[1] $x(t) = \sin(2\pi t) + 4\cos(8\pi t)$

a) Frequência Fundamental

$x(t) = x_1(t) + x_2(t)$

$f_0 = \gcd(f_1, f_2) = \gcd(1,4) = 1\,Hz$

b) Expandir na SCF

Série Clássica de Fourier:

(...), (...), onde $T = (...)$

$\to c_1 = ...,\ c_2 = ...$

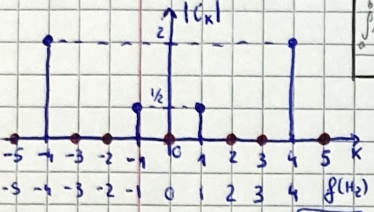
$\to$ todos os outros c_K são nulos!

c) Gráfico $|C_m|$

$|C_{\pm 1}| = ... \to$ Freq. $\pm 1 \times f_0 = ...$

(...) (...)

Nota: $F = \frac{K}{T} \Rightarrow F = K f_0$



Como a energia de $x(t)$ está espalhada pelas freq.

$\boxed{f = K f_0}$

[2] $x(t)$ periódica período T

a) Expanda na SCF

$\boxed{\sin(x) = \frac{e^{jx} - e^{-jx}}{2j}}$

$\boxed{\cos(x) = \frac{e^{jx} + e^{-jx}}{2}}$

Série Clássica de Fourier:

(...), (...), onde $T = (...)$

Integrar e obter $c_K = ...,\ K \neq 0$

$c_0 = \lim_{K \to 0} c_K = $ (L'Hopital Rule) (*)

b) Gráfico $|C_m|$

$|C_0| = ... \longrightarrow$ Freq.: $K \times f_0 = ...$

$|C_K| = ... \longrightarrow ...$

$|C_1| = \frac{2}{\pi},\ |C_2| = 0,\ |C_3| = \frac{2}{3\pi},\ |C_4| = 0$

$$C_K = \begin{cases} 0, & \text{se } K \text{ for par} \\ 1, & \text{se } K = 0 \\ \frac{2}{K\pi}, & \text{se } K \text{ for ímpar} \end{cases}$$

$x(t)$ é uma função par - $x(-t) = x(t)$, logo (...)

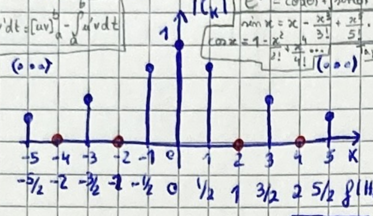
$\int_a^b u v' dt = [uv]_a^b - \int_a^b u' v\, dt$

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

$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} ...$

$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} ...$ (...) Taylor

(...)



$\boxed{f = K f_0}$

[3] $x(t) = \sin(2\pi t) + 4\cos(8\pi t)$

a) Frequência de amostragem mínima

Teorema da Amostragem (Nyquist-Shannon theorem): A frequência de amostragem tem de ser pelo menos o dobro da frequência máxima do sinal.

$\left|\frac{f_{MAX}}{f_a}\right| < \frac{1}{2}$, ou seja, $f_a > 2 \times f_{MAX}$

Assim, $f_a > 2 \times 4 = 8\,Hz$, logo, a frequência de amostragem mínima é $(8 + \varepsilon)\,Hz$ (ε infinitésimo)

b) Amostrado a 4 Hz, com que outro sinal com comp. com freq. < 2 se confunde?

$f_a = 4\,Hz,\ T_a = 1/4\,s,\ t \to mT_a$

$x(t) \to x(mT_a)$

$\sin(2\pi t) + 4\cos(8\pi t) \longrightarrow \sin(2\pi f_1 m T_a) + 4\cos(2\pi f_2 m T_a)$

$= \sin(2\pi \frac{f_1}{f_a} m) + 4\cos(2\pi \frac{f_2}{f_a} m)$,

$f_1 = 1\,Hz$ e $f_2 = 4\,Hz$.

