



DEEC

DEPARTAMENTO DE ENGENHARIA
ELECTROTÉCNICA E DE COMPUTADORES

TÉCNICO LISBOA

Electrónica Geral

José Gerald

Mestrado em Engenharia Aeroespacial
Licenciatura em Engenharia Física Tecnológica
Licenciatura em Engenharia Aeroespacial

MEAer: 1º ano, 1º semestre

LEFT: 3º ano, 1º semestre

LEAer: 3º ano, 1º semestre

2021/2022

Capítulo 7

Filtros Digitais



EG – Filtros Digitais

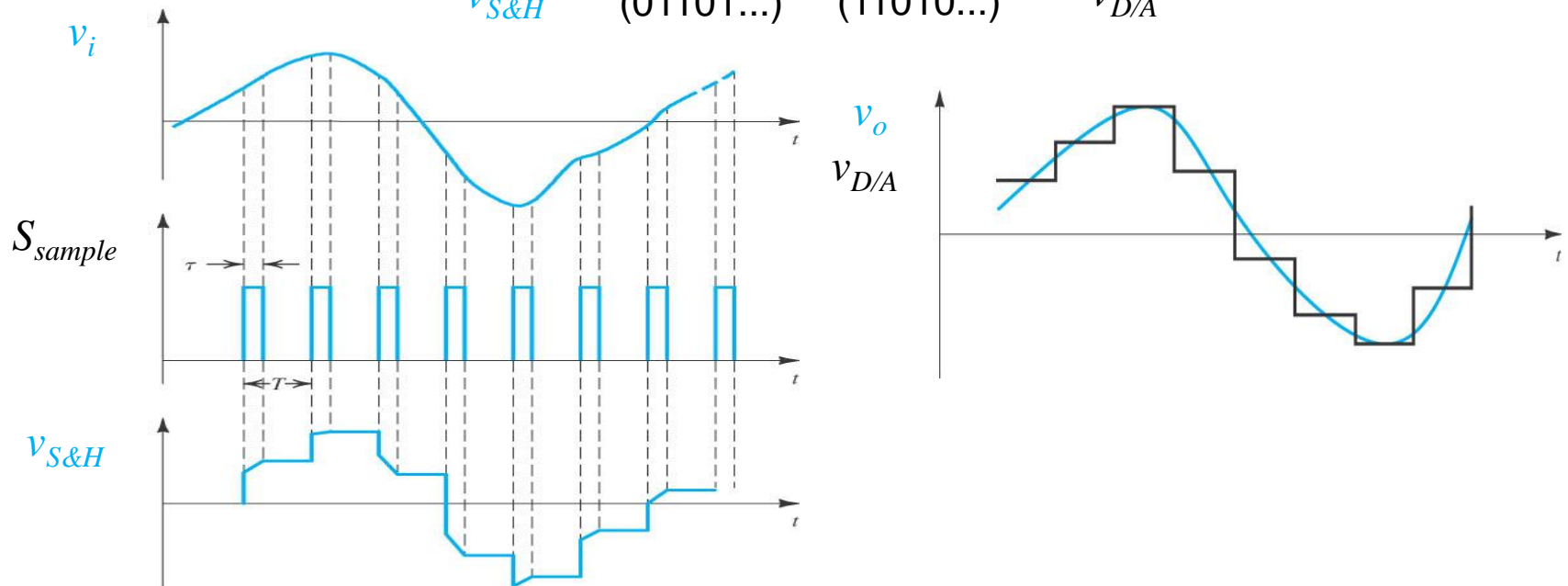
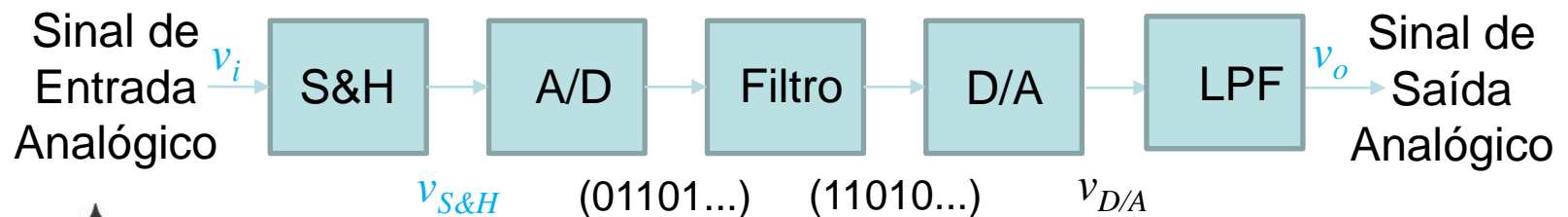
Introdução

Filtros digitais:

- Têm as vantagens do processamento digital de sinal (“Digital Signal Processing” – DSP): não variam com a temperatura, humidade, idade, etc.
- Usualmente implementados em software (via DSPs, FPGAs) o que lhes confere a capacidade de serem programáveis.
- Têm as desvantagens do processamento digital de sinal: erros de quantificação na conversão A/D e D/A e no processamento digital (precisão finita das amostras e dos registos), sinais de relógio (clock feedthrough), rapidez dos dispositivos de implementação (DSPs, etc.)
- Podem ser de 2 tipos: (i) IIR (“Infinite Impulse Response”), sempre recursivos e (ii) FIR (“Finite Impulse Response”), quase todos não recursivos.

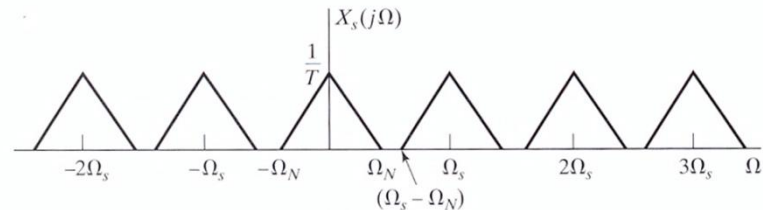
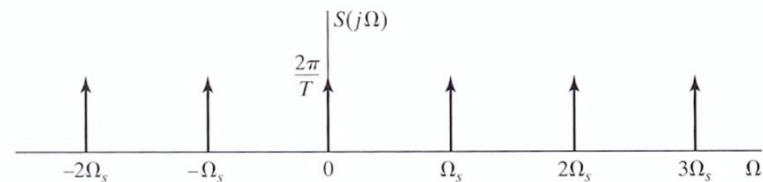
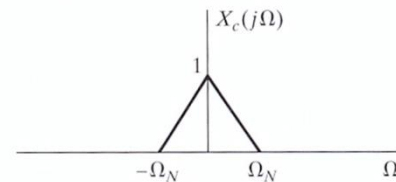
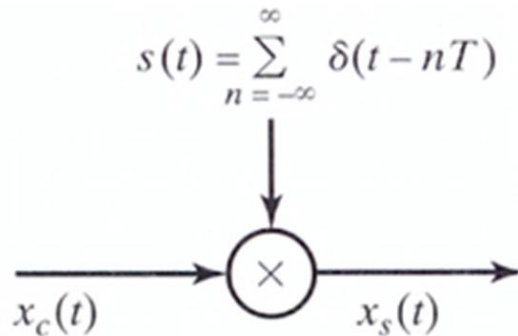
EG – Filtros Digitais

Amostragem de sinais



Amostragem de sinais (Cont.)

Na frequência:



No tempo:

$$x_s(t) = x_c(t)s(t) = \sum_{n=-\infty}^{+\infty} x_c(nt)\delta(t - nT)$$

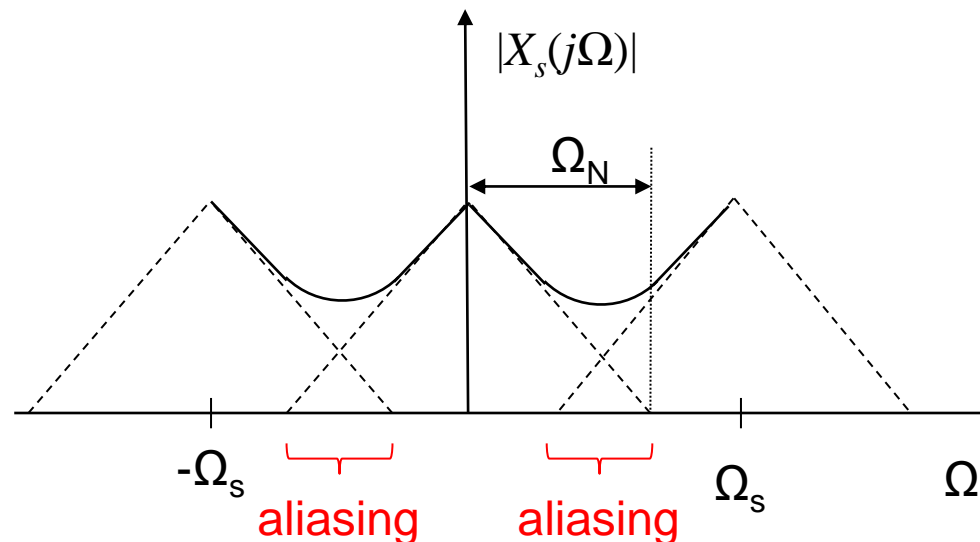


$$X_s(j\Omega) = \frac{1}{T} \sum_{n=-\infty}^{+\infty} X_c[j(\Omega - n\Omega_s)]$$

Amostragem de sinais (Cont.)

Teorema da Amostragem: Se a frequência de amostragem (Ω_s) for maior que 2 vezes a frequência máxima do sinal amostrado (Ω_N) é possível reconstituir o sinal original a partir do sinal amostrado usando um filtro passa-baixo.

Caso contrário há sobreposição de bandas (soma vectorial de sinais) designada por “aliasing”.





Amostragem de sinais (Cont.)

Amostragem Impulsiva e Transformada Z

Sinal ($x(t)$) amostrado por sequência de impulsos de Dirac ($\delta(t)$):

$$\begin{aligned}x_{\delta}(t) &= x(t) \delta(t) + x(t) \delta(t-T) + x(t) \delta(t-2T) + \dots = x(0) \delta(t) + x(T) \delta(t-T) + x(2T) \delta(t-2T) + \dots \\ &= x_0 \delta(t) + x_1 \delta(t-T) + x_2 \delta(t-2T) + \dots\end{aligned}$$

A transformada de Laplace é:

$$X_{\delta}(s) = x_0 + x_1 e^{-sT} + x_2 e^{-s2T} + \dots$$

Ou, introduzindo a variável $z = e^{sT}$ (em que $z^{-1} = e^{-sT}$ representa um atraso T)

$$X(z) = x_0 + x_1 z^{-1} + x_2 z^{-2} + \dots = \sum_{n=0}^{+\infty} x_n z^{-n}$$

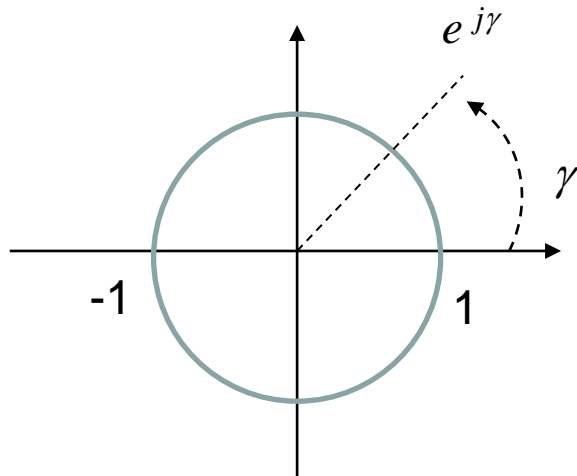
Que representa a **Transformada Z** de x_n

Amostragem de sinais (Cont.)

