

# SM - TP2

## Representação de Sinais por Sinusoides

Muitos sistemas que operam com sinal multimédia (som, img, vídeo) prezavam a forma desse sinal.

sistemas lineares

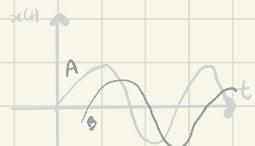
apresentam a propriedade de responder um sinal sinusoidal com outro sinal com a mesma frequência

$$A \quad x(t) = \sin(\omega_0 t)$$

sistema linear

B

$$y(t) = A \cdot \sin(\omega_0 t + \phi)$$



Esta propriedade permite:

- Decompor um determinado sinal numa soma de sinusoides;
- Analisar como é processada cada componente sinusoidal, uma-a-uma;
- Determinar o resultado do processamento do sinal original, somando os resultados das componentes sinusoidais.

NOTA

Os fenômenos oscilatórios  
só de  
Natureza  
sinusoidal

# Definir um sinal sinusoidal

$$x(t) = A \sin(\omega t + \phi)$$

Amplitude ↴

↓  
sinal

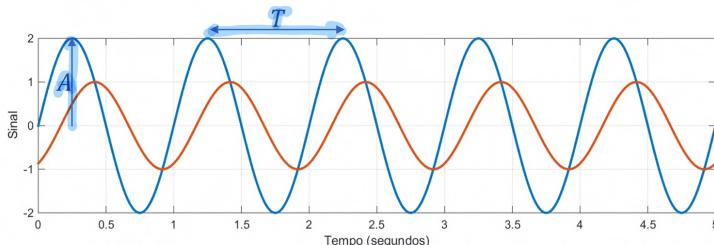
↪

vemos o deslocamento das ondas

identificam a fase

$$T = \frac{1}{f}$$

$$\omega = 2\pi \times f$$



FASE:  
"O atraso da onda  
laranja é de  
60 graus"

5

## Som como Sinal Sinusoidal Composição de Sinais Sinusoidais

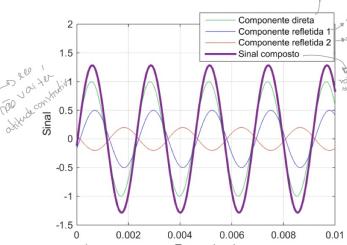
### 3. Composição de Sinais Sinusoidais – Parte I

passo analisar os sinal, sinusoide as sinusoide

- Em cenários reais, os sinais contemplam combinações de múltiplas sinusoïdes.
- Exemplo: Som num ambiente com múltiplas reflexões:



Atenção e Atraso



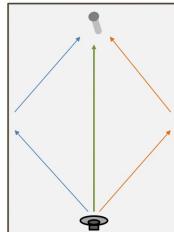
Atenção  
atraso

Componente  
refletida 1  
atraso de 100 ms  
atenuação de 0.5

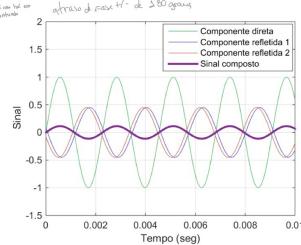
Componente  
refletida 2  
atraso de 200 ms  
atenuação de 0.2

### 3. Composição de Sinais Sinusoidais – Parte I

- Em cenários reais, os sinais contemplam combinações de múltiplas sinusoïdes.
- Exemplo: Som num ambiente com múltiplas reflexões:



Sistemas Multimídia - 2020/2021 - Telmo Cunha



9

- O somatório de  $K$  sinais sinusoidais, todos com a mesma frequência  $f_0$ , mas amplitudes  $A_k$  e fases  $\phi_k$  possivelmente diferentes, resulta num sinal sinusoidal também com a frequência  $f_0$ .

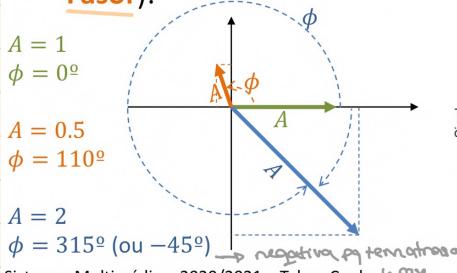
$$y(t) = \sum_{k=1}^K A_k \sin(2\pi f_0 t + \phi_k) = A \sin(2\pi f_0 t + \phi)$$

A e  $\phi$   
do sinal  
resultante  
dependem  
de  $A_k$  e  $\phi_k$

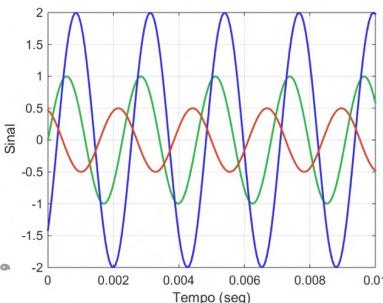
Como vamos determinar esses parâmetros de forma simples e sistematizada?  
Com uma soma vetorial!  
(e números complexos)

- Sendo várias sinusoides, com a mesma frequência  $f_0$ , caracterizadas pelos parâmetros amplitude e fase, estas podem ser conceptualmente representadas por um vetor (denominado

Fasor):



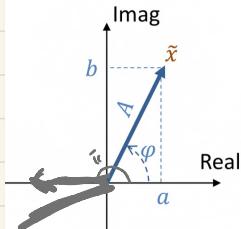
Sistemas Multimédia – 2020/2021 – Telmo Cunha da Silva



# Sinusoides e os Números Complexos

fatores (vetores) podem ser representados por números complexos

- Para tal, é adequado referir a fase ao  $\cos(\cdot)$ , em vez do  $\sin(\cdot)$ .



$$x(t) = A \cos(2\pi f_0 t + \varphi) = A \cos(\omega_0 t + \varphi)$$

$$\tilde{x} = Ae^{j\varphi} = a + jb \rightarrow \begin{matrix} \text{representação} \\ \text{cartesiana} \end{matrix}$$

$$\begin{aligned} x(t) &= \operatorname{Re}\{\tilde{x}e^{j\omega_0 t}\} = \operatorname{Re}\{Ae^{j\varphi}e^{j\omega_0 t}\} = \operatorname{Re}\{Ae^{j(\omega_0 t + \varphi)}\} \\ &= \operatorname{Re}\{A \cos(\omega_0 t + \varphi) + jA \sin(\omega_0 t + \varphi)\} \\ &\quad \hookrightarrow \text{onda resultante} \end{aligned}$$

12

Relembrando algumas relações envolvendo números complexos:

$$y = a + jb \xleftarrow{\text{Representação Cartesiana}}$$

$$\xrightarrow{\text{Representação Polar}}$$

$$\begin{aligned} A &= \sqrt{a^2 + b^2} \\ \varphi &= \operatorname{atan}(b/a) \\ a &= A \cos(\varphi) \\ b &= A \sin(\varphi) \end{aligned}$$

Soma e subtração:

$$(a_1 + jb_1) + (a_2 + jb_2) = (a_1 + a_2) + j(b_1 + b_2)$$

Multiplicação:

$$(A_1 e^{j\varphi_1})(A_2 e^{j\varphi_2}) = A_1 A_2 e^{j(\varphi_1 + \varphi_2)}$$

Conjugado:

$$(a + jb)^* = a - jb \quad (Ae^{j\varphi})^* = Ae^{-j\varphi}$$

- Multiplicação pelo conjugado:

$$yy^* = (Ae^{j\varphi})(Ae^{j\varphi})^* = A^2 = a^2 + b^2$$

- Divisão:

$$(A_1 e^{j\varphi_1}) / (A_2 e^{j\varphi_2}) = (A_1/A_2) e^{j(\varphi_1 - \varphi_2)} \quad \frac{a+jb}{c+jd} = \frac{(a+jb)(c-jd)}{c^2+d^2}$$

