



Instituto Superior Politécnico de Viseu
Escola Superior de Tecnologia de Viseu
Curso de Engenharia de Sistemas e Informática

Processamento Digital de Sinal

Aulas 5,6

4.º Ano – 2.º Semestre

Manuel A. E. Baptista, Eng.º

Manuel A. E. Baptista
Ernesto R. Afonso



Departamento de Informática

www.estv.ipv.pt

1

Programa:

1. Introdução ao Processamento Digital de Sinal
2. Representação e Análise de Sinais
3. Estruturas e Projecto de Filtros FIR e IIR
4. Processamento de Imagem
5. Processadores Digitais de Sinal

Manuel A. E. Baptista
Ernesto R. Afonso



Departamento de Informática

www.estv.ipv.pt

2

Bibliografia:

Processamento Digital de Sinal:

- Sanjit K. Mitra, “**Digital Signal Processing** – A computer based approach”, McGraw Hill, 1998
Cota: 621.391 MIT DIG
- Roman Kuc, “**Introduction to Digital Signal Processing**”, McGraw Hill, 1988.
Cota: 621.391 KUC INT
- Johnny R. Johnson, “**Introduction to Digital Signal Processing**”, Prentice-Hall, 1989.
Cota: 621.391 JOH INT
- G. Proakis, G. Manolakis, “**Digital Signal Processing – Principles, Algorithms Applications**”, 3ª Ed, P-Hall, 1996.
Cota: 621.391 PRO DIG
- James V. Candy, “**Signal Processing – The modern Approach**”, McGraw-Hill, 1988
Cota: 621.391 CAN SIG
- Mark J. T., Russel M., “**Introduction to DSP – A computer Laboratory Textbook**”, John Wiley & Sons, 1992.
Cota: 621.391 SMI INT
- James H. McClellan e outros, “**Computer-Based Exercises - Signal Proc. Using Matlab 5**”, Prentice-Hall, 1998.
Cota: 621.391 MCC COM

Processamento Digital de Imagem:

- Rafael C. Gonzalez & Richard E. Woods, “**Digital Image Processing**”, Prentice Hall, 2ª Ed., 2002.
Cota: 681.5 GON DIG.
- I. Pittas H. McClellan e outros, “**Digital Image Processing Algorithms and Applications**”, John Wiley & Sons, 2000.
Cota: 621.391 PIT.
- William K. Pratt, “**Digital image processing**”, John Wiley, 2ª Ed, 1991.
Cota: 681.5 PRA DIG
- Bernd Jähne, “**Digital image processing : concepts, algorithms, and scientific applications**”, Springer, 1997.
Cota: 681.5 JAH

Avaliação:

A avaliação é composta pela componente teórica e componente prática ponderadas da seguinte forma:

$$\text{Classificação Final} = 80\% * \text{Frequência ou exame} + 20\% * \text{Prática}$$

O acesso ao exame não está condicionado embora não tenha função de melhoria, ou seja, se o aluno entregar a prova de exame, será essa a classificação a utilizar no cálculo da média final independentemente da nota da prova de frequência obtida.

A avaliação prática é constituída por trabalhos laboratoriais a executar em MATLAB

2. Representação e Análise de Sinais (cont.)

• Análise de Fourier - Sumário

- Janelas da DFT
- Resolução da DFT - melhoramento
- Cálculo eficiente da DFT: FFT
- Considerações: Análise espectral

Manuel A. E. Baptista
Ernesto R. Afonso



Departamento de Informática

5

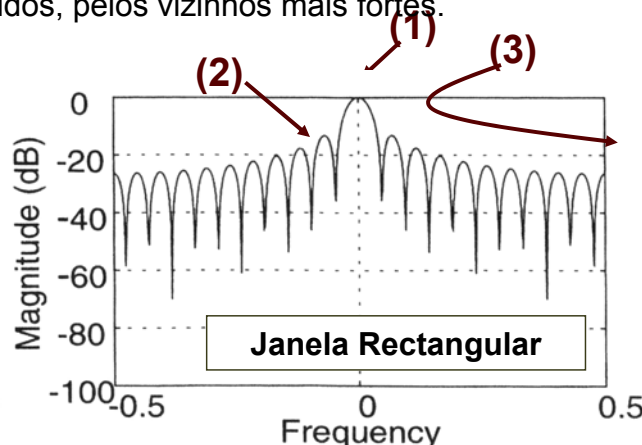
www.estv.ipv.pt

DFT – Características das Janelas

- **Sequência Discreta Finita** $\Rightarrow$ O espectro é convoluído com o espectro duma janela rectangular.
- A quantidade de *leakage* depende da janela escolhida, e da forma como o sinal se ajusta na janela.

- (1) **Resolução**: capacidade de distinção dos diferentes tons – Inversamente proporcional à largura do lóbulo principal. **Objectivo**: *a mais elevada possível*.
- (2) **Pico – nível do lóbulo lateral**: resposta máxima fora do lóbulo principal. Determina se os sinais pequenos são escondidos, pelos vizinhos mais fortes. **Objectivo**: *o mais baixo possível*.
- (3) **Decaimento dos lóbulos laterais**: decaimento dos lóbulos laterais por década. **Compromisso com** (2).

Usam-se várias janelas (*dependentes da aplicação*): **Hamming, Hanning, Blackman, Kaiser ...**



Manuel A. E. Baptista
Ernesto R. Afonso



Departamento de Informática

6

www.estv.ipv.pt

DFT das principais janelas

A Janelação (*Windowing*) reduz o *leakage* através da minimização do módulo dos lóbulos laterais.

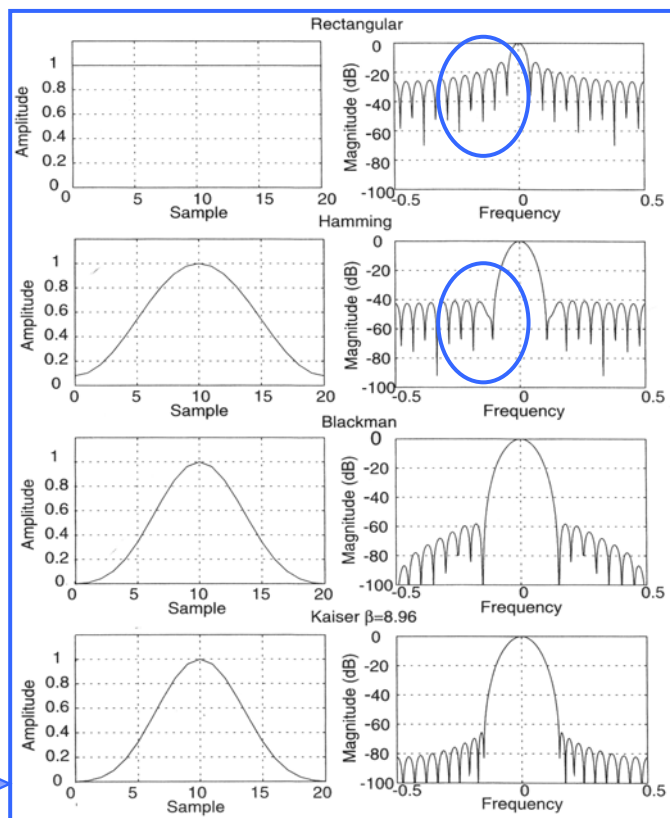
No tempo reduz as descontinuidades dos pontos terminais.

Sequência Amostrada

Não Janelada

Janelada

Algumas funções de Janela



7

DFT – Escolha da Janela

Características das Janelas mais Comuns

Tipo de Janela	-3 dB Main-lobe width [impulsos]	-6 dB Main-lobe width [impulsos]	Max sidelobe level [dB]	Sidelobe roll-off [dB/década]
Rectangular	0.89	1.21	-13.2	20
Hamming	1.3	1.81	-41.9	20
Hanning	1.44	2	-31.6	60
Blackman	1.68	2.35	-58	60

Sinal Observado

Far & strong interfering components
Near & strong interfering components
Accuracy measure of single tone

Lista de objectivos da Janela

⇒ high roll-off rate.
⇒ small max sidelobe level.
⇒ wide main-lobe

NB: Um forte componente DC pode tapar os pequenos sinais na sua vizinhança. Remova **all**!

