



**Ciências
ULisboa**

Faculdade
de Ciências
da Universidade
de Lisboa

NOTAS SOBRE DIGITALIZAÇÃO DE SINAIS ANALÓGICOS VARIÁVEIS NO TEMPO

FÍSICA EXPERIMENTAL PARA ENGENHARIA INFORMÁTICA

DEPARTAMENTO DE FÍSICA

Rui Jorge Agostinho*

1 de Dezembro, 2021

Conteúdo

1 A Discretização de um Sinal Analógico	2
2 A Taxa de Amostragem da Discretização	3
2.1 O Teorema de Nyquist-Shannon	3
2.2 Modulação da Amplitude e padrões de Moiré	4
2.3 A Taxa Ótima de Amostragem	5
3 A Digitalização dos Dados	6
3.1 Números Binários Inteiros com Sinal	6
3.2 A interface digitalizadora Science Workshop 500	8
3.3 Armazenamento dos Dados Digitais	9
3.3.1 Files de Texto	9
3.3.2 Files Binárias	9
3.3.3 Streams de Música	10

Resumo

Apresentam-se aqui os conceitos, diagramas e equações que dão apoio aos trabalhos laboratoriais deste tema[9], nomeadamente os circuitos E e OU realizados com díodos, um circuito NÃO com um transistor, os circuitos de adição e subtração de 2 bits com função de "e vai", um flip-flop D com porta de controlo como memória de 1 bit usando portas NAND e a sincronização pelo sinal de *clock*.

1 A Discretização de um Sinal Analógico

Tome-se o caso mais simples, o dum sinal analógico do tipo sinusoidal (com uma só frequência f_s)

$$V_g(t) = A_s \sin(\omega_s t) \quad (1)$$

onde A_s é a sua amplitude e $\omega_s = 2\pi f_s$ é a frequência angular do sinal.

Esta definição implica que existe uma diferença de potencial (ddp) $V_g(t)$ que varia de modo contínuo no tempo, ou seja, em qualquer instante t o sinal tem um valor bem definido e $\in \mathbb{R}$, situado entre $-A_s$ e $+A_s$ volts. Esse tipo de sinal está representado na figura 1 na pág. seguinte, com a curva a cor de laranja.

O conceito de discretização do sinal é o de abandonar a totalidade infinita dos valores contínuos que ele pode assumir ao longo do tempo, e substituí-lo por uma amostra dos seus valores possíveis, de tamanho finito, que é obtida em instantes de tempo regularmente espaçados entre si. Assim, passa-se duma variação contínua para uma tabela de valores discretos (cf. tab. 1 na pág. seguinte).

Esta medição sistemática, à intensidade instantânea do sinal, produz uma tabela com apenas alguns dos valores que $V_g(t)$ pode assumir, ou seja, constitui uma amostra do sinal real. O objetivo final é que através desta tabela de valores (a amostra completa) seja possível recuperar as características fundamentais do sinal, A_g e f_s , com a maior exatidão e o menor ruído possíveis.