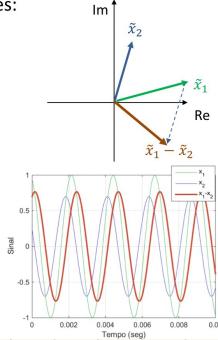
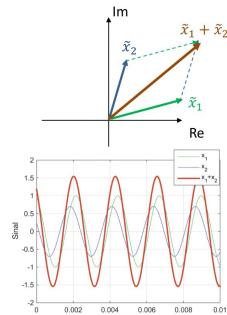
- Relação (fórmula) de Euler:

$$e^{j\varphi} = \cos(\varphi) + j \sin(\varphi)$$

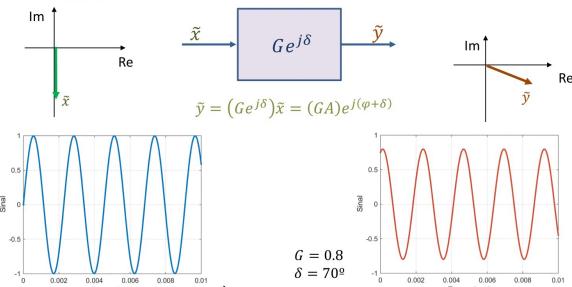
Relembra  
os N<sup>os</sup>  
Complexos

# Sinusoides & Números Complexos

A adição de sinusoides de igual frequência  $f_0$  pode ser vista através da soma dos respectivos fasores:

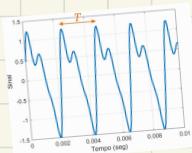


- A resposta sinusoidal de um processo linear pode ser facilmente analisada pela alteração de amplitude e de fase que tal processo introduz:



## Decomposição em Série de Sinusoides

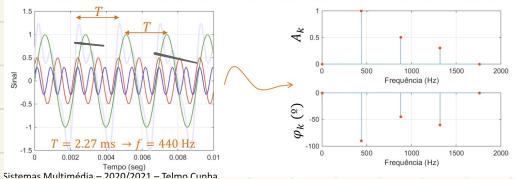
Como um processo linear mantém separadamente o comportamento a cada frequência, torna-se muito adequado decompor os sinais num somatório de componentes sinusoidais



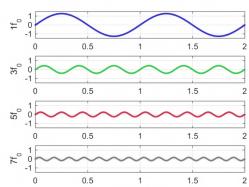
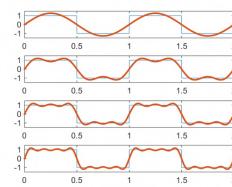
essa decomposição é muito simples para o caso de sinais periódicos (que se repetem)

- Um sinal periódico, de período  $T$ , pode ser descrito por um somatório de sinusoides de frequências múltiplas de  $f = 1/T$  (incluindo a componente constante).

$$x(t) = \sum_{k=0}^K A_k \cos(2\pi k f t + \varphi_k) = \sum_{k=0}^K A_k \sin(2\pi k f t + \phi_k)$$



Exemplo (onda quadrada):

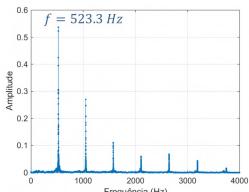
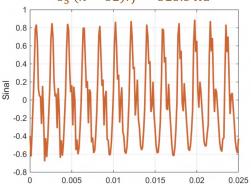


Série de Fourier

$$x(t) = \sum_{n=0}^{\infty} a_n \cos(n\pi k f t) + \sum_{n=1}^{\infty} b_n \sin(n\pi k f t)$$

Exemplo nota d' no piano

$C_5$  ( $n = 52$ ):  $f = 523.3$  Hz



# Composição de Sínusoides - Pt 2

Considerese, agora, o caso em que um sinal é constituído por um somatório de sinusoides não harmonicamente relacionadas:

$$x(t) = \sum_{k=1}^K A_k \cos(\omega_k t + \varphi_k)$$

Este sinal é também periódico se todas as frequências forem múltiplos inteiros de uma frequência comum,  $f_0 \rightarrow$  frequência do sinal composto

$$f_0 = \text{máx}(f_1, f_2, \dots, f_K)$$

**NOTA:** Se não houver um máximo divisor comum de todas as frequências, o sinal composto não é periódico

descobrir o m.d.c

$$\left\{ \begin{array}{l} f_1 = m_1 f_0 \\ f_2 = m_2 f_0 \\ f_3 = m_3 f_0 \end{array} \right. \quad \text{vamos portanto} \underline{\text{vamos portentativas}}$$

## NOTA Aula Prática 2

A amplitude não é igual à amplitude do somatório das componentes porque estas podem não atingir o máximo (e normalmente não atinge)

# TP 3

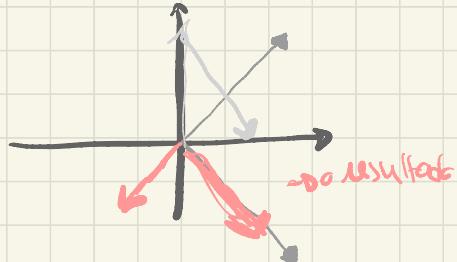
random notes

$$y(t) = 2 \cos\left(\omega_0 t + \frac{\pi}{4}\right) + 2 \cos\left(\omega_0 t - \frac{\pi}{3}\right) - 2 \cos\left(\omega_0 t + \frac{\pi}{8}\right)$$

$\underbrace{+ e^{j\frac{\pi}{4}} + e^{-j\frac{\pi}{3}} - e^{-j\frac{\pi}{8}} = \dots}$

$$1 \cdot 219 e^{-j5.918}$$

$$y(t) = 2 \cdot 219 \cos(\omega_0 t - 5.918)$$



## Séries de Fourier (aproximantes quadráticos)

qual sinal de período T pode ser representado como uma soma de sinusoides

cujas frequências são múltiplos de  $F = \frac{1}{T}$

$$x(t) = A_0 + \sum_{k=1}^{\infty} A_k \cos(\omega_k t + \phi_k)$$

$$x(t) = \sum_{k=1}^{\infty} a_k e^{j\omega_k t}$$

$$a_k = \frac{A_k}{2} e^{j\phi_k}$$

harmonicos do sinal

$$\int_{k=1}^{\infty} \frac{1}{k^2}$$

$$\int_{k=2}^{\infty} \frac{1}{k^2}$$

$$\int_{k=3}^{\infty} \frac{1}{k^2}$$

# T.P.

3 e 4

## Série e transformada de Fourier e Espectro de Sinais

### Série e Transformada Discreta de Fourier (DFT)

Mostrou-se que um sinal periódico (de período  $T$ ) pode ser decomposto numa série de senos e/ou cossenos, de frequências múltiplas de  $f = 1/T$ , denominada **Série de Fourier**:

$$x(t) = \sum_{k=0}^K a_k \cos(2\pi kft) + \sum_{k=1}^K b_k \sin(2\pi kft) = \sum_{k=0}^K A_k \cos(2\pi kft + \varphi_k)$$

### Para calcular os coeficientes:

$$a_k = \frac{2}{T} \int_0^T x(t) \cos(2\pi kft) dt; \quad b_k = \frac{2}{T} \int_0^T x(t) \sin(2\pi kft) dt; \quad \forall k > 0$$

$$a_0 = \frac{1}{T} \int_0^T x(t) dt \quad (\text{Valor médio do sinal}) \quad b_0 = 0$$

$$A_k = \sqrt{a_k^2 + b_k^2}; \quad \varphi_k = -\text{atan}2\left(\frac{b_k}{a_k}\right)$$

Contudo, na maior parte dos casos, os sinais encontram-se dividizados no tempo (não são contínuos)

No Matlab

Neste caso a Série da Fourier toma a seguinte forma:

$$x(n) = \sum_{k=0}^K a_k \cos(2\pi kfnT_a) + \sum_{k=1}^K b_k \sin(2\pi kfnT_a) = \sum_{k=0}^K A_k \cos(2\pi kfnT_a + \varphi_k)$$