"Sinais diferentes, mas indistinguíveis" $\longrightarrow \sin(2\pi \frac{f_1 + K_1 f_a}{f_a} m) + 4\cos(2\pi \frac{f_2 + K f_a}{f_a} m)$

$2\pi \frac{f_0}{f_a} m + 2\pi K m$ (periódico 2π)

Não verifica TA $\to \left|\frac{f_{MAX}}{f_a}\right| \geq 1/2$

$\to$ É sempre possível encontrar K_i de modo que $\left|\frac{f_i + K_i f_a}{f_a}\right| < 1/2,\ K_i \in \mathbb{Z}$ (Teorema da Amostragem)

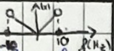
$\to$ Logo, ao reproduzir o tom, é essa frequência, entre $-\frac{f_a}{2}$ e $\frac{f_a}{2}$, que vai ser "ouvido"

Assim, o sinal amostrado confunde-se com todos os sinais da forma: $\sin(2\pi(f_1 + K_1 f_a)t) + 4\cos(2\pi(f_2 + K_2 f_a)t)$

$= \sin(2\pi(1 + 4K_1)t) + \cos(2\pi(4 + 4K_2)t),\ K_1, K_2 \in \mathbb{Z}$

(...) Resolver módulos: $|x| \leq a \Leftrightarrow x \leq a \wedge x \geq -a$!

[4] Transformada de Fourier:



a) f_a mínima amostrada

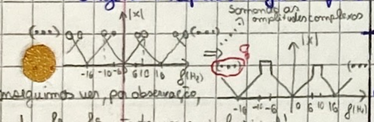
Como sabemos a TF do sinal é dada pela soma de todas as translações $X(j2\pi f)$ por múltiplos da frequência de amostragem. Em particular se $X(j2\pi f)$ for identicamente nula para $|f| \geq f_{MAX}$ então estas translações não se sobrepõem se: $f_{MAX} < f_a - f_{MAX}$ (teorema da amostragem), ou seja, $f_a > 2 \times f_{MAX}$

$\Rightarrow f_a > 2 \times (10 - \varepsilon) = 20 - \varepsilon'$, logo a frequência de amostragem mínima é $(20 - \varepsilon') + \varepsilon' = 20\,Hz$ (ε' infinitésimo)

b) TF quando amostrado a 16 Hz

$\Rightarrow \sum_{k=-\infty}^{+\infty} X(j2\pi(f - f_a k)),\ f_a = 16\,Hz$

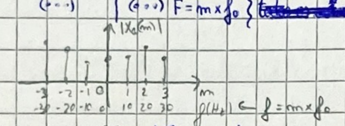
$\Rightarrow$ Original + réplicas de f_a em f_a Hz



Conseguimos ver, por observação, que de $-\frac{f_a}{2}$ a $\frac{f_a}{2}$ não temos o sinal original!

[5] $f_0 = 10\,Hz$ e $N = 10$ num período $\Rightarrow F_a = N f_0 = 100\,Hz$

|Coeficientes|	|Freq.|
(...)	(...) $F = m \times f_0$



b) $x_a(m)$ é real? Não! Existe pelo menos uma que tem parte imaginária não nula! Porque os valores da TDF não são complexos conjugados, logo ao fazermos o somatório, para obtermos o sinal no tempo, a parte imaginária não se anula. Vejamos um exemplo para $n = 0$:

$x(m) = (...)$, $x(0) = 1 + 3 - 4j + 3 + 4j + ...$ (TDF)

$\boxed{X_a(-m) = X_a(N - m)}$

$= \underbrace{...}_{\text{Real}} + \underbrace{j}_{\text{imaginária}}$!

c) Para o caso da quantização uniforme, o erro de quantização está uniformemente distribuído entre $-\frac{\Delta}{2}$ e $\frac{\Delta}{2}$, logo o erro quadrático médio de quantização é dado por: $E = \int_{-\infty}^{+\infty} e^2 f_E(e)\,de$, sendo $f_E(e) = \begin{cases} 1/\Delta, & |e| \leq \frac{\Delta}{2} \\ 0, & |e| > \Delta/2 \end{cases}$