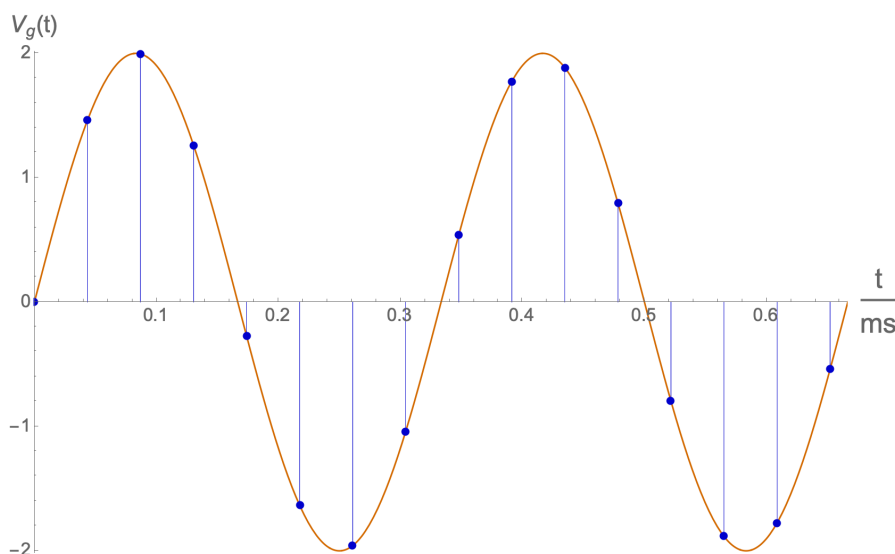


Figura 1. Sinal de frequência $f_s = 3,00$ kHz que é medido com uma taxa de amostragem $f_a = 23,0$ kHz, ou seja, a cada $\delta t_a = 1/f_a = 43,478 \mu s$ (pontos a azul).

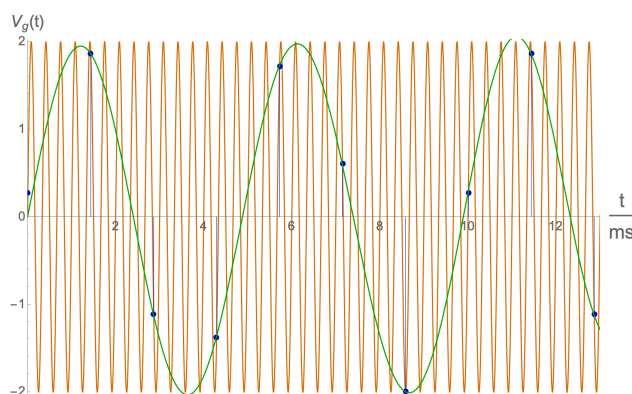
t_a (ms)	$V_g(t_a)$
0.00000	+0.00000
0.04348	+1.46167
0.08696	+1.99534
0.13046	+1.26218
0.17391	-0.27233
0.21739	-1.63394
0.26087	-1.95817
0.30435	-1.03917
0.34783	+0.53959
0.39130	+1.77577
0.43478	+1.88452
0.47826	+0.79680
0.52174	-0.79680
0.56522	-1.88452
0.60870	-1.77577
0.65217	-0.53959

Tabela 1. Os valores $V_g(t_a)$ que são medidos no sinal, nos instantes t_a .

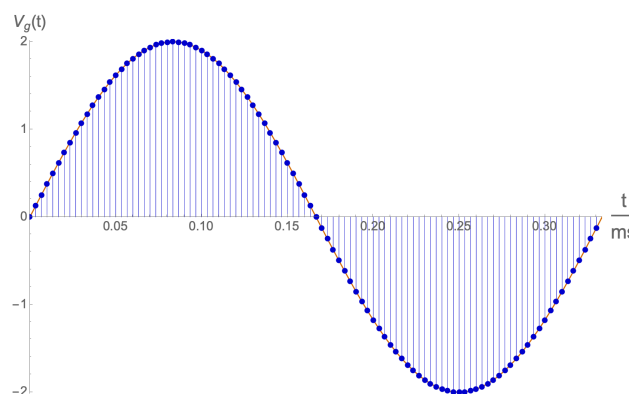
2 A Taxa de Amostragem da Discretização

É óbvio que a capacidade de atingir o objetivo proposto depende da frequência f_a com que se faz a amostragem, relativamente à frequência do sinal f_s , havendo dois casos extremos, mostrados na fig. 2:

- Se $f_a < f_s$ então não é possível recuperar f_s e obter A_g depende do tempo total de amostragem Δt_a ser muito superior ao período do sinal T_s . Mas a tabela dos dados poderá ser pequena.
- Se $f_a \gg f_s$ então é possível recuperar A_g e f_s com grande precisão, mas paga-se o preço da tabela de dados obtidos ser muito grande.



(a) Discretização a $f_a = 700$ Hz. A curva verde é o sinal recuperado.



(b) Discretização com $f_a = 300$ kHz.

Figura 2. Exemplo de possíveis resultados com **(a)** uma sub-amostragem muito pronunciada que impede a recuperação do sinal original, e **(b)** uma super amostragem, para um sinal original de frequência $f_s = 3$ kHz, a cor de laranja. Em **(b)** recupera-se otimamente o sinal original pois $f_a = 100f_s$.