$n = 0, 1, 2, \dots, N$

$$a_k = \frac{2}{N} \sum_{n=1}^N x(n) \cos(2\pi kfnT_a); \quad b_k = \frac{2}{N} \sum_{n=1}^N x(n) \sin(2\pi kfnT_a); \quad \forall k > 0$$

$$a_0 = \frac{1}{N} \sum_{n=1}^N x(n) \quad (\text{Valor médio do sinal}) \quad b_0 = 0 \quad N = \frac{T}{T_a}$$

$$A_k = \sqrt{a_k^2 + b_k^2}; \quad \varphi_k = -\text{atan}2\left(\frac{b_k}{a_k}\right)$$

$$\begin{aligned} x(t) &= \sum_{k=0}^K A_k \cos(w_{kf}t + \varphi_k) \\ &\quad \cos(wt) = \frac{e^{jwt} + e^{-jwt}}{2} \\ x(t) &= \sum_{k=0}^K A_k \frac{e^{j(2\pi kf_0 t + \varphi_k)} + e^{-j(2\pi kf_0 t + \varphi_k)}}{2} \\ \Leftrightarrow x(t) &= \sum_{k=0}^K A_k e^{j\varphi_k} e^{j2\pi kf_0 t} + \frac{1}{2} \sum_{k=0}^K A_k e^{-j\varphi_k} e^{-j2\pi kf_0 t} \\ &\quad \text{parâmetros que queremos} \\ &\quad \text{conhecendo para obter os} \\ &\quad \text{só os coeficientes} \\ &\quad \text{cosseno daquele} \\ \Leftrightarrow x(t) &= \sum_{k=0}^K \frac{1}{2} A_k e^{j\varphi_k} e^{j2\pi kf_0 t} + \sum_{k=0}^K \frac{1}{2} A_k e^{-j\varphi_k} e^{-j2\pi kf_0 t} \\ \Leftrightarrow x(t) &= \sum_{k=-K}^K A_k e^{j2\pi kf_0 t} \quad (k = -K \text{ a } K) \end{aligned}$$

Considera-se a aplicação da fórmula de Euler à Série de Fourier:

$$\begin{aligned} x(t) &= \sum_{k=0}^K A_k \cos(2\pi kf_0 t + \varphi_k) = \sum_{k=0}^K A_k \frac{e^{j(2\pi kf_0 t + \varphi_k)} + e^{-j(2\pi kf_0 t + \varphi_k)}}{2} = \\ &= \frac{1}{2} \sum_{k=0}^K A_k e^{j\varphi_k} e^{j2\pi kf_0 t} + \frac{1}{2} \sum_{k=0}^K A_k e^{-j\varphi_k} e^{-j2\pi kf_0 t} \end{aligned}$$

Esta expressão pode ser condensada na seguinte forma:

$$x(t) = \sum_{k=-K}^K C_k e^{j2\pi kf_0 t}, \quad C_k \in \mathbb{C}$$

Expliquei  
detalhado

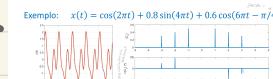
Esta é conhecida por **Série Exponencial de Fourier**.

introduz o conceito de frequências "positivas" e "negativas"

$$x(t) = \sum_{k=-K}^K C_k e^{j2\pi kf_0 t}$$

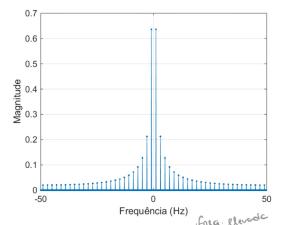
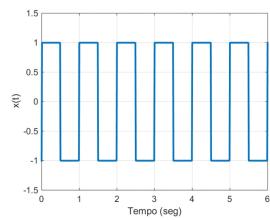
Para sinais reais verifica-se a relação:  $C_k = C_{-k}$  comprovado

Esta representação fez surgir o conceito de **Espetro de um sinal** (i.e. o seu conteúdo ao longo da frequência)

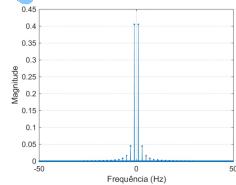
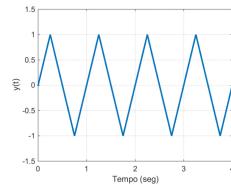


# Exemplos de aplicação da DFT

## Onda quadrada

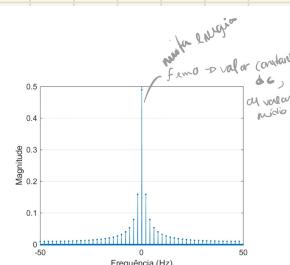
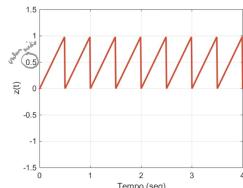


## Onda triangular

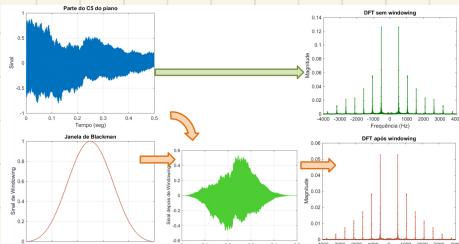


nós têm uma onda  
tão exigeante quanto à quadrangular

### Onda dente-de-serra:



+ exemplos nos slides



## em sinais Não Periódicos

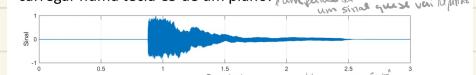
- A DFT foi definida para Sinais Periódicos.

- Mas ento, como podemos analisar o espetro de sinais não periódicos (como a recebido num sequência de áudio?)

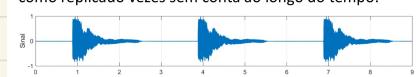
↳ Forçando o sinal recolhido a ser considerado como periódico certo?

há várias troncas, uma delas  
O windowing

Por exemplo, considere-se o registo do som produzido por se carregar numa tecla C5 de um piano:

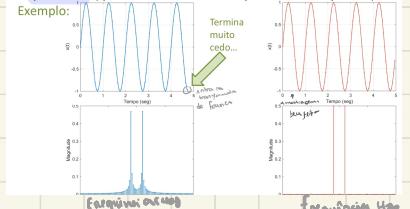


Para fins de análise do seu espetro, este sinal pode ser entendido como replicado vezes sem conta ao longo do tempo:



Então, a DFT pode ser aplicada a uma sequência de amostras, ficando implícita a repetição dessa sequência indefinidamente.

- A técnica de windowing aplica-se quando a parte inicial da sequência de amostras recolhida não apresenta a continuidade da parte final (quando se concatenam réplicas dessa sequência).



técnicas de windowing: consiste em multiplicar a sequência não circular por uma "janela" que força a forma periódica

DFT em sinais não periódicos  
(continuação)

O algoritmo FFT (Fast Fourier Transform) implementa a DFT de uma forma eficiente

$$\Rightarrow X = \text{fft}(x)/N$$

Seja  $x$  um vetor de  $N$  amostras consecutivas de um sinal e seu período de amostramento

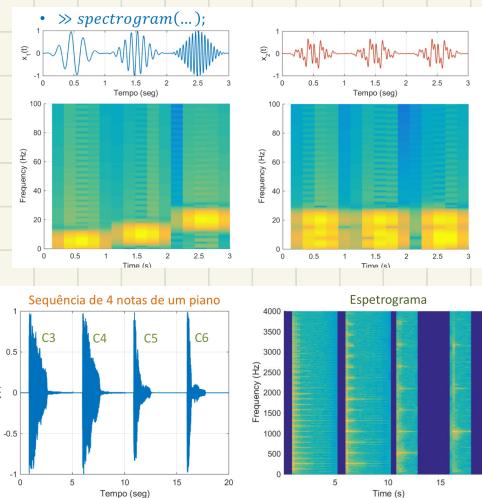
O vetor  $X$  tem também  $N$  elementos:

- um coeficiente ( $R_N \in \mathbb{C}$ ) para cada frequência da composição

$$f = [0 : df : (N-1) * df]$$

ou  $\text{fftshift}(X)$  para ordenar de  $-f_a/2$  a  $f_a/2$

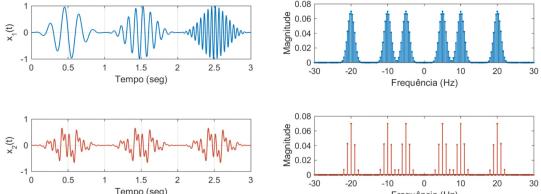
Nestes casos é usual considerar a short-time FFT, cujo resultado é conhecido por Espetrograma



## Espetrograma

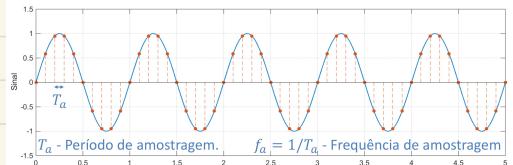
↳ evolução do conteúdo em frequência ao longo do tempo

- Por exemplo, uma sequência de tons diferentes tem um espetro idêntico ao dos mesmos tons considerados em simultâneo:



# Amostragem de Sinais

## Fundamentos da Amostragem



→ Os sinalssão caracterizados no domínio contínuo.

No entanto, o te domínio é impossível ser representado em processadores digitais.

Por isso  
para analisar sinais consideramos um domínio de tempo discreto.