Resposta em Frequência de Sistemas Amostrados

$$T(s)\big|_{s=j\omega} = T(z)\big|_{z=e^{j\omega T}} = T(z)\big|_{z=e^{j\gamma}}$$

$\gamma = \omega T$ é designada frequência digital e vem em rad



Resposta em frequência periódica!

Amostragem de sinais (Cont.)

Equação de Recorrência e Função de Sistema

Considerando o filtro digital linear, invariante no tempo e causal, podemos representar o seu processamento de sinal no tempo, ou seja, relacionar a saída y_k com a entrada x_k da forma:

$$y_n + \sum_{k=1}^N b_k y_{n-k} = \sum_{k=0}^M a_k x_{n-k} \quad \text{Equação de Recorrência}$$

Aplicando a Transformada Z a ambos os membros da equação

$$Y(z) + Y(z) \sum_{k=1}^N b_k z^{-k} = X(z) \sum_{k=0}^M a_k z^{-k} \quad \text{ou seja} \quad Y(z) \left(1 + \sum_{k=1}^N b_k z^{-k}\right) = X(z) \sum_{k=0}^M a_k z^{-k}$$

$$\Rightarrow T(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M a_k z^{-k}}{1 + \sum_{k=1}^N b_k z^{-k}} \quad \text{Função de Sistema}$$

Todos $b_k=0 \Rightarrow \text{FIR}$

Algum $b_k \neq 0 \Rightarrow \text{IIR}$

EG – Filtros Digitais

Amostragem de sinais (Cont.)

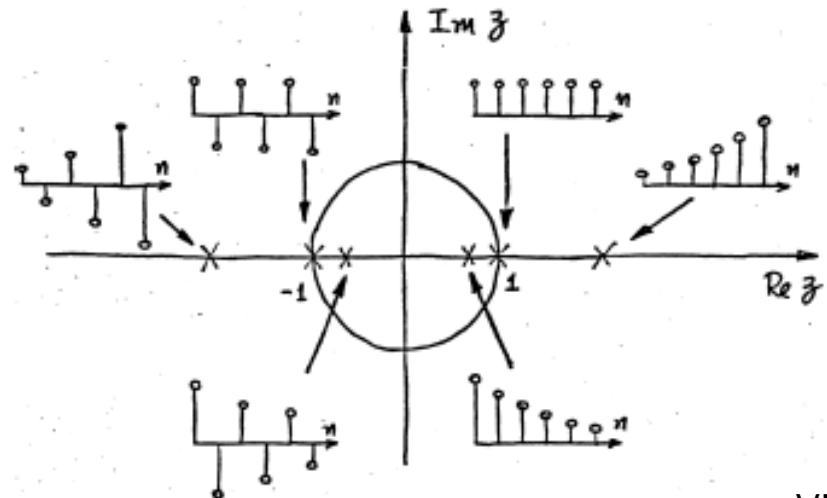
Função de Sistema e Estabilidade

$$T(z) = \frac{\sum_{k=0}^M a_k z^{-k}}{1 + \sum_{k=1}^N b_k z^{-k}} = \alpha \frac{\prod_{k=1}^M (1 - c_k z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})} \left\{ \begin{array}{l} (1 - c_k z^{-1}) \Rightarrow \text{Zero em } z=c_k \text{ e pólo em zero} \\ (1 - d_k z^{-1}) \Rightarrow \text{Pólo em } z=d_k \text{ e zero em zero} \end{array} \right.$$

Sistema Estável \Rightarrow pólos têm que estar no interior da circunferência unitária.

Exp: $T(z) = \frac{1}{(1 - az^{-1})} \rightarrow h_n = a^n u_n$

Estabilidade $\Rightarrow \sum_{n=-\infty}^{\infty} |h_n| < \infty$



Amostragem de sinais (Cont.)

Exemplo de Filtro IIR vs. FIR

IIR

Lowpass 2nd-order
Inverse-Chebyshev

$$T(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2}}$$

$$F_s = 20\text{kHz}$$

$$b_{-3\text{dB}} = 165\text{Hz}$$

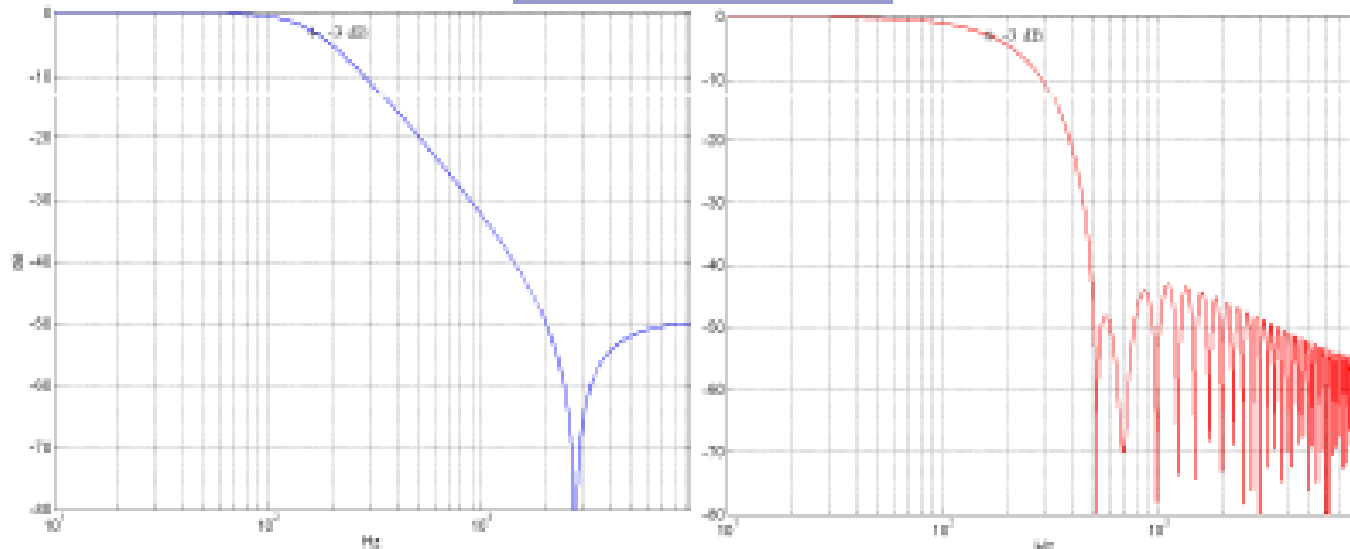
FIR

Lowpass 80th-order
Linear phase

$$T(z) = \sum_{k=0}^{80} h(k) z^{-k}$$

81 coefficients

Amplitude
Frequency Response



Filtros IIR (“Infinite Impulse Response”)

Introdução

- Maior selectividade ➡ ordens reduzidas (usualmente < 6)
- Desenhados a partir dos filtros analógicos
- Sempre recursivos
- Podem ser instáveis
- Fase não-linear

Eq. de Recorrência: $y_n = \sum_{k=0}^M a_k x_{n-k} - \sum_{k=1}^N b_k y_{n-k}$

Função de Sistema: $T(z) = \frac{\sum_{k=0}^M a_k z^{-k}}{1 + \sum_{k=1}^N b_k z^{-k}}$

Obtenção de $T(z)$

Ideal seria obter $T(s)$ e fazer $z=e^{sT}$, ou seja, $s=(1/T)\ln(z)$

➡ **função não racional em z !**

Solução: usar outras transformações $s \rightarrow z$

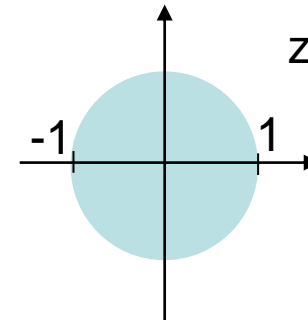
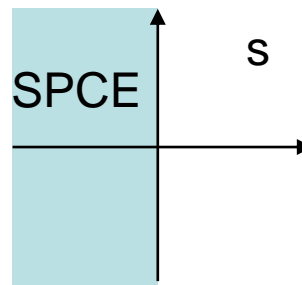
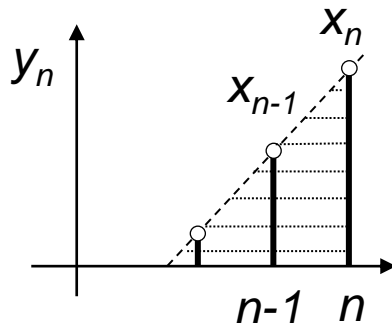
Filtros IIR (Cont.)

