

Sumários detalhados das aulas TP de Sistemas Multimédia 2023/2024

Tomás Oliveira e Silva

Estão previstas as seguintes aulas teóricas:

Data	Tema
18/09	Introdução e motivação.
25/09	Série clássica de Fourier.
02/10	Transformada discreta de Fourier (DFT).
09/10	Aula de motivação para a matéria seguinte. Uso do MATLAB para analisar um sinal de áudio.
16/10	Teorema de amostragem. Teoria dos sistemas lineares invariantes no tempo. Resposta impulsional. Resposta em frequência.
23/10	Transformadas Z. Equações de diferenças.
30/10	Outras transformadas (DCT, DST, duas dimensões).
06/11	Representações tempo-frequência.
13/11	Breve menção a espaços de cores. Quantização.
20/11	Entropia. Códigos de Huffman.
27/12	TBD
04/12	TBD
11/12	TBD
18/12	TBD

Estão previstas as seguintes aulas práticas:

Datas	Tema
20/09 e 21/09	Apresentação da Disciplina. Introdução ao MATLAB (slides + guião número 1)
27/09 e 28/09	Sinusoides e revisões sobre número complexos (guião número 2)
04/10 e 13/10	Decomposição de sinais em série de Fourier (guião número 3)
11/10 e 20/10	Transformada discreta de Fourier (guião número 4)
18/10 e 27/10	Amostragem e reconstrução de sinais (guião número 5)
25/10 e 03/11	Manipulação do espectro de sinais (guião número 6)
08/11 e 09/11	Primeiro teste prático
15/11 e 16/11	Processamento de sinais áudio (guião número 7)
22/11 e 23/11	Codificação de informação (guião número 8)
29/11 e 30/11	Codificação eficiente de informação (guião número 9)
06/12 e 07/12	Compressão de ficheiros de imagem, parte 1 (guião número 10)
13/12 e 14/12	Compressão de ficheiros de imagem, parte 2 (guião número 11)
20/12 e 21/12	Segundo teste prático

Aula do dia 18 de setembro de 2023 (tablet)

Regras de avaliação:

- Um teste teórico final, a ser feito na época de exames, a valer 50% da nota. O teste tem doze perguntas de 2 valores cada, e contam as dez melhores respostas.
- Dois testes práticos, a serem feitos em grupos de 2, um a meio do semestre e outro no fim do semestre, cada um deles valendo 25% da nota.

O conceito de sinal. Amostragem e quantização. Problemas relacionados com a amostragem demasiado lenta de sinais (rodas de carros em filmes paradas ou a rodar ao contrário, idem para pás de helicópteros).

Na amostragem uniforme substitui-se a variável t ($t \in \mathbb{R}$) por nT_a ($n \in \mathbb{Z}$), onde T_a é o período de amostragem ($f_a = 1/T_a$ é a frequência de amostragem). Assim, amostrar o sinal $x(t)$ corresponde a se obter as amostras $x(nT_a)$. Em particular, com um tom puro de f_o Hertz (ciclos por segundo), temos

$$\sin(2\pi f_o t) \rightarrow \sin(2\pi f_o nT_a) = \sin\left(2\pi \frac{f_o}{f_a} n\right).$$

Note que nesta última expressão o que importa é a razão $\frac{f_o}{f_a}$. Mais concretamente, os tons puros

$$\sin\left(2\pi \frac{f_o + kf_a}{f_a} n\right), \quad k \in \mathbb{Z}$$

são indistinguíveis. Se $\left|\frac{f_o}{f_a}\right| < \frac{1}{2}$ é possível determinar inequivocamente a frequência f_o a partir das amostras do sinal ($k = 0$). Em geral, é sempre possível encontrar um k de modo a que $\left|\frac{f_o + kf_a}{f_a}\right| < \frac{1}{2}$. Ao reproduzir o tom, é essa frequência, entre $-f_a/2$ e $+f_a/2$, que vai ser ouvida (*aliasing*).

No fim da aula foi feita uma demonstração de áudio sobre o *aliasing*. Usando o *script* MATLAB apresentado a seguir, foi criado um sinal com uma frequência que aumenta linearmente com o tempo. Inicialmente ouve-se uma tom com uma frequência a aumentar, mas a partir de uma certa altura (quando a frequência do tom ultrapassa metade da frequência de amostragem), a frequência do tom começa a diminuir.

Conteúdo do ficheiro [SM_2023_09_18.m](#):

```
f0=200; % frequency of the sinusoid
Fs=8000; % sampling frequency
t=(0:5*Fs)/Fs; % time instants for 5 seconds
x=sin(2*pi*f0*t); % the sinusoid
sound(x,Fs); % output sound

help chirp % how to use chirp()
x=chirp(t,500,2,4000); % a chirp signal
sound(x,Fs);
```

Devido à presença de um aluno Erasmus, que mais tarde optou por outra unidade curricular, a aula foi lecionada em Inglês.

Aula do dia 25 de setembro de 2023 (tablet)

Em matemática usa-se i para designar a unidade imaginária (uma das raízes quadradas de -1). Em eletrotécnica a letra i é geralmente usada para designar uma intensidade de corrente, pelo que é usual na área de processamento digital de sinal, que teve o pessoal da eletrotécnica como principais fundadores, usar em sua vez a letra j :

$$j = \sqrt{-1}.$$

É essa convenção que vamos usar em Sistemas Multimédia.

É fundamental conhecer as seguintes fórmulas:

$$e^{jx} = \cos x + j \sin x, \quad \cos x = \frac{e^{jx} + e^{-jx}}{2} \quad \text{e} \quad \sin x = \frac{e^{jx} - e^{-jx}}{2j}.$$

Estas fórmulas são uma consequência imediata dos desenvolvimentos em série de Taylor das funções exponencial, co-seno e seno.

Diz-se que uma função $x(t)$ é periódica com período T quando

$$x(t + T) = x(t), \quad \forall t \in \mathbb{R}.$$

O período fundamental é o período mínimo possível. Todos os períodos dessa função são múltiplos do período fundamental.

Quando uma função é a soma de duas funções periódicas, com períodos fundamentais T_1 e T_2 , todos os períodos da função soma serão ao mesmo tempo múltiplos inteiros tanto de T_1 como de T_2 , ou seja,

$$T = k_1 T_1 = k_2 T_2, \quad k_1, k_2 \in \mathbb{Z}.$$

Isto só é possível se T_1/T_2 for um número racional. Nesse caso, o período fundamental será o mínimo múltiplo comum de T_1 e T_2 , ou, o que é o mesmo, a frequência fundamental será o máximo divisor comum das duas frequências fundamentais.

Da unidade curricular de Cálculo 2, devem saber que uma função periódica suficientemente regular (contínua, por exemplo), pode ser expressa na seguinte soma de co-senos e senos (série clássica de Fourier):

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} a_k \cos\left(2\pi k \frac{t}{T}\right) + \sum_{k=1}^{+\infty} b_k \sin\left(2\pi k \frac{t}{T}\right).$$

Existem fórmulas para os coeficientes a_k e b_k que não vamos reproduzir aqui, pois é muito mais prático usar as exponenciais complexas $e^{j2\pi k \frac{t}{T}}$ em vez dos co-senos e senos. De facto, temos

$$x(t) = \sum_{k=-\infty}^{+\infty} c_k e^{j2\pi k \frac{t}{T}}, \quad c_k = \frac{1}{T} \int_0^T x(t) e^{-j2\pi k \frac{t}{T}} dt.$$

A convergência de

$$x_n(t) = \sum_{k=-n}^{+n} c_k e^{j2\pi k \frac{t}{T}}$$

para $x(t)$ não é, em geral, nem uma convergência uniforme nem mesmo uma convergência pontual, é uma convergência em norma.

Para quem quer saber mais

Uma maneira elegante de abordar a série clássica de Fourier consiste em definir uma classe de funções (espaço vetorial das funções periódicas de período T) e em definir um produto interno entre duas funções desse espaço vetorial:

$$\langle x(t), y(t) \rangle = \frac{1}{T} \int_0^T x(t) y^*(t) dt,$$

onde um asterisco designa conjugação complexa (mesma parte real, parte imaginária com sinal contrário). O produto interno induz uma norma,

$$\|x(t)\| = \sqrt{\langle x(t), x(t) \rangle} = \sqrt{\frac{1}{T} \int_0^T |x(t)|^2 dt},$$

e apenas as funções periódicas com norma finita pertencem ao espaço vetorial. Constatase facilmente que as funções

$$x_k(t) = e^{j2\pi k \frac{t}{T}}, \quad k \in \mathbb{Z}$$

são ortonormais, isto é, temos

$$\langle x_u(t), x_v(t) \rangle = \delta_{uv} = \begin{cases} 1, & \text{se } u = v, \\ 0, & \text{se } u \neq v. \end{cases}$$

É mais complicado provar que este conjunto de funções é completo, isto é, que qualquer função $x(t)$ que pertence ao espaço vetorial pode ser expressa por uma combinação linear destas funções:

$$x(t) = \sum_{k=-\infty}^{+\infty} c_k x_k(t). \quad (1)$$

Como as funções $x_k(t)$ são ortonormais, é muito simples calcular c_k :

$$c_k = \langle x(t), x_k(t) \rangle.$$

Como o quadrado da norma de $x(t)$, que é dada por $\sum_{k=-\infty}^{+\infty} |c_k|^2$, é finito, o quadrado da norma da aproximação de $x(t)$ por uma versão truncada de (1),

$$e_n(t) = x(t) - \sum_{k=-n}^{+n} c_k x_k(t) = \sum_{|k| > n} c_k x_k(t)$$

é $\sum_{|k| > n} |c_k|^2$ e converge para zero quando n tende para infinito. É por isso que se diz que a convergência é em norma.