Então:

$$E = \frac{1}{\Delta}\int_{-\Delta/2}^{+\Delta/2} e^2\,de = \left.\frac{e^3}{3\Delta}\right|_{-\Delta/2}^{+\Delta/2} = \frac{\Delta^2}{12}$$

, considerando $N = 2^b$ níveis e assumindo que $|x(t)| \leq A$, o $\Delta = \frac{2A}{N}$, logo para b bits o erro de quantização, em dB, é dado por:

$10\log_{10}(E) = 10\log_{10}\left(\frac{\Delta^2}{12}\right) = 10\log_{10}\left(\frac{A^2}{3N^2}\right)$

$= 20\log_{10}(A) - 10\log_{10}(3) - 20\log_{10}(2^b)$

$= 20\log_{10}(A) - \text{const.} - 6.0206b$

Assim, a redução por acrescentar 1 bit, é dada por: $E_{dB}(b) - E_{dB}(b+1) = 20\log_{10} 2 \approx 6\,dB$

Extra: $\frac{E(b+1)}{E(b)} = 0.25 \Leftrightarrow 10\log_{10}\left(\frac{E(b+1)}{E(b)}\right) = 10\log_{10}(0.25)$

Redução 75%

$\Leftrightarrow 10\log_{10}(E(b+1)) - 10\log_{10}(E(b)) = $!

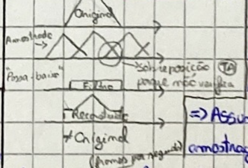
$\Leftrightarrow E_{dB}(b+1) - E_{dB}(E(b)) = -6 \Leftrightarrow -6 = -6$

c.q.m

[6] Aliasing

$\Rightarrow$ Aliasing é o fenómeno que acontece quando se amostra um sinal com frequência baixa demais. Ou seja, não verificando a condição do Teorema da Amostragem, $\left|\frac{f_{MAX}}{f_a}\right| < \frac{1}{2}$.

$\boxed{* \boxed{3} b)}$



• Um exemplo de aliasing é quando as rodas dos carros, nos filmes, começam a rodar ao contrário, este fenómeno é por vezes conhecido como "wagon-wheel effect"

$\Rightarrow$ Assumindo que a frequência de amostragem (f_a) de uma câmara é de 30 Hz (frames por segundo), qualquer frequência de sinal (f_0) maior do que a metade da freq. de amostragem (15 Hz) reproduzido por essa câmara irá sofrer desse fenómeno (rodas do carro rodam mais do que 15 vezes por segundo...)

[7] $x(t) = \sin(2\pi t) + 4\cos(8\pi t)$ quantiz. com 10 bits

$b = 10$ bits, $N = 1024$ níveis de quantização

a) $\max|x(t)| \Rightarrow$ Para a estimativa irei utilizar a soma de cada componente de $x(t)$, sabendo que, $|x(t)|$ será sempre menor ou igual à soma dos módulos dos seus componentes

$|x(t)| = |\sin(2\pi t) + 4\cos(8\pi t)| \leq |\sin(2\pi t)| + 4|\cos ...|$

Logo, $A = \max|x(t)| \leq 5$, (≤ 5 Majorante) podemos usar como estimativa o valor 5

b) Passo de quantização uniforme: Δ

$\Delta \approx \frac{2A}{N} = \frac{2 \times 5}{1024} = \frac{10}{1024} \approx 10^{-2}$ Substituir!

Sinal quantizado: $q(t) = Q(x(t))$ $(= \text{round}\left(\frac{x(t)}{\Delta}\right))$

Sinal reconstruído:

$\hat{x}(t) = Q^{-1}(Q(x(t))) = \Delta\, q(t)$

Ou seja,

Tem erro associado

$x(t) \longrightarrow \boxed{Q} \xrightarrow{q(t)} \boxed{Q^{-1}} \longrightarrow \hat{x}(t)$

Quantização Reconstrução

Erro absoluto: $\frac{\Delta}{2} = 5 \times 10^{-3}$ (metade do nível)