Transformação Bilinear

- Baseia-se na integração trapezoidal
- Aplicada ao filtro analógico mais elementar, integrador sem perdas de 1ª ordem, $T(s)=1/s$ resulta:

$$y_n = y_{n-1} + \frac{T}{2}(x_{n-1} + x_n) \quad \rightarrow \quad \frac{Y(z)}{X(z)} = \frac{T}{2} \frac{1+z^{-1}}{1-z^{-1}} = \frac{1}{s} \quad \rightarrow \quad \boxed{s = \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}}$$

Transformada Z



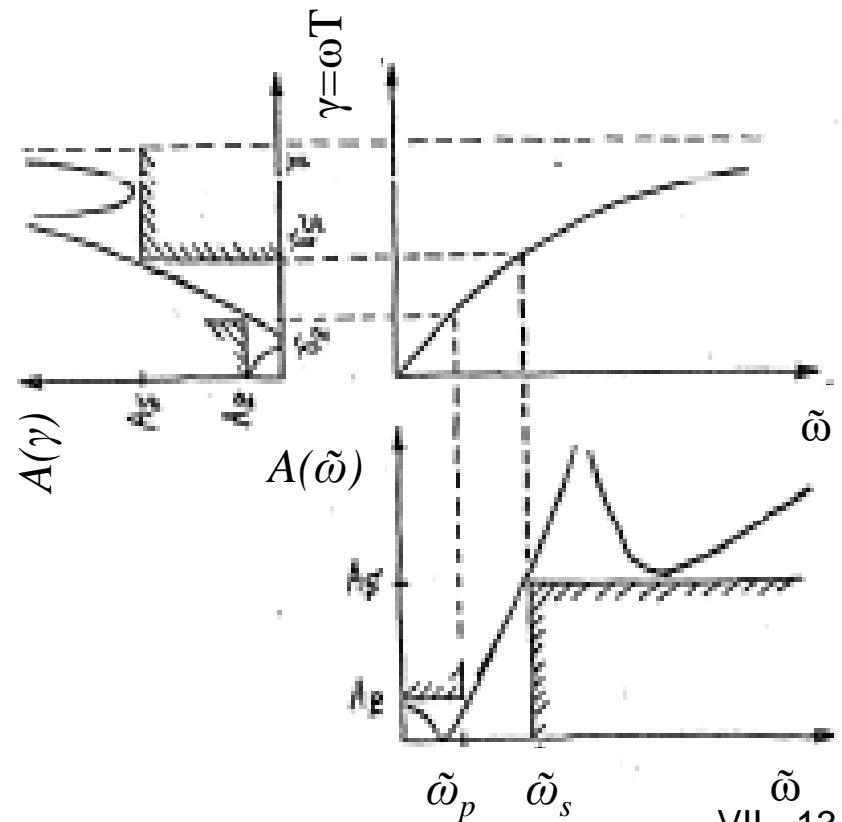
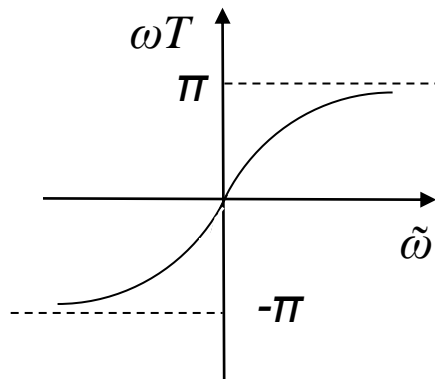
EG – Filtros Digitais

Filtros IIR (Cont.)

Deformação na Frequência da Transformação Bilinear

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \rightarrow \begin{cases} \tilde{\omega} = \frac{2}{T} \tan\left(\frac{\omega T}{2}\right) \\ \omega = \frac{2}{T} \tan^{-1}\left(\frac{\tilde{\omega} T}{2}\right) \end{cases}$$

$s = j\tilde{\omega}$ $z = e^{j\omega T}$



Filtros IIR (Cont.)

Exp:

$$T(s) = \frac{as + b}{s + c} \quad \xrightarrow{\quad} \quad T(z) = \frac{a' + b'z^{-1}}{1 - c'z^{-1}}$$

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$$

$$\left\{ \begin{array}{l} a' = \frac{(a\frac{2}{T} + b)}{(\frac{2}{T} + c)} \\ b' = \frac{(b - \frac{2}{T}a)}{(\frac{2}{T} + c)} \\ c' = \frac{(\frac{2}{T} - c)}{(\frac{2}{T} + c)} \end{array} \right.$$

Resposta de Amplitude:

$$|T(e^{j\omega T})| = \left| \frac{a' + b'e^{-j\omega T}}{1 - c'e^{-j\omega T}} \right| = \left| \frac{a' + b'\cos(\omega T) - jb'\sin(\omega T)}{1 - c'\cos(\omega T) + jc'\sin(\omega T)} \right| = \sqrt{\frac{[a' + b'\cos(\omega T)]^2 + [b'\sin(\omega T)]^2}{[1 - c'\cos(\omega T)]^2 + [c'\sin(\omega T)]^2}}$$

Resposta de Fase:

$$\arg\{T(e^{j\omega T})\} = \arg\left\{\frac{a' + b'e^{-j\omega T}}{1 - c'e^{-j\omega T}}\right\} = -\arctg\left\{\frac{b'\sin(\omega T)}{a' + b'\cos(\omega T)}\right\} - \arctg\left\{\frac{c'\sin(\omega T)}{1 - c'\cos(\omega T)}\right\}$$

Filtros IIR (Cont.)

Realização de Filtros IIR

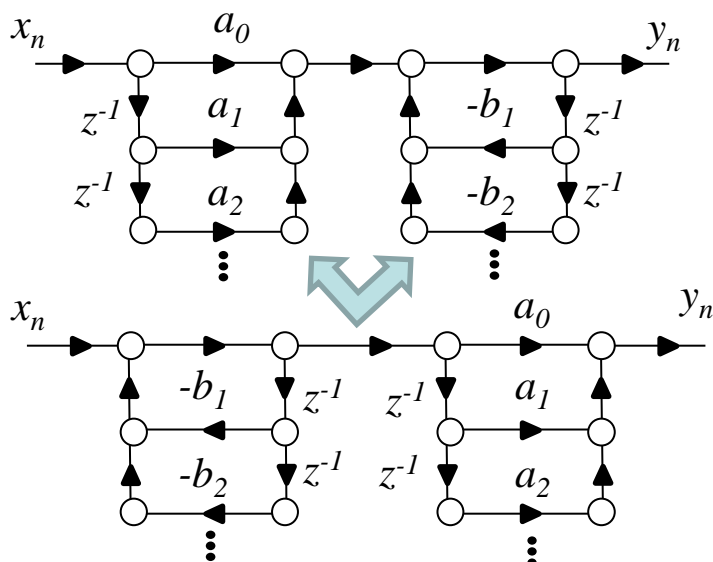
- 1 – DFS (3 operações apenas: adição, multiplicação e atraso)
- 2 – Optimização de processamento
- 3 – Implementação na tecnologia desejada

Diagrama de Fluxo de Sinal (DFS)

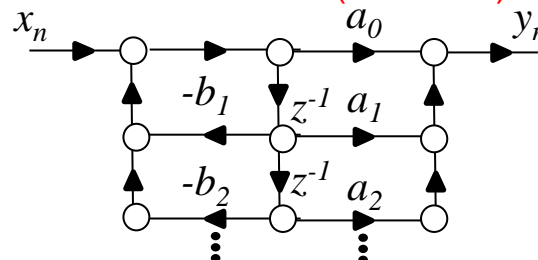
$$T(z) = \frac{\sum_{k=0}^M a_k z^{-k}}{1 + \sum_{k=1}^N b_k z^{-k}}$$

$$y_n = \sum_{k=0}^M a_k x_{n-k} - \sum_{k=1}^N b_k y_{n-k}$$

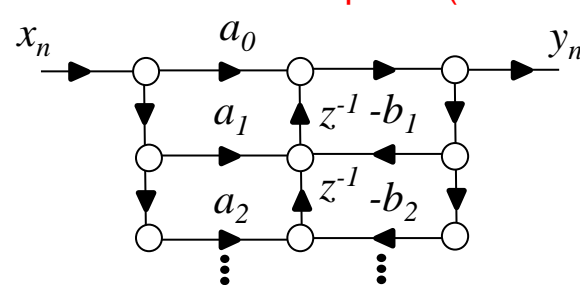
Forma Directa I



Forma Directa II (canónica)



Forma Directa II Transposta (canónica)



Filtros FIR (“Finite Impulse Response”)

Introdução

- Podem ser desenhados para terem fase linear (tipo mais usado)
- Sempre estáveis
- Quase sempre não recursivos
- Ordem elevada (usualmente >30) pois têm baixa selectividade
- Desenho complexo (aconselhável o uso de meios computacionais)

Eq. de Recorrência: $y_n = \sum_{k=0}^{N-1} h_k x_{n-k}$

Função de Sistema: $T(z) = \sum_{k=0}^{N-1} h_k z^{-k}$

Obtenção de $T(z)$

Método mais usado é o método das janelas:

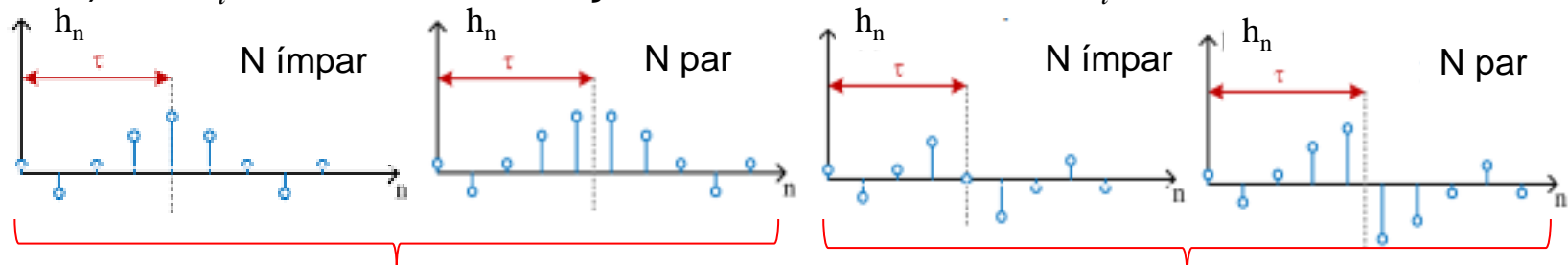
- 1) Trunca-se a resposta impulsional h_n com uma janela (temporal) finita;
 - 1a) Faz-se uma translacção dos coeficientes no tempo;
- 2) Aplica-se a Transformada Z à série resultante.

Filtros FIR (Cont.)

Fase Linear

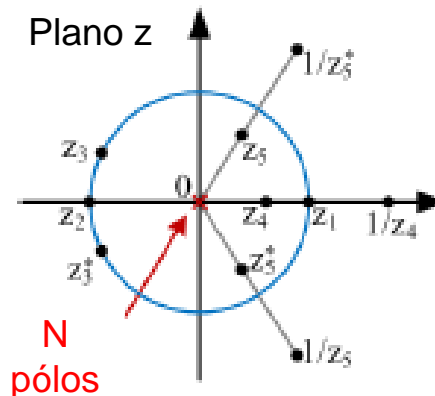
Há duas propriedades nos filtros FIR de fase linear:

- 1) Simetria (ou anti-simetria) na resposta impulsional
- 2) Se z_i é um zero da função de sistema, então $1/z_i^*$ também é.



Simetria

Anti-Simetria



Filtros FIR (Cont.)

