A DFT, a FFT e a DCT

Sistemas de Telecomunicações Definidos por Software

&

Processamento de Sinal em Tempo Real

Tópicos abordados

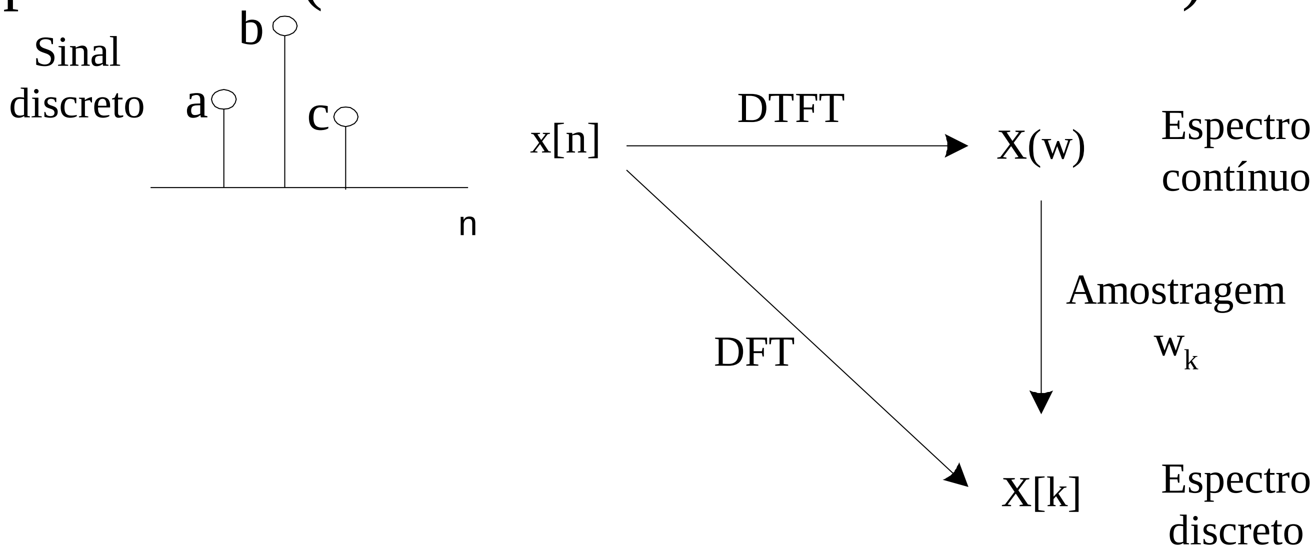
- DFT – Discrete Fourier Transform
  - Cálculo matricial
- FFT – Fast Fourier Transform
  - Algoritmo de Cooley-Tukey
  - STFT – Short Time Fourier Transform
  - Espectrograma
- DCT – Discrete Cosine Transform
  - Cálculo matricial

DFT – Discrete Fourier Transform

- Assume que o sinal é periódico com N amostras por período
- Realiza a amostragem do espectro nas frequências dadas por

$$w_k = \frac{2\pi k}{N}, \qquad k \in \{0, 1, \ldots, N-1\}.$$

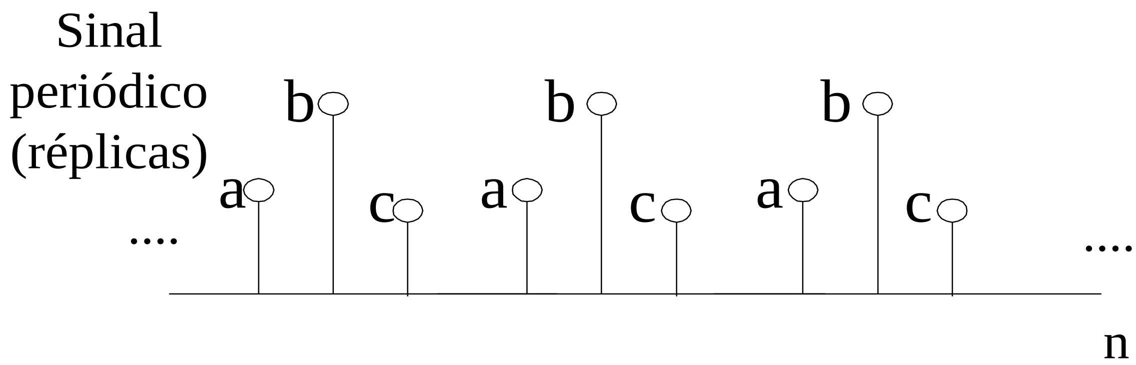
- Amostra/discretiza, de forma equi-espaçada, o espectro contínuo obtido pela DTFT (Discrete-Time Fourier Transform)



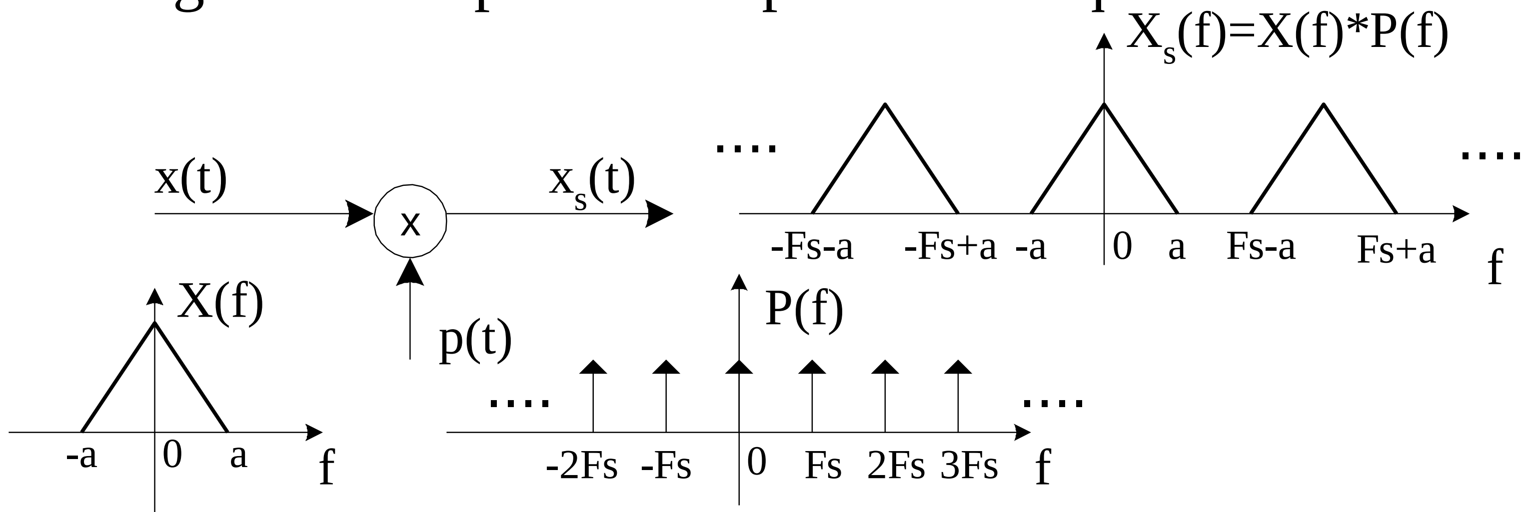
• Por exemplo, com N=4 analisam-se as frequências $\{0, \frac{\pi}{2}, \pi, -\frac{\pi}{2}\}$

Amostragem do espectro

- A amostragem na frequência causa réplicas no tempo (espectro discreto implica sinal periódico no tempo)



- A amostragem no tempo causa réplicas na frequência



DTFT – Discrete-Time Fourier Transform

Transformada directa (**DTFT**)

$$X(w) = \mathrm{DTFT}[x[n]] = \sum_{n=-\infty}^{\infty} x[n]e^{-jwn}$$

Transformada inversa (**IDTFT**)

$$x[n] = \mathrm{IDTFT}[X(w)] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(w)e^{jwn}dw$$