É possível verificar (faça-o) que

$$a_k = c_k + c_{-k} \quad \text{e que} \quad b_k = j(c_k - c_{-k}).$$

Para o sinal de período T (aqui $0 < a < \frac{T}{2}$)

$$x(t) = \begin{cases} 1, & \text{se } |t| < a, \\ 0, & a \leq |t| \leq \frac{T}{2}, \\ x(t-T), & \text{se } t > \frac{T}{2}, \\ x(t+T), & \text{se } t < -\frac{T}{2}, \end{cases}$$

os coeficientes c_k são dados por

$$\begin{aligned} c_k &= \frac{1}{T} \int_0^T x(t) e^{-j2\pi k \frac{t}{T}} dt \\ &= \frac{1}{T} \int_{-a}^{+a} e^{-j2\pi k \frac{t}{T}} dt \\ &= \frac{1}{T} \left. \frac{e^{-j2\pi k \frac{t}{T}}}{-j2\pi \frac{k}{T}} \right|_{-a}^{+a} \\ &= \frac{\sin\left(2\pi k \frac{a}{T}\right)}{\pi k}. \end{aligned}$$

Esta fórmula dá o resultado correto também para $k = 0$:

$$\lim_{k \rightarrow 0} c_k = \frac{2a}{T} = \text{valor médio da função } x(t).$$

No fim da aula foi feita uma demonstração da aproximação desta função periódica com séries truncadas de Fourier com cada vez mais termos. Usando o *script* MATLAB apresentado a seguir, verificou-se que as séries truncadas com cada vez mais termos aproximavam, em geral, cada vez melhor $x(t)$, mas que junto das descontinuidades desta função a amplitude do erro não diminuía (mas o pico ficava cada vez mais estreito). A isto chama-se fenómeno de Gibbs.

Conteúdo do ficheiro [SM_2023_09_25.m](#):

```
% signal parameters
T=2;
a=0.3;
% time instants
t=(-1000:1000)/1000*(1.5*T);
% approximation with only the n=0 term
x=c(0,T,a)*ones(size(t));
plot(t,x);
grid on;
axis([-1.5*T,+1.5*T,-0.2,+1.2]);
title('n=0');
drawnow;
for n=1:100
    % adding the n and -n terms
    x=x+c(n,T,a)*exp(1j*2*pi*n*t/T) + ...
        c(-n,T,a)*exp(1j*2*pi*(-n)*t/T);
    pause(0.1);
    plot(t,real(x));
    grid on;
    axis([-1.5*T,+1.5*T,-0.2,+1.2]);
    title(sprintf('%d <= n <= %d',-n,n));
    drawnow;
end

% the expansion coefficients
function cn = c(n,T,a)
    if n==0
        cn = 2*a/T;
    else
        cn = sin(2*pi*n*a/T)/(pi*n);
    end
end
```

Desafio (observe experimentalmente). Qual é a amplitude do pico do erro da aproximação junto da descontinuidade? Esse pico depende do número de parcelas usadas na aproximação? Como é que a largura desse pico se comporta quando o número de parcelas aumenta?

Aula do dia 2 de outubro de 2023 (tablet)

Foi descrito como se calcula o período fundamental de uma função que é a soma de duas funções periódicas (ver detalhes no material da aula anterior).

O que é que acontece na série clássica de Fourier quando se amostra o sinal $x(t)$, de período T , N vezes num período? Neste caso usa-se um período de amostragem de $T_a = T/N$ e podemos fazer a seguinte aproximação (note que na aproximação fazemos $t = nT_a$, pelo que dt deve ser substituído pelo salto T_a que é dado nos valores de t)

$$\begin{aligned} c_k &= \frac{1}{T} \int_0^T x(t) e^{-j2\pi k \frac{t}{T}} dt \\ &\approx \frac{1}{T} \sum_{n=0}^{N-1} x(nT_a) e^{-j2\pi kn \frac{T_a}{T}} T_a \\ &= \frac{1}{N} \sum_{n=0}^{N-1} x(nT_a) e^{-j2\pi \frac{kn}{N}} \\ &= \hat{c}_k \end{aligned}$$

A partir dos coeficientes \hat{c}_k seria depois possível obter uma aproximação de $x(t)$. Note no entanto que apareceu um problema: estes novos coeficientes são periódicos (com período N). De facto, é muito fácil confirmar que $\hat{c}_{N+k} = \hat{c}_k$ para todos os valores inteiros de k . Só podemos pois, usar valores de k de 0 a $N-1$ (um período), ou, melhor ainda, para $|k| < \frac{N}{2}$.

É, no entanto, possível tornar as coisas exatas.¹ Para isso, considere que amostrou uma função $x(t)$, periódica ou não, obtendo apenas as N amostras $x(0), x(T_a), \dots, x((N-1)T_a)$. Sem perda de generalidade, nas fórmulas que vamos apresentar a seguir vamos ignorar o período de amostragem (fica implícito). Assim as amostras passam a ser $x(0), x(1), \dots, x(N-1)$, onde o argumento é o número da amostra

¹O que se segue é uma dedução simplificada da Transformada Discreta de Fourier, em Inglês *Discrete Fourier Transform*, ou DFT. Não foi feito na aula teórica por falta de tempo, mas apresentamos aqui a teoria porque não é complicado.

(ou índice de um *array*). A partir destas N amostras vamos calcular o equivalente a \hat{c}_k (na literatura sobre este assunto é usual usar $X(m)$ em vez de \hat{c}_k , pelo que é isso que fazemos aqui; é a mesma fórmula mas usando uma notação ligeiramente diferente):

$$X(m) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{mn}{N}}, \quad m = 0, 1, \dots, N-1.$$

Note que $X(m)$ é uma função periódica de período N . Passa-se dos N valores de $x(n)$ para os N valores de $X(m)$ fazendo uma transformação linear.² Fazendo $W_N = e^{j2\pi/N}$, temos

$$\underbrace{\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix}}_{\mathbf{X}} = \frac{1}{N} \underbrace{\begin{bmatrix} W_N^{-0} & W_N^{-0} & \cdots & W_N^{-0} \\ W_N^{-0} & W_N^{-1} & \cdots & W_N^{-(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ W_N^{-0} & W_N^{-(N-1)} & \cdots & W_N^{-(N-1)(N-1)} \end{bmatrix}}_{\mathbf{W}} \underbrace{\begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}}_{\mathbf{x}}$$

Em notação matricial temos $\mathbf{X} = \frac{1}{N} \mathbf{W} \mathbf{x}$. A $\frac{1}{\sqrt{N}} \mathbf{W}$ chama-se matriz de Fourier. A inversa esta matriz é o seu conjugado, pelo que é fácil obter \mathbf{x} a partir de \mathbf{X} . O ficheiro [SM_2023_10_02_F.m](#), cujo conteúdo se apresenta a seguir, ilustra isso.

```
N=10; % value of N
mn=(0:N-1)'*(0:N-1); % exponents
F=exp(-2i*pi/N*mn)/sqrt(N); % Fourier matrix
norm(inv(F)-conj(F)) % inv(F) is almost
% equal to conj(F)
```

A menos de um fator de escala, basta pré-multiplicar \mathbf{X} por esse conjugado (ou seja, é usar W_N^{-1} em vez de W_N !). Em termos de somatórios, e tendo em atenção o fator de escala, obtemos

$$x(n) = \sum_{m=0}^{N-1} X(m) e^{+j2\pi \frac{mn}{N}}, \quad n = 0, 1, \dots, N-1. \quad (2)$$

²Sim, o que aprenderam em Álgebra Linear e Geometria Analítica é útil!

Note que $x(n)$ fica, por causa desta fórmula, também uma função periódica de período N .

No MATLAB a DFT — $x(n) \rightarrow X(m)$ — pode ser calculada usando a função `fft`, que, cuidado, não faz a divisão por N , e a transformada inversa — $X(m) \rightarrow x(n)$ — pode ser calculada usando a função `ifft`, que, essa sim, faz a divisão por N .

Do que foi exposto atrás, podemos associar $X(m)$ a \hat{c}_m . Ora \hat{c}_m , que é uma aproximação de c_m , está associado ao sinal (lembre-se que $T = NT_a$)

$$e^{j2\pi m \frac{t}{T}} = e^{j2\pi m \frac{t}{NT_a}} = e^{j2\pi \left(\frac{m}{N} f_a\right) t},$$

ou seja, $X(m)$ está associado à frequência $\frac{m}{N} f_a$. Note que devido à periodicidade de $X(m)$, deve-se interpretar o valor $m = N - 1$ como sendo $m = -1$, $m = N - 2$ como $m = -2$ e assim por diante.