2.1 O Teorema de Nyquist-Shannon

Entre os extremos que se podem escolher para a frequência de amostragem f_a , vulgarmente designada por taxa de amostragem, existe um valor mínimo que garante a recuperação da frequência original do sinal, f_s . Isso

foi demonstrado por Nyquist[1], depois por Shannon[2], e constitui o "teorema da amostragem de Nyquist-Shannon"[3], que se pode formular assim (na versão de Shannon):

– Se uma função $x(t)$ contém frequências não superiores a $f_{s,max}$ (Hz), então a função pode ser completamente determinada pelo valor das suas ordenadas numa série de pontos espaçados de $\frac{1}{2f_{s,max}}$ segundos.

Deduz-se daqui que a taxa mínima de amostragem que permite recuperar o sinal original — designa-se por taxa de Nyquist — é:

$$f_{a,min} = 2f_{s,max} = f_{Nyquist} \quad (2)$$

Por outro lado, se um sinal $v(t)$ for discretizado com uma taxa de amostragem f_a , então a maior frequência que é possível detetar em $v(t)$, e que se designa por frequência de Nyquist, será

$$f_{Nyquist} = f_a/2 \quad (3)$$

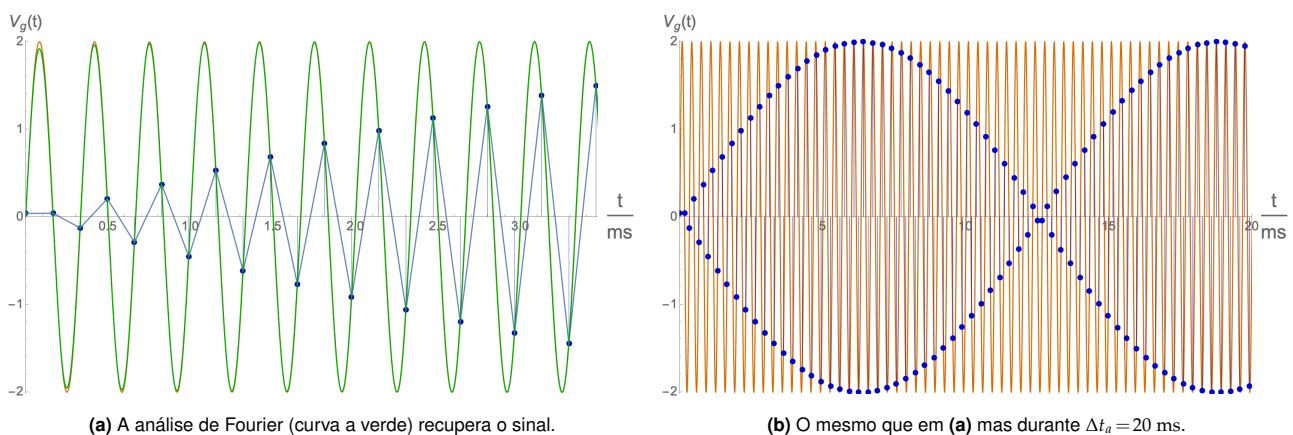


Figura 3. Uso Taxa de amostragem de Nyquist, aqui a $f_a = 6,015$ kHz, para um sinal original de frequência $f_s = 3$ kHz, a cor de laranja. Em (a) recupera-se o sinal e, em (b), vê-se o fenómeno de Moiré na sequência dos pontos da amostragem.

A análise da fig. 3a mostra que a transformada de Fourier inversa (curva a verde) recupera sem grandes problemas o sinal original com $f_s = 3$ kHz. Porém, se se usar a simples conexão dos pontos da amostra com segmentos de reta (a azul), o resultado é fazer aparecer um sinal com a frequência correta (a portadora) mas com um "modulação em amplitude", como bem se vê na fig. 3b.

Se a instrumentação que recupera o sinal original não o fizer corretamente, então esse batimento em amplitude é espúrio e, por exemplo, num sinal de áudio surge como uma frequência extra no som que não existia na realidade (o original).