8

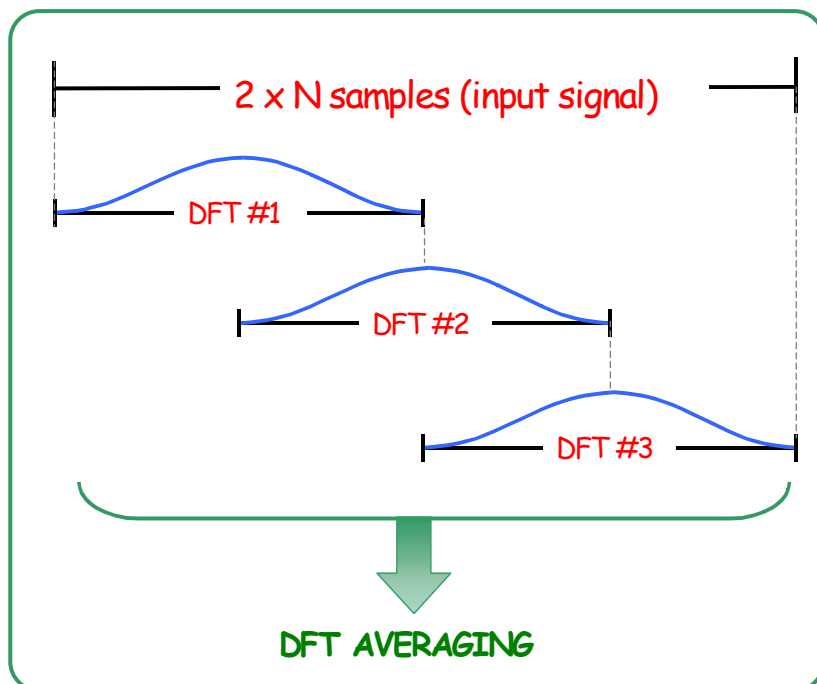
DFT – Correção das Perdas da Janela

Smooth data-tapering windows cause information loss near edges.

Solução:

sliding (overlapping) DFTs.

- Attenuated inputs get next window's full gain & leakage reduzido.
- Normalmente um *overlap* de 50% ou 75% (depende da largura do lóbulo principal).

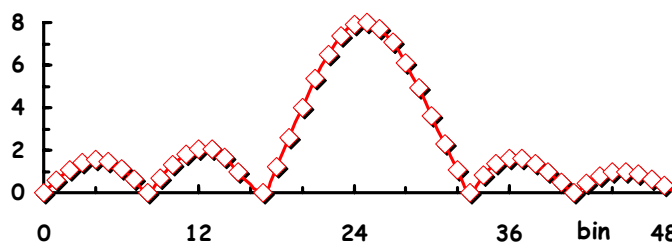
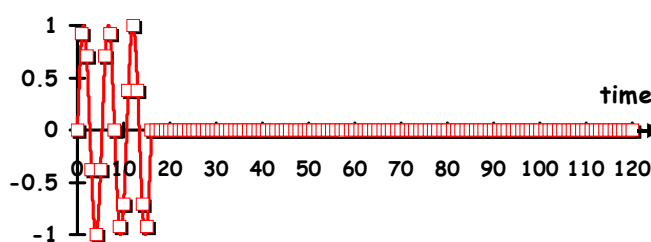
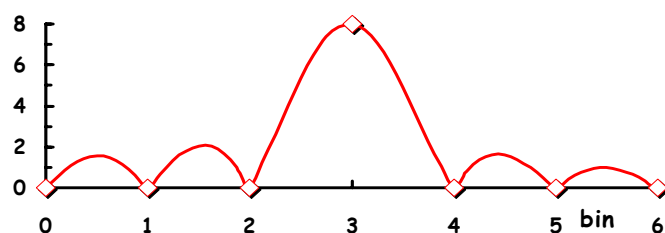
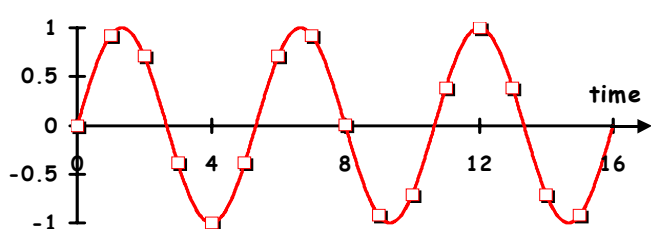


Inconveniente: incremento do tempo de processamento total.

9

DFT - Adição de Zeros

Melhora os espaçamento inter-amostras na frequência da DFT ("resolução").



Após a introdução de impulsos

@ frequências

$$f_m = \frac{m \cdot f_s}{N_S + L}$$

N_S = amostras originais, L = padded.

10

DFT - Adição de Zeros (2)

Resolução espectral da DFT

Capacidade de distinguir duas frequências muito próximas: não é melhorada pela introdução de zeros!

Espaçamento inter-amostras na frequência: incrementado pela adição de zeros ("espalhamento da frequência" da DFT devido à mesma frequência f_s)

- A adição de zeros no domínio da frequência, aumenta a taxa de amostragem no domínio do tempo. **Nota:** só funciona se o teorema da amostragem for satisfeito!
- Outras razões para a adição de zeros: para obter um número de amostras igual a uma potência de 2 (ver FFT).

NOTA: Introduza zeros **após** o janelamento (se for o caso)! Caso contrário, os zeros irão distorcer parcialmente a função da janela.



DFT - scalloping loss (SL)

A frequência de entrada f_0 entre os centros dos impulsos provoca perda de módulo

$$SL = 20 \log_{10}(|c_{r+k_{\max}}|/|c_{k_{\max}}|)$$

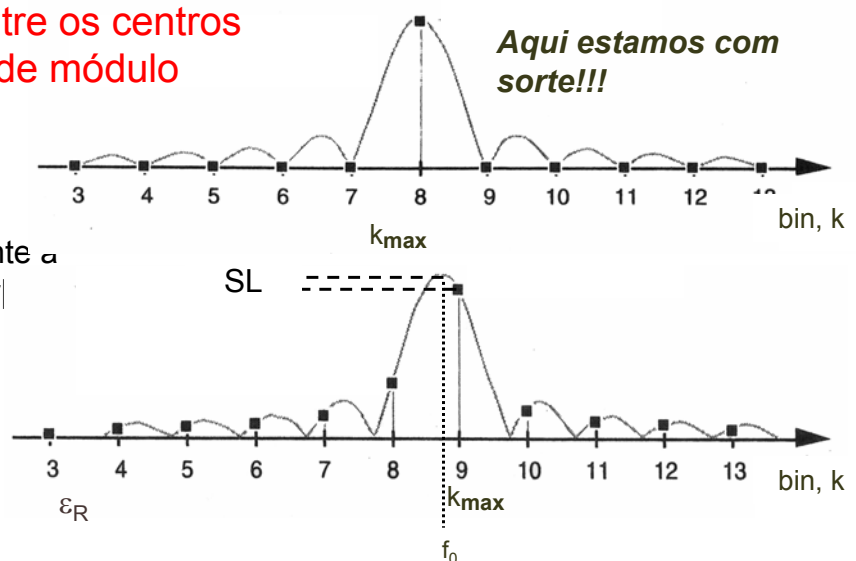
O pior caso quando f_0 cai exactamente a meio entre 2 impulsos sucessivos ($|r|$)

$$f_0 = (k_{\max} + r) f_s / N \quad |r| \leq 1/2$$

Erro na frequência: $\varepsilon_f = r f_s / N$,

Erro relativo: $\varepsilon_R = \varepsilon_f / f_0 = r / [(k_{\max} + r)] \leq 1 / (1 + 2 k_{\max})$

Pode ter impacto na interpretação dos dados (f_0 errado!)