O ficheiro [SM_2023_10_16.m](#), cujo conteúdo se apresenta a seguir, ilustra a associação dos coeficientes $X(M)$ às respetivas frequências (sem usar frequências negativas).

```
N=1000;           % number of samples
Ta=0.001;         % sampling period
fa=1/Ta;          % sampling frequency
fo=20;            % signal frequency
t=(0:N-1)*Ta;     % time instants
x=sin(2*pi*fo*t); % signal
X=fft(x)/N;       % DFT
f=(0:N-1)/N*fa;   % DFT frequencies
fh=figure(1);     % get figure handle
subplot('Position',[0.05 0.05 0.9 0.9]);
plot(f,abs(X));    % graph abs(DFT)
grid on;
fh.PaperType='A4';
fh.PaperOrientation='landscape';
print('DFT.pdf','-dpdf','-noui','-fillpage');
```

Vê-se um pico com amplitude 0.5 para a frequência 20Hz e outro para a frequência $f_a - 20 = 980\text{Hz}$, mas que na realidade corresponde à frequência -20Hz . Se fosse um co-seno o gráfico seria igual (a diferença

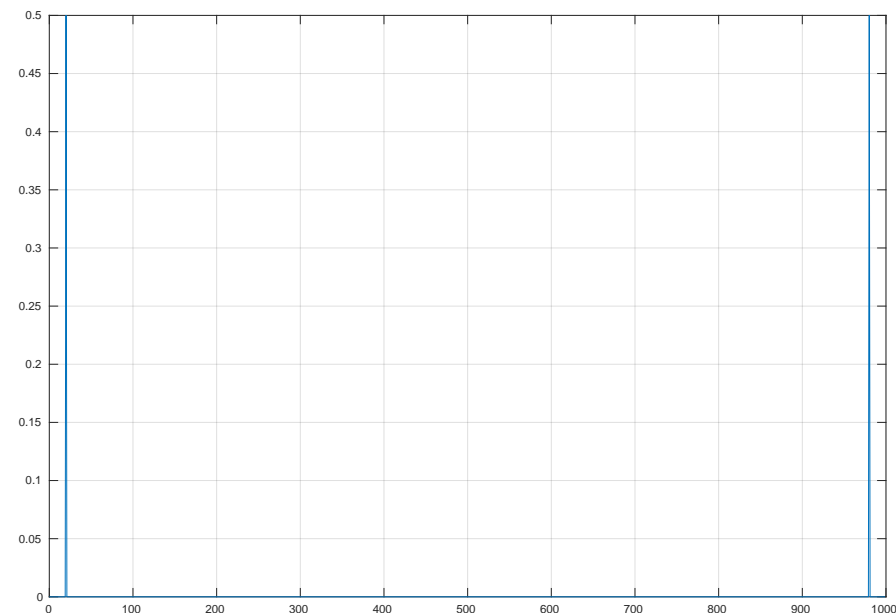


Figura 1: Gráfico do valor absoluto da DFT de um segundo de um seno de 20Hz amostrado a 1000Hz.

entre a DFT do seno e a do co-seno está na fase — lembre-se das fórmulas apresentadas perto do fim da página 2).

Aula do dia 9 de outubro de 2023

Nesta aula foram feitas algumas demonstrações de análise e manipulação de sinais de áudio no MATLAB. Foi explicado detalhadamente o funcionamento dos ficheiros [SM_2023_10_03_explore.m](#) e [SM_2023_10_03_ding.m](#), cujos conteúdos não apresentamos aqui. Foram também apresentados alguns exemplos de filtragem de sinais (comandos `firpm`, `freqz` e `filter` do MATLAB), servindo estes exemplos para motivar os alunos para a teoria de sistemas lineares invariantes no tempo que será o tema da aula seguinte.

Aula do dia 16 de outubro de 2023 (tablet)

Foram tratados dois assuntos: breve menção ao teorema da amostragem ([wikipedia](#)) e teoria de sistemas lineares invariantes no tempo.

A transformada de Fourier pode ser utilizada para estudar como é que a energia de um sinal está espalhada pelas frequências. (A transformada discreta de Fourier, que já estudámos à duas aulas, pode ser considerada uma aproximação da transformada de Fourier.) Como vimos logo na primeira aula, a amostragem do sinal

$$\cos(2\pi f_o t) = \frac{e^{+j2\pi f_o t} + e^{-j2\pi f_o t}}{2},$$

que podemos dizer ser composto por dois sinais, um com frequência $+f_o$ e outro com frequência $-f_o$, dá origem ao sinal

$$\cos\left(2\pi \frac{f_o}{f_a} n\right).$$

Ora este sinal é indistinguível de cada um dos sinais

$$\cos\left(2\pi \frac{f_o}{f_a} n + 2\pi k n\right) = \cos\left(2\pi \frac{f_o + k f_a}{f_a} n\right), \quad k \in \mathbb{Z}.$$

Portanto, em geral, o que se passa para a frequência f_o do sinal não amostrado terá de se refletir nas frequências $f_o + k f_a$ do sinal amostrado, já que não as conseguimos distinguir. Se a transformada de Fourier de $x(t)$, que é dada por

$$X(j2\pi f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi f t} dt$$

(é a transformada bilateral de Laplace avaliada para $s = j2\pi f$) só não for nula para uma dessas frequências, e se soubermos qual é a frequência para a qual não se anula, então, conceptualmente como não há ambiguidades, não deverá haver perda de informação, e o sinal amostrado deverá ter a mesma informação que o sinal não amostrado. E assim

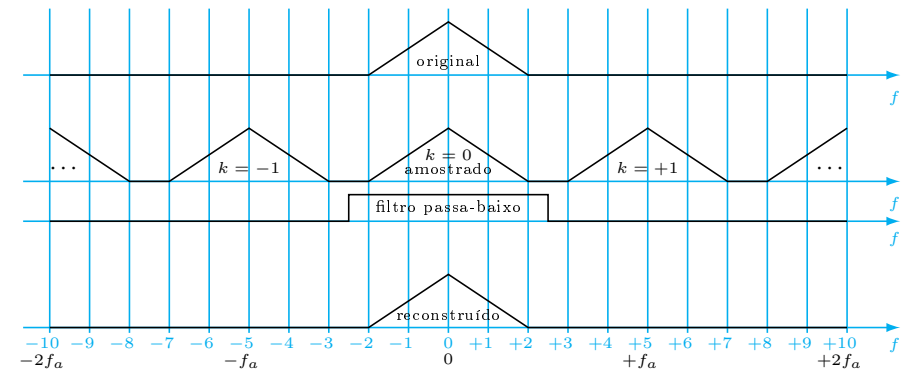


Figura 2: Ilustração do que acontece à transformada de Fourier de um sinal quando este é amostrado e depois reconstruído.

é. Pode-se provar que a transformada de Fourier do sinal amostrado, a menos de um possível fator de escala, é dada por

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

ou seja, é a soma de todas as translações de $X(j2\pi f)$ por múltiplos da frequência de amostragem. Em particular, se $X(j2\pi f)$ for identicamente nula para $|f| \geq f_{\max}$ então essas translações não se sobrepõem se

$$f_{\max} < f_a - f_{\max}$$

(no lado esquerdo temos a frequência máxima possível para $k = 0$ e no lado direito temos a frequência mínima possível para $k = 1$) ou seja, se

$$f_a > 2f_{\max}.$$

Neste caso, como não há sobreposição, uma filtragem passa baixo (multiplicação na frequência por um sinal rectangular) eliminaria todas as translações com $k \neq 0$, recuperando desta maneira o sinal original. A figura 2 ilustra o que acontece num exemplo fictício, no qual $f_{\max} = 2$ e $f_a = 5$.

Apesar de não ter sido lecionado na aula teórica, fica aqui registado como se reconstroi o sinal não amostrado a partir do sinal amostrado

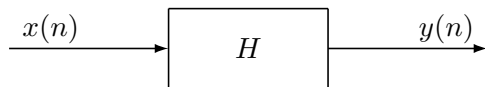


Figura 3: Diagrama de blocos de um sistema.

(série cardinal, em Inglês *cardinal series*):

$$x(t) = \sum_{k=-\infty}^{+\infty} x(kT_a) \operatorname{sinc}\left(\frac{t}{T_a} - k\right),$$

onde (sinc significa *sine cardinal*)

$$\operatorname{sinc}(x) = \begin{cases} 1, & \text{se } x = 0, \\ \frac{\sin(\pi x)}{\pi x}, & \text{se } x \neq 0. \end{cases}$$

Note que quando $t = nT_a$ todos os argumentos das sines do somatório são os números inteiros $n - k$ e nesse caso a sinc é igual a 1 se o número inteiro for 0 (isto é, se $k = n$), e é igual a 0 se não o for (isto é, se $k \neq n$).

Passemos agora para o tema sistemas lineares. Um sistema recebe um ou mais sinais de entrada, manipula-os, e, como resultado dessa manipulação, produz um ou mais sinais de saída. A figura 3 ilustra isso de uma forma conceptual, designada por diagrama de blocos, para um sistema com um sinal de entrada $x(n)$ e um sinal de saída $y(n)$. Como vamos descrever o que se passa quando se trabalha com sinais amostrados, n designa o número da amostra. A transformação de $x(n)$ em $y(n)$ (o sistema) é representada na figura pela letra H dentro de um retângulo. As setas indicam o sentido do fluxo de informação.

[Linearidade.] Para $k = 1, 2, \dots, K$, suponha que se sabe que a resposta de um dado sistema ao sinal $x_k(n)$ é $y_k(n)$. Sendo $\alpha_1, \alpha_2, \dots, \alpha_K$ escalares (números reais), diz-se que um sistema é linear se a sua resposta ao sinal $\sum_{k=1}^K \alpha_k x_k(n)$, que é uma combinação linear dos K sinais de entrada, for $\sum_{k=1}^K \alpha_k y_k(n)$, que é a mesma combinação linear, mas agora dos sinais de saída.

[Invariância temporal.] Seja Δ um número inteiro. Diz-se que um sistema é invariante no tempo se a resposta ao sinal $x_k(n - \Delta)$ for $y_k(n - \Delta)$, ou seja se atrasar/adiantar o sinal de entrada corresponder a atrasar/adiantar pelo mesmo valor o sinal de saída.

[Consequências.] Se um sistema for linear e invariante no tempo é muito simples caracterizar matematicamente a transformação que ele faz ao sinal de entrada para obter o sinal de saída. Para conseguir isso, seja

$$\delta(n) = \begin{cases} 1, & \text{se } n = 0; \\ 0, & \text{se } n \neq 0. \end{cases}$$

A este sinal elementar dá-se o nome de impulso. Munidos deste sinal podemos decompor o sinal de entrada do sistema, $x(n)$, na soma de impulsos

$$x(n) = \sum_{k=-\infty}^{+\infty} x(k) \delta(n - k).$$

Note que nesta decomposição de $x(n)$ numa combinação linear de um número infinito de impulsos os sinais são os $\delta(n - k)$ e os α_k da combinação linear são os valores das amostras $x(k)$. Ignorando para já problemas de convergência por termos uma combinação linear com um número infinito de sinais, pelos factos de o sistema ser linear e de ser invariante no tempo, o sinal de saída será

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k) h(n - k) = \sum_{k=-\infty}^{+\infty} h(k) x(n - k),$$

onde $h(n)$ é a resposta do sistema ao sinal $\delta(n)$. Como $\delta(n)$ é um impulso chama-se a $h(n)$ resposta impulsional. Os matemáticos dão à fórmula anterior o nome de convolução, pelo que o sinal de saída é a convolução do sinal de entrada com a resposta impulsional (ou vice versa, já que a convolução é comutativa).