2.2 Modulação da Amplitude e padrões de Moiré

Vem a propósito dizer, *mas fugindo ao tema*, que esta técnica é usada na emissão de rádio AM, de Amplitude Modulada. O sinal é emitido numa frequência portadora, entre os 540 kHz e os 1700 kHz, mas a informação pretendida (áudio da voz e música) está contida na variação da amplitude da portadora. A frequência desta variação não é mais do que a frequência da voz ou da música que se quer transmitir. Obtém-se este sinal com a equação

$$A_{port} \cos(2\pi f_{port} t) + A_{aud} \sin(2\pi f_{aud} t) \quad (4)$$

onde os A_x são as amplitudes dos sinais e f_x as respetivas frequências. Repare que $f_{aud} \equiv f_{aud}(t)$ é variável no tempo (a música). Veja-se um exemplo na fig. 4 na pág. seguinte.

A estrutura de modulação visível na imagem na fig. 3b, em alguns casos poderá constituir um padrão de Moiré [4] em 2D, como bem se vê na fig. 5b na pág. seguinte, na sequência espacial dos pontos azuis. O efeito corresponde ao aparecimento de padrões visualmente distintos, mas que não existem no sinal original. Estas podem ser estudadas como um "efeito de interferência" entre o sinal original e a grelha de observação que é utilizada (sobreposta), neste caso a taxa de amostragem.

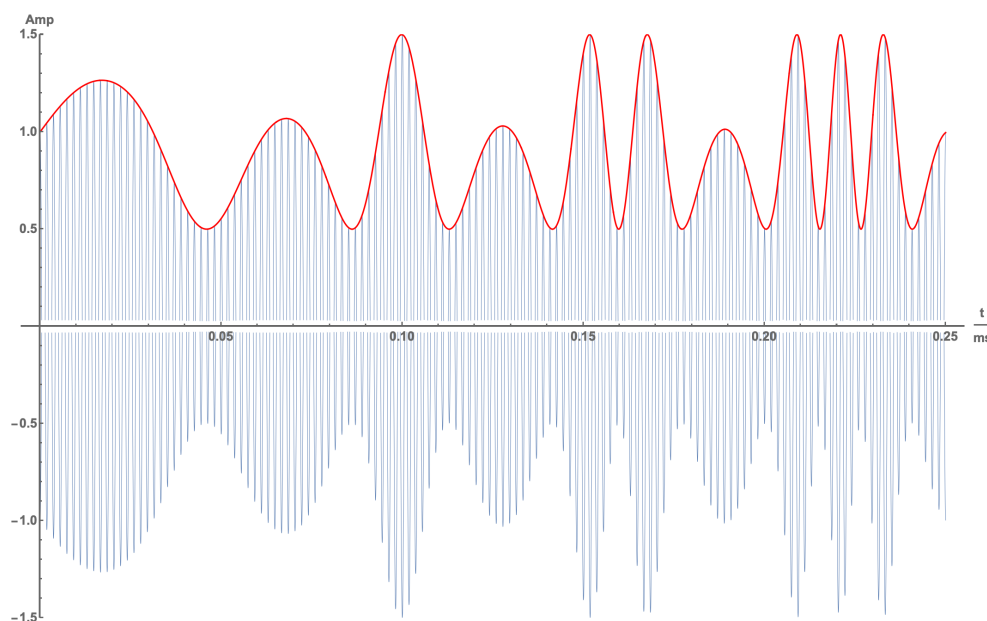


Figura 4. Sinal de rádio nos 540 kHz (AM) com modulação em amplitude de música, em torno de 8 kHz, sinal vermelho.