Este processo de "discretização" de um sinal chama-se amostragem.

↓  
consiste em adquirir amostras do sinal (tipicamente um sinal periódico)

Pode parecer fácil, mas não é, na prática um sinal com muitas variações que é amostrado com um intervalo de tempo elevado não vai soar como o original e um que seja praticamente constante amostrado com um intervalo reduzido vai gerar amostras desnecessárias

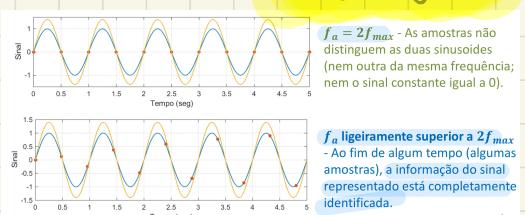
Assim, estabelece-se um compromisso entre o nº de amostras e a sua representatividade

3

## Critério de Nyquist

Para que um sinal seja amostrado sem perda de informação temos de considerar uma frequência de amostragem superior ao dobro da frequência mais elevada do sinal.

( $f_a > 2f_{max}$ )



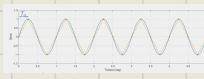
$f_a = 2f_{max}$  - As amostras não distinguem as duas sinusoides (nem outra da mesma frequência; nem o sinal constante igual a 0).

$f_a$  ligeiramente superior a  $2f_{max}$  - Ao fim de algum tempo (algumas amostras), a informação do sinal representado está completamente identificada.

Muitas amostras  
requerem muita memória  
(é só pouca "novidade" de amostra para amostra)



Mas poucas amostras  
podem não ser suficientemente representativas

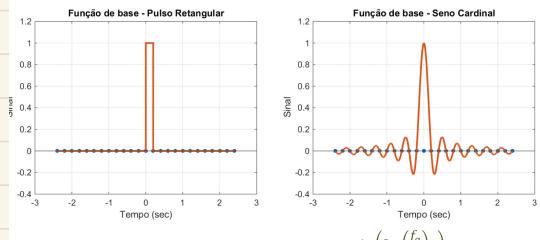


# Reconstrução de Sinais Amostrados

1 faz-se combinar na função Seno Radial ( $\frac{\sin(\omega t)}{\omega}$ )

$$h(t) = \frac{\sin(\omega \pi \frac{f_a}{2} t)}{\omega \pi \frac{f_a}{2} t} = \frac{\sin(\pi f_a t)}{\pi f_a t}$$

A reconstrução ideal deve considerar como função de base, em vez de pulso retangular.

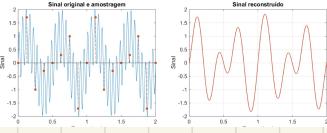


## Aliasing

Quando a amostragem origina um espectro de frequências com alguma(s) fora da janela de Nyquist, ocorre aliasing.

- Como exemplo, considere-se o caso de um sinal composto por dois tons, um correspondente a um som audível ( $f_1 = 2\text{kHz}$ ) e outro a um som inaudível ( $f_2 = 19\text{kHz}$ ), tendo ambos a mesma potência.

- A frequência de amostragem foi considerada para o sinal audível, tendo sido escolhida  $f_a = 8\text{kHz}$ .

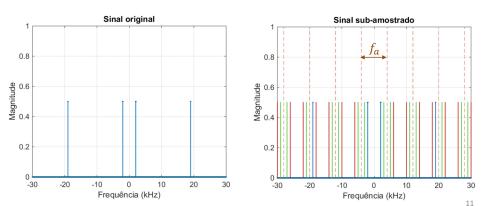


para evitar este fenômeno o processo de amostragem deve ser precedido de um filtro anti-aliasing

2 elimina as frequências superiores a  $f_a$

Mas pq ocorre aliasing?

S deve-se ao facto de o espetro do sinal se repetir em cada intervalo da largura  $f_a$ .



## Quantificação da Amplitude dos Sinais

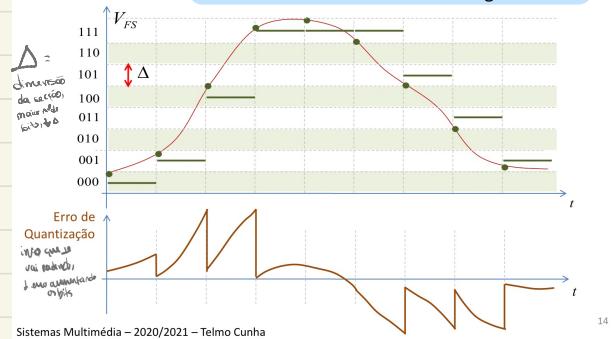
Não é só o tempo que é digitizado, a amplitude dos sinais também. Cada amostra do sinal é adquirida e armazenada numa palavra digital que apresenta um limite de  $n$  bits.

Logo, cada amostra adquire um valor de um conjunto (finito) de códigos possíveis.

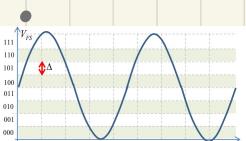
| Número de Bits | Número de Códigos |
|----------------|-------------------|
| 3              | 8                 |
| 4              | 16                |
| 5              | 32                |
| ...            | ...               |
| N              | $2^N$             |

Este n° pode condicionar o processo

- Para ilustrar o efeito da quantização da amplitude dos sinais, considere-se o caso do armazenamento em código de 3 bits:



- A relação sinal-ruído é uma métrica muito importante, pois indica o esforço necessário para se distinguir a informação útil contida num sinal, quando na presença de ruído.



- A relação sinal-ruído é usualmente apresentada em decibéis (dB):  
 $SNR_{dB} = 10 \log_{10}(SNR)$

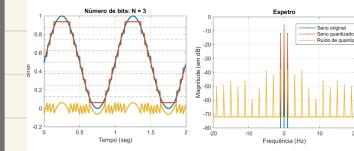
- Notando que  $\Delta = V_{FS}/2^N$ , verifica-se que a SNR entre a sinusoide de referência e o ruído de quantização é:

$$SNR_{dB} = 10 \log_{10}\left(\frac{3}{2} \cdot 2^N\right) = N \cdot 20 \log_{10}(2) + 10 \log_{10}\left(\frac{3}{2}\right) \rightarrow$$

$$SNR_{dB} \approx 6.02N + 1.76$$

fórmula. **N bits**, melhor  
relação sinal-ruído, maior variação a  
potência do sinal útil comparada ao  
sinal ruído.  
**Exemplo:**

- Exemplo: Quantização de um seno com resolução de  $N = 3$  bits:



- Exemplo: Quantização de uma imagem:



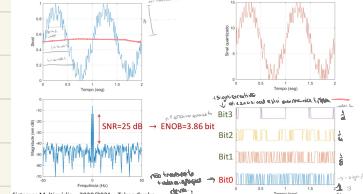
**Nº de bits Efetivo**  
naciocínio inverso!

- Este é o conceito de Número de Bits Efectivos (ENOB – Effective Number of Bits):

$$ENOB \approx \frac{SNR_{dB} - 1.76}{6.02}$$

- Neste conceito, o ruído pode incluir diferentes componentes (quantização, ruído térmico, ruído de medição, de processo, ...).