[Estabilidade.] Se o sinal de entrada for limitado em amplitude, isto é, se $|x(n)| < A$ para todos os valores de n , e se $\sum_{k=-\infty}^{+\infty} |h(k)|$ existir e não for superior a B , é muito simples demonstrar que o sinal de saída

$y(n)$ também será limitado em amplitude, e teremos $|y(n)| < AB$ para todos os valores de n . Diz-se neste caso que o sistema é estável.

[Resposta em frequência.] Quando um sinal periódico passa por um sistema linear e invariante no tempo, o que é que lhe acontece? Para responder a esta pergunta vamos ver o que acontece para o sinal elementar $x(n) = e^{j2\pi f_o n}$. A resposta do sistema a este sinal de entrada é

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{+\infty} h(k) e^{j2\pi f_o(n-k)} \\ &= \sum_{k=-\infty}^{+\infty} h(k) e^{j2\pi f_o n} e^{-j2\pi f_o k} \\ &= \underbrace{e^{j2\pi f_o n}}_{x(n)} \underbrace{\sum_{k=-\infty}^{+\infty} h(k) e^{-j2\pi f_o k}}_{\text{resposta em frequência}} \end{aligned}$$

que é um sinal bem definido se o sistema for estável. Note que o sinal de saída é o sinal de entrada multiplicado pela resposta em frequência, que é a função

$$H(e^{j2\pi f_o}) = \sum_{k=-\infty}^{+\infty} h(k) e^{-j2\pi f_o k}$$

(veremos na próxima aula por que é que se usa esta notação para o argumento da função). Resumindo, quando um sinal de entrada tem apenas uma única frequência (no caso anterior, f_o), o sinal de saída é um sinal com essa mesma frequência, mas a sua amplitude e fase são alteradas pelo fator $H(e^{j2\pi f_o})$. Se o módulo de $H(e^{j2\pi f_o})$ for menor do que um o sinal é atenuado, e se for maior do que um é amplificado. A fase de $H(e^{j2\pi f_o})$, se for diferente de zero, provoca um desalinhamento temporal entre o sinal de entrada e o de saída.

A frequência f_o usada acima é uma frequência normalizada (deverá estar entre $-\frac{1}{2}$ e $+\frac{1}{2}$). Deve ser substituída por f_o/f_a se a quisermos expressar em Hertz (reveja a este respeito o que foi escrito acerca da

amostragem de um seno logo na nossa primeira aula; reveja também o início da aula do dia 16 de outubro).

[Exemplo.] Considere um sistema que calcula o seu sinal de saída usando a fórmula

$$y(n) = x(n) + 2x(n-1) + x(n-2).$$

Para este sistema temos $h(0) = h(2) = 1$, $h(1) = 2$, e todos os outros valores de $h(k)$ são zero. A resposta em frequência deste sistema é

$$\begin{aligned} H(e^{j2\pi f_o}) &= 1 + 2e^{-j2\pi f_o} + e^{-j4\pi f_o} \\ &= e^{-j2\pi f_o} (e^{+j2\pi f_o} + 2 + e^{-j2\pi f_o}) \\ &= e^{-j2\pi f_o} (2 + 2\cos(2\pi f_o)) \end{aligned}$$

Para este sistema temos $|H(e^{j2\pi f_o})| = 2 + 2\cos(2\pi f_o)$. Para frequências baixas — $f_o \approx 0$ — temos $|H(e^{j2\pi f_o})| \approx 4$, ou seja as frequências baixas do sinal de entrada são amplificadas por um fator de aproximadamente quatro. Para frequências próximas de metade da frequência de amostragem — $f_o \approx \frac{1}{2}$ — temos $|H(e^{j2\pi f_o})| \approx 0$, ou seja, essas frequências são fortemente atenuadas, pelo que praticamente desaparecem no sinal de saída. Trata-se pois de um sistema passa-baixo, que deixa passar as baixas frequências, mas que não deixa passar as altas frequências.

Aula do dia 23 de outubro de 2023 ([tablet](#))

No fim da aula anterior introduzimos a transformada Z, cujo estudo foi terminado nesta aula. Apresentamos aqui o resumo do que foi feito nas duas aulas.

A transformada Z do sinal $x(n)$ é designada por $X(z)$ e é definida por

$$x(n) \quad \leftrightarrow \quad X(z) = \sum_{n=-\infty}^{+\infty} x(n) z^{-n}.$$

É importante mencionar que z é uma variável complexa e que a fórmula para $X(z)$ dada acima poderá apenas convergir numa região anular, do tipo $r < |z| < R$, do plano complexo.

Como consequência deste definição temos imediatamente as seguintes propriedades:

linearidade

$$\sum_{k=1}^K \alpha_k x_k(n) \quad \leftrightarrow \quad \sum_{k=1}^K \alpha_k X_k(z),$$

efeito de um atraso ou avanço

$$x(n - \Delta) \quad \leftrightarrow \quad z^{-\Delta} X(z), \quad \Delta \in \mathbb{Z},$$

impulso

$$\delta(n) \quad \leftrightarrow \quad 1, \quad z \in \mathbb{C},$$

exponencial (esta por lapso não foi mencionada na aula)

$$\begin{cases} a^n, & n \geq 0, \\ 0, & n < 0, \end{cases} \quad \leftrightarrow \quad \frac{1}{1 - az^{-1}} = \frac{z}{z - a}, \quad |z| > |a|.$$

Como explicado na aula anterior, se um sistema for linear, invariante no tempo (e estável), o seu sinal de saída é a convolução do seu sinal de entrada com a sua resposta impulsional. Aplicando transformadas Z à fórmula da convolução obtem-se

convolução (em baixo, temos $m = n - k$)

$$\sum_{k=-\infty}^{+\infty} h(k)x(n-k) \quad \leftrightarrow \quad \sum_{n=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} h(k)x(n-k)z^{-(n-k)-k}$$

$$\underbrace{\sum_{k=-\infty}^{+\infty} h(k)z^{-k}}_{H(z)} \underbrace{\sum_{m=-\infty}^{+\infty} x(m)z^{-m}}_{X(z)}$$

ou seja, a transformada da convolução de dois sinais é o produto das respetivas transformadas.

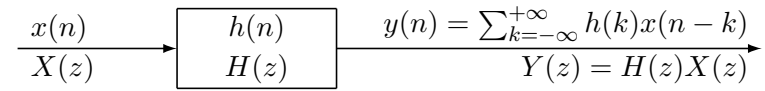


Figura 4: Efeito de um sistema linear e invariante no tempo (no domínio dos tempos em cima, no da transformada Z em baixo). Note que para sistemas causais, que não respondem antes de se aplicar um estímulo, temos $h(k) = 0$ para $k < 0$.

Como consequência desta propriedade, para um sistema linear e invariante no tempo, a razão entre a transformada Z do seu sinal de saída e a transformada Z do seu sinal de entrada é fixa (é a transformada Z da sua resposta impulsional). A essa razão chama-se função de transferência. A figura 4 resume o que acontece a um sinal quando este passa por um sistema linear e invariante no tempo.

É interessante comparar a fórmula apresentada na aula anterior para a resposta em frequência de um sistema com a fórmula para a sua função de transferência, que é a transformada Z da sua resposta impulsional. Constata-se que a resposta em frequência para uma frequência normalizada de f_o é, nem mais nem menos, que a função de transferência avaliada para $z = e^{j2\pi f_o}$. Em particular, para $f_o = 0$, correspondente à frequência não normalizada de 0 Hz, temos $z = 1$, e para $f_o = 1/2$, correspondente à frequência não normalizada de $f_a/2$ Hz (f_a é a frequência de amostragem), temos $z = -1$. Para um quarto da frequência de amostragem temos $z = j$.

Como exemplo, considere que a equação que descreve como se calcula o sinal de saída de um dado sistema é

$$y(n) = x(n) + 2x(n-1) + x(n-2).$$

Pretende-se calcular a função de transferência e esboçar a resposta em frequência desse sistema. Aplicando transformadas Z obtemos

$$Y(z) = X(z) + 2z^{-1}X(z) + z^{-2}X(z)$$

pelo que a função de transferência pretendida é

$$H(z) = \frac{Y(z)}{X(z)} = 1 + 2z^{-1} + z^{-2}.$$

Para esboçar a resposta em frequência é mais simples usar $\omega = 2\pi f$ (deixamos cair o ω), porque as fórmulas ficam com menos tralha. Fazendo $z = e^{j\omega}$ obtemos

$$H(e^{j\omega}) = 1 + 2e^{-j\omega} + e^{-j2\omega} = e^{-j\omega}(e^{j\omega} + 2 + e^{-j\omega}) = e^{-j\omega}(2 + 2\cos\omega).$$

Verifica-se assim que $|H(e^{j\omega})| = 2 + 2\cos\omega$. Com base nesta fórmula é depois fácil esboçar a resposta em frequência deste sistema (faça-o, é tipo passa-baixo porque atenua as frequências altas).

Se o sistema fosse descrito pela equação

$$y(n) = x(n-1) - 0.9y(n-2)$$

teríamos

$$Y(z) = z^{-1}X(z) - 0.9z^{-2}Y(z),$$

pelo que a função de transferência seria

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z^{-1}}{1 - 0.9z^{-2}}$$

Note que neste caso por no lado direito da equação aparecer o próprio sinal de saída (atrasado) a função de transferência para a ter um denominador.