Fase Linear

Verificação de condição necessária e suficiente:


$$T(e^{j\gamma}) = \sum_{n=0}^{N-1} h_n e^{-jn\gamma}$$

N ímpar
$$T(e^{j\gamma}) = \sum_{n=0}^{(N-3)/2} h_n \left[e^{-jn\gamma} + e^{-j(N-1-n)\gamma} \right] + h_{\frac{N-1}{2}} e^{-j\frac{N-1}{2}\gamma}$$

$$= e^{-j\frac{N-1}{2}\gamma} \left\{ h_{\frac{N-1}{2}} + \sum_{n=0}^{(N-3)/2} 2h_n \cos \left[\left(n - \frac{N-1}{2} \right) \gamma \right] \right\}$$

N par
$$T(e^{j\gamma}) = \sum_{n=0}^{\frac{N-1}{2}} h_n \left[e^{-jn\gamma} + e^{-j(N-1-n)\gamma} \right]$$

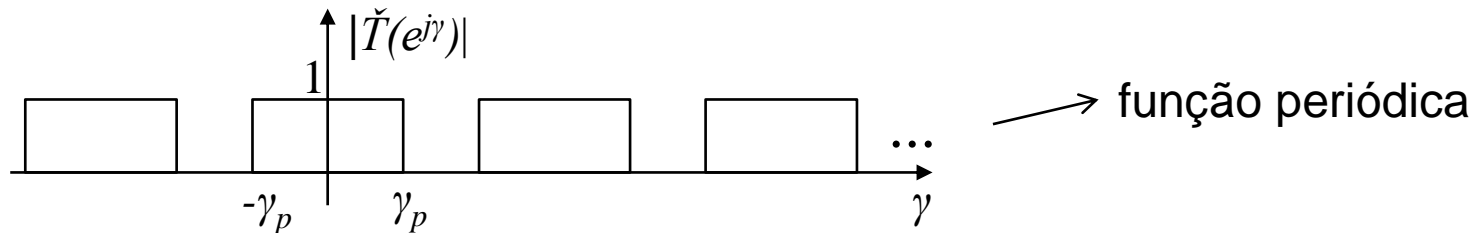
$$= e^{-j\frac{N-1}{2}\gamma} \left\{ \sum_{n=0}^{\frac{N-1}{2}} 2h_n \cos \left[\left(n - \frac{N-1}{2} \right) \gamma \right] \right\}$$

A fase (linear) vem: $\phi(\gamma) = -\frac{N-1}{2}\gamma$  O atraso (constante) vem: $\tau(\gamma) = \frac{N-1}{2}$

Filtros FIR (Cont.)

Método das Janelas (“Windows”)

Resposta ideal (por exp LP) desejada



$$\bar{T}(e^{j\gamma}) = \sum_{n=-\infty}^{\infty} \hat{h}_n e^{-j\gamma n} = \text{série de Fourier com coeficientes } \hat{h}_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} \bar{T}(e^{j\gamma}) e^{j\gamma n} d\gamma$$

Resposta impulsional ideal, não causal, de duração infinita!

Para obter o filtro FIR trunca-se \hat{h}_n , ou seja, a resposta do filtro fica $h_n = \hat{h}_n W_n$

$$\text{Janela} = \begin{cases} W_n \neq 0, & n=0,1,2,\dots,N-1 \\ W_n = 0, & n < 0 \text{ ou } n \geq N \end{cases}$$

Nota: Para aumentar a ordem do filtro (maior selectividade) e manter fase linear, aumenta-se a janela (centrada em zero) e faz-se translação dos coeficientes para que comecem em $n=0$.

Filtros FIR (Cont.)

Fenómeno de Gibbs

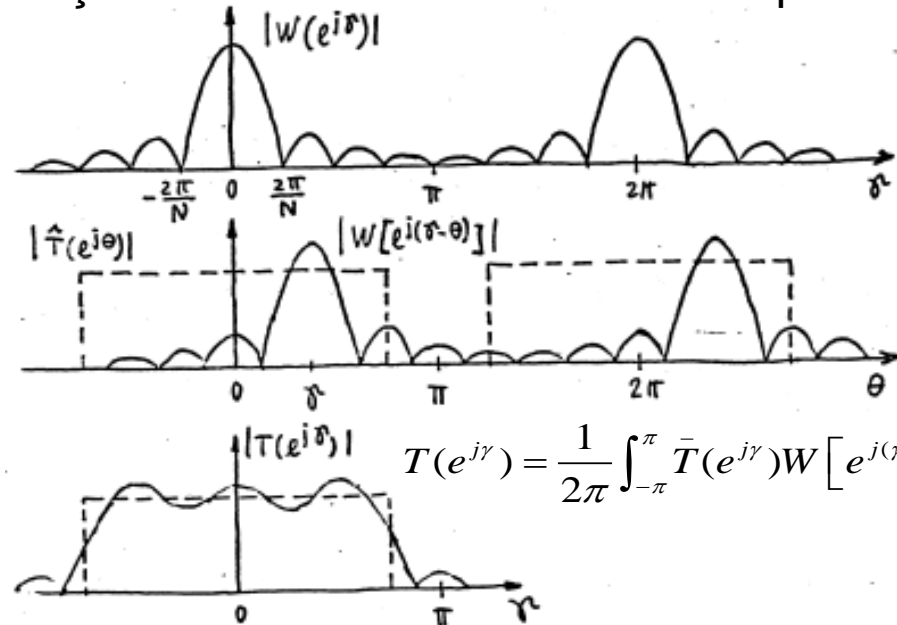
- A truncatura da resposta impulsional ideal provoca ondulação não desejada na resposta de amplitude! Minimização passa por utilizar janelas não rectangulares.
- Produto $\hat{h}_n W_n$ resulta na convolução dos sinais no domínio da frequência.
- Exp. para janela rectangular:

$$\begin{cases} W_n = 1, & n=0,1,2,\dots,N-1 \\ W_n = 0, & n<0 \text{ ou } n>N \end{cases}$$

$$W(e^{j\gamma}) = \sum_{n=0}^{N-1} e^{-jn\gamma} = \frac{1 - e^{-j\gamma N}}{1 - e^{-j\gamma}} =$$

$$= e^{-j\frac{N-1}{2}\gamma} \frac{\sin(\gamma N/2)}{\gamma N/2}$$

$$\bar{T}(e^{j\gamma}) = \sum_{n=-\infty}^{\infty} \hat{h}_n e^{-j\gamma n}$$

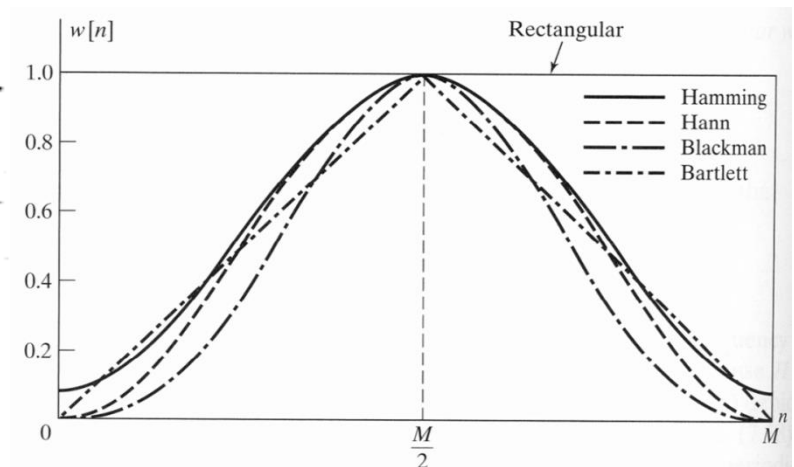
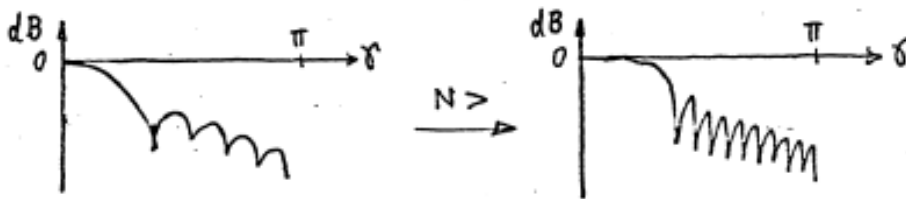


$$T(e^{j\gamma}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \bar{T}(e^{j\gamma}) W[e^{j(\gamma-\theta)}] d\theta$$

Filtros FIR (Cont.)

Fenómeno de Gibbs

- N grande conduz a lóbulos com largura menor e mesma área, ou seja, ondulação mais rápida mas com a mesma amplitude.
- Usando lóbulos laterais com menor amplitude vem lóbulo principal mais largo, logo menor ondulação mas banda de transição mais larga (menor selectividade).
- A banda de transição (\approx largura do lóbulo principal) $\Delta\gamma = A/N$, A é uma constante dependente da janela.