Erro quadrático médio: $\boxed{c)}$

[9] Sistema linear $x(m) \longleftrightarrow X(z) = \sum_{m=-\infty}^{+\infty} x(m) z^{-m}$

$H(z) = \frac{Y(z)}{X(z)}$, func. transf. $y(m)$ (...)

a) Calcular função transferência $\boxed{H(z) = 1 - z^{-2}}$

b) Resposta em frequência $\Rightarrow z = e^{j\omega},\ \omega = 2\pi \frac{f_0}{f_a}$

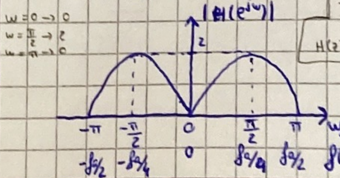
$H(e^{j\omega}) = 1 - e^{-2j\omega} = e^{-j\omega}(2j\sin(\omega))$

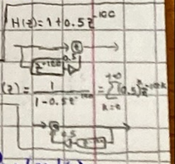
$|H(e^{j\omega})| = |2j\sin(\omega)| = 2|\sin(\omega)|$

$\omega = 0 \to 0$

$\omega = \frac{\pi}{2} \to 2$

$\omega = \pi \to 0$





Função par: $x(-t) = x(t)$ [cosseno]

Função ímpar: $x(-t) = -x(t)$ [seno]

$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} + ...$

$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - ...$

$e^x = 1 + \frac{x}{1} + \frac{x^2}{2!} + \frac{x^3}{3!} + ...$

$\cos(A+B) = \cos(A)\cos(B) - \text{sen}(A)\text{sen}(B)$

$\text{sen}(A+B) = \text{sen}(A)\cos(B) + \text{sen}(B)\cos(A)$

$e^{j\theta} = \cos\theta + j\sin\theta \quad \cos\theta = \frac{e^{j\theta} + e^{-j\theta}}{2} \quad \sin\theta = \frac{e^{j\theta} - e^{-j\theta}}{2j}$

$\log_a b = \frac{\log_c b}{\log_c a}$

[10] Huffman codes ⇒ Número de letras = 1600

$p(A) = \frac{7}{16}$, $p(B) = \frac{1}{16}$, $p(C) = \frac{3}{16}$, $p(D) = 1/16$

$P(E) = \frac{2}{16}$, $P(F) = \frac{2}{16}$

a) Entropia ⇒ $H = -\sum_{k=1}^{N} p_k \log_2(p_k)$

(...) $H = \frac{35.596}{16}$ $\underline{BPS}$ ↳ Bits por símbolo

b) Código

~~c) Entropia~~ $C \to 111$, $D \to 1011$, $E \to 100$, $F \to 110$

$A \to 0$, $B \to 1010$,

$N_m = 1 \times \frac{7}{16} + 4 \times \frac{1}{16} + 3 \times \frac{3}{16} + 4 \times \frac{1}{16} \ldots$
$= 2.25$ BPS

Para 1600 letras ⇒ $1600 \times 2.25 = 3600$ bits ↳ $\boxed{H \leq N_m < H+1}$

c) 1 erro de um bit, possível recuperar?

Sim! Os códigos de Huffman têm a propriedade de se auto-sincronizarem após um erro de um bit na receção. Essa auto-sincronização pode demorar mais ou menos bits, dependendo do código e da info. a codificar. Exemplo:

$\underbrace{0}_{A}\ \underbrace{111}_{C}\ \underbrace{1011}_{D}\ \underbrace{100}_{E}\ \underbrace{110}_{F}\ \underbrace{0}_{A}$

erro!

$\underbrace{0}_{A}\ \underbrace{1\boxed{0}1\ 1}_{D}\ \underbrace{0}_{A}\ \underbrace{11\ 1}_{C}\ \underbrace{0}_{A}\ \underbrace{0}_{A}\ \underbrace{110}_{F}\ \underbrace{0}_{A}$ → Propagação do Erro!