Para quem quer saber mais

Se todos os zeros do denominador tiverem módulo menor do que um, condição necessária para que um sistema causal seja estável, a sua resposta impulsional decairá exponencialmente para zero ao ritmo de r^n , sendo r o maior raio dos zeros do denominador.

Os zeros do numerador são chamados zeros (da função de transferência) do sistema, e os zeros do denominador são chamados pólos do sistema.

É possível representar graficamente funções de transferência usando diagramas de blocos. Ilustramos isso na figura 5, que apresenta diagramas de blocos dos sistemas

$$H_1(z) = 1 + 0.5z^{-10000}$$

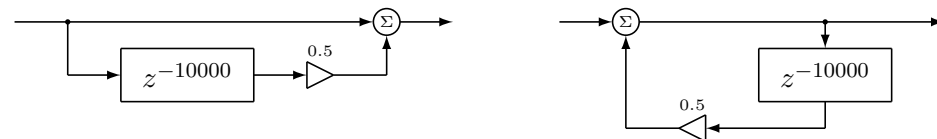


Figura 5: Diagramas de blocos dos sistemas $H_1(z)$, à esquerda, e $H_2(z)$, à direita. Um triângulo representa um ganho (multiplicação por uma constante).

(que acrescenta um eco a um som, com um atraso de 10000 amostras e com uma amplitude que é metade da do sinal original) e

$$H_2(z) = \frac{1}{1 - 0.5z^{-10000}} = \sum_{k=0}^{+\infty} (0.5)^k z^{-10000k}$$

(que acrescenta um número infinito de ecos, cada um atrasado 10000 amostras em relação ao anterior, e com metade da amplitude).

Aula do dia 30 de outubro de 2023 (tablet)

Nesta aula foram descritas superficialmente as transformadas de cossenos e de senos, e foi descrito como se podem aplicar todas as transformadas apresentadas até agora a sinais bidimensionais.

[**Outras transformadas**] Seja N um número inteiro positivo, seja

$$\mathbf{x} = [x(0) \ x(1) \ \dots \ x(N-1)]^T$$

um vetor coluna com N elementos (T designa transposta de um vetor ou matriz), e, para $m = 0, 1, \dots, N-1$, sejam

$$\mathbf{x}_m = [e^{j2\pi \frac{m}{N}0} \ e^{j2\pi \frac{m}{N}1} \ \dots \ e^{j2\pi \frac{m}{N}(N-1)}]^T.$$

Com esta notação, a fórmula (2) da transformada discreta inversa de Fourier pode ser escrita na forma

$$\mathbf{x} = \sum_{m=0}^{N-1} X(m)\mathbf{x}_m,$$

ou seja, o sinal cujas amostras estão guardadas em \mathbf{x} é uma combinação linear dos N sinais elementares \mathbf{x}_m . A transformada discreta de Fourier pode pois ser considerada como uma maneira de determinar os coeficientes dessa combinação linear. Como cada um desses coeficientes está associado a um sinal com uma frequência bem determinada, a DFT é muito útil para analisar o conteúdo espectral 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 \mathbf{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 de $X(m)$ são pequenos. Substituindo esses valores por zero ou por versões quantizadas com muitos poucos bits permite aumentar a taxa de compressão. Isso é em geral conseguido nas transformadas de co-senos e nas de senos, que em vez das exponenciais complexas da DFT usam-se ou apenas co-senos ou apenas senos (mas começando com apenas meio período em vez de um período completo como da DFT).

Das várias transformada possíveis, apresentamos, meramente a título ilustrativo, uma delas (DCT-II):

$$X(m) = \sum_{n=0}^{N-1} x(n) \cos \left(\frac{\pi}{N} \frac{2n+1}{2} m \right).$$

(Note que é π/N e não $2\pi/N$ e que é $(n+1/2)m$ e não nm com é na DFT.) A respetiva transformada inversa (que é a DCT-III com um fator de escala diferente) é dada por

$$x(n) = \frac{1}{N} X(0) + \frac{2}{N} \sum_{m=1}^{N-1} X(m) \cos \left(\frac{\pi}{N} \frac{2n+1}{2} m \right)$$

A figura 6 mostra os primeiros sinais \mathbf{x}_m , tanto para a DFT como para a DCT-II ([código MATLAB usado para fazer a figura](#)).

Note-se que na DFT assume-se que $x(n)$ é um sinal periódico de período N , o mesmo acontecendo a $X(m)$. Nas transformadas de co-senos

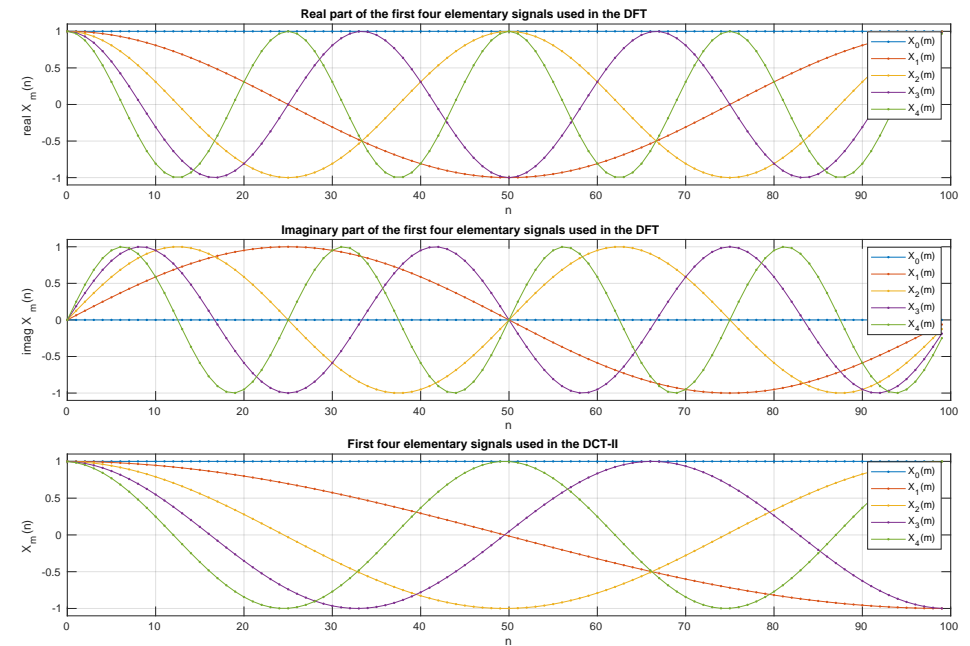


Figura 6: Os primeiros sinais elementares usados na DFT (em cima e no meio) e na DCT-II (em baixo).

e nas de senos, continua a existir periodicidade em $x(n)$, mas agora o período passa a ser $2N$; é pressuposto que as amostras de $x(n)$ entre N e $2N$ sejam imagens de espelho do que se passa entre 0 e N . Como existem várias maneiras de especificar essas imagens de espelho, existem várias transformadas.

[Transformadas bidimensionais] Consideremos agora um sinal bidimensional (por exemplo uma imagem) $x(n_1, n_2)$, amostrado numa grelha retangular, com N_1 amostras na primeira dimensão (por exemplo n_1 representa uma coordenada no eixo dos xx 's) e N_2 amostras na segunda dimensão (por exemplo n_2 representa uma coordenada no eixo dos yy 's). Por analogia com o que acontece no caso unidimensional, podemos definir a transformada didimensional discreta de Fourier (DFT-2D), da

seguinte maneira:

$$X(m_1, m_2) = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) e^{-j2\pi \frac{n_1 m_1}{N_1}} e^{-j2\pi \frac{n_2 m_2}{N_2}}$$

Como vamos ver já de seguida, esta definição corresponde a fazer primeiro DFTs-1D em ordem a uma das variáveis, por exemplo n_1 , obtendo-se

$$X_1(m_1, n_2) = \frac{1}{N_1} \sum_{n_1=0}^{N_1-1} x(n_1, n_2) e^{-j2\pi \frac{n_1 m_1}{N_1}}, \quad n_2 = 0, 1, \dots, N_2 - 1,$$

e depois fazer DFTs-1D em ordem à outra das variáveis, neste caso n_2 , obtendo-se

$$X(m_1, m_2) = \frac{1}{N_2} \sum_{n_2=0}^{N_2-1} X_1(m_1, n_2) e^{-j2\pi \frac{n_2 m_2}{N_2}}, \quad m_1 = 0, 1, \dots, N_1 - 1.$$

Do ponto de vista computacional, esta maneira de efetuar os cálculos reduz bastante o número de multiplicações e somas que é necessário efetuar. Na aplicação direta da fórmula da DFT-2D, temos de avaliar $N_1 N_2$ pares de valores de (m_1, m_2) , sendo que o somatório duplo precisa de $N_1 N_2$ multiplicações. Logo, no total $(N_1 N_2)^2$ multiplicações. Fazendo os cálculos de forma fazeada, avaliamos primeiro N_2 DFTs-1D de N_1 pontos ($N_2 N_1^2$ multiplicações usando diretamente a fórmula), e depois N_1 DFTs-1D de N_2 pontos ($N_1 N_2^2$ multiplicações), para um total de apenas $N_1 N_2 (N_1 + N_2)$ multiplicações. Para $N_1 = N_2 = N$ com este simples expediente passa-se de uma complexidade computacional de $\mathcal{O}(N^4)$ para $\mathcal{O}(N^3)$.

Para as transformadas de co-senos e para as de senos, a maneira de proceder é a mesma. Para um array rectangular, fazem-se primeiro transformadas unidimensionais linha a linha, guardando os resultados nas mesmas linhas, e depois fazem-se transformadas unidimensionais coluna a coluna, guardando os resultados nas mesmas colunas. (É claro que a ordem pode ser a inversa; primeiro por colunas e depois por linhas, tanto faz).