- Exemplo da quantização (N=4 bits) de um sinal com ruído:

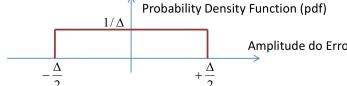


↓ erro quando há maior nº de bits



para cada código corresponde a um  
intervalo de amplitude igualado  
 $\Delta = \frac{V_{FS}}{2^N}$  → valor de fim da escala  
ou seja, 3 bits representam o intervalo de amplitude do sinal

- Admitindo que o valor do erro de quantização pode adquirir, com igual probabilidade, qualquer valor nesse intervalo:



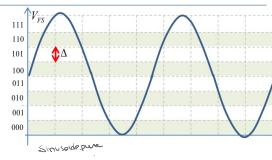
- Então, a potência (normalizada) do erro de quantização é:

$$P(\epsilon) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} \epsilon^2(t) dt = \int_{-\infty}^{+\infty} \epsilon^2 pdf(\epsilon) d\epsilon = \frac{1}{\Delta} \int_{-\Delta/2}^{+\Delta/2} \epsilon^2 d\epsilon \rightarrow P(\epsilon) = \frac{\Delta^2}{12}$$

16

- Considerando um sinal sinusoidal que percorre toda a escala de amplitude:

$$x(t) = \frac{V_{FS}}{2} \sin(\omega t)$$



- A potência associada a essa onda sinusoidal é, então:

$$P_x = \frac{1}{T} \int_0^T x(t)^2 dt = \frac{1}{2} \cdot \left(\frac{V_{FS}}{2}\right)^2 = \frac{V_{FS}^2}{8}$$

- E a relação sinal-ruído (SNR – Signal to Noise Ratio) é:

Processo de  
quantização

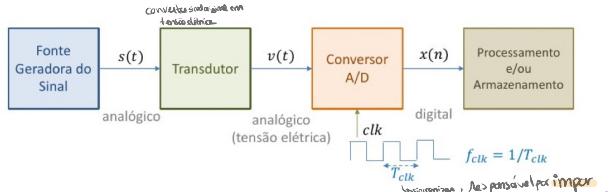
$$SNR = \frac{P_x}{P_\epsilon} = \frac{\frac{V_{FS}^2}{8}}{\frac{\Delta^2}{12}} = \frac{3}{2} \cdot \frac{V_{FS}^2}{\Delta^2}$$

17

# Conversão Analógico/Digital

## Analógico/Digital

Como vimos anteriormente para processar um sinal de forma **digital** precisamos de o converter do seu domínio contínuo no tempo para o discreto, fazendo assim uma **amostragem do tempo e uma quantização da sua amplitude**. Este procedimento é realizado por um **conversor analógico/digital**, que segue a seguinte arquitetura:

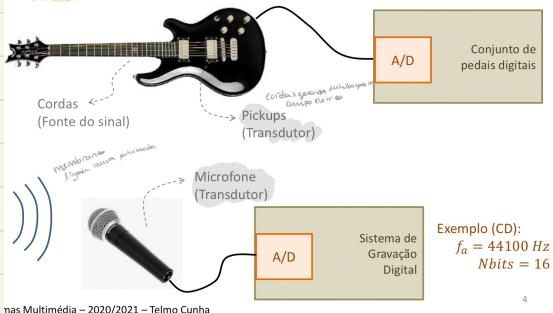


Nestes dispositivos, a **pressão do ar** (sinal analógico) entra no **transdutor** onde é **convertida em tensão elétrica** (analogica). Por sua vez, esta passa pelo **conversor A/D**, que é ativado às transições positivas do **clock** e a cada transição converte a tensão num sinal digital, fechando e abrindo rapidamente o circuito, armazenando no condensador a tensão que é posteriormente convertida em sinal digital.

Por ser ativado à transição positiva do **clock**, o **tempo de amostragem (Ta)** é exatamente o **tempo entre as transições** e a frequência (fa) o seu inverso.

Como a fa é definida *a priori*, é necessário **aplicar um filtro anti-aliasing** antes do sinal ser processado no conversor A/D, que vai filtrar as componentes cuja frequência é superior a metade da fa.

- Exemplo de sistemas de conversão analógico-digital aplicado ao áudio:

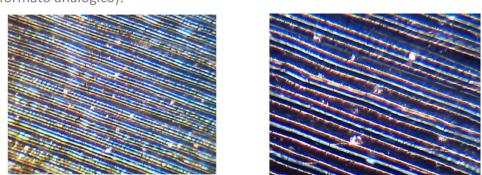


- A gravação de áudio (e não só) no formato digital (aparecimento do CD nas décadas de 80/90) veio contribuir para um aumento significativo da qualidade dos equipamentos de áudio.



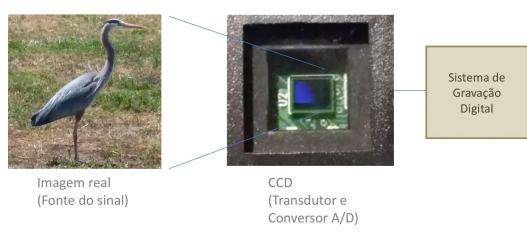
- A gravação de áudio (e não só) no formato digital (aparecimento do CD nas décadas de 80/90) veio contribuir para um aumento significativo da qualidade dos equipamentos de áudio.

Imagen, ao microscópio, dos sulcos num disco de vinil (armazenamento no formato analógico):

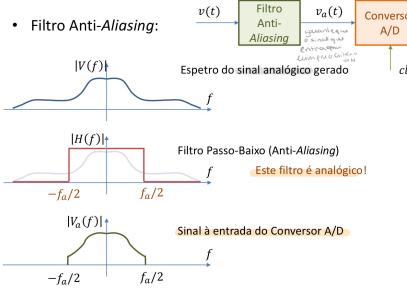


As partículas de pó (evidentes na imagem) causam ruído na reprodução do som armazenado no disco de vinil.

- Exemplo de sistema de conversão analógico-digital aplicado a imagem:



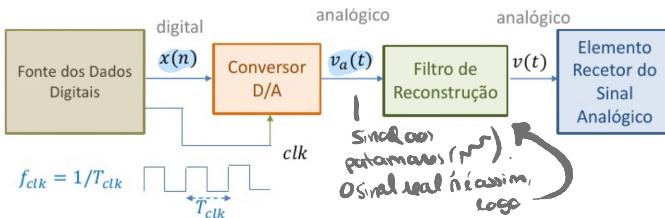
## 1. Conversão Analógico/Digital (A/D)



# Conversão digital/analógico

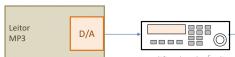
## Digital/Analógico

A conversão digital/analógico é praticamente o inverso da analógico/digital, seguindo a seguinte arquitetura:



O **conversor D/A**, com o **mesmo intervalo entre as transições do relógio**, converte o sinal digital em analógico, que depois é reconstruído no **filtro de reconstrução**, que **elimina as transições abruptas entre as diferentes amostras**.

## Exemplo de conversão digital/analogica aplicada ao áudio:

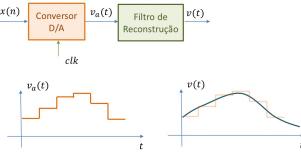


## Exemplo de conversão digital/analogica aplicada à imagem:



## 2. Conversão Digital/Analógica (D/A)

- O conversor D/A necessita do Filtro de Reconstrução (é um filtro analógico) para eliminar as transições abruptas entre os valores das amostras consecutivas.



Veja questão no p.p., pode sair na tarefa!

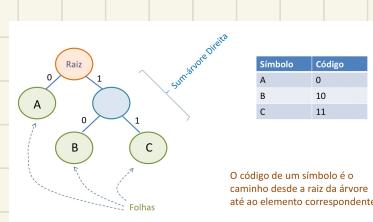


# Árvores Binárias

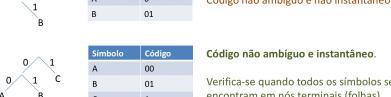
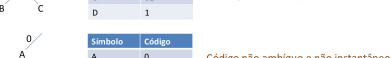
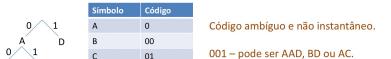
haver com  
os símbolos

**Esquema de codificação binária**, constituída por uma sequência de ramificações duplas, em que o ramo da esquerda corresponde ao bit 0 e o da direita ao 1 e que culminam nos símbolos a codificar (as folhas das árvores).

O código de cada símbolo é o percurso seguido desde a raiz (topo) até esse símbolo.

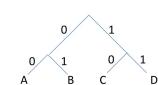


- Ambiguidade e Instantaneidade em árvores binárias:



- Exemplo:

| Símbolo | Código |
|---------|--------|
| A       | 00     |
| B       | 01     |
| C       | 10     |
| D       | 11     |



| Símbolo | Código |
|---------|--------|
| A       | 0      |
| B       | 10     |
| C       | 110    |
| D       | 1110   |



- Por exemplo, seja a seguinte mensagem (de 40 elementos):

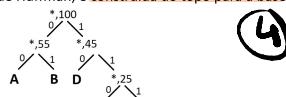
② CABADAAEBBBBEAACBCABBBACDDDDDEEDDDABAEAA

| Símbolo | Frequência (%) |
|---------|----------------|
| A       | 30 (12/40)     |
| B       | 25 (10/40)     |
| C       | 10 (4/40)      |
| D       | 20 (8/40)      |
| E       | 15 (6/40)      |

| Símbolo | Freq. (%) |
|---------|-----------|
| A       | 30        |
| B       | 25        |
| C       | 10        |
| D       | 20        |
| E       | 15        |

55% equilíbrio!