[11] Codificação aritmética:

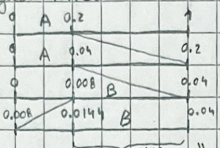
Na codificação aritmética a prob. de ocorrência de cada símbolo é representada em intervalos. Ou seja, parte-se o intervalo $[0, 1[$ e nele identifica-se o sub-intervalo ao qual corresponde o primeiro símbolo lido. Para cada símbolo subsequente, subdivide-se o intervalo atual em sub-intervalos proporcionais às probabilidades da tabela de intervalos, e encontra-se novamente o intervalo que corresponde ao próximo símbolo. No final do processo, teremos um intervalo que corresponde à probabilidade de ocorrência de todos os símbolos na ordem correta. A saída do algoritmo é então um valor que esteja contido nesse intervalo e que possa ser representado com o menor número de bits. Para descodificar, basta expandir o intervalo correspondente e repetir esse raciocínio até que se atinja o limite de resolução imposta pelo número de casas decimais do valor fracionário correspondente aos bits recebidos. Como os códigos aritméticos não se auto-sincronizão, pois tratam o conjunto de carateres como uma unidade única, não é possível recuperar a informação após a ocorrência de um erro. Embora os códigos sejam mais eficientes que os códigos de Huffman, N_m mais próximo da entropia, eles não se auto-sincronizão (ao contrário dos de Huffman)

e.g.

Símbolo	Probabilidade	Intervalo
A	0.2	[0; 0.2[
B	0.8	[0.2; 1[

Na prática seria em base 2

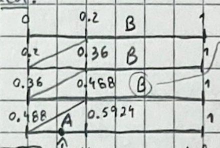
Codificar "AABB":



Para codificar "AABB" basta enviar um valor em $[0.0144, 0.0400[$, e.g. envio $\boxed{0.0200}$

Vamos supor que existe um erro no primeiro bit decimal, $0 \to 1$, ou seja o valor recebido é $\boxed{0.5200}$

Ao descodificar:



Ao contrário dos códigos de Huffman depois de atingir o símbolo correto ele não auto-sincroniza

Mensagem codificada: A A B B

Mensagem descodificada: B B B A

× × ✓ ×

[12] LZW:

A tecnica LZW trata-se de uma tecnica de compressão sem perdas de dados baseada num dicionário e no registro de padrões da estrutura, esses padrões permitem a compressão ser eficiente (eg. o formato GIF utiliza este tipo de compressão). Segue um exemplo, passo por passo para codificar "AABAAAB", sabendo que os símbolos possíveis são "A", "B" e "C".

[1] Iniciar o dicionário com todos os símbolos possíveis

[2] Repetir até chegar ao fim dos símbolos para codific.

[2a)] C = próximo símbolo

[2b)] Se $\boxed{PC}$ está no dicionário:

→ P = P concatenação com C

→ voltar [2a)]

[3] Enviar índice de $\boxed{P}$

[2c)] Enviar para a saída o índice de $\boxed{P}$ e adicionar $\boxed{PC}$ ao dicionário

	P	C	Símbolo	Índice	Saída
	Ø	Ø	A	1	Ø
	Ø	Ø	B	2	Ø
[1]	Ø	Ø	C	3	Ø
[2]	Ø	A	—	—	—
	A	A	AA	4	1
	A	B	AB	5	1
	B	A	BA	6	2
	A	A	—	—	—
	AA	A	AAA	7	4
	A	B	—	—	—
	AB	Ø	—	—	5
[3]					

Notar que:

a compressão na prática vai se adaptando, ficando cada vez maior, número de bits por "número" aumenta

$\underbrace{\text{"1", "1", "2", "4", "1"}}_{\text{5 "números"}}$, considerando o número de bits de cada "número" constante e igual ao número de bits por "símbolo":

Rácio $= \frac{7}{5} = 1.4$

Ou seja, nesta pequena sequência (não tinha muito proveito de padrões) conseguimos comprimir 1.4 unidades de informação do fluxo de dados original numa única unidade de informação no fluxo de dados comprimidos

RLE (Run Length Encoding): este tipo de compressão é muito utilizado essencialmente quando existem poucos símbolos e quando a probabilidade de repetição é muito elevada. A ideia é compactar uma sequência de símbolos contando-se quantas vezes aparecem consecutivamente e envia-se o símbolo e a respetiva contagem.