Para terminar, por exemplo na compressão de imagens usando o formato JPEG, a imagem é subdividida em blocos independentes de 8x8 pixels, a cada bloco é aplicada uma transformada bidimensional de co-senos, sendo depois os coeficientes quantizados (de acordo com a qualidade de compressão pretendida) e finalmente comprimidos e armazenados. Em aulas futuras vamos olhar para os processos de quantização e de compressão.

Aula do dia 6 de novembro de 2023 ([tablet](#))

Nesta aula o tema foi representações tempo frequência.

Começamos por descrever a transformada “tempo-curto” de Fourier (*Short-Time Fourier Transform*). A ideia é usar uma função bem localizada no tempo, designada habitualmente por janela (temporal), por exemplo

$$w(t) = \begin{cases} 1, & \text{se } |t| < T, \\ 0, & \text{se } |t| \geq T, \end{cases}$$

ou

$$w(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2}$$

para “localizar” uma função $x(t)$ em torno do instante de tempo t_0 multiplicando-a por $w(t - t_0)$. A transformada “tempo-curto” de Fourier é a transformada de Fourier dessa localização:

$$X(j2\pi f, t_0) = \int_{-\infty}^{+\infty} x(t) w(t - t_0) e^{-j2\pi f t} dt$$

Infelizmente, quando mais localizada está uma função no tempo, mais dispersa está na frequência, e vice versa. (A janela Gaussiana apresentada acima é a que está mais bem localizada ao mesmo tempo no tempo e na frequência.)

Podemos também fazer o mesmo para sinais amostrados. Na prática, subdivide-se o sinal que se pretende analisar em segmentos (com sobreposição ou não), e, para cada um deles aplica-se a janela e calcula-se

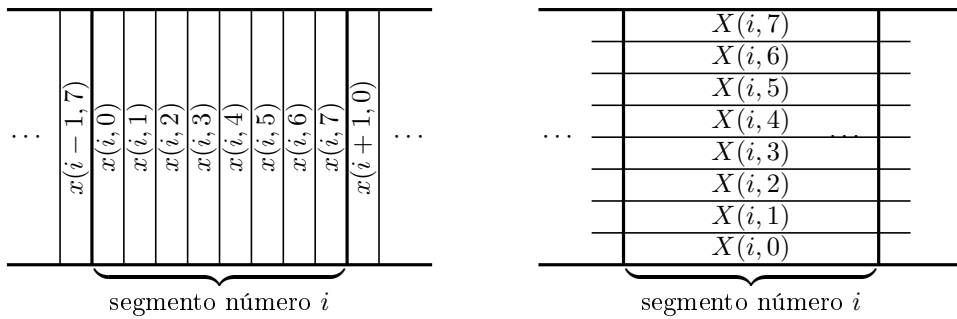


Figura 7: Representação tempo frequência usando a DFT (tempo no eixo horizontal, frequências no vertical); i é o número do segmento. Nesta ilustração cada segmento contém 8 amostras e não há sobreposição de segmentos; no lado esquerdo está só a informação no tempo, no lado direito está no tempo (por segmentos) e na frequência (dentro de cada segmento).

depois uma DFT (transformada discreta de Fourier). Para cada segmento passe-se então das N amostras $x(n)$ temporais, que não têm nenhuma localização na frequência, para N amostras $X(m)$, que não têm nenhuma localização no tempo dentro do segmento que está a ser analisado. Com segmentos de 8 amostras sem sobreposição, passa-se assim da situação do lado esquerdo da figura 7 para a situação do lado direito.

O MATLAB tem uma função, `spectrogram`, que permite automatizar estes cálculos, e que apresenta o resultado na forma de uma imagem. Cores quentes, como o vermelho e o amarelo representam valores de $|X(i, m)|$ grandes, e cores frias, como o azul escuro, representam valores baixos. O código seguinte (ficheiro [SM_2023_11_06.m](#)) ilustra a sua utilização:

```
clear all; close all; clc;
% https://www.101soundboards.com/boards/
% 10634-r2-d2-sounds-star-wars
[x,fs]=audioread('22.mp3');
sound(x,fs);
fig=figure(1);
subplot('Position',[0.10 0.10 0.85 0.85]);
```

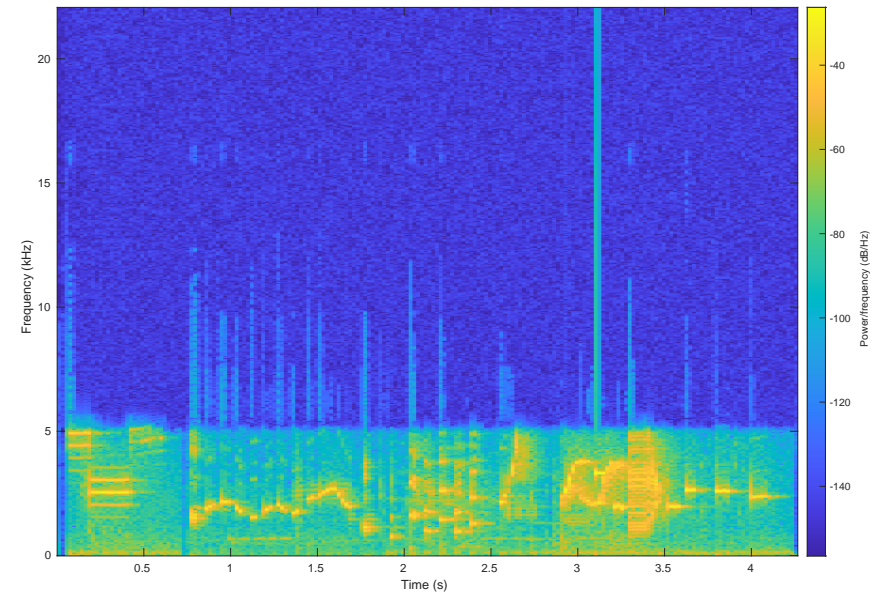


Figura 8: Análise tempo-frequência de um som.

```
spectrogram(x,hanning(1024),64,1024,fs,'yaxis');
fig.PaperType='A4';
fig.PaperOrientation='landscape';
fig.PaperUnits='points';
print('spectrogram.pdf','-dpdf','-noui','-fillpage');
```

A figura 8 mostra o aspeto do resultados da utilização deste comando para analisar um som produzido pela R2D2.

Aula do dia 13 de novembro de 2023 ([tablet](#))

Nesta aula o tema principal foi a quantização de sinais unidimensionais, com breve referência ao que pode ser feito para sinais bidimensionais.

[Quantização] No processo de amostragem transforma-se uma função real de uma variável real numa função real de variável inteira. No

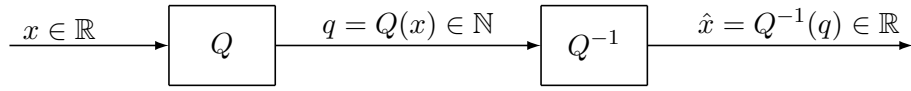


Figura 9: Quantização, $x \rightarrow q = Q(x)$, e reconstrução, $q \rightarrow \hat{x} = Q^{-1}(q)$; x e \hat{x} são números reais, e q é um número inteiro. Q é a função inversa de Q^{-1} , ou seja $q = Q(Q^{-1}(q))$. Em geral, pretende-se reduzir o mais possível o erro de quantização, dado por $e = \hat{x} - x = Q^{-1}(Q(x)) - x$. Note que só temos $x = Q^{-1}(Q(x))$, ou seja, um erro de quantização nulo, quando x é um dos níveis de quantização. Na prática, o número de níveis de quantização é finito.

caso da amostragem uniforme, faz-se $t = nT_a$, onde T_a é o período de amostragem e n é o número da amostra. Isto já foi estudado em aulas anteriores. Mas, para guardar um bocado de um sinal num computador isso não é suficiente. É preciso também discretizar os valores da própria função, ou seja o resultado final da amostragem mais quantização tem de ser uma função inteira de variável inteira. Cada valor inteiro da função representa um nível do processo de quantização (ver figura 10). Para reconstruir o sinal, obtendo-se em geral uma aproximação ao sinal original, cada valor quantizado (um número inteiro) tem de ser convertido no nível correspondente. A figura 9 ilustra conceptualmente todo este processo.

Sejam $x(n)$ as amostas do sinal que pretendemos quantizar. Considere que se pretendem N níveis de quantização, isto é, que $x(n)$ vai ser substituído pelo número inteiro $q(n) = Q(x(n))$, que apenas pode assumir N valores distintos (por exemplo, de 1 a N). O sinal reconstruído será $\hat{x}(n) = Q^{-1}(q(n))$. Existem várias maneiras de otimizar o processo de quantização. O objetivo é geralmente o de minimizar o erro de quantização $e(n) = \hat{x}(n) - x(n)$. Em particular, é habitual tentar minimizar a distorção quadrática média (erro quadrático médio)

$$D = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^{+N} e^2(n)$$

já que isso é matematicamente fácil de fazer. Usando a função densidade de probabilidade $f_X(x)$ para modelar a distribuição dos valores de $x(n)$, e pressupondo que se modela $x(n)$, para os vários valores de n ,

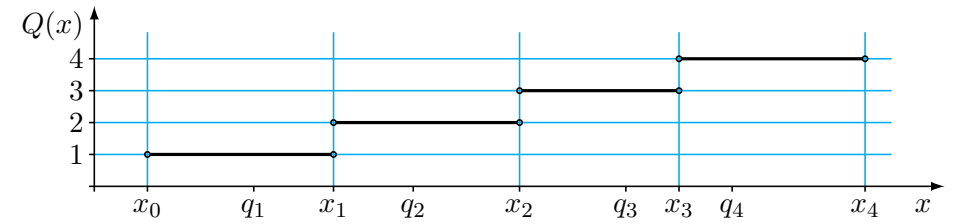


Figura 10: Forma genérica de uma função de quantização (para $N = 4$). Em geral, x_0 pode ser $-\infty$ e x_N pode ser $+\infty$. Nos pontos de descontinuidade de $Q(x)$, tanto faz escolher o valor da função à esquerda, como o valor à direita.

por variáveis aleatórias independentes³ todas com função densidade de probabilidade $f_X(x)$, o valor médio da distorção quadrática será dado por

$$E[e^2(n)] = \int [Q^{-1}(Q(x)) - x]^2 f_X(x) dx.$$

Sejam q_1, q_2, \dots, q_N os níveis de quantização, diferentes uns dos outros e ordenados por ordem crescente. A função $Q(x)$ poderá ter apenas um desses valores. Faz todo o sentido que essa função seja constante por secções, isto é, que o seu gráfico seja tipo escada, como se ilustra na figura 10.