Exemplos:

$$x[n] = 3\delta[n] - 2\delta[n-1] \qquad X(w) = 3 - 2e^{-jwn}$$

$$y[n] = \alpha^n u[n], \quad |\alpha| < 1 \qquad Y(w) = \frac{1}{1-\alpha e^{-jw}}$$

$$z[n] = \frac{W_o}{\pi}\,\mathrm{sinc}\left(\frac{W_o}{\pi}n\right) \qquad Z(w) = \prod\left(\frac{w}{2W_o}\right)$$

DFT e IDFT: definição

- Considerando que o sinal x[n] é periódico com N amostras por período, temos um espectro **discreto**:

$$X(w_k) = \sum_{n=0}^{N-1} x[n] e^{-j w_k n}$$

com $w_k = \frac{2\pi k}{N}, \; k \in \{0, 1, \ldots, N-1\}$

Equação de análise ou transformada directa (**DFT**): tempo para a frequência

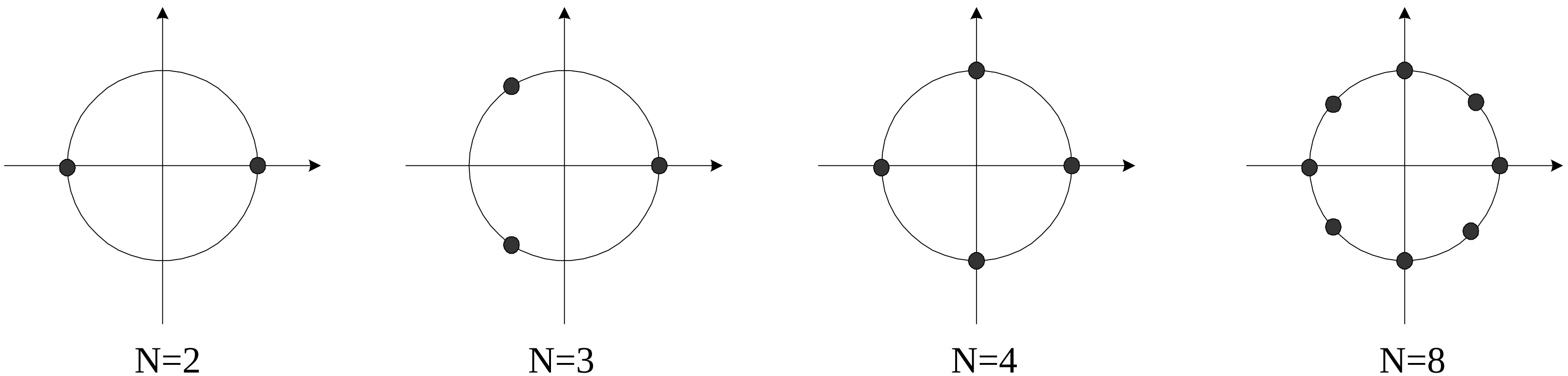
$$X[k] = \mathrm{DFT}[x[n]] = \sum_{n=0}^{N-1} x[n] e^{-j w_k n} = \sum_{n=0}^{N-1} x[n] e^{-\frac{j 2\pi k n}{N}}$$

Equação de síntese ou transformada inversa (**IDFT**): frequência para o tempo

$$x[n] = \mathrm{IDFT}[X[k]] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{\frac{j 2\pi k n}{N}}$$

DFT: frequências analisadas

Análise gráfica das frequências amostradas, em função do número de pontos N



N	Frequências
2	$\{0, \pi\}$
3	$\{0, \frac{2\pi}{3}, -\frac{2\pi}{3}\}$
4	$\{0, \frac{\pi}{2}, \pi, -\frac{\pi}{2}\}$
8	$\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, -\frac{3\pi}{4}, -\frac{\pi}{2}, -\frac{\pi}{4}\}$

Frequências analisadas:

$$w_k = \frac{2\pi k}{N}$$

$$k \in \{0, 1, \ldots, N-1\}.$$

Cálculo da DFT a 4 pontos

- Os coeficientes da DFT_4 do sinal x[n] são obtidos por:

$$X[0] = \sum_{n=0}^{3} x[n]e^{-\frac{j2\pi 0n}{4}} = \sum_{n=0}^{3} x[n] = x[0] + x[1] + x[2] + x[3]$$

$$X[1] = \sum_{n=0}^{3} x[n]e^{-\frac{j2\pi 1n}{4}} = x[0] + x[1]e^{-\frac{j2\pi 1}{4}} + x[2]e^{-\frac{j2\pi 2}{4}} + x[3]e^{-\frac{j2\pi 3}{4}}$$

$$X[2] = \sum_{n=0}^{3} x[n]e^{-\frac{j2\pi 2n}{4}} = x[0] + x[1]e^{-\frac{j2\pi 2}{4}} + x[2]e^{-\frac{j2\pi 2.2}{4}} + x[3]e^{-\frac{j2\pi 2.3}{4}}$$

$$X[3] = \sum_{n=0}^{3} x[n]e^{-\frac{j2\pi 3n}{4}} = x[0] + x[1]e^{-\frac{j2\pi 3}{4}} + x[2]e^{-\frac{j2\pi 3.2}{4}} + x[3]e^{-\frac{j2\pi 3.3}{4}}$$

Cálculo da DFT a 4 pontos

Analisando os coeficientes e os vectores de base tem-se:

$$X[0] = [1 \;\; 1 \;\; 1 \;\; 1] \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

$$X[1] = \left[1 \;\; e^{-\frac{j2\pi 1}{4}} \;\; e^{-\frac{j2\pi 2}{4}} \;\; e^{-\frac{j2\pi 3}{4}}\right] \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

$$X[2] = \left[1 \;\; e^{-\frac{j2\pi 2}{4}} \;\; e^{-\frac{j2\pi 2.2}{4}} \;\; e^{-\frac{j2\pi 2.3}{4}}\right] \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

$$X[3] = \left[1 \;\; e^{-\frac{j2\pi 3}{4}} \;\; e^{-\frac{j2\pi 3.2}{4}} \;\; e^{-\frac{j2\pi 3.3}{4}}\right] \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

Cálculo matricial da DFT

Agrupando os vectores base em matriz, tem-se:

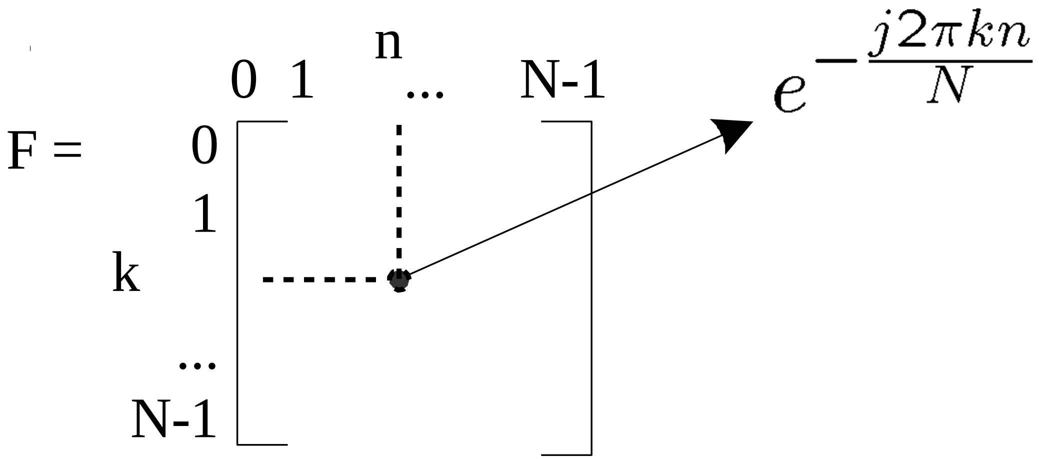
$$\mathbf{y} = \mathrm{DFT}_4[\mathbf{x}] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{-\frac{j2\pi 1}{4}} & e^{-\frac{j2\pi 2}{4}} & e^{-\frac{j2\pi 3}{4}} \\ 1 & e^{-\frac{j2\pi 2}{4}} & e^{-\frac{j2\pi 2.2}{4}} & e^{-\frac{j2\pi 2.3}{4}} \\ 1 & e^{-\frac{j2\pi 3}{4}} & e^{-\frac{j2\pi 3.2}{4}} & e^{-\frac{j2\pi 3.3}{4}} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