- Se este processo for desenvolvido ao longo da direção vertical, este consiste na criação de uma árvore binária, que, ao contrário da do método de Huffman, é construída do topo para a base).



# Ródigo de Huffman

É um algoritmo que define a forma de construção de árvores binárias e consiste em:

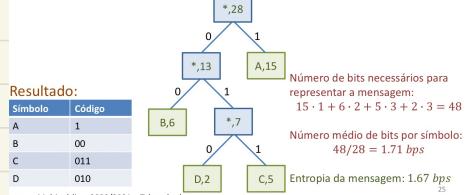
- Ordenar os símbolos por ordem crescente de ocorrência;
- Agrupar os dois símbolos de menor ocorrência e cria-se o elemento do nível imediatamente superior com a soma das ocorrências dos dois;
- Repete-se o ponto anterior com os dois símbolos de menor ocorrência (incluindo o novo nó criado), até que se esgotem os símbolos.

É importante realçar que o **símbolo de menor ocorrência situa-se sempre à esquerda!**

## 6. Código de Huffman

| Símbolo | Ocorrências |
|---------|-------------|
| A       | 15          |
| B       | 6           |
| C       | 5           |
| D       | 2           |

- Repete-se o procedimento até se esgotar os símbolos.



# Ródigo Shannon-Fano

## Código de Shannon-Fano

Este código é semelhante ao de Huffman, considerando também a frequência de ocorrência de cada símbolo. No entanto, existem casos em que o **nível de compressão é menor**, mas também existem outros em que é **mais eficiente**. Processa-se da seguinte maneira:

- Ordenam-se os símbolos por ordem decrescente de ocorrência;
- De seguida, dividem-se os grupos de forma a que a probabilidade de ocorrência de cada um seja a mais parecida possível;
- Atribui-se o bit 0 ao conjunto com maior probabilidade e 1 ao outro. Repete-se o processo para cada um dos grupos, até que cada subgrupo tenha apenas um elemento.

Este processo pode ser feito através de uma árvore binária, mas ao contrário do código de Huffman, esta é construída do topo para a base.

O código é menos eficiente quando há grandes assimetrias entre a probabilidade de ocorrência de cada subgrupo.

## 7. Codificação de Shannon-Fano

- Atribui-se, então, o bit 0 ao primeiro conjunto de símbolos, e o bit 1 ao segundo conjunto.

- Em seguida, repete-se o processo para cada um dos grupos, até que todos os subgrupos formados contenham apenas um símbolo.

- Por exemplo, seja a seguinte mensagem (de 40 elementos):

CABADAAEBBBBEAACBCABBBACDDDDDEEDDDABAEAA

| Símbolo | Freq. (%) |
|---------|-----------|
| A       | 30        |
| B       | 25        |
| C       | 10        |
| D       | 20        |
| E       | 15        |

| Símbolo | Código |
|---------|--------|
| A       | 00     |
| B       | 01     |
| C       | 10     |
| D       | 110    |
| E       | 111    |

- No exemplo desta mensagem:

CABADAAEBBBBEAAACBCABBBACDDDEEDDDABAEAA

- Número total de bits usados após codificação de Shannon-Fano:

$$2 \cdot 12 + 2 \cdot 10 + 2 \cdot 8 + 3 \cdot 6 + 3 \cdot 4 = 90 \text{ bits}$$

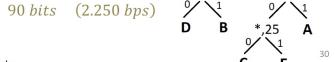
- Número médio de bits por símbolo:

$$90/40 = 2.250 \text{ bps} \text{ (bits por símbolo)}$$

- Entropia: 2.237 bps

- Com codificação de Huffman:

90 bits (2.250 bps)



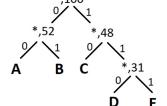
Sistemas Multimédia – 2020/2021 – Telmo Cunha

*Este código pode não ser tão eficiente quando a prob. apresenta assimetria acentuada entre dois subgrupos)*

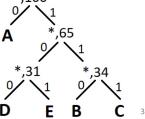
Exemplo:

| Símbolo | Frequência (%) |
|---------|----------------|
| A       | 35             |
| B       | 17             |
| C       | 17             |
| D       | 16             |
| E       | 15             |

Shannon-Fano:  
2.31 bps (em média)



Huffman:  
2.30 bps (em média)



Sistemas Multimédia – 2020/2021 – Telmo Cunha

## Codificação aritmética

- ↳ codificação superior aos métodos anteriores
- ↳ não usa árvores binária
- ↳ visa representar uma mensagem (ou partes) através de um número real entre 0 e 1

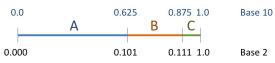
Por exemplo, considere-se a mensagem:

ABAAC

produzida pelo alfabeto cujas probabilidades de ocorrência são:

| Símbolo | Probabilidade (%) |
|---------|-------------------|
| A       | 62.5              |
| B       | 25.0              |
| C       | 12.5              |

Divide-se o intervalo  $[0; 1]$  em partes de dimensões iguais às probabilidades de cada símbolo:



Multimédia – 2020/2021 – Telmo Cunha

- Subdivide-se os intervalos correspondentes a cada símbolo à medida que estes ocorrem:

ABAAC  
Base 10

Qualquer número  $q$  neste intervalo representa a mensagem ABAAC:  
 $0.444 \le q < 0.452$

Por exemplo:  
 $0.0111001_b$  ( $0.4453_{10}$ )

Sistemas Multimédia – 2020/2021 – Telmo Cunha

- Em cada passo deste procedimento, a tabela de probabilidades de ocorrência dos símbolos pode ser alterada de uma forma pré-definida.

- Naturalmente, o processo de descodificação terá que seguir esse mesmo procedimento (que se designa por Modelo).

- Por exemplo, na codificação de texto em língua portuguesa, se surgir o símbolo 'q' será muito provável que o símbolo seguinte seja um 'u', pelo que a tabela de probabilidades pode (pontualmente) ser modificada (conduzindo a uma maior taxa de compressão).

- Podem ser criadas diversas regras que definem, assim, o modelo a ser usado (naturalmente, o aumento da compressão acarreta uma maior complexidade computacional do processo).

35

33

# Capítulo 6

## Codificação de Dados sem Perda

No capítulo anterior foi abordada a codificação de símbolos. No entanto, existem outros níveis de codificação que podemos utilizar para comprimir informação, nomeadamente ao nível dos dados (que podem ser ficheiros, mensagens...).

Nunca ficheiro de texto em que uma palavra ocorre muitas vezes, podemos representá-la com um símbolo único, reduzindo exponencialmente a quantidade de informação necessária para o representar.

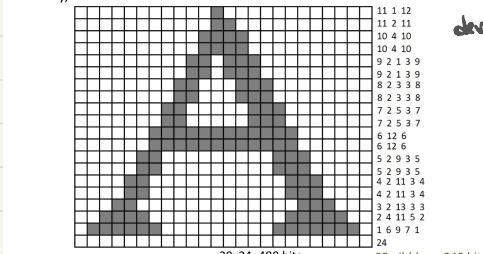
### Codificação Run-Length Encoding

Consiste em representar cada ocorrência de um símbolo, seguido do número de vezes que este ocorre naquela posição.

É mais útil quando existem muitos símbolos repetidos na mensagem.

- Por exemplo, a sequência de símbolos:  
AAAAAAAABBBBCCCCCCCCBBABBB
- Seria representada por:  
A8B4C10B2A1B3
- Evidentemente, esta representação só se torna útil para casos em que o número de símbolos é baixo, surgindo, com elevada probabilidade, muitos símbolos repetidos ao longo dos dados.

Uma aplicação onde esta técnica tem bastante utilidade é na codificação de imagens monocromáticas (imagens a preto-e-branco), como é o caso do fax.



### Codificação baseada em dicionários

Codificação baseada em dicionário

Este tipo de compressão é aplicado por vários codificadores, que têm em comum a utilização de um dicionário de símbolos, onde são associados a um número (índice ou endereço), que os representa na codificação.

Dependendo dos códigos, o dicionário pode ser fixo ou dinâmico, sendo construído à medida que a mensagem é codificada.