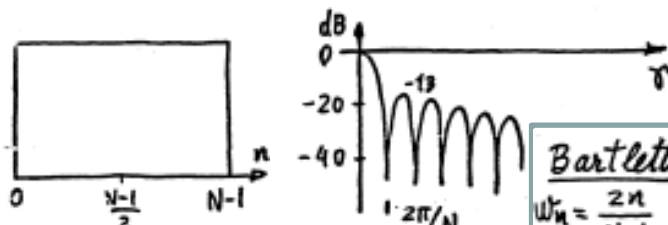
EG – Filtros Digitais

Filtros FIR (Cont.) *Janelas Mais Usadas*

Rectangular

$$w_n = 1$$

$$n = 0 \dots N-1$$



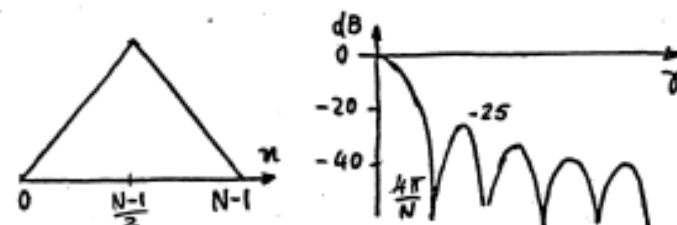
Bartlett

$$w_n = \frac{2n}{N-1}$$

$$n = 0 \dots (N-1)/2$$

$$= 2 - \frac{2n}{N-1}$$

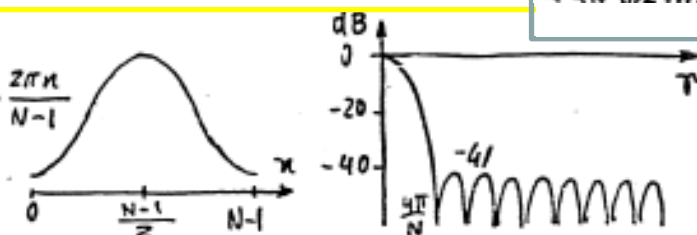
$$n = (N-1)/2 \dots N-1$$



Hamming

$$w_n = .54 - .46 \cos \frac{2\pi n}{N-1}$$

$$n = 0 \dots N-1$$

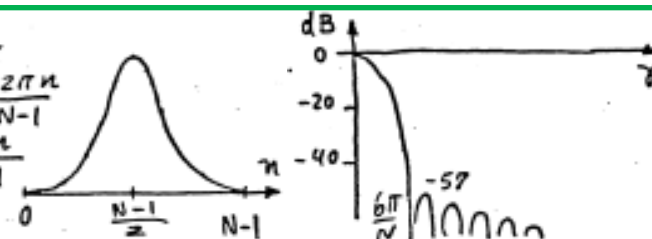


Blackman

$$w_n = .42 - .5 \cos \frac{2\pi n}{N-1}$$

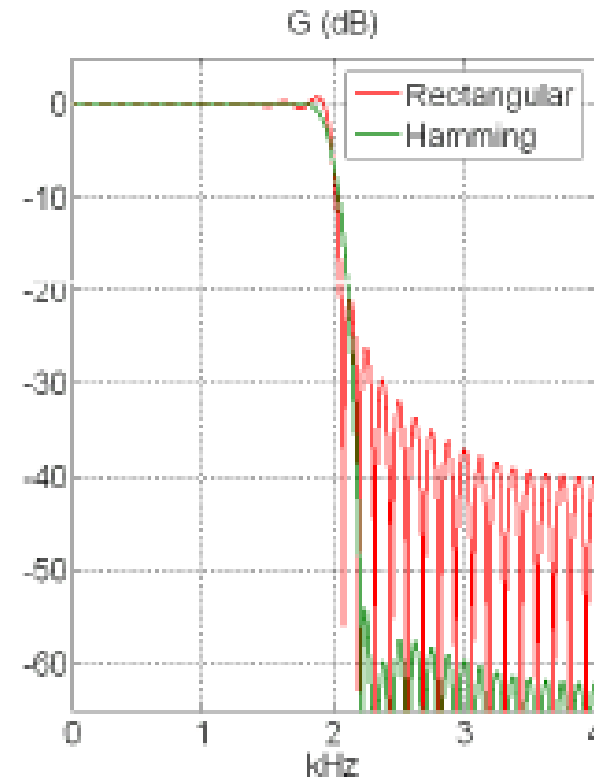
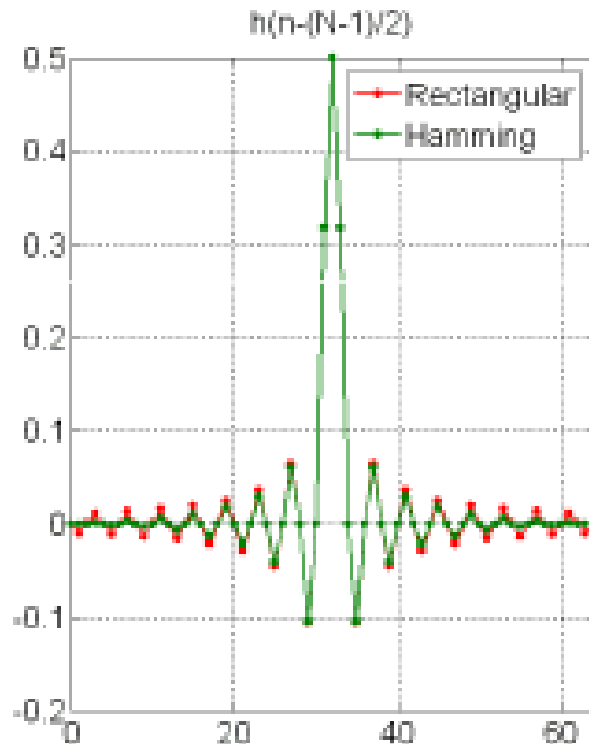
$$+ .08 \cos \frac{4\pi n}{N-1}$$

$$n = 0 \dots N-1$$



Filtros FIR (Cont.)

Exemplo: FIR LP de $N=64$, $f_s=8\text{kHz}$, $f_p=2\text{kHz}$, janelas rectangular e Hamming.



Filtros FIR (Cont.)

Realização de Filtros FIR

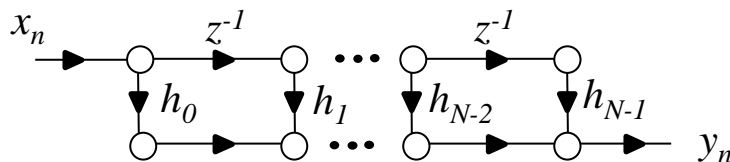
- 1 – DFS
- 2 – Optimização de processamento
- 3 – Implementação na tecnologia desejada

Diagrama de Fluxo de Sinal (DFS)

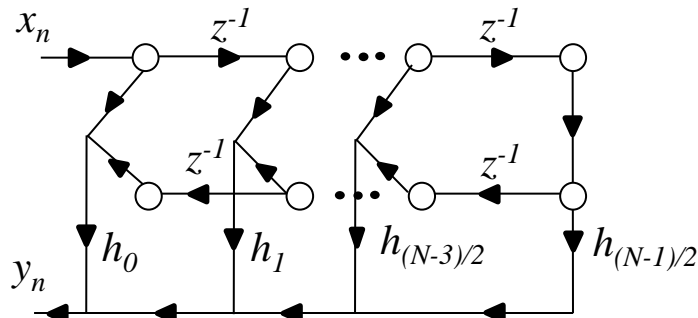
$$T(z) = \sum_{n=0}^{N-1} h_n z^{-n}$$

$$y_n = \sum_{k=0}^{N-1} h_k x_{n-k}$$

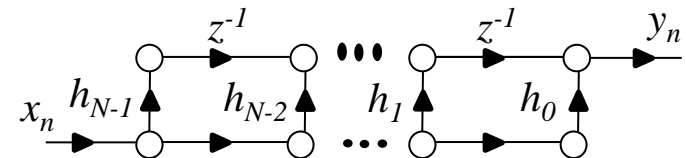
Forma Directa I



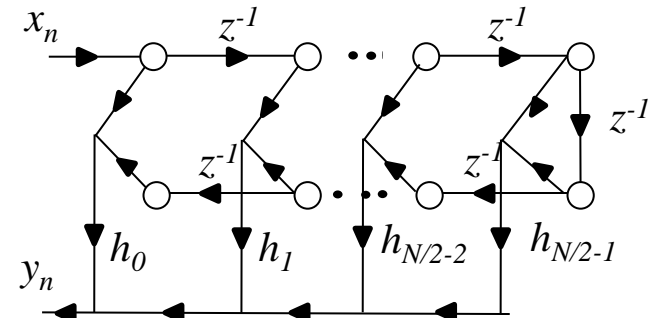
F.D. I para FIR de Fase Linear (N ímpar)



Forma Directa I Transposta



F.D. I para FIR de Fase Linear (N par)



Realização de Filtros Digitais

Há dois métodos mais usuais na realização de filtros digitais:

- 1) Implementação em hardware (ROM, ASIC, etc.)
- 2) Implementação em software (DSP, FPGA)

1) Implementação em ROM com Aritmética Distribuída

Eq. de Recorrência: $y_n + \sum_{k=1}^N b_k y_{n-k} = \sum_{k=0}^M a_k x_{n-k}$

Genericamente: $y = \sum_{k=1}^K a_k x_k$ ou seja $y = \sum_{k=1}^K a_k (-x_k^0 + \sum_{j=1}^b x_k^j 2^{-j})$


$$= -\sum_{k=1}^K a_k x_k^0 + \sum_{k=1}^K \sum_{j=1}^b a_k x_k^j 2^{-j}$$

$$\sum_{j=1}^b 2^{-j} \sum_{k=1}^K x_k^j a_k$$

Definindo

$$F(x_1^j, x_2^j, \dots, x_K^j) = \sum_{k=1}^K x_k^j a_k \quad j=1 \dots b$$

$$= -\sum_{k=1}^K x_k^0 a_k \quad j=0$$



$$y = \sum_{j=0}^b 2^{-j} F(x_1^j, x_2^j, \dots, x_K^j)$$

Realização de Filtros Digitais (cont.)

Exemplo para representação em complemento para 2 com 4 bits

$$y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} + a_3 x_{n-3}$$

$$\begin{aligned}
 &= a_0 (-x_n^0 + 2^{-1} x_n^1 + 2^{-2} x_n^2 + 2^{-3} x_n^3) + \\
 &\quad a_1 (-x_{n-1}^0 + 2^{-1} x_{n-1}^1 + 2^{-2} x_{n-1}^2 + 2^{-3} x_{n-1}^3) + \\
 &\quad a_2 (-x_{n-2}^0 + 2^{-1} x_{n-2}^1 + 2^{-2} x_{n-2}^2 + 2^{-3} x_{n-2}^3) + \\
 &\quad a_3 (-x_{n-3}^0 + 2^{-1} x_{n-3}^1 + 2^{-2} x_{n-3}^2 + 2^{-3} x_{n-3}^3)
 \end{aligned}$$

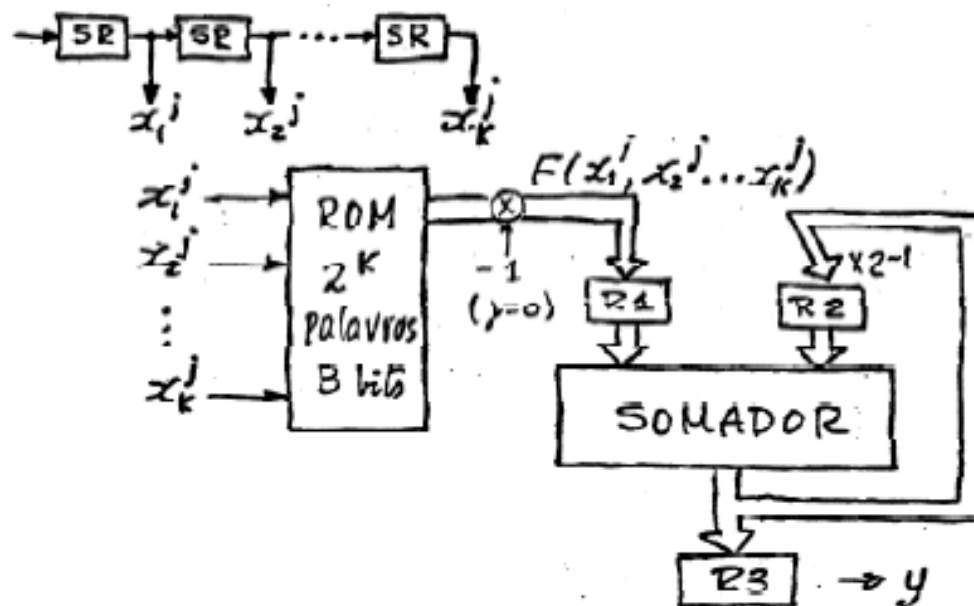
$F(x^0)$
 $F(x^1)$
 $F(x^2)$
 $F(x^3)$

$$\Rightarrow y_n = \sum_{j=0}^3 2^j F(x_n^j, x_{n-1}^j, x_{n-2}^j, x_{n-3}^j) \quad , (-F) \text{ para } j=0$$

EG – Filtros Digitais

Realização de Filtros Digitais (cont.)