Simplificando, tem-se o cálculo matricial da DFT:

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

DFT: cálculo matricial

- Genericamente, colocando o vector **x** em evidência, estabelece-se a matriz F (operador DFT)



Cálculo matricial da DFT

x é o sinal no tempo

y são os coeficientes da DFT

$$\mathbf{y} = DFT[x] = \mathbf{F}x$$

- Por exemplo, para N=2, 3 e 4 tem-se, respectivamente:

$$\mathbf{F}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \mathbf{F}_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-j\frac{2\pi}{3}} & e^{-j\frac{2\pi 2}{3}} \\ 1 & e^{-j\frac{2\pi 2}{3}} & e^{-j\frac{2\pi 4}{3}} \end{bmatrix} \qquad \mathbf{F}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

IDFT: cálculo matricial

- De forma análoga estabelece-se o operador que calcula a IDFT
- Verifica-se que este é dado por $\mathbf{F}^{-1}$: $\quad \mathbf{x} = \mathbf{F}^{-1}\mathbf{y}$
- Por sua vez $\mathbf{F}^{-1} = \frac{1}{N}\mathbf{F}^*$
- Tem-se assim o cálculo matricial da IDFT:

$$\mathbf{x} = \mathrm{IDFT}[y] = \frac{1}{N}\mathbf{F}^* y$$

$$F^{-1} = \frac{1}{N}\begin{array}{c} \\ 0 \\ 1 \\ n \\ \vdots \\ N-1 \end{array}\begin{array}{c} 0 \quad 1 \quad k \quad \dots \quad N-1 \\ \left[\begin{array}{ccccc} & & & & \\ & & & & \\ & & e^{\frac{j2\pi kn}{N}} & & \\ & & & & \\ & & & & \end{array}\right] \end{array}$$

Cálculo matricial da IDFT

x é o sinal no tempo

y são os coeficientes da DFT

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

Número de multiplicações complexas para vectores com N pontos:

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

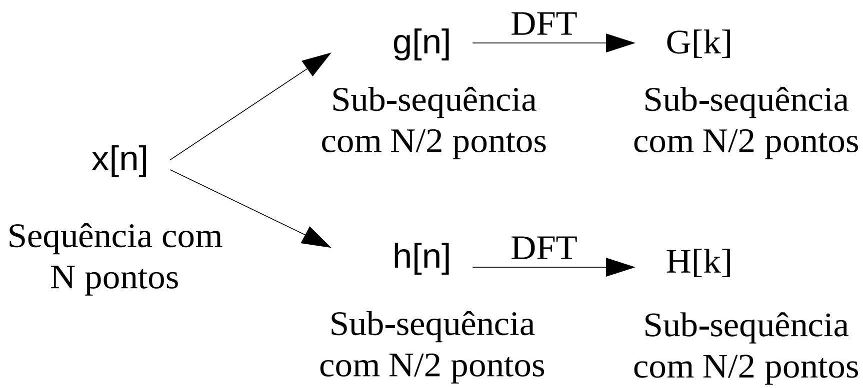
FFT: Exemplos de aplicações

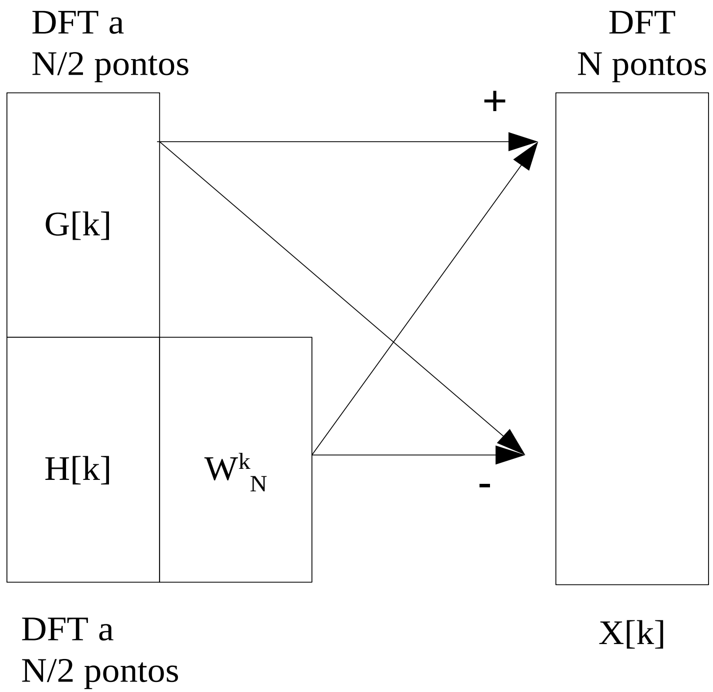
- Convolução circular (e linear através da circular)
- Métodos de convolução rápida:
  - overlap and add; overlap and save;
- Detecção de componentes de frequência
- Filtragem