LZ77

Um código que aplica um dicionário dinâmico é o LZ77, e constrói-se considerando uma janela deslizante e uma janela de observação, sendo o código criado com base na busca de padrões equivalentes na segunda na primeira janela, da seguinte forma:

et d'adicionar

- O código é inicializado com a janela deslizante inicial (dicionário inicial);
- Procura-se na janela de observação pela máxima sequência cumum com a deslizante, a partir do primeiro carácter (que tem fazer parte da sequência);
- Adiciona-se ao código o par posição-comprimento que representa a localização do padrão na janela deslizante, tendo o primeiro o bit mais significativo a 1 (adicionamos 128);
- As janelas avançam o número de caracteres adicionados ao código e volta-se ao ponto 2, até que não hajam mais caracteres para codificar.

Deve ser assumido um compromisso em relação ao nível de compressão e a quantidade de informação para representar o código comprimido, pois uma janela de observação muito pequena faz com que seja menos provável encontrar padrões, mas uma muito grande obriga à utilização de muitos bits para representar a posição.

info representada por símbolos

quantidade de info bem desejada

### Exemplo

$N=8 \rightarrow$  tamanho da janela de janela

$R=4 \rightarrow$  janela observação

A E A B C E H C | D A R C | H D E A B C D

deslizante

observação

1º identificar janela deslizante

2º janela observação

3º copiar e ver se tem igual

A E A B C E H C

ti existe nenhum bloco de comprimento em 111 de 4, basta o número 3, normal, normal e avançarmos

Vai video!

Aula da minha

### Codificador LZ77:

- Exemplo ilustrativo (com  $N = 8$  e  $M = 4$ ):

#### Sequência a codificar:

A E A B C E H C D A B C H D E A B C D

#### Sequência:

- A E A B C E H C D (2,3) H D      J. deslizante: J. de obs.: H C D A B C H D      E A B C
- A E A B C E H C D (2,3) H D E      C D A B C H D E      A B C D
- A E A B C E H C D (2,3) H D E (3,3)      B C H D E A B C      D
- A E A B C E H C D (2,3) H D E (3,3) D

### Codificador LZ77:

- Muitas vezes, o índice indicado não é referente à posição inicial da janela deslizante mas sim à posição do carácter que se está analisar.
- Ou seja, no par distância-comprimento, a distância é interpretada como o número de caracteres que precedem o carácter em análise.
- Se a dimensão da janela deslizante é pequena, torna-se fácil perder repetições de sequências que estejam um pouco afastadas no texto.
- Se essa dimensão é elevada, os índices de distância requerem um número elevado de bits (retirando eficiência à compressão).

# Codificador LZW

Versão melhorada da LZ78

Considera desde o início um dicionário

Algoritmo:

- Constitui-se o dicionário com todos os símbolos do código.
- Começa-se a percorrer os dados até que a presente sequência w mais o caráter seguinte não esteja no dicionário.
- Adiciona-se w mais o caráter seguinte ao dicionário (o dicionário vai aumentando de tamanho).
- Sequências encontradas no dicionário são substituídas pelo índice correspondente.

Codificador LZW (Lempel-Ziv-Welch):

Exemplo:

TOBEORNOTTOBEORTOBEORNOT#

Dicionário inicial:

| Símbolo | Índice |
|---------|--------|
| #       | 0      |
| A       | 1      |
| B       | 2      |
| ...     | ...    |
| Z       | 26     |

Com 5 bits  
Correspondem a 32 símbolos  
5 bits por símbolo

- Início da análise dos dados: a sequência TO não está no dicionário, pelo que se adiciona ao dicionário.

TO 27

O... 28

14

Codificador LZW (Lempel-Ziv-Welch):

Exemplo:

TOBEORNOTTOBEORTOBEORNOT#

- No final, a sequência de saída fica:

|    |    |   |   |    |    |    |    |    |    |    |    |     |    |    |    |   |
|----|----|---|---|----|----|----|----|----|----|----|----|-----|----|----|----|---|
| 20 | 15 | 2 | 5 | 15 | 18 | 14 | 15 | 20 | 27 | 29 | 31 | 36  | 30 | 32 | 34 | 0 |
| T  | O  | B | E | O  | R  | N  | O  | T  | TO | BE | OR | TOB | EO | RN | OT | # |

- Número total de bits:  $6 \times 5\text{bit} + 11 \times 6\text{bit} = 96\text{ bits}$
- Antes da compressão:  $25 \times 5\text{bit} = 125\text{ bits}$
- Reduz-se em cerca de 22% o volume de dados.

Ver slides  
Exemplo

Mensagem: A B A A A B B A A A A

sequência de saída: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

Resposta: nova versão do dicionário

Dicionário:

A 1 B 2 AB 3 BA 4 AA 5 AAB 6 BB 7 BAA 8

Sequência codificada: 3 3 2 4 4 6 1

Mensagem original: T E S T E T E T E S T E

Dicionário: E 1 S 2 T 3 TE 4 ES 5 ST 6 TET 7 TES 8 STE 9

Note: o dicionário tem limite!

3

Sequência codificada: 1 1 2 6 3 5 8

Mensagem: A A B B C A B C A

Dicionário: A 1 B 2 C 3 AA 4 AB 5 B 6 BBC 7 CA 8

Notas

9

## 4. Codificadores para Imagem sem Perdas

- Existem vários algoritmos de compressão, sem perda de informação, destinados à compressão de imagens.
- Alguns exemplos são:
  - JBIG:** compressão para imagens binárias (fax, por exemplo) – implementa RLE, seguido de codificação de Huffman.
  - JPEG-LS:** Usa predição linear entre pixels adjacentes e codificação de Huffman sobre o erro.
  - GIF:** Usa o código LZW, mas restringe a paleta de cores ao máximo de 256.
  - PNG:** Usa predição linear e o algoritmo DEFLATE (que se baseia no LZ77 e no LZW) para comprimir o erro. Admite "true color".

# Capítulo 7

## Codificação com perdas!

### 5. Geralmente

a ideia é sempre tentar expressar com maior taxa de compressão mas descartar a informação que não seja muito relevante!

Exemplo prático transmissão de vídeo em tempo real não seria possível com compressão sem perdas, nem que tenha alguma perda para que aconteça!

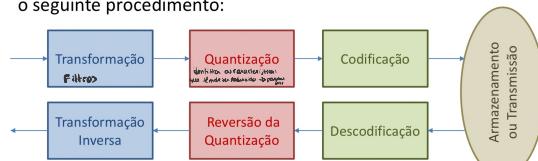
As técnicas de compressão com perdas consideram informação específica sobre a fonte geradora dos sinais, e também sobre o receptor (destinatário da informação).

### Exemplo:

O ser humano faz uso dos 5 sentidos, não sendo especialmente bom em nenhum particularmente, podendo estas limitações ser usadas para descartar informação cuja falta é imperceptível por este, o destinatário da informação com que estamos a trabalhar.

## Conclusão: a informação pouco relevante basta

- Muitos processos de compressão de dados com perdas seguem o seguinte procedimento:

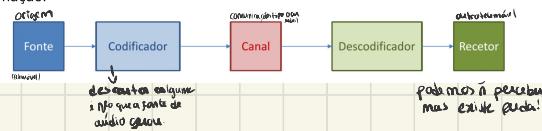


- Transformação:** Operação no domínio da frequência (por exemplo, filtrar determinadas bandas).
- Quantização:** Alocar diferentes números de bits a determinadas bandas de frequência (usado no áudio, por exemplo).
- Codificação:** Aplicação de códigos de Huffman, ou aritmético.

Começa por ser **transformado**, eliminando dados que são desnecessários para a compreensão do destinatário, seguido da **quantização**, que consiste na **atribuição de bits** aos símbolos a representar e por fim a **codificação**, a aplicação de código como os estudados anteriormente.

## 2. Compressão de Áudio

Neste tipo de codificação são tidas em conta a **redundância**, sendo eliminados os componentes redundantes do sinal e a **irrelevância**, a eliminação de elementos que cuja representação é irrelevante tendo em conta a forma como o receptor interpreta a informação.



pode nos à percepção mas evita ruído!

**Redundância** - associada ao gerador. Há características que não são úteis → descartá-las.

- L
- A eliminação de componentes redundantes no sinal implica conhecimento sobre as características de geração desse sinal.
  - A eliminação de informação irrelevante pressupõe conhecimento sobre a forma como o receptor interpreta a informação.

a codificação de música de flauta quando compondo um canto cuja intensidade não é tão alta quanto a humana!