Quando se tem N níveis de quantização, a distorção quadrática média pode ser posta na forma

$$E[e^2(n)] = \sum_{n=1}^N \int_{x_{n-1}}^{x_n} (q_n - x)^2 f_X(x) dx.$$

Para um dado valor de x verifica-se que $(q_k - x)^2$ toma o menor valor possível para o valor de q_k mais próximo de x . Isso implica que para minimizar a distorção média os limites x_1, x_2, \dots, x_{N-1} dos intervalos dentro dos quais $Q(x)$ é constante devem satisfazer

$$x_n = \frac{q_n + q_{n+1}}{2}, \quad n = 1, 2, \dots, N-1. \quad (3)$$

³Isto é irrealista mas torna o problema da quantização mais simples. Veja mais adiante o muito breve resumo sobre quantização vetorial.

[Quantização uniforme] Na quantização uniforme a função de quantização é da forma⁴

$$Q(x) = \text{round}\left(\frac{x}{\Delta}\right) = \left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor$$

onde Δ é o passo de quantização; a notação $\lfloor \cdot \rfloor$ designa a função floor. A respetiva função inversa será simplesmente

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

pelo que temos

$$\hat{x} = \Delta \text{round}\left(\frac{x}{\Delta}\right).$$

A figura 11 ilustra todo este processo (código MATLAB usado para fazer a figura).

Sejam $x(n)$ as amostras de um sinal de amplitude A , isto é, temos $|x(n)| \leq A$ para todos os valores de n . Se tivermos N níveis de quantização deveremos usar $\Delta \approx \frac{2A}{N}$. Considerando que o erro de quantização está uniformemente distribuído entre $-\frac{\Delta}{2}$ e $+\frac{\Delta}{2}$, o erro quadrático médio da quantização (a distorção média) será

$$D = \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{+\frac{\Delta}{2}} e^2 de = \frac{e^3}{3\Delta} \Big|_{-\frac{\Delta}{2}}^{+\frac{\Delta}{2}} = \frac{\Delta^2}{12} = \frac{A^2}{3N^2}.$$

Se os valores forem quantizados com n bits⁵, teremos $N = 2^b$ (ou eventualmente, $N = 2^b - 1$, mas esse detalhe não é muito relevante para o que se pretende). O erro quadrático médio de quantização, em dB, será

$$\begin{aligned} 10 \log_{10} E &= 20 \log_{10} A - 10 \log_{10} 3 - 20 \log_{10} N \\ &= 20 \log_{10} A - 10 \log_{10} 3 - 20b \log_{10} 2 \\ &= 20 \log_{10} A - 4.77121 - 6.0206 b. \end{aligned}$$

Verifica-se assim que o erro quadrático médio de quantização diminui em cerca de 6 dB por cada bit extra de quantização.

⁴Existem outras possibilidades, mas a aqui apresentada é a mais simples.

⁵Neste caso, a representação dos valores quantizados deverá ser feita em complemento para 2, já que $Q(x)$ pode ser um número inteiro negativo.

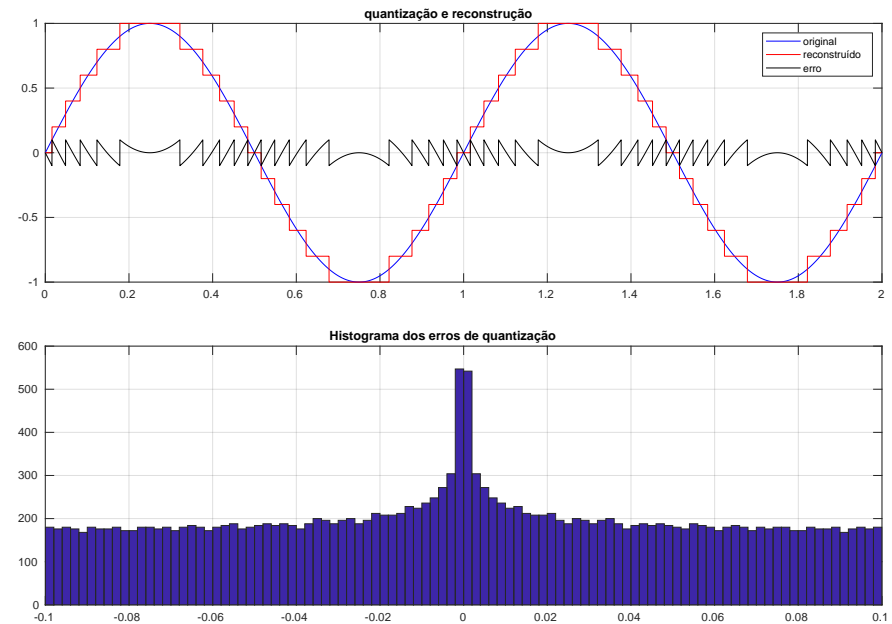


Figura 11: Quantização uniforme de uma senoide usando $\Delta = 0.2$. Note que o erro de quantização apenas se desvia significativamente da distribuição uniforme para erros próximos de zero.

[Quantização não-uniforme (algoritmo de Llyod-Max)] Como o nome sugere, na quantização não uniforme os níveis de quantização não estão igualmente espaçados uns dos outros. Isso pode ser vantajoso se $f_X(x)$ se desviar muito da distribuição uniforme, já que nesse caso a distorção quadrática média da quantização uniforme estará longe da menor distorção possível.

Em 1957, Lloyd inventou um algoritmo para, dado $f_X(x)$ e o número N de níveis de quantização, calcular os níveis ótimos de quantização. Em 1960, Max reinventou o mesmo algoritmo (Lloyd não o tinha publicado), pelo que esse algoritmo é hoje conhecido como algoritmo de Llyod-Max. A ideia do algoritmo é muito simples:

1. Começa-se com escolher, de forma arbitrária, os N níveis de quantização (todos diferentes e ordenados por ordem crescente).

2. Depois usa-se a equação (3) para ajustar os pontos de descontinuidade (limiares de decisão) da função $Q(x)$. Os limites exteriores, x_0 e x_N são fixos e dependem da região de suporte de $f_X(x)$. Se a função densidade de probabilidade for simétrica, deve-se aqui garantir que os valores de x_k também o são. Isso evita problemas de convergência do algoritmo relacionados com erros numéricos ao efetuar os cálculos.
3. Para caso intervalo $]x_{k-1}, x_k[$, com $k = 1, 2, \dots, N$, calcula-se o nível de quantização que minimiza a distorção quadrática quando x está dentro desse intervalo. A minimização de

$$\int_{x_{k-1}}^{x_k} (x - q_k)^2 f_X(x) dx$$

em ordem a q_k dá

$$q_k = \int_{x_{k-1}}^{x_k} x f_X(x) dx,$$

ou seja, q_k é o valor médio de x dentro do intervalo em causa. Se estes novos valores dos q_k não diferirem muito dos anteriores, considera-se que o algoritmo convergiu pelo que este é terminado. Caso contrário, volta-se atrás para o passo número 2, ou seja, repetem-se os passos 2 e 3 até o algoritmo convergir.

O ficheiro [SM_2023_11_13_lloyd_max.m](#) implementa este algoritmo e mostra-o em acção.

[Quantização vetorial] O algoritmo de Lloyd-Max pode ser generalizado para sinais multidimensionais. A ideia é a mesma. Escolhe-se um conjunto de N pontos do 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 (o 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 chame-se diagrama de Voronoi, ilustrado na figura 12 ([código MATLAB usado para fazer a figura](#)).

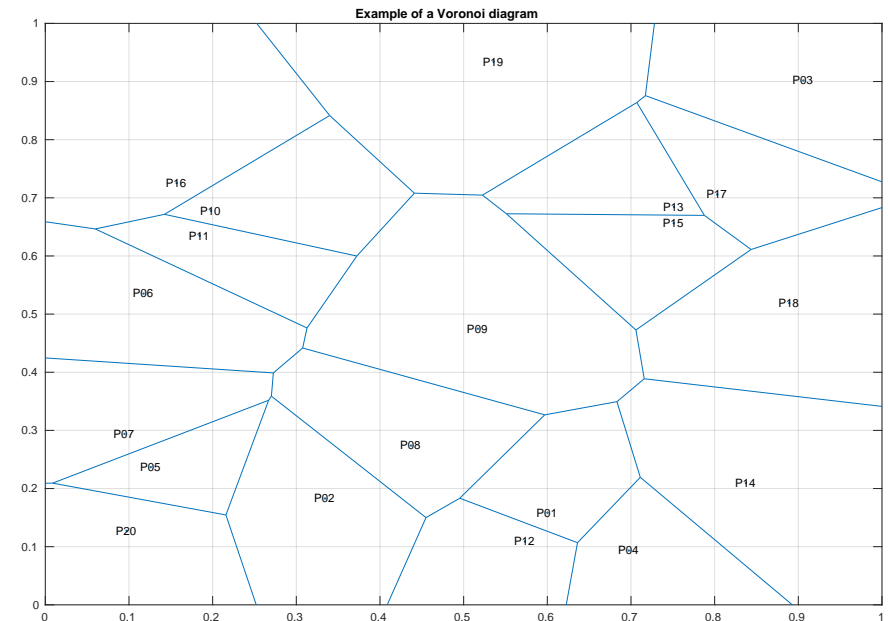


Figura 12: Regiões de Voronoi.