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 <u>par</u>

- h[n], sub-sequência com as amostras de x[n] que estão em índice <u>ímpar</u>

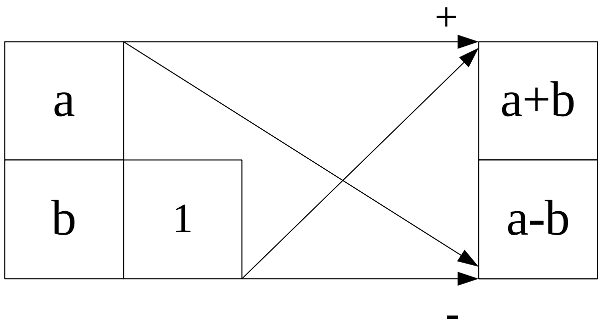
$$X[k] = G[k] + W_N^K H[k]$$

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

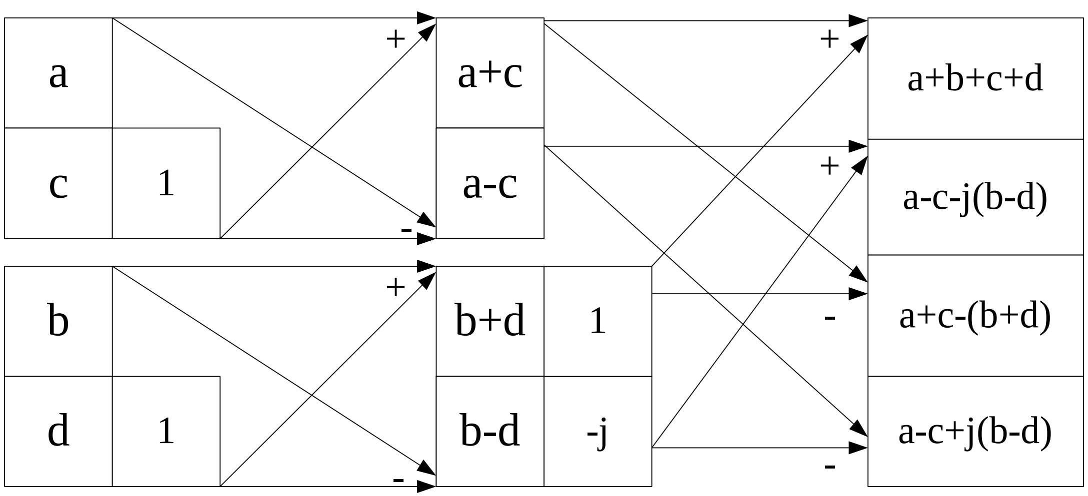
$$W_N^K = e^{-j\frac{2\pi k}{N}}$$

FFT: Algoritmo Cooley-Tukey

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



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



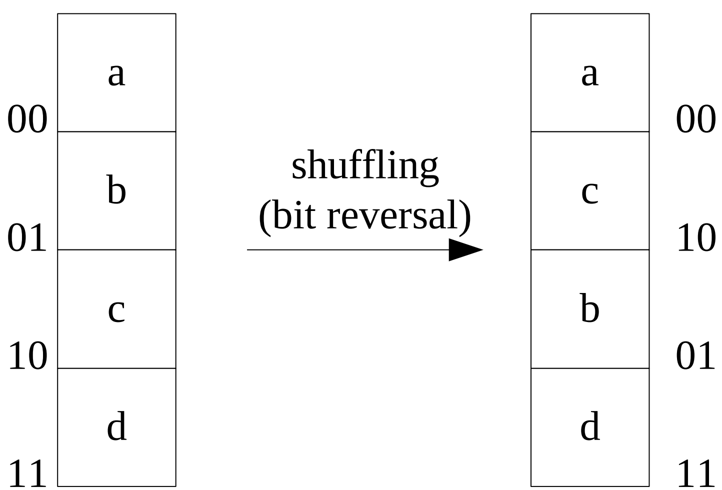
$$(=)\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

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

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.

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

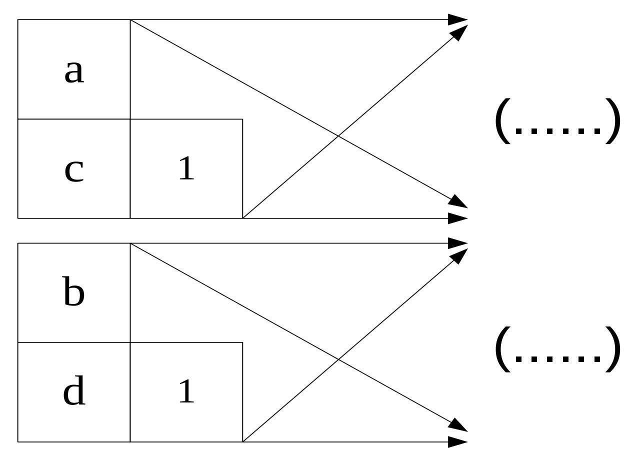


$$g[n] = \begin{bmatrix} a \\ c \end{bmatrix}$$

(índice par)

$$h[n] = \begin{bmatrix} b \\ d \end{bmatrix}$$

(índice ímpar)



A IFFT: cálculo via FFT

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

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

$$= \frac{1}{N}\mathrm{DFT}[X[k]^*]^*$$

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

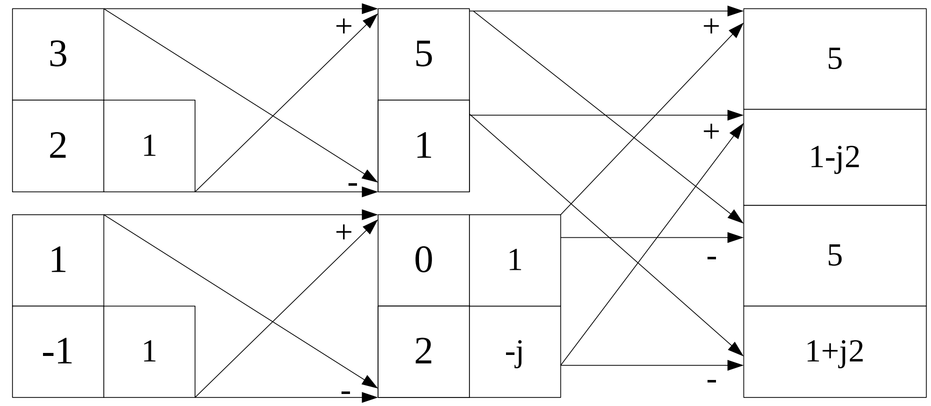
Exemplos de cálculo da FFT e IFFT

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

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

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

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

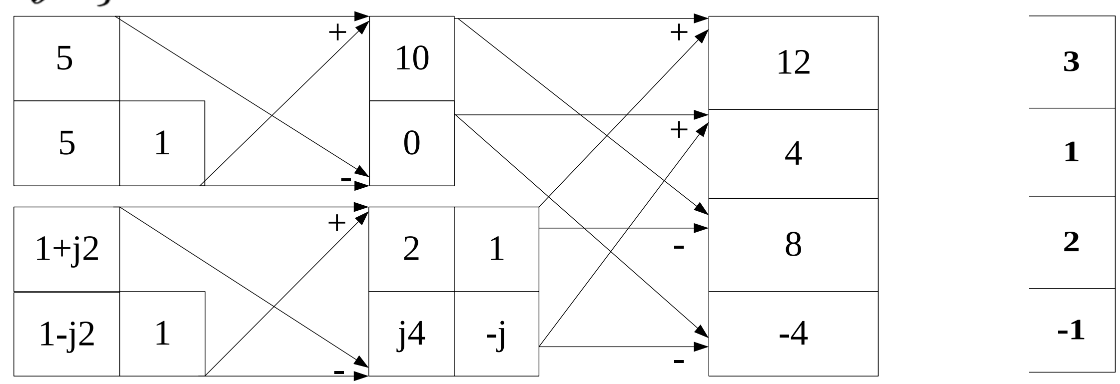


1. Conjugar

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

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

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



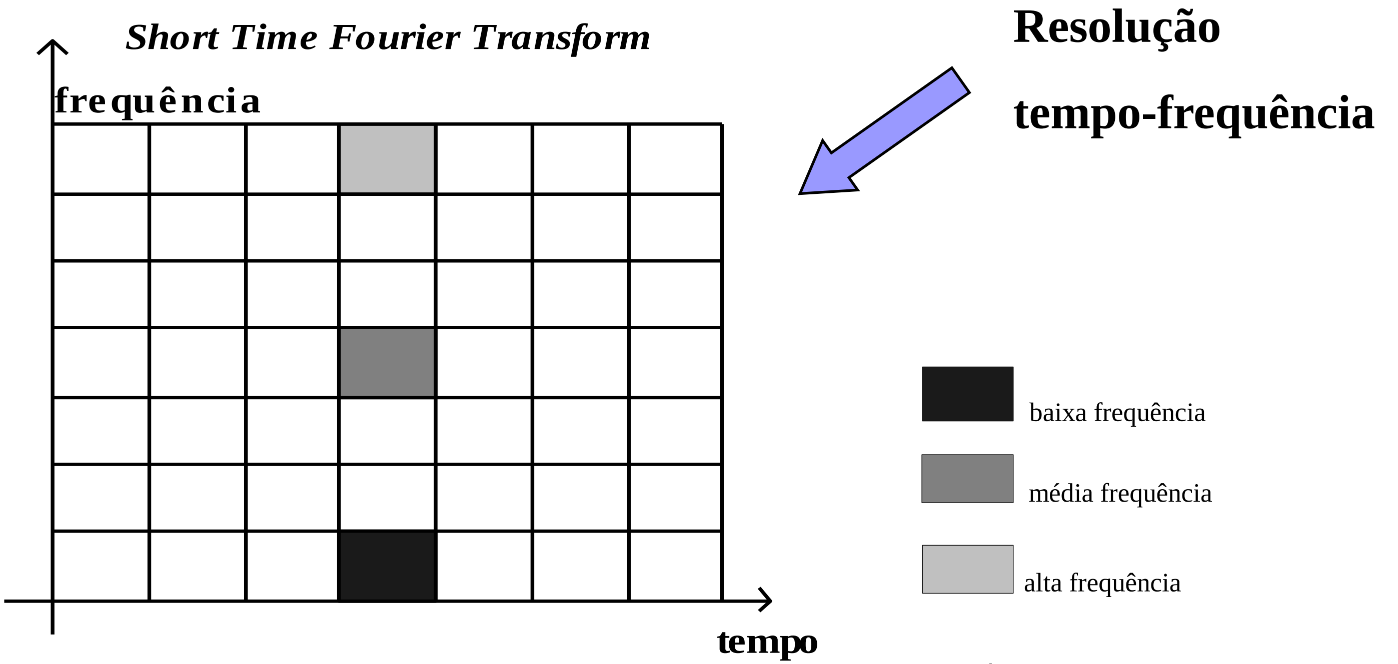
2. Calcular a FFT

3. Dividir e conjugar

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

STFT - Short Time Fourier Transform

- A ideia base é efectuar o cálculo da transformada de Fourier (FFT) em tempos localizados (janelas)
- Motivação: as componentes de frequência dominantes variam com o tempo, tal como acontece por exemplo num sinal de audio (voz)