Exemplo: AAADBCAAA

Codificado: A(3), B(2), C(1), A(3)

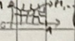
Variante: Assumindo codificação de 1 byte por símbolo. Envia-se 2 tipos de informação:

[1] Para repetição $\boxed{N|S}$, $N = 0, \ldots, 127$, $(N+1)$ repetições de S

[2] Sem repetição $\boxed{N|S_1|S_2|\ldots|S_{N-126}}$, $N = 128 \ldots 255$ $N-126$ símb. (pelo menos 2)

[8] Lloyd-Max:

O algoritmo de Lloyd-Max é um algoritmo de quantização não-uniforme, ou seja, os níveis de quantização não estão igualmente espaçados uns dos outros. A ideia deste algoritmo é, dado $f_x(x)$ e o número de níveis de quantização (N), calcular os níveis ótimos de quantização. Como funciona: [1] Escolher de forma arbitrária, os N níveis de quantização (todos diferentes e por ordem crescente) [2] Ajustar os pontos de descontinuidade (limiares de decisão) da função $Q(x)$, usando $x_m = \frac{q_m + q_{m+1}}{2}$, $m = 1, 2, \ldots, N-1$. Os limites exteriores, x_0 e x_N são fixos e dependem da região de suporte de $f_x(x)$. Deve-se também garantir que os valores de x_k são simétricos, no caso de $f_x(x)$ também o ser. [3] Para cada intervalo $]x_{k-1}, x_k[$, com $k = 1, 2, \ldots, N$, calcula-se o nível de quantização que minimiza a distorção quadrática quando se está dentro desse intervalo. A minimização de $\int_{x_{k-1}}^{x_k} (x - q_k)^2 f_x(x)\,dx$, onde $q_k = \int_{x_{k-1}}^{x_k} x f_x(x)\,dx$, ou seja, q_k é o valor médio de x em $]x_{k-1}, x_k[$. Agora, com base num "threshold" verificamos se os valores de q_k diferem menos que esse limiar e terminamos o algoritmo. Caso contrário, voltamos ao ponto [2] e repetimos até o algoritmo convergir.

Quantização vetorial: Para além disso, este algoritmo pode ser generalizado para sinais multidimensionais. A ideia é a mesma. Escolhe-se um conjunto de N pontos no espaço multidimensional para serem os valores quantizados, para cada valor quantizado determina-se a região do espaço que está mais perto desse valor do que de todos os outros, para cada região substitui-se o valor quantizado pelo centro de massa da região, designado por "centroide", e repete-se até convergir. A duas dimensões a subdivisão do plano nas regiões que estão mais próximas de um ponto do que de todos os outros chama-se diagrama de Voronoi.  → Regiões de Voronoi. A quantização de cores (espaço de cores) numa imagem ou filme é um caso onde a quantização vetorial pode ser útil. Definimos paletes de cores (eg. 256 cores) e precisamos de escolher as cores para cada região, quantização vetorial pode ser utilizada para fazer isso. Um espaço de cores muito conhecido é o espaço de cores aditivo, o famoso "RGB", usado para emitir diretamente uma imagem.

TF para compressão: A transformada discreta de Fourier pode ser considerada como uma maneira de determinar os coeficientes da combinação linear dos N sinais elementares x_m. Como cada um desses coeficientes ~~será~~ está associado a um sinal com uma frequência bem determinada, a DFT é muito útil para analisar o conteúdo espetral de um sinal. Mas, quando o que se pretende é armazenar o sinal numa forma comprimida, o uso de números complexos não é muito conveniente. Seria preferível usar apenas números reais. Para efeitos de compressão de sinais, os sinais x_m a usar devem permitir representações esparsas dos sinais que aparecem com mais frequência na prática, isto é, representações em que se a maior parte dos valores $X(m)$ são pequenos, substituir esses valores por zero ou por uma versão quantizada com poucos bits permite aumentar a taxa de compressão.