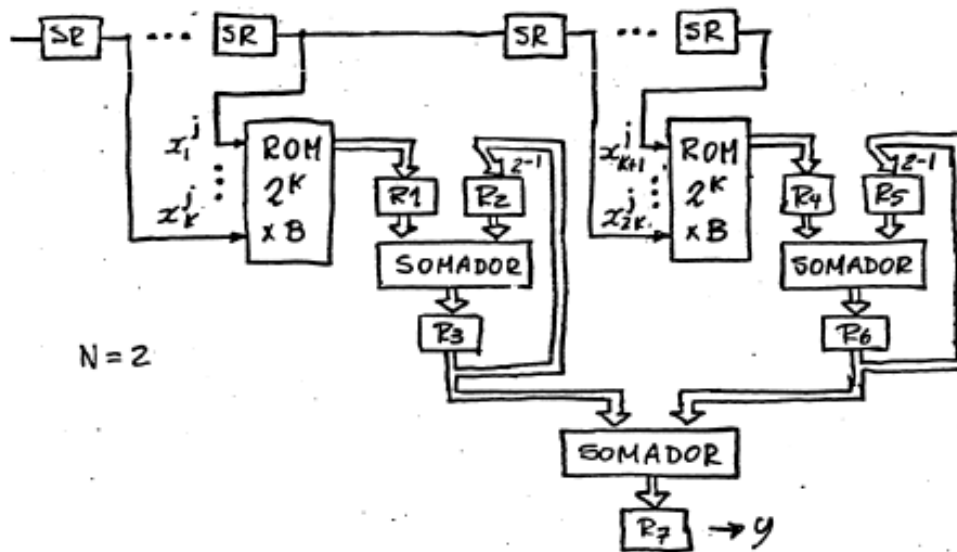
$$y_n = \sum_{j=0}^3 2^j F(x_n^j, x_{n-1}^j, x_{n-2}^j, x_{n-3}^j) \quad , (-F) \text{ para } j=0 \quad K=4, B=4$$



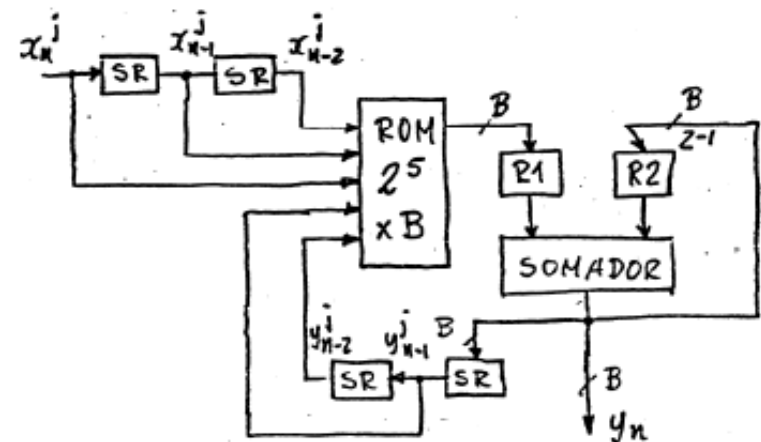
EG – Filtros Digitais

Realização de Filtros Digitais (cont.)

Arquitectura paralela – aumento de rapidez



Secção biquadrática



Realização de Filtros Digitais (cont.)

2) Implementação em DSP

Aplicações dos DSPs (Fonte: Texas Instrument)

	Digital Media Processors	OMAP Applications Processors	C6000 Digital Signal Processors	C5000 Digital Signal Processors	C2000 Microcontrollers	MSP430 Microcontrollers	Stellaris 32-Bit ARM Cortex-M3 MCUs
Audio							
Automotive							
Communications							
Industrial							
Medical							
Security							
Video							
Wireless							
Key Feature	Complete tailored video solution	Low power and high performance	High performance	Power-efficient performance	Performance, Integration for greener industrial applications	Ultra-low power	Open architecture software, rich communications options

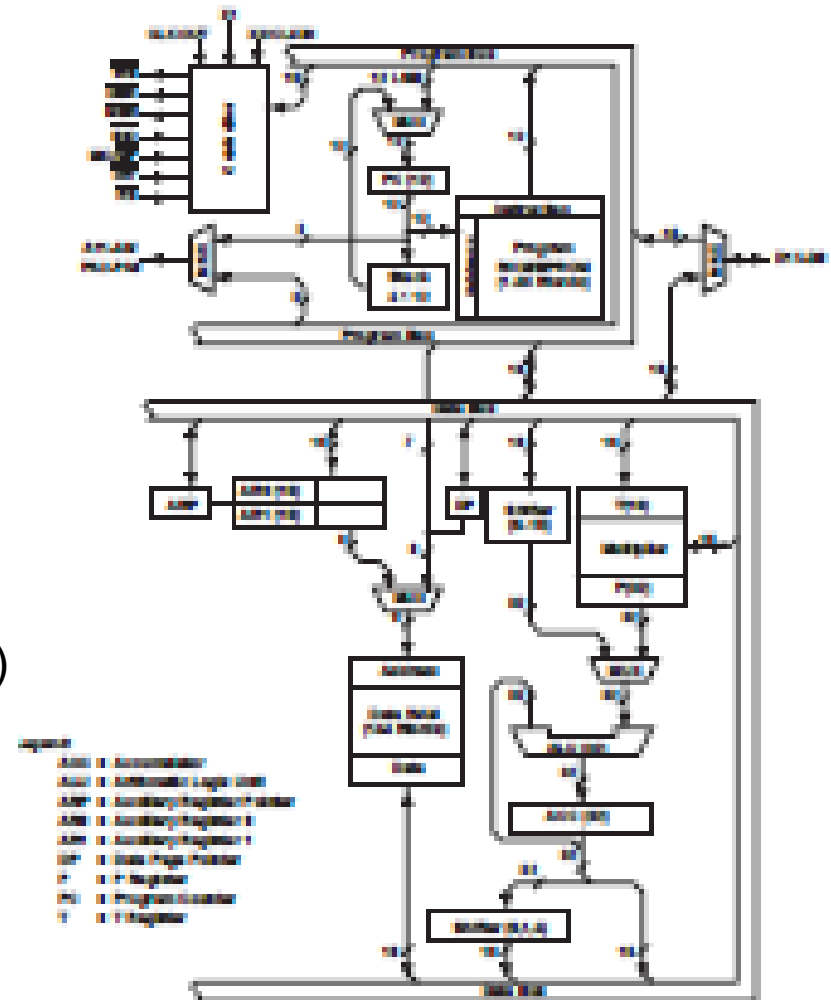
OMAP- Open Multimedia Applications Platform.

EG – Filtros Digitais

Real. de Filt. Dig. (cont.)

TI 32010 (1983)

- 5 MIPS
- Instruction Cycle 200 ns
- Palavra de 16 bits
- ALU 16x16 bits vírgula fixa
- Relógio de 20 MHz
- Memória:
 - 144 palavras RAM (2,3 kb)
 - 1,5 k palavras ROM (24 kb, programa)
- 40 pins
- Alimentação +5V/0V





EG – Filtros Digitais

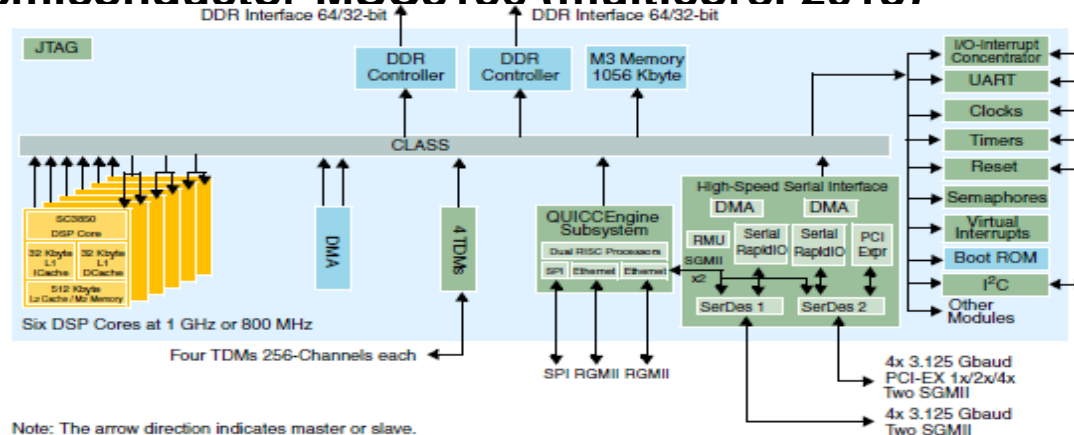
Realização de Filtros Digitais (cont.)

TMS320C25	TMS320C40	TMS320C62	TMS320C6748
Fins de 80	Meados de 90	Inícios de 2000	2014
Essencialmente vírgula fixa	Vírgula flutuante	Vírgula flutuante	Vírgula flutuante
Instruction cycle: 100 ns → 10 MIPS (≈ 4 operações por iteração) ≈ 40 MOPS	Instruction cycle: 40 ns → 275 MOPS	Vírgula flutuante: 1800 MOPS Vírgula fixa: 8000 MOPS	2746 MFLOPS (MFLOPS: <u>M</u> illion <u>F</u> loating <u>P</u> oint <u>O</u> perations <u>p</u> er <u>S</u> econd)
Principalmente operações com 16 bits	Operações com 32 bits	Operações com 32 bits	Operações com 32 bits e com 64 bits
1 ALU + 1 auxiliar ALU	1 ALU + 2 auxiliar ALUs	6 ALUs	6 ALUs
68 pinos	325 pinos	256 pinos	361 pinos

EG – Filtros Digitais

Realização de Filtros Dig (cont.)