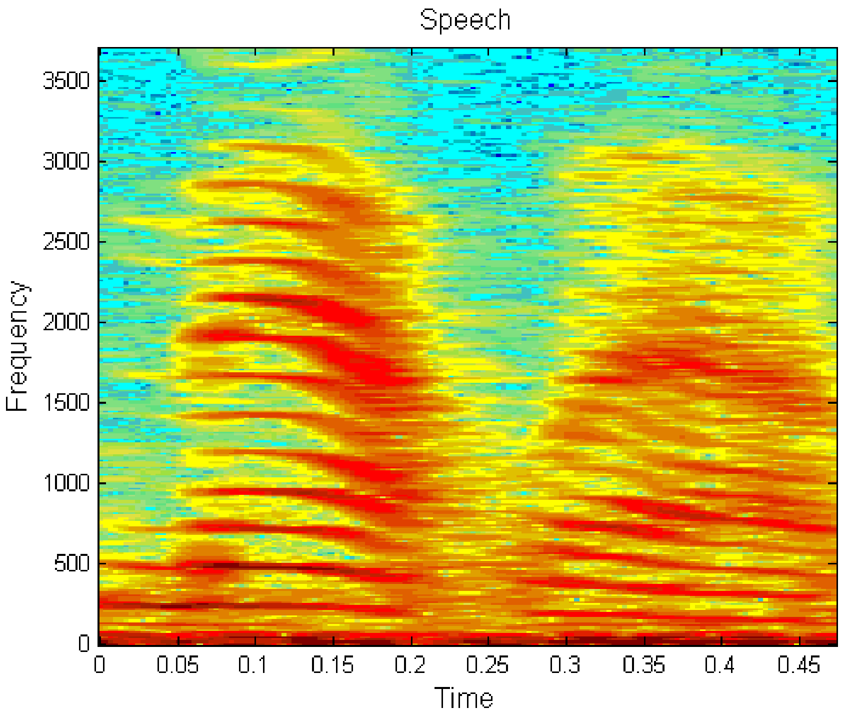


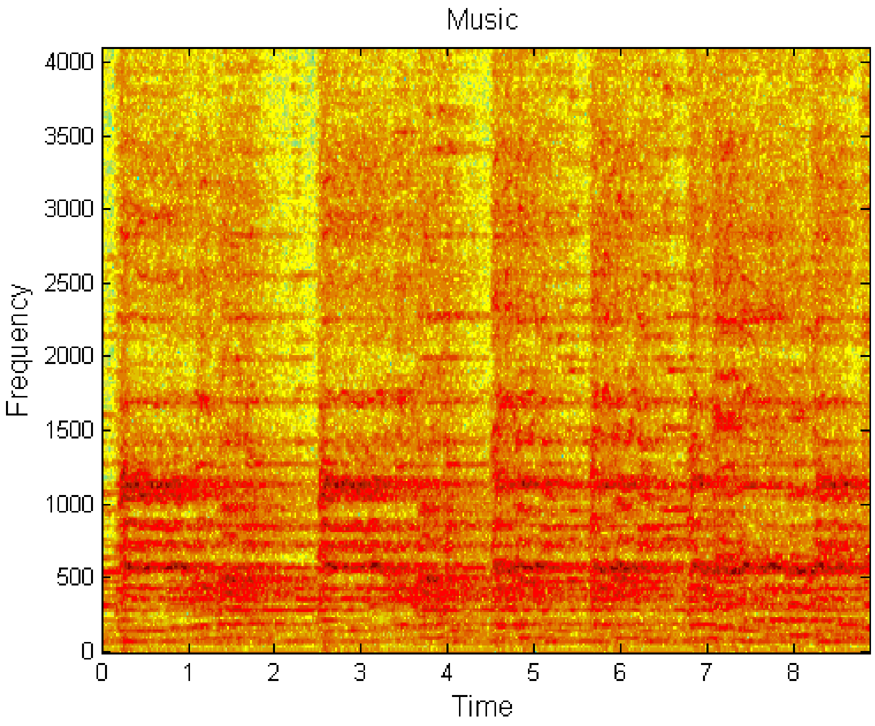
Espectrograma: tempo-frequência

Cálculo da STFT e representação tempo-frequência do sinal

Sinal de fala

Sinal de música

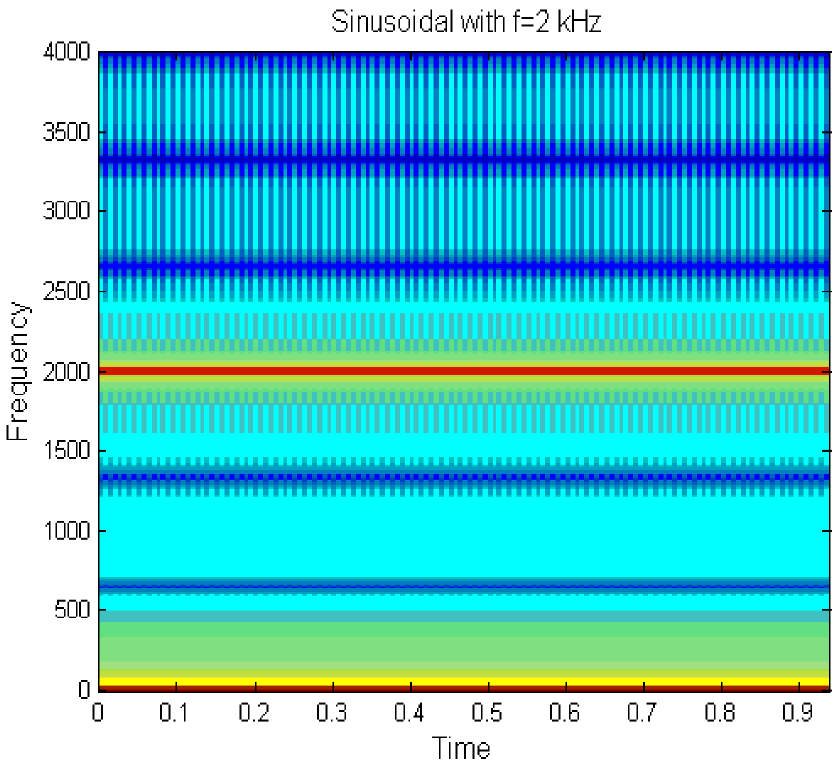


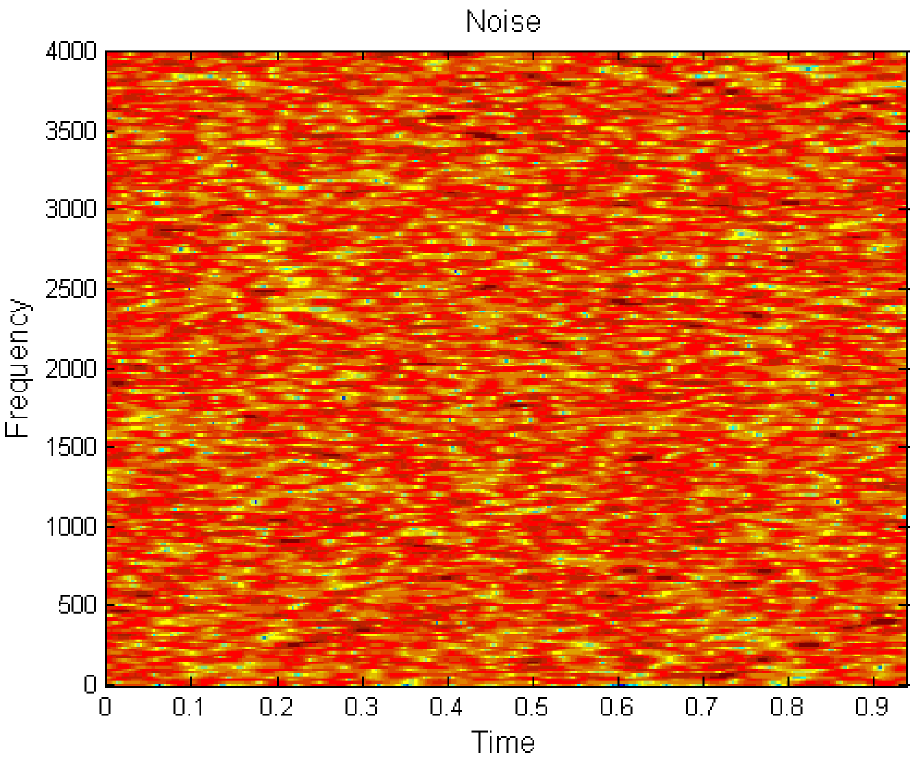


Espectrograma: tempo-frequência

Sinusóide de frequência 2 kHz, com DC *offset*

Sinal de ruído





DCT – Discrete Cosine Transform

- Não é a parte real da DFT
- Base ortnormada de funções co-seno
- Transformada real
- O cálculo pode ser realizado através da FFT

DCT 1D a N pontos

DCT: $$X[k] = A[k]\sum_{n=0}^{N-1} x[n]\cos\left(\frac{(2n+1)k\pi}{2N}\right)$$

$$A[k] = \begin{cases} \frac{1}{\sqrt{N}}, & k = 0 \\ \sqrt{\frac{2}{N}}, & k \neq 0 \end{cases}$$

IDCT: $$x[n] = \sum_{k=0}^{N-1} A[k]X[k]\cos\left(\frac{(2n+1)k\pi}{2N}\right)$$

DCT 1D a 4 pontos

DCT: $$X[k] = A[k]\sum_{n=0}^{3} x[n]\cos\left(\frac{(2n+1)k\pi}{8}\right)$$

$$A[k] = \begin{cases} \frac{1}{2}, & k = 0 \\ \frac{1}{\sqrt{2}}, & k \neq 0 \end{cases}$$

IDCT: $$x[n] = \sum_{k=0}^{3} A[k]X[k]\cos\left(\frac{(2n+1)k\pi}{8}\right)$$