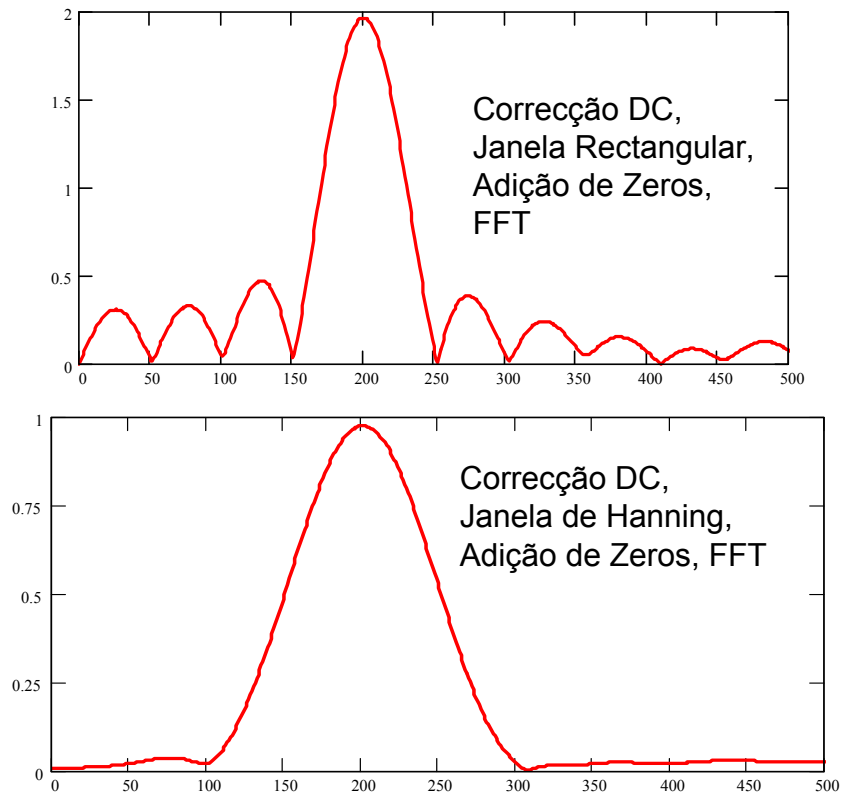
Nota: As janelas não rectangulares alargam o lóbulo principal da DFT $\Rightarrow$ SL menos severo

A correcção depende da janela usada!!!

DFT - SL : Exemplo

SL - correcção

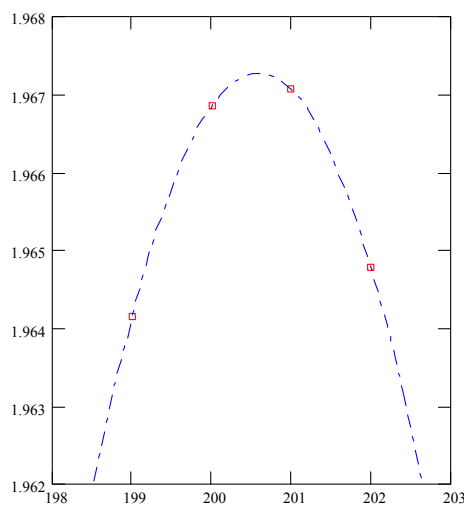
- aumentando N (?)
- melhorando o janelamento,
- adicionando zeros,
- interpolação à volta de $k_{\max}$.



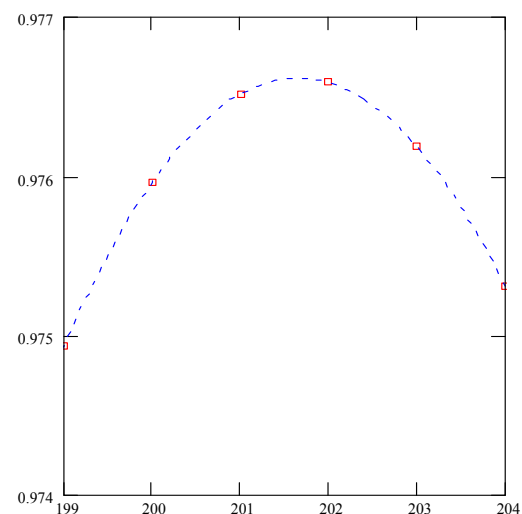
13

DFT – Interpolação Parabólica

Janela Rectangular



Janela de Hanning



- A interpolação parabólica, é em geral suficiente para determinar a posição do pico (i.e. frequência).
- Existem outros algoritmos dependendo dos dados.

14

DFT - Média

Média Incoerente

$$\overline{\text{DFT}}_{\text{INCOH}}(k) = \frac{|DFT_1(k)| + \dots + |DFT_M(k)|}{M}$$

M = N°. de DFT para a média

k = número impulsos, k=0, 1 .. N-1

➡ Reduz as flutuações do ruído de fundo, e a energia média do ruído não é alterada.

Média Coerente

$$\overline{\text{DFT}}_{\text{COH}}(k) = \frac{\text{Re}\{DFT_1(k)\} + \dots + \text{Re}\{DFT_M(k)\}}{M} + j \cdot \frac{\text{Im}\{DFT_1(k)\} + \dots + \text{Im}\{DFT_M(k)\}}{M}$$

➡ A energia do ruído de fundo é reduzida.

Manuel A. E. Baptista
Ernesto R. Afonso



DFT - Cálculo eficiente: FFT

DFT

$$\tilde{c}_k = \frac{1}{N} \sum_{n=0}^{N-1} s[n] \cdot e^{-j \frac{2\pi kn}{N}} = \frac{1}{N} \sum_{n=0}^{N-1} s[n] \cdot W_N^{kn}$$

$W_N^{n,k}$ = *twiddle factors*

k = 0, 1 .. N-1

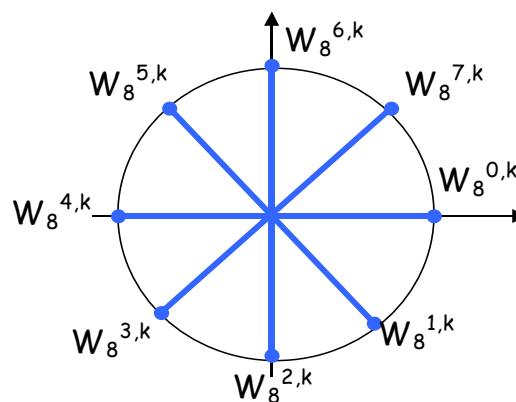
Cálculo directo da DFT ➡ redundância

W_N^{kn} função periódica calculada muitas vezes.

O cálculo directo da DFT necessita de $\sim N^2$ multiplicações complexas.

complexidade $O(N^2)$

Muito Mau !



Algoritmos (= **FFT - Fast Fourier Transform**) desenvolvido para calcular N pontos da DFT com $\sim N \log_2 N$ multiplicações (complexidade $O(N \log_2 N)$).

Manuel A. E. Baptista
Ernesto R. Afonso



DFT – Cálculo eficiente

Complexidade Computacional:

Medida através do número de $\times$, $+$, e é proporcional ao tempo gasto para executar um algoritmo.

Porém: outros factores: quantidade de memória requerida operações transcendentais, raiz, log, etc.

Em VLSI: consumo, área de chip são factores importantes para escolha de um algoritmo.

Algoritmos de FFT: revolucionaram a área de processamento de sinais

DFT - Cálculo eficiente

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{kn} \quad k = 1, 2, \dots, N-1$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot W_N^{-kn} \quad n = 1, 2, \dots, N-1$$

$$W_N = e^{-j\frac{2\pi}{N}}$$

Como as equações diferem apenas do factor de escala N e do sinal do expoente de W_N , a teoria utilizada p/ calculo da DFT aplica se também à IDFT

Cálculo directo:

- como $x[n]$ pode ser um sinal complexo, para calcular N amostras do sinal $X[k]$ são necessárias N^2 multiplicações complexas e $N(N-1)$ adições complexas

ou

- $4N^2$ multiplicações reais e $N(4N-2)$ somas reais, além de mais memória para armazenamento de N amostras complexas de $x[n]$ e dos coeficientes W_N

DFT - Cálculo eficiente

A maioria dos algoritmos de FFT exploram as seguintes características:

1) **Simetria complexa conjugada:** $W_N^{k[N-n]} = W_N^{-kn} = (W_N^{kn})^*$

2) **Periodicidade em k e n :** $W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}$

Exploram ainda a decomposição de uma DFT de N pontos em DFTs de comprimentos menores

Algoritmos:

- Goertzel(1958): $O(N^2)$
- Cooley - Tukey(1965): Deu origem à decimação no tempo
- Sande - Tukey(1966): Deu origem à decimação em frequência.