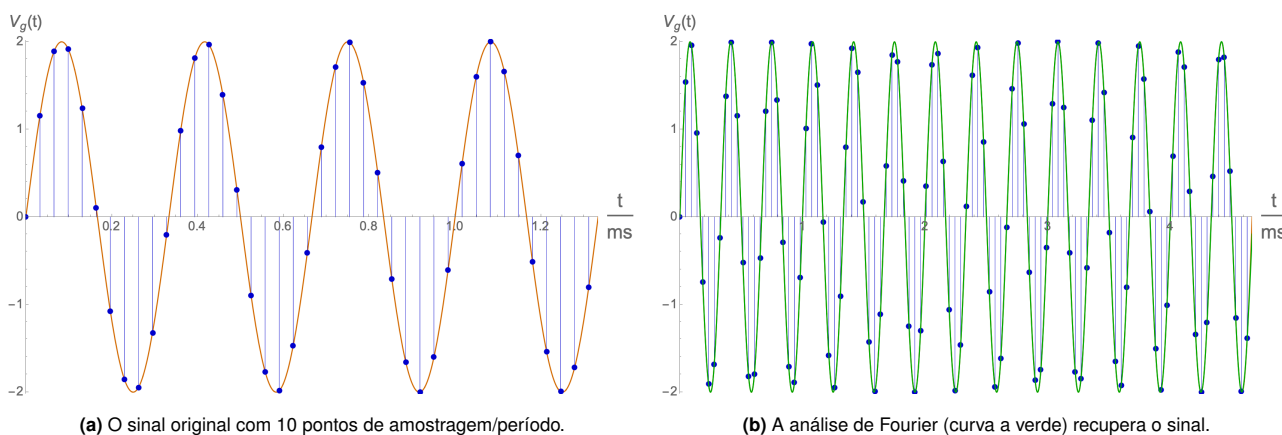
Esta situação não é exclusiva dos sinais elétricos variáveis no tempo ($V_g(t)$ volts) pois o princípio matemático é de âmbito genérico. Imaginemos que o sinal $S(x)$ é a variação de intensidade luminosa $S(x) \equiv I(x)$ ao longo duma direção x , num cenário à nossa frente que se pretende fotografar.

Um efeito semelhante ao da fig. 3 (pág. anterior) pode aparecer quando se discretiza a imagem do mundo real: a passagem da variação contínua de $I(x)$ para a imagem pixelizada (descontínua em $x = x_1, x_2, \dots, x_n \Rightarrow I(x_1), I(x_2), \dots, I(x_n)$), faz aparecer padrões fictícios na variação de intensidade luminosa $I(x_n)$ ao longo dessa linha, agora transformada em píxeis x_n discretos. E isto torna-se bem visível.

2.3 A Taxa Ótima de Amostragem

Estes problemas são minimizados (ou mesmo anulados) quando se usa uma boa taxa (frequência) de amostragem f_a , do sinal com frequência f_s , de modo a reproduzi-lo (à posteriori) com a máxima fidelidade. Sem entrar nos detalhes matemáticos, já se consegue um bom resultado quando

$$f_a > 10f_s \quad (5)$$



(a) O sinal original com 10 pontos de amostragem/periód.

(b) A análise de Fourier (curva a verde) recupera o sinal.

Figura 5. (a) Uso da Taxa de amostragem $f_a = 30,05$ kHz para um sinal original de frequência $f_s = 3$ kHz, a cor de laranja. Em (b) nota-se a recuperação do sinal sem quaisquer problemas, mas verifica-se o fenómeno de Moiré, só visual.

Note-se que mesmo com uma frequência de amostragem $f_a = 10f_s$ a amplitude apresenta uma pequena flutuação de uns poucos % (fig. 5b). Para reduzir ao máximo este problema (se o for) então a solução será aumentar ainda mais a taxa de amostragem f_a .

É por estas razões que o osciloscópio Rigol 1001B, usado nos laboratórios, tem uma taxa máxima $f_{a,max} = 1 \text{ Gs/s}$ (giga samples per second), o que permite obter uma ótima descrição dos sinais com frequências até $f_s = f_{a,max}/20 = 50 \text{ MHz}$. Assim, mesmo a $f_s = 50 \text{ MHz}$, como $f_a = 20f_s \Rightarrow$ um período do sinal conterá 20 pontos de amostragem, o que é ótimo!

3 A Digitalização dos Dados

Todos os dados obtidos com a amostragem do sinal original, como os da tab. 1 na pág. 3, são transformados para a forma binária (base 2), ao serem guardados numa memória qualquer (um registo no CPU, RAM, CD, DVD, etc.), ou para cálculo no CPU ou no Processador de Sinal.