Para cada valor de k, existe um vector base associado (tal como na DFT)

Para k=0: $$C_{0,n} = \frac{1}{2}\cos\left(\frac{(2n+1)0\pi}{8}\right) = \left\{\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right\}$$

Para k=1: $$C_{1,n} = \frac{1}{\sqrt{2}}\cos\left(\frac{(2n+1)1\pi}{8}\right) = \{0.6533, \quad 0.2706, \quad -0.2706, \quad -0.6533\}$$

Cálculo matricial da DCT e IDCT

- **C** é o operador que calcula a DCT a 4 pontos
- Cada linha de **C** é um vector base (co-seno): $c_{k,n}$

$$C = \begin{bmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ 0.6533 & 0.2706 & -0.2706 & -0.6533 \\ 0.5000 & -0.5000 & -0.5000 & 0.5000 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{bmatrix}$$

- Cálculo matricial da DCT: $\mathbf{y} = DCT[\mathbf{x}] = \mathbf{C}\mathbf{x}$

 x e y são vectores coluna; x é o sinal no tempo e y são os coeficientes da DCT.

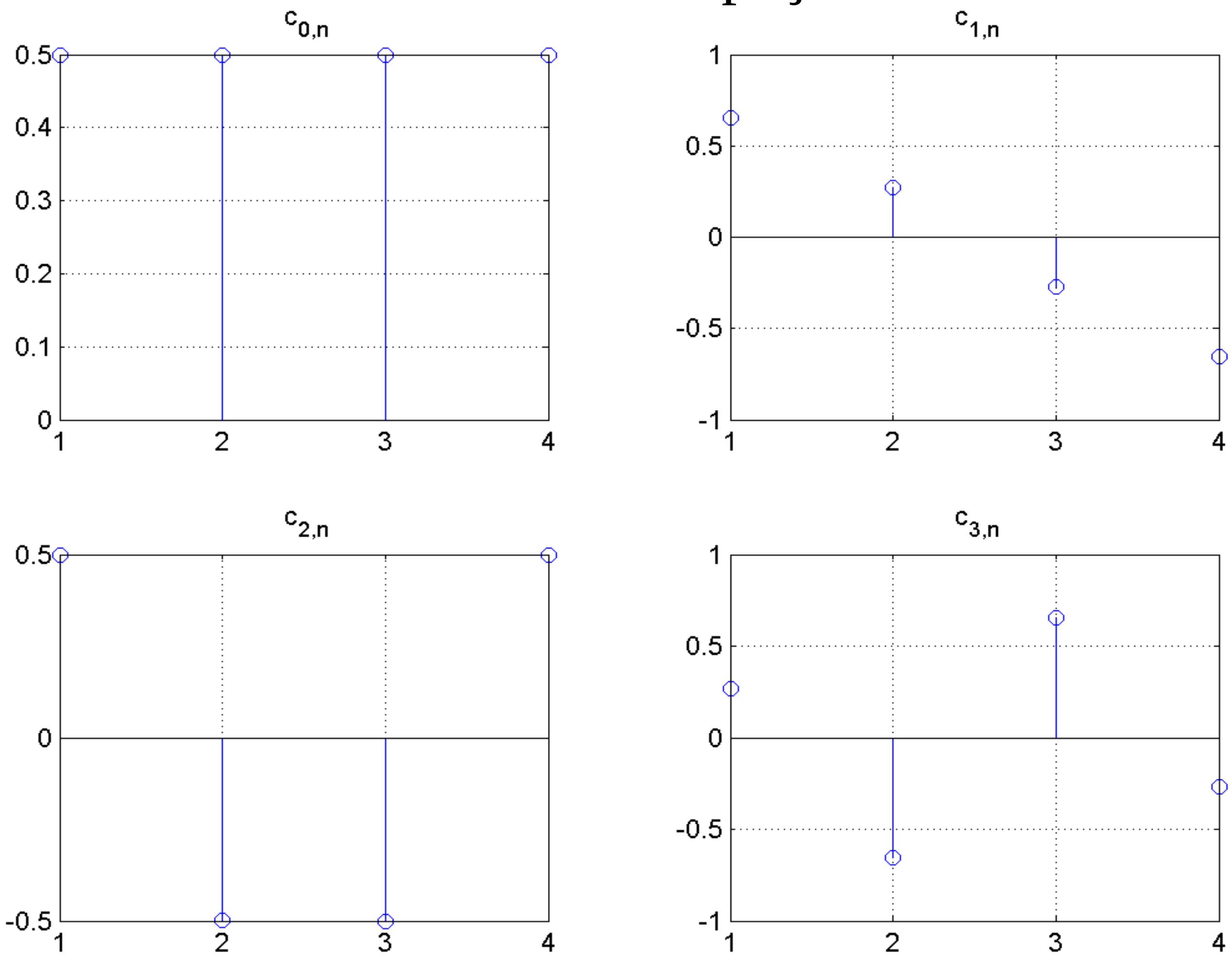
- A IDCT é calculada através de $\mathbf{x} = IDCT[\mathbf{y}] = \mathbf{C}^{\mathbf{T}}\mathbf{y}$
- A matriz **C** é ortogonal $\mathbf{C}^{\mathbf{T}} = \mathbf{C}^{-1}, \quad \mathbf{C}^{\mathbf{T}}\mathbf{C} = \mathbf{C}\mathbf{C}^{\mathbf{T}} = \mathbf{I}$