DFT – Cálculo Eficiente - Algoritmos de Decimação no Tempo

Decomposição sucessiva de $x[n]$ em parcelas menores

Diversos tipos: mais clássico: para N potências de 2

$x[n]$ de N pontos é dividido em 2 sequências de $N/2$ pontos compostas pelos n ímpares e n pares

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{kn}$$

$$X[k] = \sum_{n \text{ par}} x[n] \cdot W_N^{nk} + \sum_{n \text{ ímpar}} x[n] \cdot W_N^{nk}$$

DFT – Cálculo Eficiente - Algoritmos de Decimação no Tempo

$$X[k] = \sum_{n \text{ par}} x[n] \cdot W_N^{nk} + \sum_{n \text{ ímpar}} x[n] \cdot W_N^{nk}$$

Mudando as variáveis: $n=2r$ para n par
 $n=2r+1$ para n ímpar

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r] \cdot W_N^{2rk} + \sum_{r=0}^{(N/2)-1} x[2r+1] \cdot W_N^{(2r+1)k}$$

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r] \cdot (W_N^2)^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x[2r+1] \cdot (W_N^2)^{rk}$$

Como: $W_N^2 = W_{N/2}$ podemos reescrever:

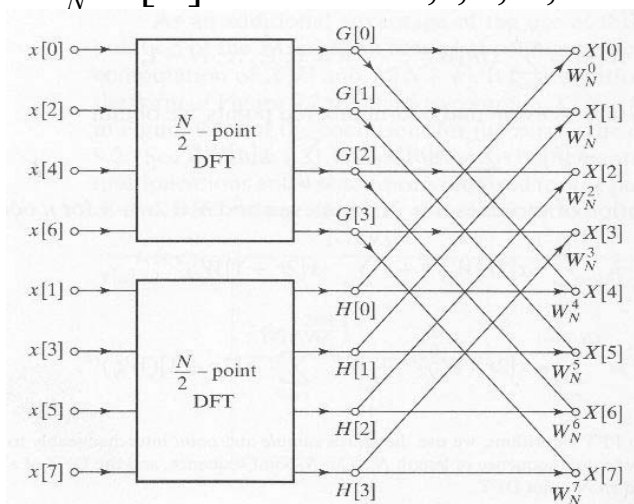
21

DFT – Cálculo Eficiente - Algoritmos de Decimação no Tempo

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r] \cdot (W_N^2)^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x[2r+1] \cdot (W_N^2)^{rk}$$

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r] \cdot W_{N/2}^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x[2r+1] \cdot W_{N/2}^{rk}$$

$$X[k] = G[k] + W_N^k H[k] \quad k = 0, 1, 2, \dots, N-1$$



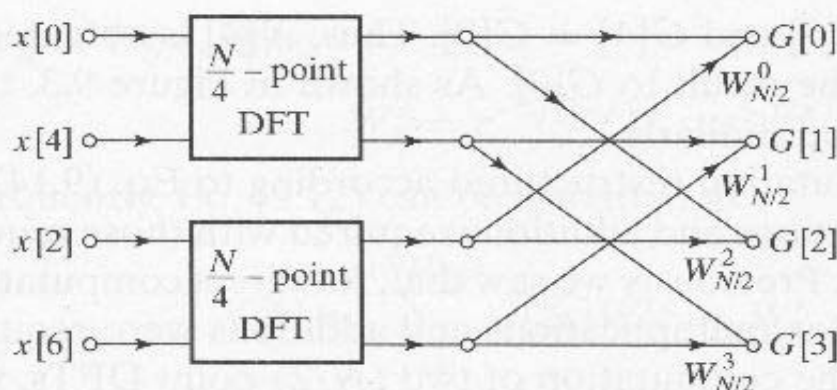
22

DFT – Cálculo Eficiente - Algoritmos de Decimação no Tempo

Aplicando o mesmo princípio para o cálculo de $G[k]$ e $H[k]$ $\rightarrow$ DFT($N/2$)

$$G[k] = \sum_{l=0}^{(N/4-1)} g[2l] \cdot W_{N/4}^{lk} + W_{N/2}^k \sum_{l=0}^{(N/4-1)} g[2l+1] \cdot W_{N/4}^{lk}$$

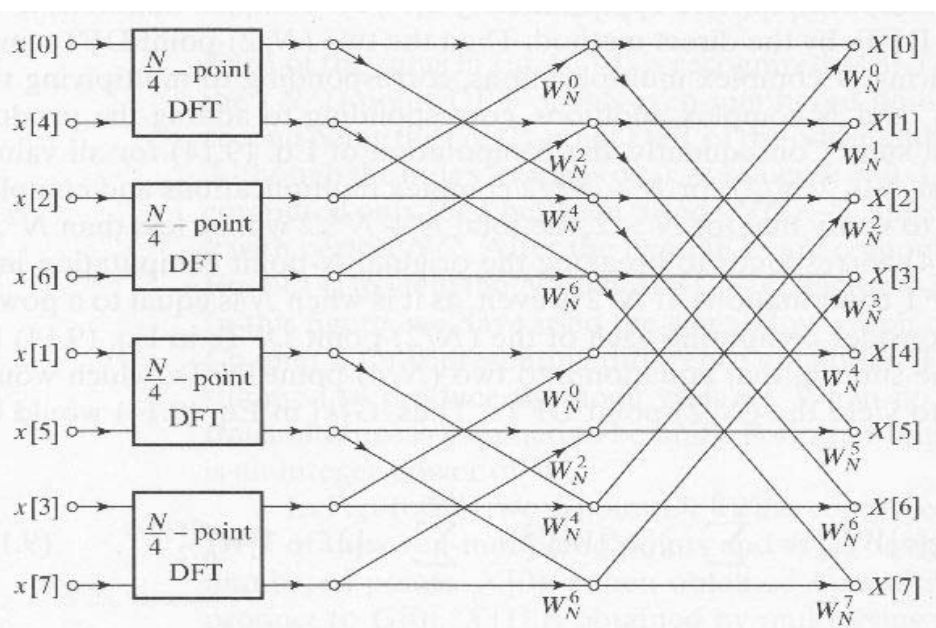
$$H[k] = \sum_{l=0}^{(N/4-1)} h[2l] \cdot W_{N/4}^{lk} + W_{N/2}^k \sum_{l=0}^{(N/4-1)} h[2l+1] \cdot W_{N/4}^{lk}$$



23

DFT – Cálculo Eficiente - Algoritmos de Decimação no Tempo

Temos:



E assim sucessivamente até chegar ao cálculo da DFT(2)

24

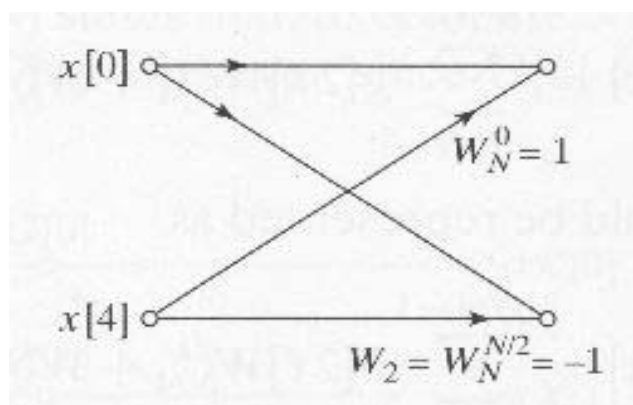
DFT – Cálculo Eficiente - Algoritmos de Decimação no Tempo

DFT de 2 pontos:

$$X[k] = \sum_{n=0}^1 x[n] \cdot W_N^{kn}$$

$$X[0] = x[0] \cdot W_2^{0 \cdot 0} + x[1] \cdot W_2^{0 \cdot 1} = x[0] + x[1]$$

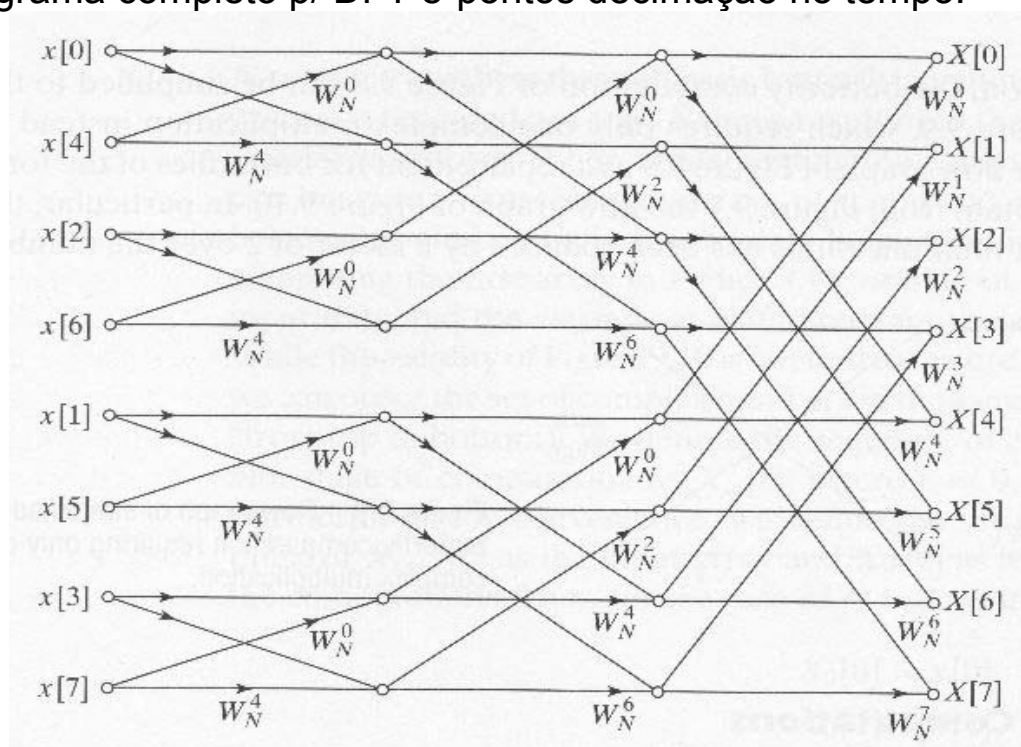
$$X[1] = x[0] \cdot W_2^{1 \cdot 0} + x[1] \cdot W_2^{1 \cdot 1} = x[0] - x[1]$$