Na realidade, os valores obtidos deixam de pertencer ao conjunto dos números Reais (com dízimas infinitas e variação contínua) e reduzem-se a uma coleção finita de N_{tot} valores possíveis, que são associados a Números Inteiros Binários. A quantidade N_{tot} é controlada pela totalidade de bits usados no formato digital escolhido, para estes números inteiros. O total de valores diferentes que é possível guardar numa palavra de N_b bits (sem sinal, ou seja, todos positivos), é dado por

$$N_{tot,+} = 2^{N_b} \quad (6)$$

Para *um sinal todo positivo* ($V_g(t) \geq 0 \text{ V}$) que varie até ao máximo $A_s > 0 \text{ V}$, a *resolução da digitalização* obtida fica limitada a

$$\Delta V_{rsl,+} = \frac{A_s}{2^{N_b} - 1} \text{ volts} \quad (7)$$

que é a capacidade de discriminar dois valores de tensão próximos entre si.

Vejamos o caso simples de guardar os valores de $V_g(t)$ (tab. 1 na pág. 3) em variáveis de $N_b = 4$ bits. Então só estarão disponíveis $N_{digit} = 2^4 = 16$ valores de tensão elétrica, distintos entre si. Como o sinal varia entre $\pm V_{g,max} = \pm 2 \text{ volt}$, então haverá uma separação (um intervalo) entre estes valores de:

$$\Delta V_{rsl,+} = \frac{2 - (-2)}{2^4 - 1} = 0,2666(6) \text{ volts} \quad (8)$$

Deste modo, a *diferença mínima de tensão que permite discriminar* dois valores próximos entre si (de V_g) é $\Delta V_{rsl,+}$. Por isso é que se designa por *resolução da digitalização*.

Dito pela negativa, se dois valores tiverem uma diferença $V_g(t_2) - V_g(t_1) < \Delta V_{rsl,+}$ então não podem ser distinguidos como diferentes, ou "resolvidos" (separados) entre si. Neste caso *é-lhes atribuído o mesmo valor digitalizado*.

Há uma consequência importante desta técnica: o que se guarda em memória é o binário inteiro $n_{bin,i}$ associado ao valor real de $V_g(t_i)$, que foi medido no instante t_i da amostragem. Para este exemplo, o respetivo valor de tensão recupera-se com

$$V_{g,digit}(t_i) = V_{g,max} - n_i \Delta V_{rsl,+} \leq V_g(t_i) \quad (9)$$

n	n_{bin}	V_{dig}/V
0	0000	-2.0000
1	0001	-1.7333
2	0010	-1.4667
3	0011	-1.2000
4	0100	-0.9333
5	0101	-0.6667
6	0110	-0.4000
7	0111	-0.1333
8	0000	+0.1333
9	1001	+0.4000
10	1010	+0.6667
11	1011	+0.9333
12	1100	+1.2000
13	1101	+1.4667
14	1110	+1.7333
15	1111	+2.0000

Tabela 2. Valores digitais.

3.1 Números Binários Inteiros com Sinal

Há que fazer uma correção ao exemplo antes dado. Quando a intensidade do sinal $V_g(t)$ pode ser positiva ou negativa, devem usar-se *binários inteiros com sinal*. Existem três maneiras de representar um número binário X

com sinal, mas todas elas usam o bit mais à esquerda (o mais significativo, ou *MSB* de "Most Significant Bit") para indicar o sinal. A regra é que:

$$\text{se } X \geq 0 \Rightarrow \text{MSB} = 0 \quad \text{e se } X < 0 \Rightarrow \text{MSB} = 1 \quad (10)$$

Por exemplo, o valor $X = +107_{10}$ apresentado em 8 bits é $X = 0|01101011_2$, onde o traço vertical | é apenas uma ajuda visual, porque não existe na realidade. É apenas mais um bit na eletrónica.

1. Uma das maneiras é usar a codificação "*sinal | valor absoluto*". Por exemplo $-X = -107_{10} = 1|01101011_2$. Nesta codificação tanto existe o valor $+0_{10} = 0|0000000_2$ como há o $-0_{10} = 1|0000000_2$.
2. Outra maneira é codificar pelo "*complemento para 1*" (*cp1*), que se obtém pela inversão ($0 \rightarrow 1, 1 \rightarrow 0$) de cada bit do número. Designemos $\boxed{cp1(X) \equiv \bar{X}}$. Assim $cp1(X) = \bar{X} = -107_{10} = 1|10010100_2$. Note-se que o resultado corresponde à operação $1|1111111_2 - 0|01101011_2 = 1|10010100_2$. Nesta codificação tanto existe o valor $+0_{10} = 0|0000000_2$ como o $-0_{10} = 1|1111111_2 = cp1(+0)$.
3. A mais importante é a do "*complemento para 2*" (*cp2*). A operação define-se assim: $cp2(X) = cp1(X) + 1$, ou seja, invertem-se os bits ($1 \rightarrow 0, 0 \rightarrow 1$) e depois adiciona-se 1. Designemos $\boxed{cp2(X) \equiv \bar{\bar{X}}}$.