Vectores base da DCT1D a 4 pontos

- São as linhas da matriz C

- Constituem base ortonormada no espaço de dimensão 4

Componente DC



Alta frequência

Cálculo da DCT e IDCT

$$x_1 = \begin{bmatrix} 3 & 5 & 9 & 11 \end{bmatrix}^T$$

$$y_1 = DCT\ [x_1] = Cx_1 = \begin{bmatrix} 14.00 & -6.30 & -0.00 & 0.44 \end{bmatrix}^T$$

$$z_1 = \begin{bmatrix} 14.00 & -6.30 & -0.00 & \mathbf{0} \end{bmatrix}^T \cong y_1$$

$$w_1 = IDCT\ [z_1] = C^T z_1 = \begin{bmatrix} 2.87 & 5.29 & 8.70 & 11.12 \end{bmatrix} \cong x_1$$

Sinal de baixa frequência

Coeficientes significativos

Sinal de alta frequência

$$x_2 = \begin{bmatrix} -3 & 5 & -9 & 11 \end{bmatrix}^T$$

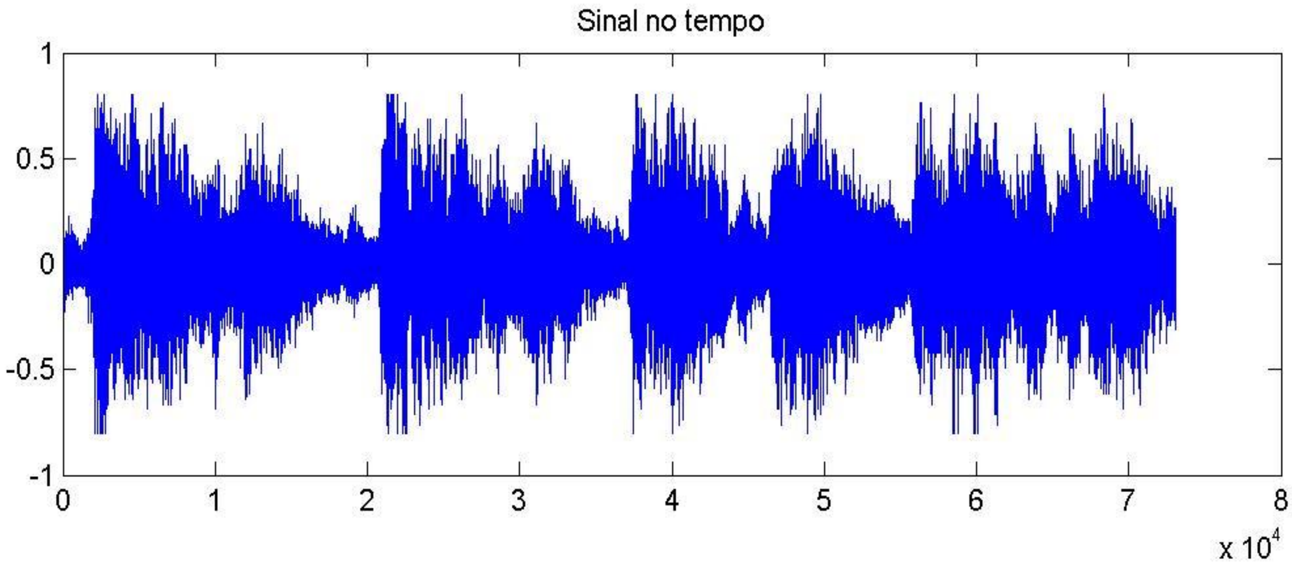
$$y_2 = DCT\ [x_2] = Cx_2 = \begin{bmatrix} 2.0 & -5.35 & 6.00 & -12.93 \end{bmatrix}^T$$

$$z_2 = \begin{bmatrix} \mathbf{0} & -5.35 & 6.00 & -12.93 \end{bmatrix}^T \cong y_2$$

$$w_2 = IDCT\ [z_2] = C^T z_2 = \begin{bmatrix} -4.0 & 4.00 & -10.00 & 10.00 \end{bmatrix} \cong x_2$$

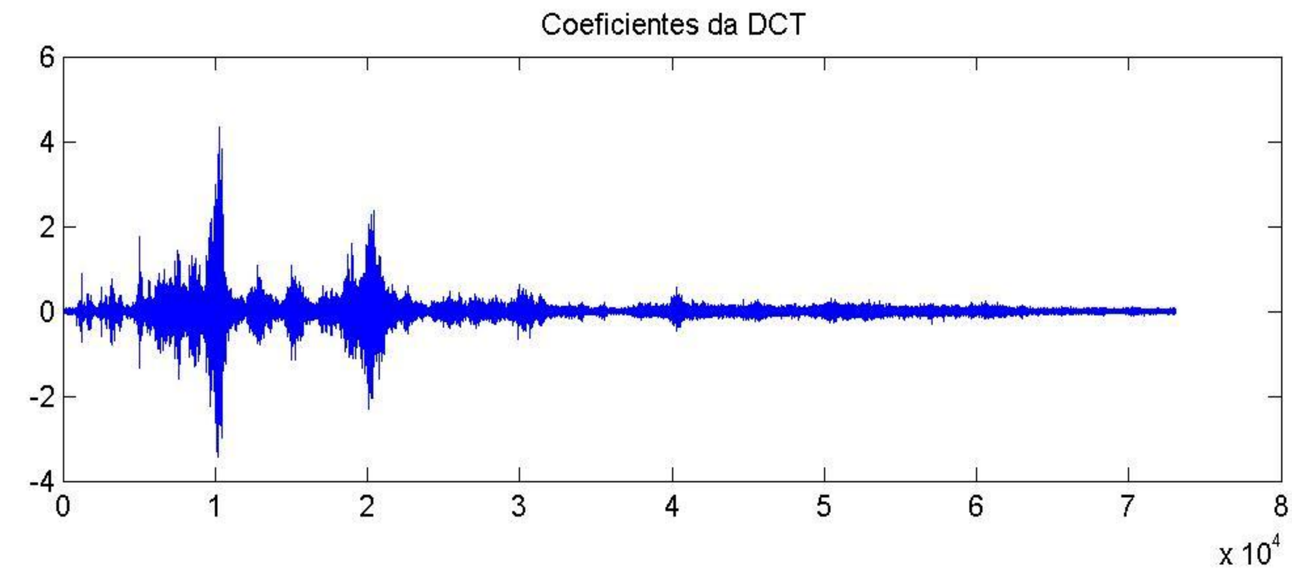
Sinais próximos dos originais

Sinal no tempo e coeficientes da DCT



Energia dispersa no tempo

(vector x)



