M%/2)) / (I%-M%/2)
280   H[I%] = H[I%] * (0.54 - 0.46*COS(2*PI*I%/M%) )
290 NEXT I%
300 '
310 SUM = 0                        'Normalize the low-pass filter kernel for
320 FOR I% = 0 TO 100              'unity gain at DC
330   SUM = SUM + H[I%]
340 NEXT I%
350 '
360 FOR I% = 0 TO 100
370   H[I%] = H[I%] / SUM
380 NEXT I%
390 '
400 FOR J% = 100 TO 4999           'Convolve the input signal & filter kernel
410   Y[J%] = 0
420   FOR I% = 0 TO 100
430     Y[J%] = Y[J%] + X[J%-I%] * H[I%]
440   NEXT I%
450 NEXT J%
460 '
470 END
```

Exemplo:
Se $f_s = 1000$ Hz
$f_c = 0.14 \times 1000$ Hz
$f_c = 140$ Hz

Cria a resposta ao impulso (kernel) do filtro

Convolui com o sinal (executa a filtragem em si)

Windowed-Sinc Filter

EXERCÍCIO

- Projetar um filtro (seu código no computador) que deve "eliminar" frequências abaixo de 70 Hz de um sinal digitalizado a uma taxa de amostragem de 500Hz, implementando um *windowed sinc filter,* utilizando o janelamento de Hamming. A faixa de transição (*BW*) deve ser de no máximo 20Hz.

Windowed-Sinc Filter

EXERCÍCIO

- Projetar um filtro (seu código no computador) que deve "eliminar" frequências abaixo de 70 Hz de um sinal digitalizado a uma taxa de amostragem de 500Hz, implementando um *windowed sinc filter,* utilizando o janelamento de Hamming. A faixa de transição (*BW*) deve ser de no máximo 20Hz.

- Filtro passa-altas
- $f_c = 70$ Hz ➔ $f_s = 70/500 = 0{,}14$
- $BW = 20$ Hz ➔ $f_s = 20/500 = 0{,}04$
- $M \approx 4/BW \approx 4/0{,}04 \approx 100$
- $M = 101$

Windowed-Sinc Filter

EXERCÍCIO

- Filtro **passa-baixas**, com janelamento Hamming:

$$h[i] = K\frac{\sin(2\pi f_c(i - M/2))}{i - M/2}\left[0.54 - 0.46\cos(2\pi i/M)\right]$$

Obs.: ignore K (fator de normalização), por enquanto

- para $\mathbf{f_c = 0,14}$ e $\boldsymbol{M} \mathbf{= 101}$

$$h[i] = K\frac{\sin(0.8796i - 44.4198)}{i - 50.5}\left[0.54 - 0.46\cos(0.0622i)\right]$$

(Criamos a resposta ao impulso do filtro passa-baixas com esta equação...)

Windowed-Sinc Filter

EXERCÍCIO

- Filtro **passa-baixas**, com **f_c = 0,14** e ***M* = 101**

$$h[i] = K \frac{\sin(0.8796i - 44.4198)}{i - 50.5}\left[0.54 - 0.46\cos(0.06221i)\right]$$

- *K* é obtido programaticamente, normalizando-se os valores do kernel do filtro
    - Obtém-se o somatório dos valores de todos os pontos
    - Divide-se cada ponto do kernel pelo somatório obtido

Windowed-Sinc Filter

EXERCÍCIO

- Filtro **passa-baixas**, com **f_c = 0,14** e ***M* = 101**

$$h[i] = K \frac{\sin(0.8796i - 44.4198)}{i - 50.5}\left[0.54 - 0.46\cos(0.06221i)\right]$$

- Mas queremos um filtro **passa-altas**
  - Podemos criá-lo, *invertendo espectralmente* o kernel do filtro passa-baixas acima. Como?
    - Invertendo o sinal de todos os pontos do kernel
    - Somando 1 ao ponto central

$$\left(x[n] \longrightarrow \boxed{\delta[n] - h[n]} \longrightarrow y[n] \right)$$

Windowed-Sinc Filter

EXERCÍCIO

- De posse da resposta ao impulso (kernel) do filtro passa-altas com $\mathbf{f_c = 0{,}14}$ e $\boldsymbol{M} \mathbf{= 101}$, convolui-se o kernel com o sinal...

```
for(j=100; j<2499; j++)
{
   y[j] = 0;
   for(i=0; i<100; i++){
      y[j] = y[j] + x[j-i]*h[i];
}
```