25

DFT – Cálculo Eficiente - Algoritmos de Decimação no Tempo

Diagrama completo p/ DFT 8-pontos decimação no tempo:



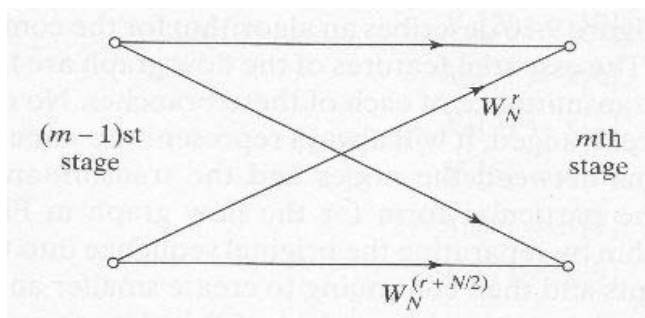
Notar que a complexidade computacional é: $N \cdot \log(N)$

26

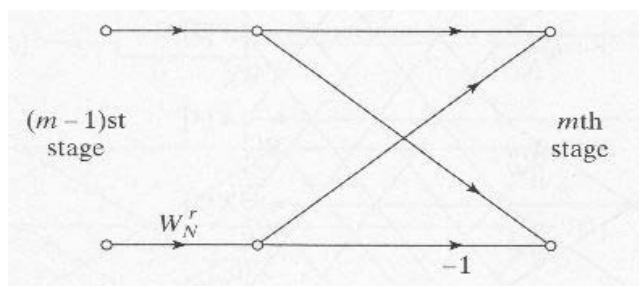
DFT – Cálculo Eficiente - Algoritmos de Decimação no Tempo

Reduzindo ainda mais a complexidade computacional:

Célula básica de computação: **butterfly**



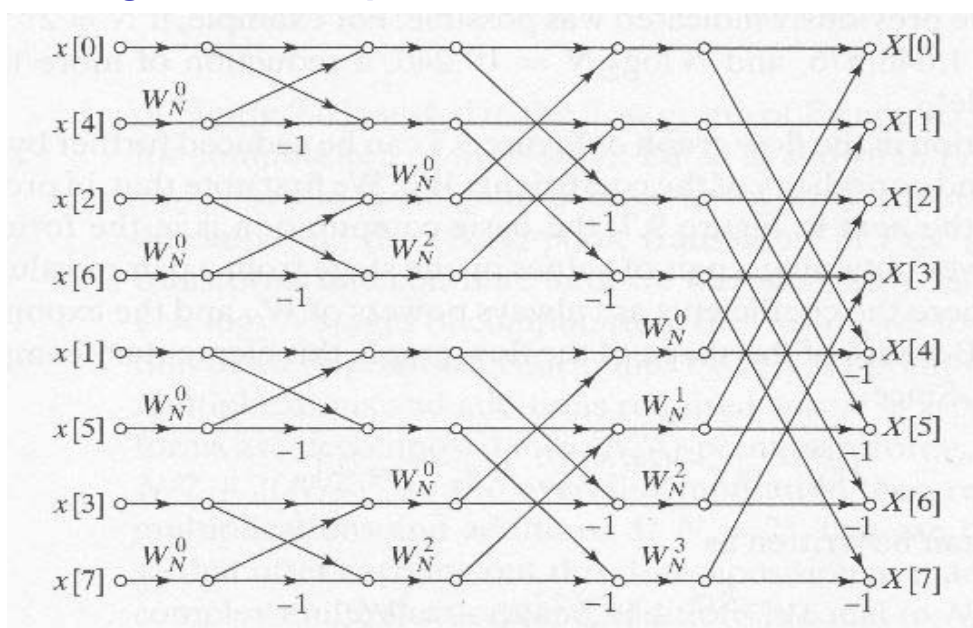
Como: $W_N^{(r+N/2)} = W_N^r \cdot W_N^{N/2} = W_N^r \cdot (-1)$



27

DFT – Cálculo Eficiente - Algoritmos de Decimação no Tempo

Assim, temos o **Algoritmo Completo**



Obs:

- Complexidade computacional, $O(N \cdot \log(N))$;
- Computação **In-Place**, uso da mesma memória p/ entrada e saída;
- Ordem do sinal de entrada $x[n]$

28

DFT – Cálculo Eficiente - Algoritmos de Decimação no Tempo

Ordenação Bit-Reversa

$$X[0] = x[0]$$

$$X[1] = x[4]$$

$$X[2] = x[2]$$

$$X[3] = x[6]$$

$$X[4] = x[1]$$

$$X[5] = x[5]$$

$$X[6] = x[3]$$

$$X[7] = x[7]$$

$$X[000] = x[000]$$

$$X[001] = x[100]$$

$$X[010] = x[010]$$

$$X[011] = x[110]$$

$$X[100] = x[001]$$

$$X[101] = x[101]$$

$$X[110] = x[011]$$

$$X[111] = x[111]$$

DFT – Cálculo Eficiente - Algoritmos de Decimação na Frequência

Decomposição sucessiva de $X[k]$ em parcelas menores

Diversos tipos: mais clássico: p/ N potência de 2

$X[k]$ de N pontos é dividido em 2 sequências de $N/2$ pontos compostas por k ímpares e k pares

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{nk}$$

$$X[2r] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{n(2r)}$$

$P/ X[\text{pares}]$

Que podemos escrever como:

DFT – Cálculo Eficiente - Algoritmos de Decimação na Frequência

$$X[2r] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{n(2r)} \quad \text{P/ } X[\text{pares}]$$

Que podemos escrever como:

$$X[2r] = \sum_{n=0}^{N/2-1} x[n] \cdot W_N^{2nr} + \sum_{n=N/2}^{N-1} x[n] \cdot W_N^{2nr}$$

Substituindo as variáveis no 2º somatório

$$X[2r] = \sum_{n=0}^{N/2-1} x[n] \cdot W_N^{2nr} + \sum_{n=0}^{N/2-1} x[n + N/2] \cdot W_N^{2(n+N/2)r}$$

Notando que: $W_N^{2r(n+N/2)} = W_N^{2rn} \cdot W_N^{rN} = W_N^{2rn} \cdot 1$

Logo:
$$X[2r] = \sum_{n=0}^{N/2-1} x[n] \cdot W_N^{2rn} + \sum_{n=0}^{N/2-1} x[n + N/2] \cdot W_N^{2rn}$$

31

DFT – Cálculo Eficiente - Algoritmos de Decimação na Frequência

Lembrando que: $W_N^{2rn} = W_{N/2}^{rn}$

Temos que:

$$X[2r] = \sum_{n=0}^{N/2-1} x[n] \cdot W_N^{2rn} + \sum_{n=0}^{N/2-1} x[n + N/2] \cdot W_N^{2rn}$$

Pode ser escrito como:

$$X[2r] = \sum_{n=0}^{N/2-1} (x[n] + x[n + N/2]) \cdot W_{N/2}^{rn}$$

De modo análogo p/ k ímpares podemos escrever:

$$X[2r + 1] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{n(2r+1)} \quad \text{P/ } X[\text{ímpares}]$$

Que podemos escrever como:

32

DFT – Cálculo Eficiente - Algoritmos de Decimação na Frequência

$$X[2r+1] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{n(2r+1)} \quad \text{P/ } X[\text{ímpares}]$$

Que podemos escrever como:

$$X[2r+1] = \sum_{n=0}^{N/2-1} x[n] \cdot W_N^{n(2r+1)} + \sum_{n=N/2}^{N-1} x[n] \cdot W_N^{n(2r+1)}$$