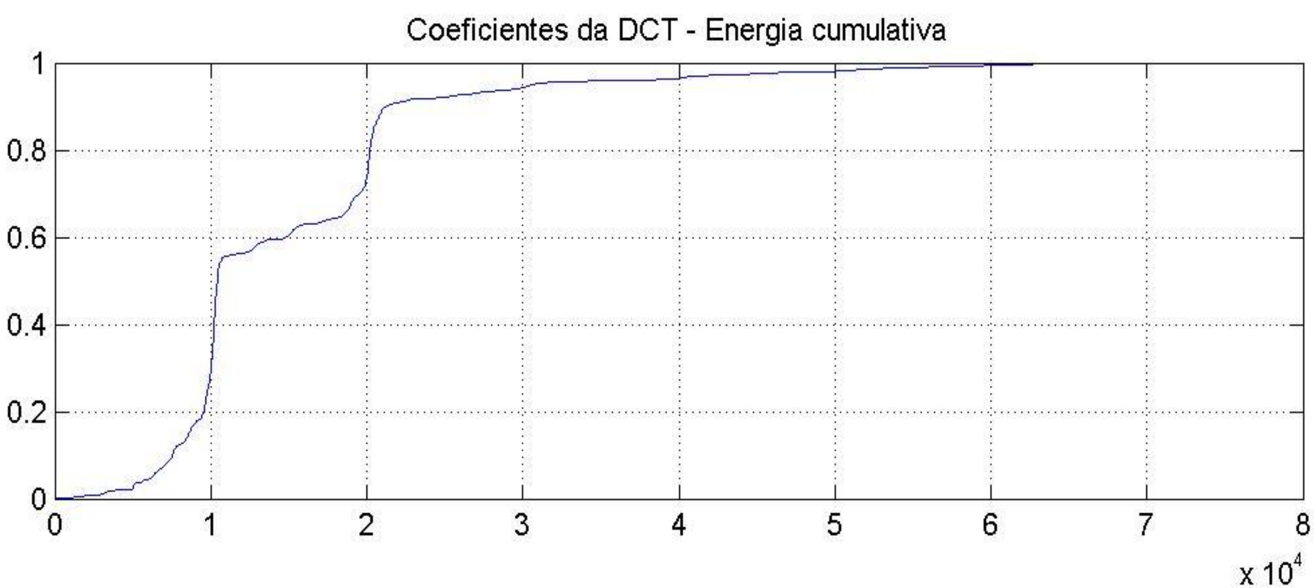
Energia concentrada num número reduzido de coeficientes da transformada

(vector y)

predominam as baixas frequências)

Percentagem de energia acumulada

Domínio do tempo (x)



Domínio da transformada (y)

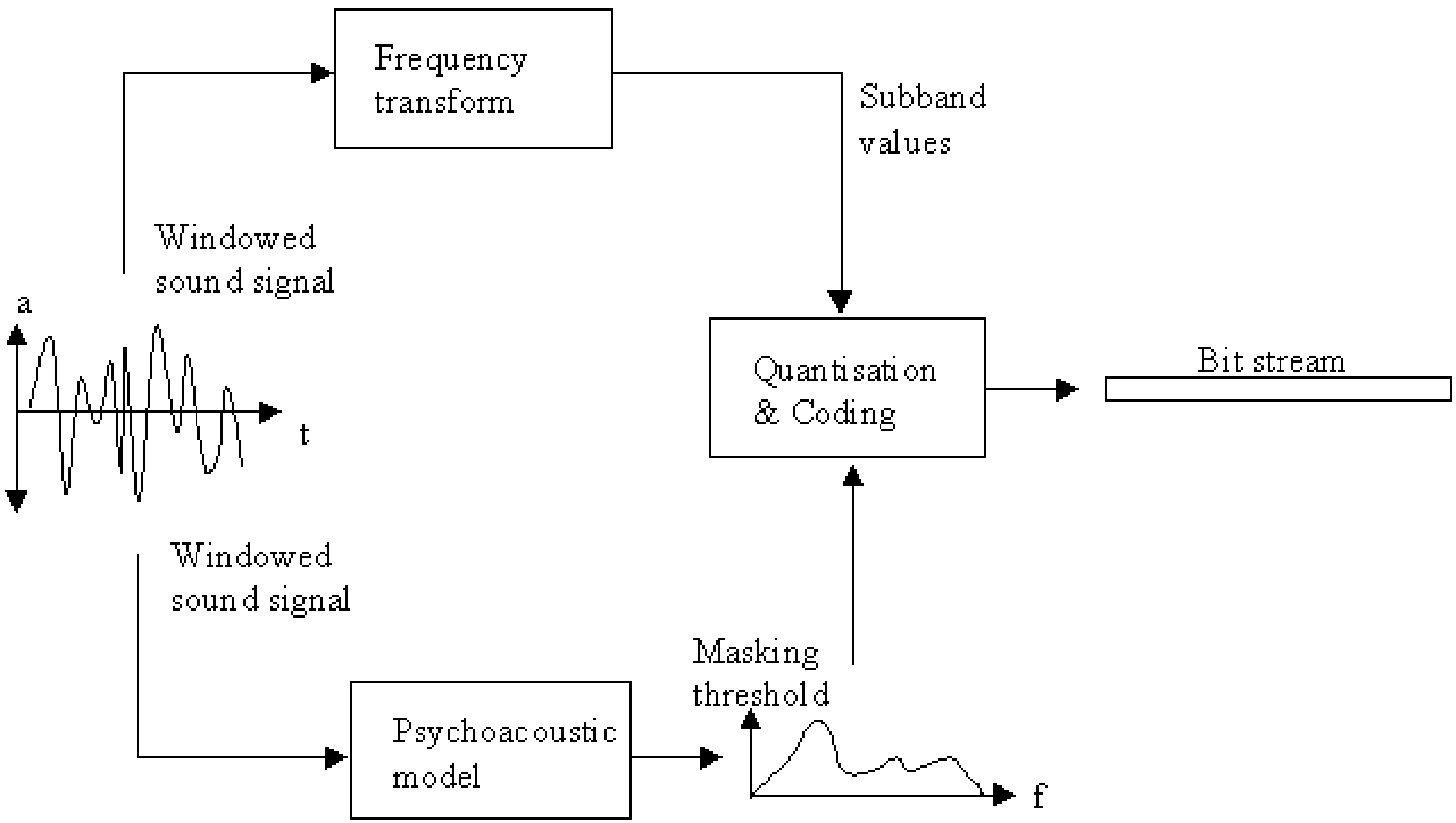
Troço de código da função dct

A função MATLAB **dct** calcula a DCT1D, através da **fft (1D)**

```
function b=dct(a,n)
%DCT  Discrete cosine transform.
%
%   Y = DCT(X) returns the discrete cosine transform of
      X.
%   The vector Y is the same size as X and contains the
%   discrete cosine transform coefficients.
(...)

% Compute weights to multiply DFT coefficients
ww = (exp(-i*(0:n-1)*pi/(2*n))/sqrt(2*n)).';
ww(1) = ww(1) / sqrt(2);
if rem(n,2)==1 | ~isreal(a), % odd case
  % Form intermediate even-symmetric matrix
  y = zeros(2*n,m);
  y(1:n,:) = aa;
  y(n+1:2*n,:) = flipud(aa);
```

```
 % Compute the FFT and keep the appropriate portion:
  yy = fft(y);
  yy = yy(1:n,:);


else % even case
  % Re-order the elements of the columns of x
  y = [ aa(1:2:n,:); aa(n:-2:2,:) ];
  yy = fft(y);
  ww = 2*ww;  % Double the weights for even-length
      case
end

% Multiply FFT by weights:
b = ww(:,ones(1,m)) .* yy;
if isreal(a), b = real(b); end
if do_trans, b = b.'; end
```

Codificação áudio: aplicação de transformadas



Referências e links

1. A. Oppenheim, R. Schafer, J. Buck, ***Discrete-time Signal Processing***, second edition, Prentice-Hall, 1999.

2. S. Kuo, B. Lee, **Real-Time Digital Signal Processing - Implementations Applications and Experiments with the TMS320C55x**, Wiley, 2001.

Curso on-line sobre a DFT:
http://www.spd.eee.strath.ac.uk/users/interact/moved/dft/dftideas.html

Curso on-line sobre a FFT:
http://www.spd.eee.strath.ac.uk/users/interact/moved/fft.html