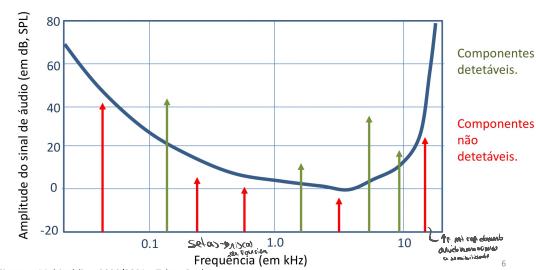
irrelevância → assim o áudio só é percebido pelo receptor

## Psicofisiologia

É uma disciplina que estuda a fisiologia auditiva e a sua relação com a acústica, através de espetrogramas, destacando-se três níveis de pesquisa:

### 1. Limiar Auditivo

**Limiar auditivo** o nível mínimo de pressão acústica necessária para provocar uma sensação auditiva, num ambiente silencioso;



### 2. Mascaramento da Frequência

**Mascaramento na Frequência:** A presença de sinais sonoros em determinadas bandas de frequência diminui a capacidade do ouvido para detetar sinais a outras frequências.



### 3. Tratamento no tempo

**Mascaramento no tempo** quando um sinal de elevada amplitude ocorre, a percepção de sinais de amplitudes mais baixas num determinado intervalo de tempo seguinte é imperceptível. Também é difícil percebermos sons que têm uma duração menor reduzida.

Nos casos em que se verifique alguma destas propriedades, podemos utilizar menos bits para codificar estas partes do sinal que não são tão perceptíveis, ou mesmo descartá-las.

Exemplos de compressão de áudio com perdas são o MPEG-1/2/3 (MP3).

Estes algoritmos entram em conta como espetrograma digital

**MPEG-I:** *prévisões, detalhes, rotulados*, não necessário saber

Modelo psico-acústico

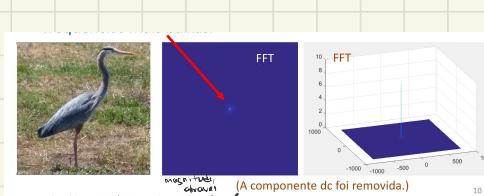
| Camada (Layer) | Máscara no Tempo | Máscara na Frequência | kbit/s    | Notas                    |
|----------------|------------------|-----------------------|-----------|--------------------------|
| 1              |                  | x                     | 30 - 448  | Philips DCC              |
| 2              | x                | x                     | 64 - 256  | DVD, DAB                 |
| 3 (MP3)        |                  | Modelo mais complexo  | 64 (mono) | Maior taxa de compressão |

### 3. Compressão de imagem

Uma abordagem possível é remover as componentes de frequências elevadas, uma vez que as predominantes são de baixa, sendoas as diferenças quase imperceptíveis ao olho humano.

Outra é a utilização de um modelo diferente do RGB, pois o olho humano é pouco sensível à cor (crominância), comparado com a intensidade da luz de cada pixel (luminância). Passa-se assim a utilizar o modelo YCbCr, sendo a primeira uma matriz de luminância e as duas últimas de crominância, cuja variação é menor, o que faz com que a quantidade de informação armazenada seja também menor (subamostragem).

Exemplos de compressão de imagens com perdas são os formatos JPEG, JPEG2000 e TIFF.



#### Alguns Formatos Standard:

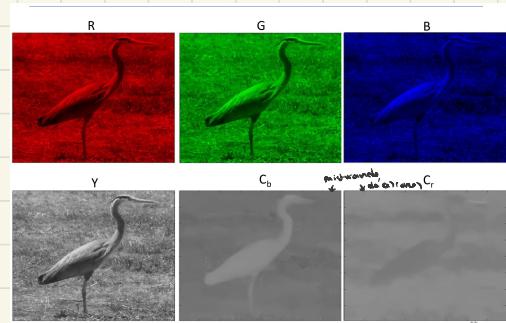
- **JPEG – Joint Photographic Experts Group**  
Efetua compressão com perda de informação.  
Obtém imagens com elevada taxa de compressão.
- **JPEG2000**  
Pode efetuar compressão com ou sem perda de informação.
- **TIFF – Tagged Image File Format**  
Pode efetuar compressão com ou sem perda de informação.

Também as características do sensor óptico humano são consideradas para se proceder a uma maior compressão de imagens, sem que se note uma diminuição da qualidade da imagem.

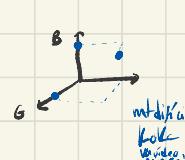
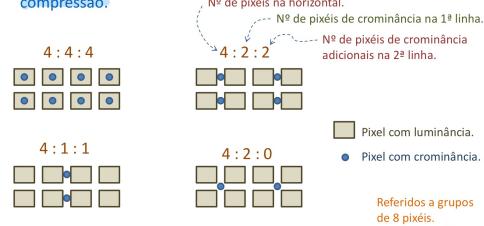
Por exemplo, o olho humano não é igualmente sensível às cores R, G e B, pelo que se tira melhor proveito se se usar uma outra forma de representar as cores, em vez do RGB.

O olho humano é mais sensível à intensidade de luz de cada pixel (luminância) do que à sua cor (crominância).

Assim, é usual transformar a representação RGB para a YCbCr (uma matriz de luminância (Y) e duas de crominância (Cb e Cr)).



- Aproveitando a menor variabilidade da crominância, e o facto de o olho humano não ser muito sensível a variações de cor, a **compressão da crominância** é muitas vezes subamostrada na compressão:



13

# 4. Compressão de Vídeo

Este tipo de formato não seria possível sem a compressão imagens com perdas, tendo em conta a quantidade de imagens que são armazenadas por cada segundo de vídeo.

Um filme com uma hora necessitaria de 112GB para ser armazenado!

Em acréscimo às técnicas de compressão utilizadas para comprimir as imagens com perdas, tira-se partido da **redundância temporal**, uma vez que *frames* consecutivos apresentam muita informação em comum. Existem assim dois tipos de algoritmos que utilizam este conceito:

**Codificadores preditivos** criam modelos matemáticos para descrever o movimento da imagem com base num *frame* modelo e codificam apenas os seus parâmetros;

**Codificadores de movimento** comparam *frames* consecutivos e codificam apenas informação das diferenças. Pode existir também um *frame* de referência, com o qual os seguintes são comparados.



- É usual um vídeo ser composto por diferentes tipos de *frame*, por exemplo:
  - **Frame de referência** – imagem em bruto, sem ser o resultado do processamento da informação de *frames* anteriores.
  - **Frame preditiva** – resultado de um *modelo preditivo* (usualmente complexo) que opera com base na *frame* de referência anterior, e nas diferenças entre *frames*.
- Operando sobre a evolução das diferenças, nomeadamente nas componentes de luminância e crominância separadamente, conseguem-se taxas de compressão muito significativas.
- Os modelos preditivos são muitas vezes suportados por algoritmos de identificação de entidades nas imagens diferença.

Sempre que há uma mudanças radical nas imagens em estudo (por exemplo de cenário), reiniciam-se os processos, sendo definida uma nova *frame* modelo.

Existem ainda outros processos que consideram uma maior compressão da imagem, com perdas visíveis, mas às quais o ser humano não presta atenção.

Um exemplo deste tipo de técnicas é a substituição da relva por retângulos verdes na transmissão de um jogo de futebol.

No entanto, estes tipos de compressão são muito direcionados ao tipo de conteúdo que está a ser codificado, pois a forma como o ser humano interpreta e processa as imagens de um jogo de futebol é muito diferente de uma novela ou de um concerto.

Neste aspecto a compressão de áudio é muito mais simples, pois os conceitos estudados pela psicoacústica podem ser aplicados de forma geral, havendo uma sendo possível uma abstração do que está a ser representado.

Exemplos da codificação de vídeo com perdas de informação são MPEG-2 (TV e HDTV) e 4 (MP4), Motion JPEG.

| Algoritmos de compressão de vídeo |  |
|-----------------------------------|--|
| Motion JPEG                       | Cada frame é comprimido em JPEG.<br>Não considera a redundância temporal.<br>É, por exemplo, usada em algumas máquinas fotográficas. |
| MPEG-2                            | Considera desde 352x288@25fps até 1920x1250@60fps (HDTV).<br>Usada em transmissão de TV e HDTV.                                      |
| H.261, H.263 e H.263+             | Vídeo-conferência sobre rede telefónica fixa (ISDN) ou móvel.  |
| H.264 (MPEG-4 AVC)                | Atualmente muito usado em transmissão de vídeo.  |

21