Substituindo variáveis no 2º somatório

$$X[2r+1] = \sum_{n=0}^{N/2-1} x[n] \cdot W_N^{n(2r+1)} + \sum_{n=0}^{N/2-1} x[n + N/2] \cdot W_N^{(n+N/2)(2r+1)}$$

Notando que: $W_N^{(n+N/2)(2r+1)} = W_N^{n(2r+1)} \cdot W_N^{(2r+1)N/2} = W_N^{n(2r+1)} \cdot (-1)$

Logo: $X[2r+1] = \sum_{n=0}^{N/2-1} x[n] \cdot W_N^{n(2r+1)} - \sum_{n=0}^{N/2-1} x[n + N/2] \cdot W_N^{n(2r+1)}$

33

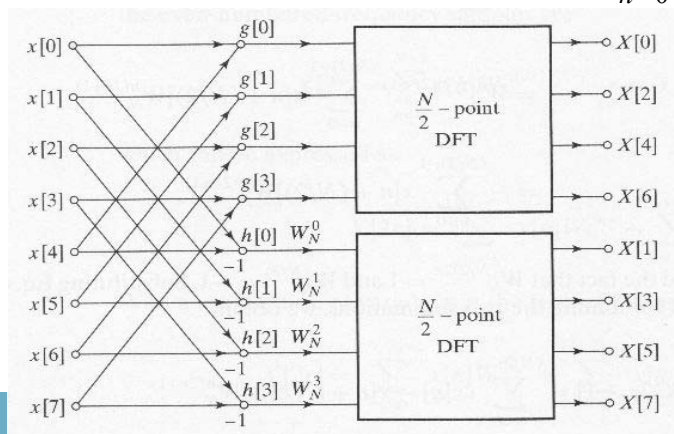
DFT – Cálculo Eficiente - Algoritmos de Decimação na Frequência

$$X[2r+1] = \sum_{n=0}^{N/2-1} (x[n] - x[n + N/2]) \cdot W_N^{n(2r+1)}$$

Para **k ímpares**:

$$X[2r+1] = \sum_{n=0}^{N/2-1} (x[n] - x[n + N/2]) \cdot W_N^{2nr} \cdot W_N^n$$

$$X[2r+1] = \sum_{n=0}^{N/2-1} (x[n] - x[n + N/2]) \cdot W_N^n \cdot W_N^{nr}$$



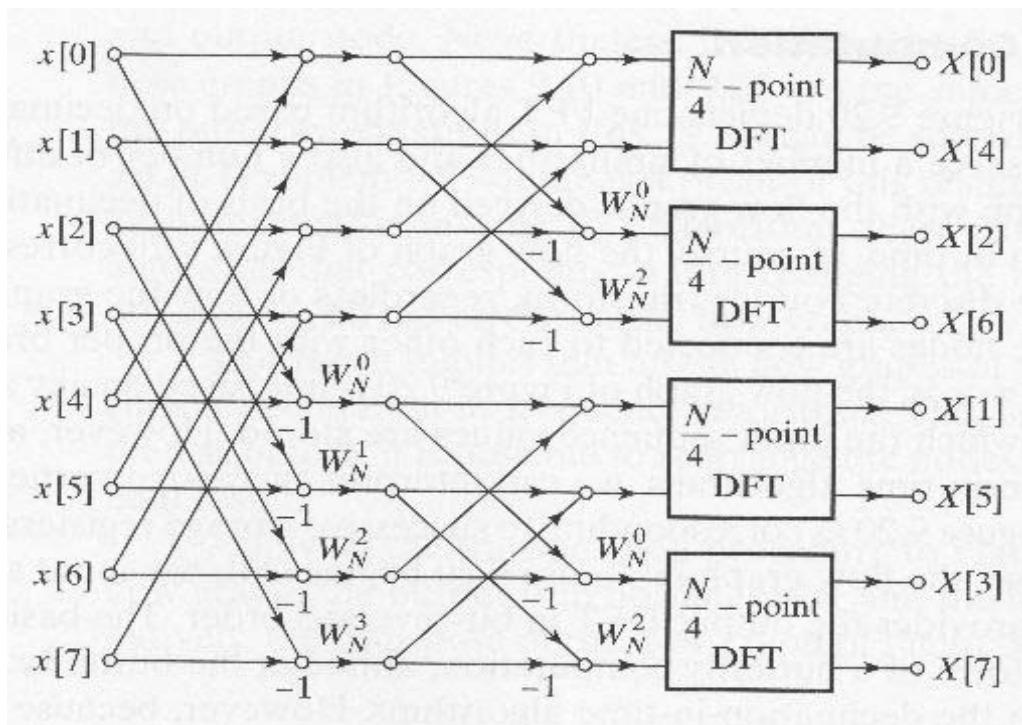
Para **k pares**:

$$X[2r] = \sum_{n=0}^{N/2-1} (x[n] + x[n + N/2]) \cdot W_N^{rn}$$

34

DFT – Cálculo Eficiente - Algoritmos de Decimação na Frequência

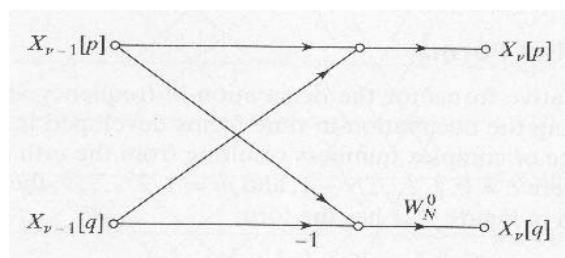
Aplicando o mesmo procedimento para o cálculo da DFT N/2 pontos



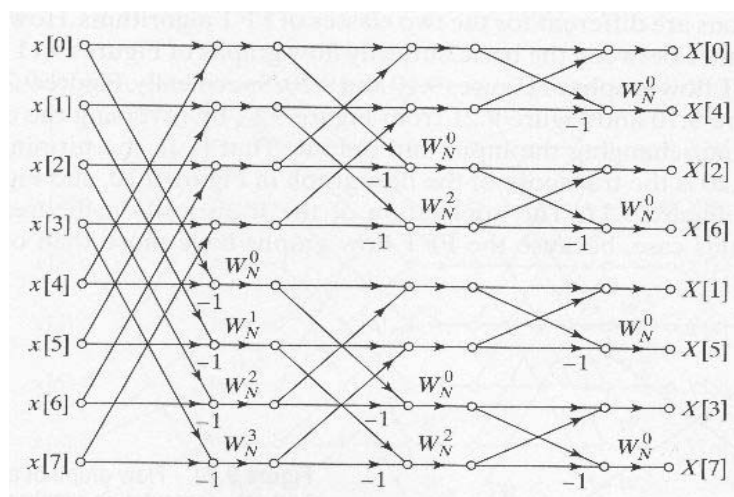
35

DFT – Cálculo Eficiente - Algoritmos de Decimação na Frequência

E assim sucessivamente até a DFT de 2 pontos, calculada por:



Algoritmo completo p/ DFT(8) decimação em Frequência:



Obs:

- O(N.log(N))
- Computação **In-Place**
- Saída bit-reverso

36

DFT – Cálculo Eficiente - Convolução

Método Directo:

$$y[n] = \sum_{k=0}^{N-1} x[k] \cdot h[n-k]$$

$$N \geq N_1 + N_2 - 1 = 2N - 1$$

Complexidade: $O(2N^2)$

Por FFT:

$$\begin{aligned} x[n] &\xrightarrow{FFT(2N)} X[k] & Y[k] &= X[k] \cdot H[k] \\ h[n] &\xrightarrow{FFT(2N)} H[k] & Y[k] &\xrightarrow{IFFT(2N)} y[n] \end{aligned}$$

Complexidade: $O(3.2N \cdot \log(2N) + 2N)$

TABLE 13-1 COMPUTING TIMES IN SECONDS

N	Direct Method	FFT Method	Speed Factor
16	0.0008	0.003	0.27
32	0.003	0.007	0.43
64	0.012	0.015	0.8
128	0.047	0.033	1.4
256	0.19	0.073	2.6
512	0.76	0.16	4.7
1024	2.7	0.36	7.5
2048	11.0	0.78	14.1
4096	43.7	1.68	26.0

37

FFT: Fast Fourier Transform

- É um algoritmo com complexidade logarítmica para o cálculo da DFT; não é outra transformada
- Para valores elevados de N é muito mais eficiente do que a DFT
- Existem algoritmos de decimação no tempo e na frequência (já visto!!!)