[Espaços de cores] A quantização de cores numa imagem ou filme é um caso onde a quantização vetorial pode ser útil. Considere, por exemplo, o caso do formato de compressão gif (para imagens). Neste formato, a cada pixel da imagem está associado apenas um número inteiro. Esse número inteiro é um índice para uma tabela (a paleta de cores), onde está armazenada a informação completa da cor a usar. No máximo, neste formato apenas podemos ter 256 cores. Para converter uma imagem para este formato, é pois necessário escolher as 256 cores a usar. A quantização vetorial pode ser usada para fazer isso.

Nos espaços de cores aditivos, usados quando se está a emitir diretamente uma imagem (ecrã ou projetor, por exemplo), é habitual especificar uma cor como sendo um vetor com três componente: R (de *red*), G (de *green*) e B (de *blue*). Zero corresponde à intensidade mínima e um à máxima. É habitual quantizar cada uma destas três componentes com 8 bits, e nesse caso é habitual de tanto R, como G e como B tenham uma

gama de values de 0 a 255. Um outro espaço de cores bastante usado é o HSV, no qual cada cor é (também) um vetor com três compomentes: H (de *hue*), S (de *saturation*) e V (de *value*). Variando apenas o H obtem-se uma espécie de arco-íris. Quanto mais pequeno for S mais próximo de um tom de cinzento é a cor. Quanto maior for V mais brilhante é a cor. O ficheiro [SM_2023_11_13_hsv_example.m](#) ilustra isso para S fixo e igual a um:

```
for v=0.1:0.1:1
    for a=0:10:359
        [r,g,b]=hsv2rgb([a/360,1.0,v])
        plot(v*cos(pi*a/180),v*sin(pi*a/180),...
            'o','MarkerSize',3+7*v,...
            'MarkerFaceColor',[r,g,b],...
            'MarkerEdgeColor',[r,g,b]);
    hold on;
end
end
hold off;
grid on;
axis([-1.1,1.1,-1.1,1.1]);
axis('equal');
```

Para quem quer saber mais

As seguintes páginas da wikipédia contêm informação adicional:

- [Espaços de cores.](#)
- [HSV.](#)
- [Amostragem de imagens.](#)

Aula do dia 20 de novembro de 2023 ([tablet](#))

Nesta aula começou-se a explicar como se pode comprimir informação.

[O conceito de alfabeto] A um item indivisível de informação é dado o nome de símbolo. O alfabeto é o conjunto de todos os símbolos.

Por exemplo, um alfabeto pode ser o conjunto de todas as letras. Um outro alfabeto, bem maior, pode ser o conjunto de todas as palavras (cada palavra, cada símbolo). No caso da quantização, o alfabeto pode ser o conjunto de todos os níveis de quantização (ou melhor, a sua representação por números inteiros). O nosso objetivo nesta aula é o de comprimir uma mensagem, que é uma sequência de símbolos (do mesmo alfabeto).

[Entropia] Considere uma variável aleatória que apenas pode assumir N valores distintos, digamos $1, 2, \dots, N$. Considere ainda que a probabilidade de ocorrer o valor k é p_k . A entropia dessa variável aleatória é dada por

$$H = - \sum_{k=1}^N p_k \log_2 p_k$$

e mede-se em bits. O seu valor máximo ocorre quando as várias probabilidades são todas iguais, isto é, quando $p_k = \frac{1}{N}$. Nesse caso, se $N = 2^b$, a entropia da variável aleatória será b , que é o número de bits necessários para distinguir cada um dos valores possíveis da variável aleatória. Quando alguns valores são mais pequenos que outros, a entropia será menor do que $\log_2 N$, e é isso que torna possível comprimir informação.

Considere um alfabeto com N símbolos, que vamos designar por S_1, S_2, \dots, S_N . Considere também o símbolo S_k ocorre com uma probabilidade p_k . Finalmente, considere que uma mensagem é uma sequência destes símbolos, e que a probabilidade de ocorrência de um símbolo é independente dos símbolos que o precedem, isto é, que os vários símbolos que compõem a mensagem podem ser considerados como variáveis aleatórias independentes, todas com a mesma distribuição. Nestas condições, em teoria, será possível comprimir a mensagem usando um código que gasta, em média, H bits por símbolo. Esse valor não pode ser melhorado, isto é, não existem códigos que conseguem comprimir usando em média menos do que H bits por símbolo. Como vamos ver a seguir, os códigos de Huffman comprimem a mensagem usando entre H e $H + 1$ bits por símbolo, ou seja, nunca podem diferir do melhor código possível em mais do que um bit por símbolo.

[**Códigos de Huffman**] Considere que o símbolo S_k é codificado (em binário) pela sequência de bits B_k . Se B_k tiver L_k bits, o número médio de bits por símbolo será

$$L = \sum_{k=1}^N L_k p_k,$$

Mais uma vez, estamos a considerar que existem N símbolos, e que a probabilidade de S_k é p_k . Nos códigos de Huffman, que vamos descrever a seguir, temos, como já mencionada atrás,

$$H \leq L < H + 1,$$

É fundamental que as sequências de bits B_k não criem ambiguidades. Por exemplo, se tivermos três símbolos com códigos

$$B_1 = 0, \quad B_2 = 00 \quad \text{e} \quad B_3 = 1$$

Não será possível distinguir a ocorrência de $S_1 S_1$ da de S_2 , já que ambos os casos são codificados da mesma maneira: dois bits a zero. Também é importante que a descodificação seja instantânea, isto é, logo que a sequência B_k é recebida, podermos decidir que foi enviado S_k . Por exemplo, no código

$$B_1 = 00, \quad B_2 = 10 \quad \text{e} \quad B_3 = 1$$

isso não é possível: depois de receber um bit a 1 temos de esperar pelo bit seguinte para determinar qual foi o símbolo recebido (neste caso será S_2 se o próximo bit for 0 e será S_3 se for 1).

Uma maneira de garantir uma descodificação sem ambiguidades e instantânea consiste em usar códigos de prefixos. A ideia é construir uma árvore binária que tem como folhas os símbolos, e para a qual cada ramo para a esquerda corresponde a um bit a 0 e cada ramo para a direita corresponde a um bit a 1. A sequência de bits do código

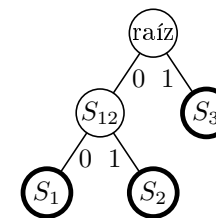


Figura 13: Árvore de decisão de um código de prefixos. O código (B_1) de S_1 é 00, o de S_2 é 01 e o de S_3 é 1. O símbolo S_{12} corresponde a uma descodificação parcial; corresponde à união dos símbolos S_1 e S_2 e o seu código é apenas 0. Quando se acaba de descodificar um símbolo volta-se à raiz para se começar a descodificar o símbolo seguinte.

de cada símbolo é obtida percorrendo a árvore desde a sua raiz até ao símbolo e concatenando os bits assim obtidos. Essa árvore pode e deve ser usada para descodificar a informação. Como o caminho da raiz para cada folha é único o código é instantâneo (logo que se chega à folha ficamos a conhecer o símbolo), e não tem ambiguidades se os bits usados nos dois ramos de cada nó interior da árvore forem diferentes. A figura 13 ilustra tudo isto.

A construção de um código de Huffman consiste em construir a árvore de decisão a partir das suas folhas. Em cada fase da construção da árvore selecionam-se dois nós ativos para serem aglomerados num único nó (mais perto da raiz). Um dos ramos é etiquetado com um 0 e o outro com um 1. Esses dois nós deixam de estar ativos, passando a estar ativo o nó que foi criado. Inicialmente, as folhas (S_1 a S_N) são os únicos nós da árvore que existem e estão todos ativos. Continua-se este processo até apenas restar um nó, que será a raiz da árvore. Por exemplo, para o exemplo da figura 13, inicialmente apenas existem os nós S_1 , S_2 e S_3 , estando todos eles ativos. Foram primeiro selecionados os nós S_1 e S_2 para serem aglomerados, dando isso origem ao nó S_{12} . Depois foram aglomerados os nós S_{12} e S_3 , dando origem ao nó S_{123} , que, sendo o único nó ativo, é a raiz da árvore.

Para terminar a descrição falta mencionar um aspeto muito importante, que é o que torna o código o mais eficiente possível: em cada fase do processo, os dois nós que são selecionados para serem agloremados

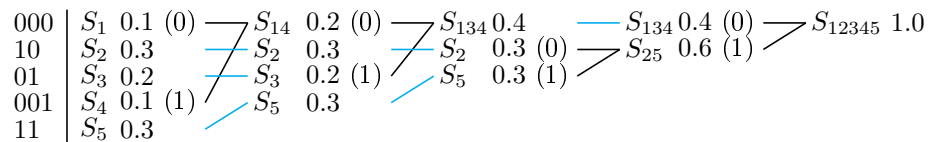


Figura 14: Exemplo de construção de um código de Huffman. Em cada etapa as duas entradas menos prováveis são aglomeradas. Depois do processo completo o código de cada símbolo é lido da direita para a esquerda. O número médio de bits por símbolo deste código é 2.2. Como a entropia é neste caso ligeiramente superior a 2.17095, o código de Huffman está muito perto do que é teoricamente possível fazer (usando, por exemplo, um codificador aritmético, que será um dos temas a estudar na próxima aula).

dos, são os que têm menor probabilidade. É claro que ao aglomerar, o novo nó terá como probabilidade a soma das probabilidades dos dois nós aglomerados. Logo, a raiz terá sempre probabilidade 1. Na figura 14 ilustra-se todo o processo para um caso concreto com 5 símbolos.