Freescal Semiconductor MSC8156 (multicore. 2013)



- 783 pins
- Palavra de 128 bits
- Alimentação -0,3V a 1,1V
- 6 DSPs SC3850

- Relógio até 1GHz
- Memória: 2GB DDR3 RAM
- Um 3850 tem 8 ALU 16x16 bits

Intelcore i7-6700k (2015)

- Quad-core
- 64 GB (máx) DDR4
- 64 bits

- 1366 pins (saem do socket)
- Alimentação 1,35 V
- Relógio de 4 GHz



EG – Filtros Digitais

Filtros Adaptativos

1. Introdução

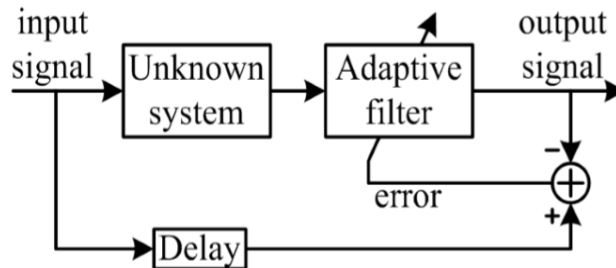
Sistemas (contínuos ou discretos cujos parâmetros variam no tempo e apresentam as seguintes características:

- Adaptação automática
- Possibilidade de treino para filtragens específicas
- Não necessitam de procedimentos relativos à síntese (aproximação), pelo contrário, desenham-se a si mesmos
- Podem identificar modelos, após um período de treino
- Em certa medida, podem reparar-se a si próprios
- Em geral, podem considerar-se sistemas não lineares de parâmetros variáveis no tempo
- Em geral, são mais difíceis de analisar que os sistemas não adaptativos, mas possibilitam um melhor desempenho quando o sinal de entrada tem características desconhecidas ou variantes no tempo

Filtros Adaptativos

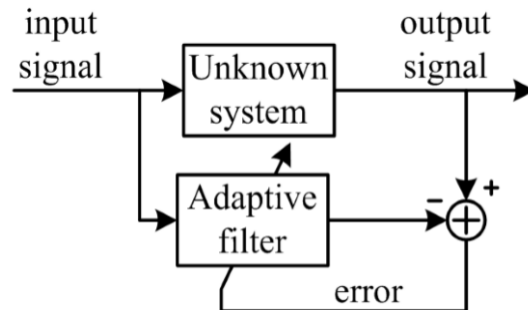
2. Modos de Operação

Correção (modelação inversa, circuito aberto)



$$\hat{H}_{opt}(z) = H^{-1}(z)$$

Identificação (modelação direta, circuito fechado)



$$\hat{H}_{opt}(z) = H(z)$$



EG – Filtros Digitais

Filtros Adaptativos

3. Áreas de Aplicação

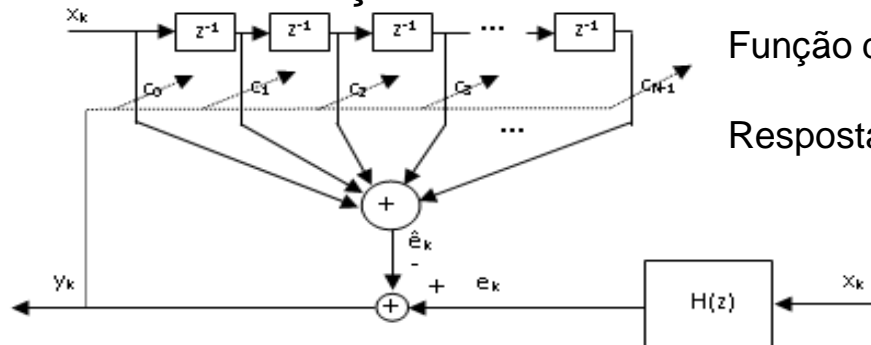
- Identificação de sistemas
- Igualização de canal
- Digitalização/síntese de voz
- Cancelamento de eco (dados e acústico)
- Branqueamento espectral

Filtros Adaptativos

3. Áreas de Aplicação (Cont.)

Exemplos:

- Identificação de sistemas

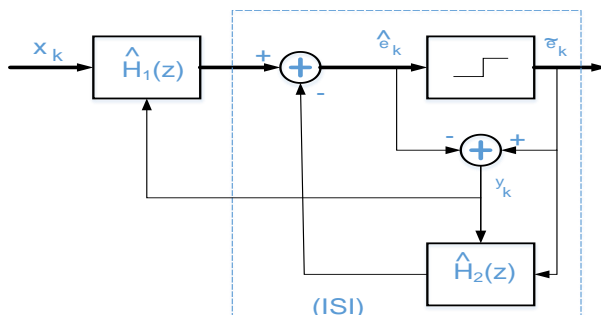


Função de sistema: $\hat{H}(z) = \hat{c}_0 + \hat{c}_1 z^{-1} + \hat{c}_2 z^{-2} + \dots + \hat{c}_N z^{-N}$

Resposta impulsional: $\hat{h}_n = \hat{c}_0, \hat{c}_1, \hat{c}_2, \dots, \hat{c}_N$

$$Y(z) = E(z) - \hat{E}(z) = X(z) [H(z) - \hat{H}(z)]$$

- Igualização Adaptativa – Decision Feedback Equaliser (DFE)



$$X(z) = H(z)E(z)$$

$$\begin{aligned} Y(z) &= \tilde{E}(z) - \hat{E}(z) = \frac{X(z)}{H(z)} - \left[X(z)\hat{H}_1(z) - \frac{X(z)}{H(z)}\hat{H}_2(z) \right] = \\ &= X(z) \left[\frac{1 + \hat{H}_2(z)}{H(z)} - \hat{H}_1(z) \right] \end{aligned}$$

EG – Filtros Digitais

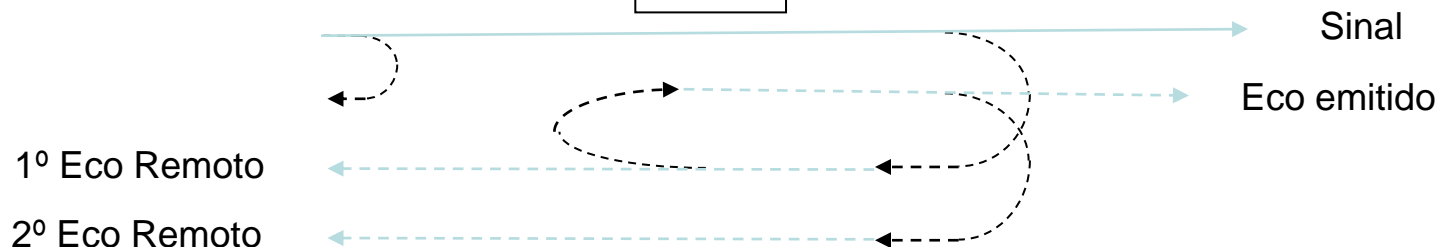
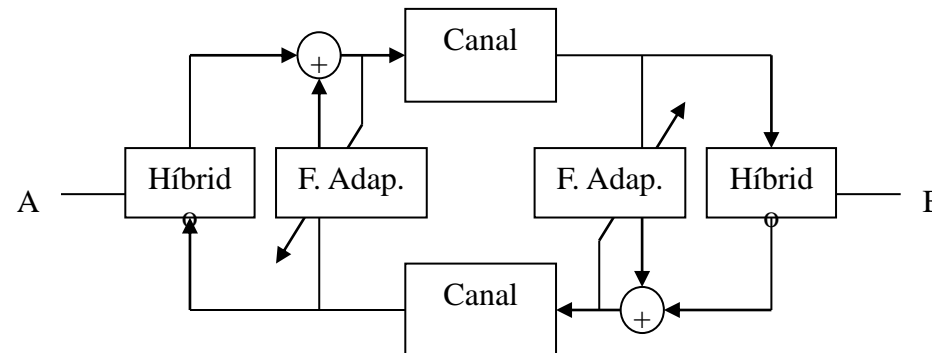
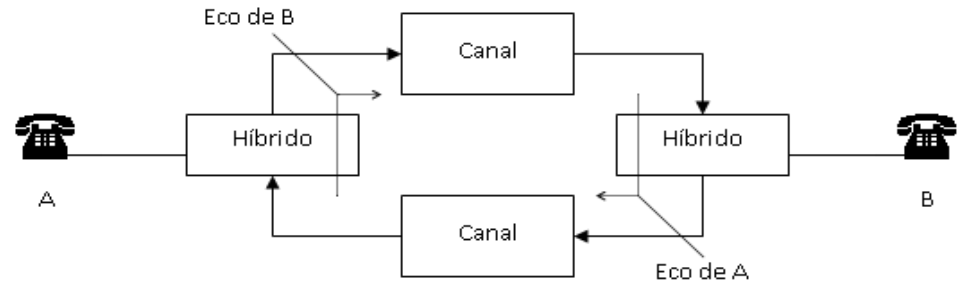
Filtros Adaptativos

3. Áreas de Aplicação (Cont.)

- Cancelamento de Eco em Transmissão de Voz

Eco remoto:

Grande atraso (satélite > 0,5 s) e pequena amplitude

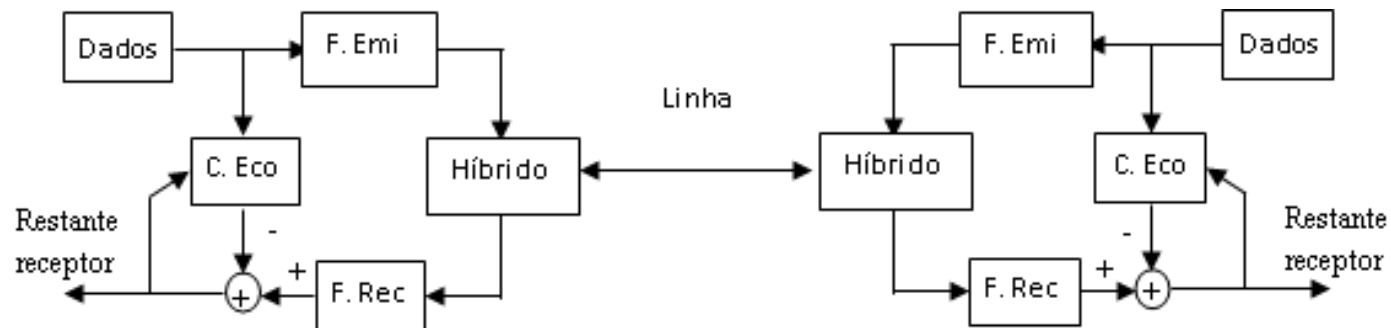


EG – Filtros Digitais

Filtros Adaptativos

3. Áreas de Aplicação (Cont.)