Número de multiplicações complexas para vectores com N pontos:

- DFT - N^2
- FFT - $N \log_2(N)$

38

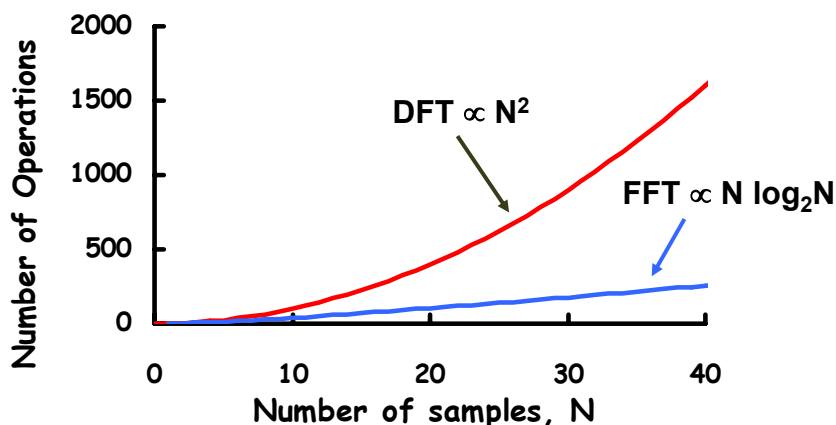
FFT: Exemplos de aplicações

- Convolução circular (e linear através da circular)
- Métodos de convolução rápida:
 - overlap and add; overlap and save;
- Detecção de componentes de frequência
- Filtragem

Manuel A. E. Baptista
Ernesto R. Afonso



FFT - Vantagens



N	DFT	Radix-2
4	16	4
32	1024	80
128	16384	448
1024	1048576	5120

O uso de DSPs e PLDs influenciou o desenho dos algoritmos nos anos **60** e **70**: a contagem das multiplicações como “factor de qualidade”. **Agora**: o número de adições/acesso à memória (SW) e os custos de comunicação (HW) são também importantes.

Manuel A. E. Baptista
Ernesto R. Afonso



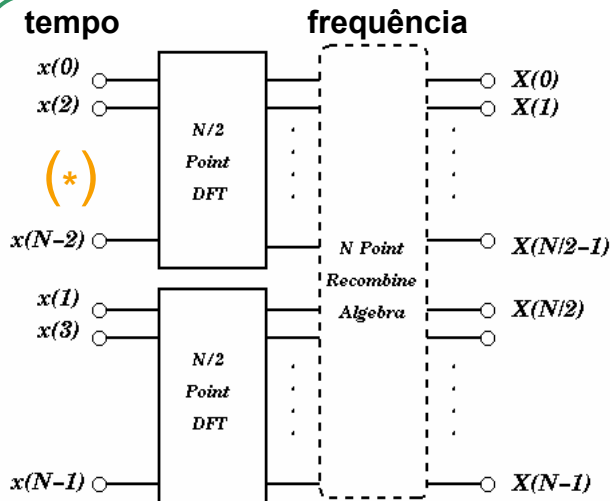
Filosofia da FFT

Filosofia geral (a ser aplicada recursivamente): **divide & conquer**.

$$\underbrace{\sum \text{custo}(\text{sub-problemas}) + \text{custo}(\text{mapeamento})}_{\text{custo total}} < \text{custo}(\text{problema original})$$

Os diferentes algoritmos apresentam custos distintos.

Exemplo: Algoritmo - Decimação no Tempo



Passo 1: Decomposição no Domínio do Tempo.

N points signal $\rightarrow$ N, 1-point signals (interlace decomposition). Shuffled input data (bit reversal). **$\log_2 N$ estados**.

(*) : apresentada apenas a 1.^a decomposição.

Passo 2: 1 point input spectra calculation.

(Nothing to do!)

Passo 3: Frequency domain synthesis.

N spectra synthesised into one.

FFT – Árvore de Família

Divide & conquer

$$N : \text{MDC}^{(*)}(N_1, N_2) = 1$$

N_1, N_2 co-prime. Ex: $240 = 16 \cdot 3 \cdot 5$

Cost: **SUB-PROBLEMAS**.

- No twiddle-factors calculations.
- Easier mapping (permutations).
- Some algorithms:
Good Thomas, Kolba,
Parks, Winograd.

$$N : \text{MDC}(N_1, N_2) > 1$$

Ex: $N = 2^n$

Cost: **MAPEAMENTO**.

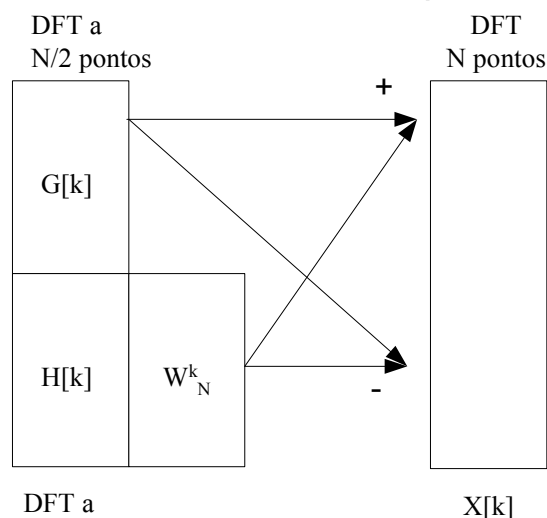
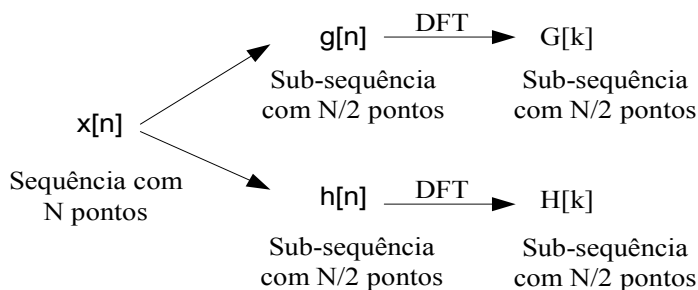
- Twiddle-factors calculations.
- Easier sub-problems.
- Some algorithms:
Cooley Tukey,
Decimation in time / in frequency
Radix 2 Radix 4,
Split radix.

(*) MDC= Máximo Divisor Comum

FFT: Algoritmo Cooley-Tukey

Decimação no tempo

A DFT de N pontos é sucessivamente dividida em duas DFT de N/2 pontos



- $g[n]$, sub sequência com as amostras de $x[n]$ que estão em índice par
- $h[n]$, sub sequência com as amostras de $x[n]$ que estão em índice ímpar

$$X[k] = G[k] + W_N^k H[k]$$

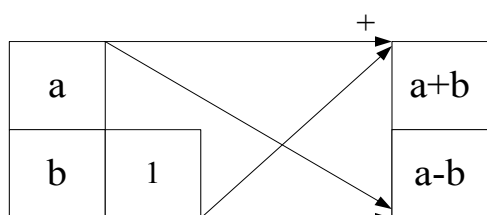
$$X[k + \frac{N}{2}] = G[k] - W_N^k H[k]$$

$$W_N^k = e^{-j \frac{2\pi k}{N}}$$

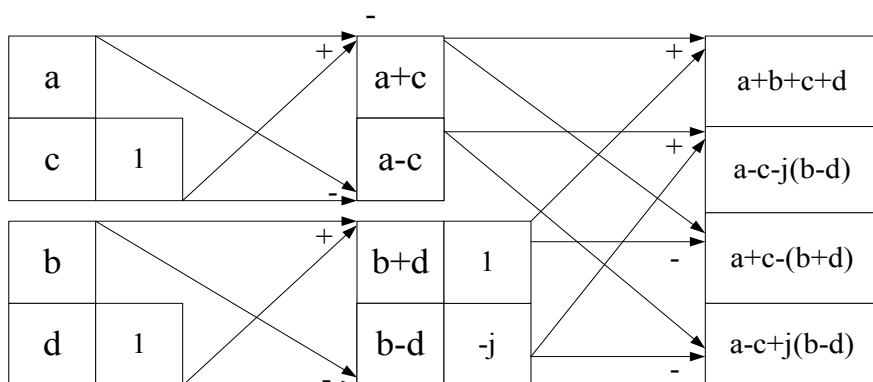
43

FFT: Algoritmo Cooley-Tukey

- A demonstração deste algoritmo pode ser encontrada em [1] e [2]
- As butterflies: cálculo da FFT a 2 e 4 pontos



$$(\Rightarrow) \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a+b \\ a-b \end{bmatrix}$$