Do exemplo anterior: $cp2(X) = \bar{\bar{X}} = -107_{10} = 1|10010101_2$.

A implementação das operações de subtração, em hardware, são feitas com recurso à soma com o *cp2*. Exemplo com *sinal + 3 bits*: se $B = 0|101_2 \Rightarrow \bar{B} = 1|010_2 + 0|001_2 = 1|011_2$. Logo $B + \bar{B} = 10|000_2 = 2^{3+1}$.

Nos 1+3 bits (*sinal | mantissa*) o resultado é $0|000_2 = B + (-B)$. O bit inicial "1" é um *overflow* e descarta-se.

Note-se que se X_2 tem *sinal + nb* bits de mantissa o $cp2(X_2)$ é o que falta para 2^{nb+1} .

Uma vantagem desta codificação é que existe um valor positivo de $+0_{10} = 0|0000000_2$, auto consistente pois o $cp2(+0) = 1|1111111_2 + 1_2 = 0|0000000_2$. O mesmo acontece com o valor negativo $-0_{10} = 1|0000000_2$, que tem $cp2(-0) = 0|1111111_2 + 1_2 = 1|0000000_2$.

Usemos o formato "*sinal-valor*" para o exemplo que se pretende explorar, utilizando uma palavra de $N_b = 4$ bits. Repare-se que devido à existência de dois valores nulos $+0_{10} = 0|000_2$ e $-0_{10} = 1|000_2$, o total de valores diferentes que existem é dado por

$$\boxed{N_{tot,\pm} = 2^{N_b} - 1} \quad (11)$$

Neste caso, serão $2^4 - 1 = 15$ valores possíveis, que designaremos cada um por k_i .

Há alguns aspetos a realçar quando se faz uma digitalização e veja-se o exemplo na tab. 3 (pág. seguinte):

- Para um sinal que se pretende digitalizar a N_b bits e que pode assumir valores extremos de $\pm A_s$, a resolução digital (discriminação) calcula-se por

$$\boxed{\Delta V_{rsl,\pm} = \frac{2A_s}{N_{tot} - 1} = \frac{2A_s}{2^{N_b} - 2}} \quad (12)$$

o que, neste exemplo, se torna em $\Delta V_{rsl,\pm} = 4/(2^4 - 2) = 0,2857 \text{ V}$ (cf. na tab. 3a).

- Ora, todos os valores de $V_g(t_{a,i})$ (tab. 3b) serão arredondados, na realidade são truncados, pelo Conversor Analógico-Digital (ADC do inglês) para coincidirem com um dos valores na tab. 3a.
- Assim, os verdadeiros valores digitalizados $V_{i,digit}$ estão na tab. 3c (pág. seguinte) e têm $erro_i$ em relação aos valores originais $V_g(t_{a,i})$ exatos (tab. 3b na pág. seguinte), obtidos pela amostragem (discretização).
- Como a resolução digital $\Delta V_{rsl,\pm}$ é uma constante (uma vez escolhidos N_b e A_s) e a taxa de amostragem f_a fixa os instantes de amostragem $t_{a,i}$ (sempre intervalados de $\delta t_a = 1/f_a$), então num CD de música (uma file com música) basta guardar toda a sequência de valores $k_{i,bin} \leftrightarrow V_g(t_{a,i})$ para se reconstruir o *sinal(t)* à posteriori (cf. sec. 3.3.3 na pág. 10).