- Cancelamento de Eco em Transmissão de Dados



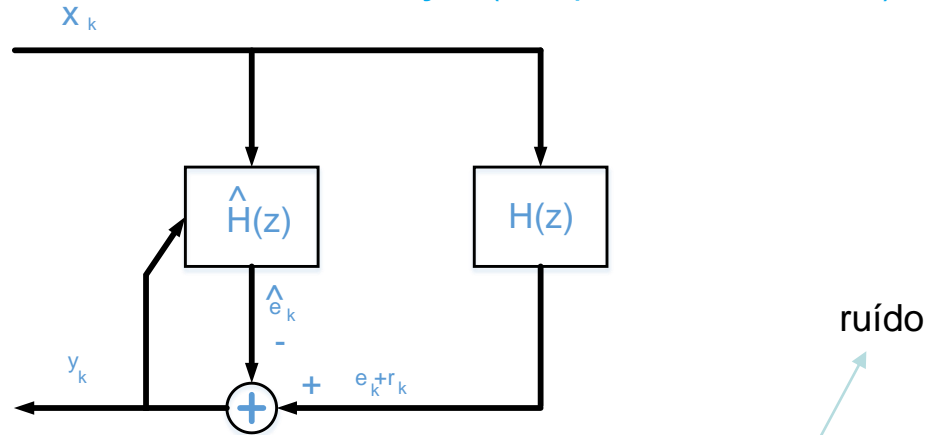
Eco local:

Pequeno atraso e grande amplitude (mais de -15 dB abaixo do sinal emitido)

EG – Filtros Digitais

Filtros Adaptativos

4. Sistema de Identificação (“Output Error Criterion”)



e_k – Sinal de saída do sistema desconhecido

\hat{e}_k – Sinal de saída do filtro adaptativo
(estimativa de e_k)

x_k – Sinal de entrada

y_k – Sinal de erro (e possível sinal de saída)

r_k – Ruído aditivo

ruído

$$Y(z) = E(z) + R(z) - \hat{E}(z) = X(z) [H(z) - \hat{H}(z)] + R(z)$$

Minimização de $E[y_k^2] \Rightarrow \hat{H}(z) = H(z)$

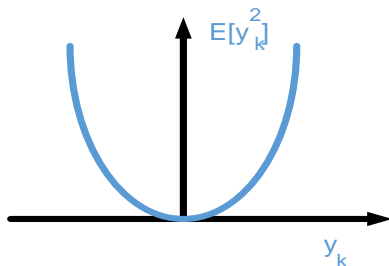
Características:

- y_k é geralmente adequado ao restante recetor
- Sistema de identificação mais utilizado
- Mas para filtros adaptativos recursivos conduz geralmente a superfícies de erro com mínimos locais

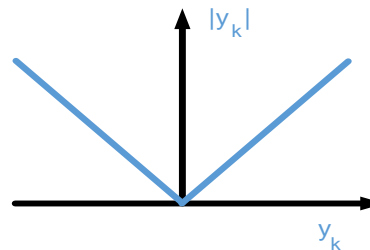
Filtros Adaptativos

5. Formas de Erro (Funções de Custo)

Função do
Erro Quadrático Médio

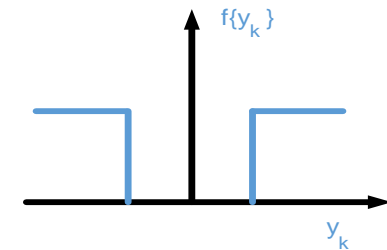


Função do
Módulo



Outras Funções
Exp: Não Linear com

Limiar



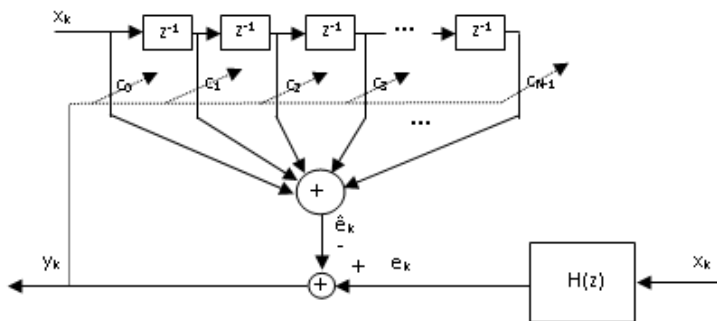
5.6 Estimadores Lineares Ótimos

- Não Recursivo – Filtro de Wiener, produz uma estimativa baseada numa soma pesada de amostras em número finito.
- Recursivo – Filtro de Kalman, produz uma estimativa baseada num modelo recursivo (AR – “Autoregressive”) continuamente otimizado a partir de cada nova amostra.

EG – Filtros Digitais

Filtros Adaptativos

7. Filtro de Wiener



Superfície de erro é um hiper-paraboloide invertido (mínimo global)

$$\text{Mínimo} \rightarrow \frac{\partial E[y_k^2]}{\partial \hat{\mathbf{C}}^T} = 0$$

$$\underbrace{E[e_k \mathbf{X}_k^T]}_P = \hat{\mathbf{C}}_{\text{otimo}}^T \underbrace{E[\mathbf{X}_k \mathbf{X}_k^T]}_R$$

Equação de Wiener-Hopf:

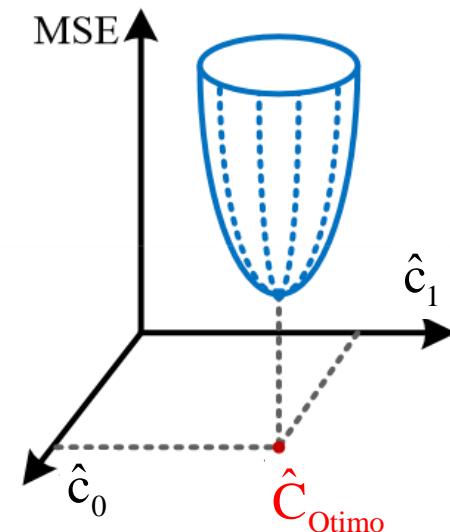
$$\hat{\mathbf{C}}_{\text{otimo}}^T = \mathbf{R}^{-1} \mathbf{P}$$

$$\hat{e}_k = \sum_{i=0}^N x_{k-i} \hat{c}_i = \mathbf{X}_k^T \hat{\mathbf{C}} = \hat{\mathbf{C}}^T \mathbf{X}_k$$

$$\mathbf{X}_k = [x_k, x_{k-1}, \dots, x_{k-N}]^T$$

$$\hat{\mathbf{C}} = [\hat{c}_0, \hat{c}_1, \dots, \hat{c}_N]^T$$

$$\text{MSE} = E[y_k^2] = E[(e_k - \hat{\mathbf{C}}^T \mathbf{X}_k)^2]$$





EG – Filtros Digitais

Filtros Adaptativos

8. Algoritmos de Adaptação

Algoritmos de gradiente – baseados no filtro de Wiener, estimam o gradiente das superfícies de nível. Bons para superfícies de erro “bem comportadas”.

Exp: LMS e derivados.

Algoritmos de mínimos quadrados – baseados no filtro de Kalman, determinísticos, minimizam a soma dos quadrados dos erros. Não são directamente afectados pelos mínimos locais. Geralmente apresentam maior rapidez de convergência, maior complexidade e problemas de estabilidade.

Exp: RLS, Fast-Kalman e derivados.

8.1 Algoritmos de Gradiente

Algoritmo do gradiente determinístico

$$\text{Gradiente} = \nabla_k^T = \frac{\partial E[y_k^2]}{\partial \hat{\mathbf{C}}^T} = \frac{\partial \epsilon}{\partial \hat{\mathbf{C}}^T} = \left[\frac{\partial \epsilon}{\partial \hat{c}_0}, \frac{\partial \epsilon}{\partial \hat{c}_1}, \dots, \frac{\partial \epsilon}{\partial \hat{c}_N} \right] = 2R\hat{\mathbf{C}}_k - 2P$$

Fazendo $\nabla_k^T = 0$ vem

$$\hat{\mathbf{C}}_{k+1} = \hat{\mathbf{C}}_k - \mu \nabla_k$$

μ é o passo de adaptação, regula a estabilidade e rapidez de convergência



Filtros Adaptativos

8.1 Algoritmos de Gradiente (Cont.)

Algoritmo do gradiente estocástico (LMS – “Least Mean Squares”)

Utiliza o valor instantâneo de y_k^2 como estimativa da média

$$\hat{\nabla}_k = \frac{\partial y_k^2}{\partial \hat{\mathbf{C}}^T} = \left[\frac{\partial y_k^2}{\partial \hat{c}_0}, \frac{\partial y_k^2}{\partial \hat{c}_1}, \dots, \frac{\partial y_k^2}{\partial \hat{c}_N} \right] = -2y_k \mathbf{x}_k$$

Algoritmo LMS: $\hat{\mathbf{C}}_{k+1} = \hat{\mathbf{C}}_k + 2\mu y_k \mathbf{x}_k$

(Variantes: algoritmos do sinal, LMS normalizado, etc)

8.2 Algoritmos de Mínimos Quadrados

Minimizam a soma $J_k = \sum_{i=a}^k y_i^2$

Exps: RLS (“Recursive Least Squares”, $a=1$), “Fast Kalman, etc.



EG – Filtros Digitais

Filtros Adaptativos

8.3 Comparação da Complexidade (Peso Computacional)

Algoritmo	Multi./Div.	Somas/Sub	Total	Exp: N=30
LMS	$2(N+1)$	$2(N+1)$	$4N+3$	123
RLS	$3N^2+11N+8$	$2N^2+6N+4$	$5N^2+17N+1$ 2	5022
Fast	$8N+5$	$7N+2$	$15N+7$	457

N – Ordem do filtro (N+1 baixadas)

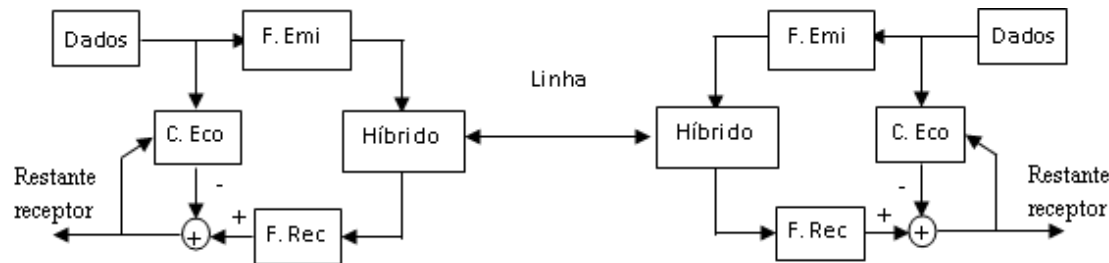
Estrutura transversal (usualmente designada por filtro FIR)

Estrutura Processamento do filtro também incluído

EG – Filtros Digitais

Filtros Adaptativos

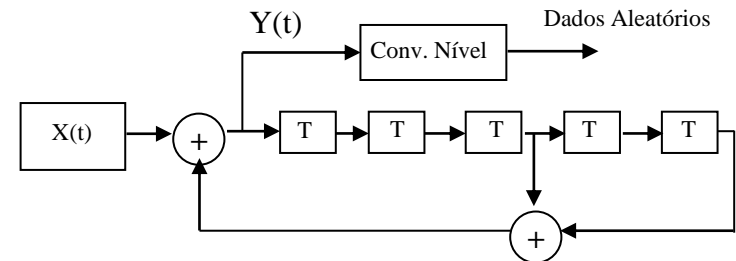
3º Trabalho de Laboratório: Cancelamento de eco em transmissão de dados



Algoritmo LMS: $c_{i,k+1} = c_{i,k} + 2 \mu y_k x_{k-i}$

ERLE:
$$ERLE = \frac{E[e_k^2]}{E[(e_k - \hat{e}_k)^2]} \text{ dB}$$

Baralhador de dados (scrambler)



Arquitetura do filtro adaptativo