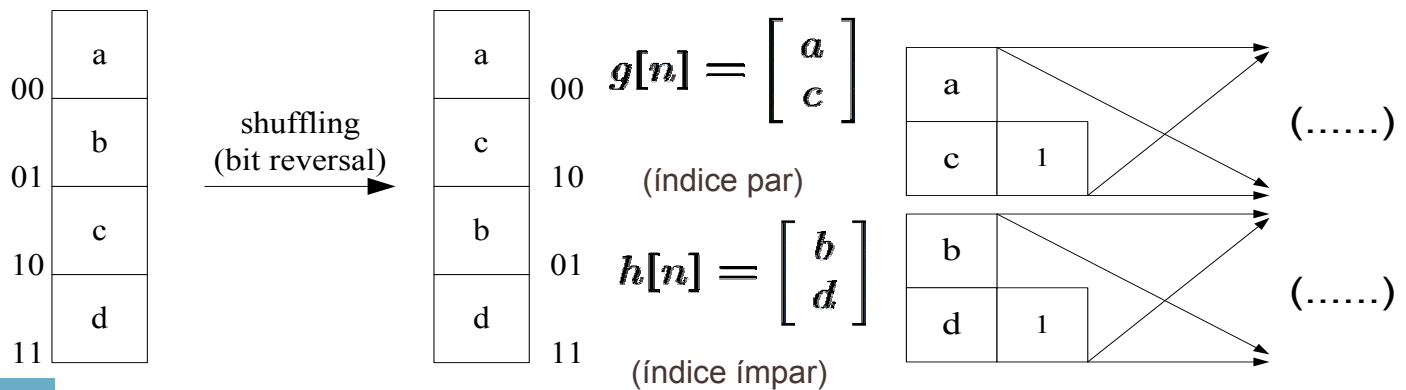
$$(\Rightarrow) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} a+b+c+d \\ a-c-j(b-d) \\ a+c-(b+d) \\ a-c+j(b-d) \end{bmatrix}$$

44

FFT: Algoritmo Cooley-Tukey

O *shuffling* realizado através de *bit-reversal* separa a sequência nas duas sub-sequências com as amostras de índice par e ímpar, designadas por $g[n]$ e $h[n]$, respectivamente.

$$x[n] = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \xrightarrow{\text{DFT}_4} X[k] = \begin{bmatrix} a + b + c + d \\ a - c - j(b - d) \\ a + c - (b + d) \\ a - c + j(b - d) \end{bmatrix}$$



A IFFT: cálculo via FFT

A estrutura de cálculo da FFT (que calcula a DFT) também pode ser utilizada para determinar a IDFT

$$x[n] = \text{IDFT}[X[k]] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N} = \frac{1}{N} \left(\sum_{k=0}^{N-1} X^*[k] e^{j2\pi kn/N} \right)^* \\ = \frac{1}{N} \text{DFT}[X[k]^*]^*$$

Para calcular a IDFT, via IFFT:

1. Conjugar o espectro $X[k]$ e obter $X[k]^*$;
2. Calcular a FFT de $X[k]^*$ (realizando o respectivo *shuffling*);
3. Conjugar o resultado da FFT e dividi-lo por N .

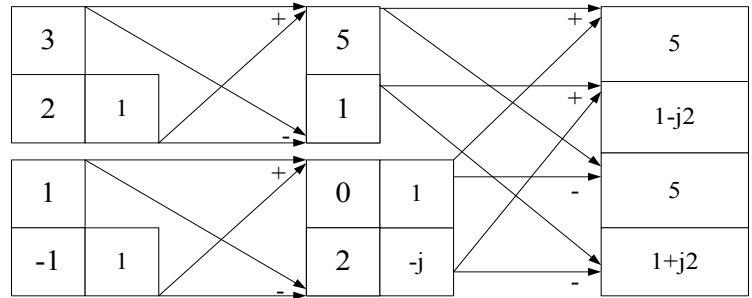
Exemplos de cálculo da FFT e IFFT

$$x[n] = \{3, 1, 2, -1\}$$

$$g[n] = \{3, 2\}$$

$$h[n] = \{1, -1\}$$

$$X[k] = \{5, 1 - j2, 5, 1 + j2\}$$



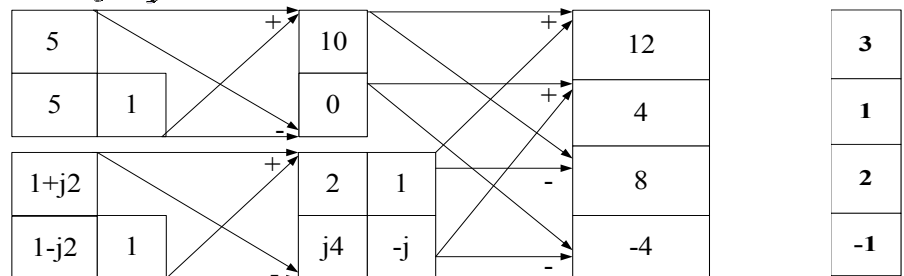
1. Conjugar

$$X^*[k] = \{5, 1 + j2, 5, 1 - j2\}$$

3. Dividir e conjugar

$$G[k] = \{5, 5\}$$

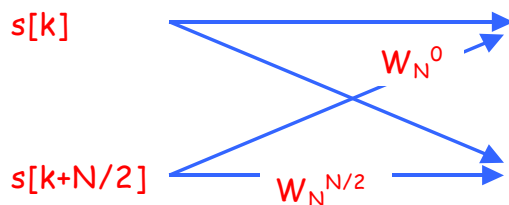
$$H[k] = \{1 + j2, 1 - j2\}$$



2. Calcular a FFT

$$x[n] = \{3, 1, 2, -1\}$$

FFT- (Alguns) conceitos & notas



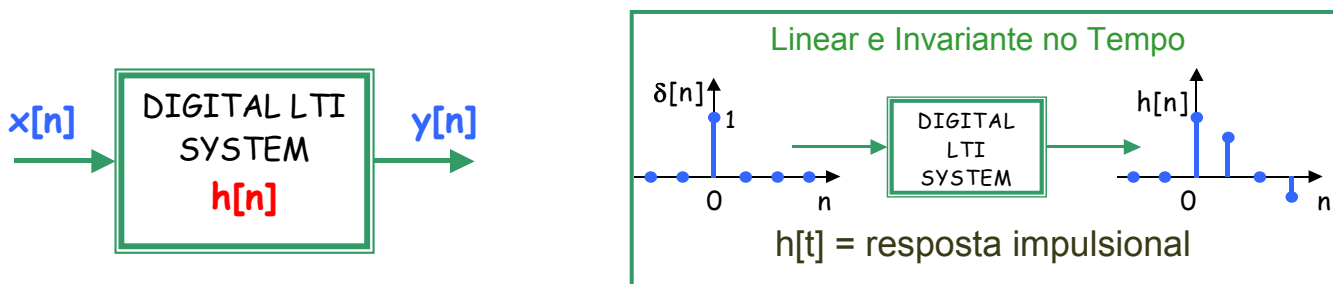
Butterfly: elemento básico do cálculo da FFT.

Aproximação Dual: dados a serem reordenados no tempo e na frequência!

1. Decimação no Tempo $\Rightarrow$ time data shuffling.
2. Decimação na Frequência $\Rightarrow$ frequency data shuffling.
3. Computação In-place: no auxiliary storage needed, allowed by most algorithms.
4. DFT pruning: only few bins needed or different from zero $\Rightarrow$ only they get calculated (ex: algoritmo de Goertzel).
5. Caso de dados reais: Mirroring effect in DFT coeffs. $\Rightarrow$ only half of them calculated.
6. N Potências de 2: Many common FFT algorithms work with power-of-two number of inputs. When they are not $\Rightarrow$ pad inputs with zeroes.

FFT - Análise espectral (considerações)

Análise dum Sistema: medição da relação entrada-saída.



$$\begin{array}{ccccc}
 x[n] & h[n] & y[n] = x[n] * h[n] = \sum_{m=0}^{\infty} x[n-m] \cdot h[m] & \boxed{y[n] \text{ obtida de } \{ x[n], h[t] \}} \\
 \updownarrow & \updownarrow & \updownarrow & \\
 X(f) & H(f) & Y(f) = X(f) \cdot H(f) &
 \end{array}$$

$H(f)$: Função de Transferência LTI

➡ A Função de Transferência, pode ser estimada através de $Y(f) / X(f)$