k_{10}	k_{bin}	volts	\Rightarrow	$t_{a,i}$ (ms)	$V_g(t_{a,i})$		$k_{i,10}$	$k_{i,bin}$	$V_{i,digit}$	$erro_i$
+7	0111	+2.0000		0.00000	+0.0000	\rightarrow	+0	0000	+0.0000	-0.0000
+6	0110	+1.7143		0.04348	+1.4617	\rightarrow	+5	0101	+1.4286	-0.0331
+5	0101	+1.4286		0.08696	+1.9953	\rightarrow	+6	0110	+1.7143	-0.2810
+4	0100	+1.1429		0.13046	+1.2622	\rightarrow	+4	0100	+1.1429	-0.1193
+3	0011	+0.8571		0.17391	-0.2723	\rightarrow	-0	1000	-0.0000	+0.2723
+2	0010	+0.5714		0.21739	-1.6339	\rightarrow	-5	1101	-1.4286	-0.2053
+1	0001	+0.2857		0.26087	-1.9582	\rightarrow	-6	1110	-1.7143	+0.2439
+0	0000	+0.0000		0.30435	-1.0392	\rightarrow	-3	1011	-0.8571	+0.1821
-0	1000	-0.0000		0.34783	+0.5396	\rightarrow	+1	0001	+0.2857	-0.2539
-1	1001	-0.2857		0.39130	+1.7758	\rightarrow	+6	0110	+1.7143	-0.0615
-2	1010	-0.5714		0.43478	+1.8845	\rightarrow	+6	0110	+1.7143	-0.1702
-3	1011	-0.8571		0.47826	+0.7968	\rightarrow	+2	0010	+0.5714	-0.2254
-4	1100	-1.1429		0.52174	-0.7968	\rightarrow	-2	1010	-0.5714	+0.2254
-5	1101	-1.4286		0.56522	-1.8845	\rightarrow	-6	1110	-1.7143	+0.1702
-6	1110	-1.7143		0.60870	-1.7758	\rightarrow	-6	1110	-1.7143	+0.0615
-7	1111	-2.0000	0.65217	-0.5396	\rightarrow	-1	1001	-0.2857	+0.2539	

(a) Os k_i binários inteiros com sinal e os valores de tensão associados.

(b) Os 16 valores exatos da amostragem do sinal $V_g(t)$ da tabela 1.

(c) Os valores digitalizados que são atribuídos a $V_g(t_{a,i})$, com $erro_i = V_{i,digit} - V_g(t_{a,i})$ volts.

Tabela 3. Digitalização em inteiros com sinal e $N_b = 4$ bits, dum sinal sinusoidal de $A_s = 2$ V e $f_s = 3$ kHz, com $f_a = 23$ kHz.

- A recuperação das características do sinal, temporais e de tensão, obtém-se a partir da sequência dos valores digitalizados:

$$V_g(t) \rightarrow V_g(t)_{digit} \equiv (i \cdot \delta t_a ; \Delta V_{rsl, \pm} \cdot k_i) \quad \text{com} \quad i = 0 \dots i_{max} \quad (13)$$

A tabela de valores digitalizados $V_g(t)_{digit}$ é diferente do sinal original $V_g(t)$ e, por isso, apresenta pequenas variações em amplitude que não estão em $V_g(t)$. Estas diferenças ao longo do tempo traduzem-se (na recuperação) num espectro de frequências ligeiramente diferente do original, ou seja, aparecem frequências parasita.

Obviamente quanto maior for N_b melhor será a resolução (em tensão) da digitalização, o que melhora a fidelidade entre o sinal digitalizado e o original, ou seja, diminui o $|erro_i| < \Delta V_{rsl}$ e o espectro recuperado é mais fiel ao original.

3.2 A interface digitalizadora Science Workshop 500

A interface *Science Workshop 500* (da PASCO) para aquisição de dados está ligada à porta série (USB) do PC e permite a medição simultânea de 3 sinais de tensão, com taxa de amostragem f_a regulável e tempo total de medição Δt_{tot} escolhidos pelo utilizador, através do software *Data Studio* que faz output da tabela de valores medidos, além de gráficos e a Transformada de Fourier Rápida (FFT, do inglês).

Das 3 entradas analógicas A, B e C que possui, deve usar-se a A que é diferencial (tal como um voltímetro), o que significa que os dois condutores de input estão isolados das entradas B e C, que têm o condutor da massa em comum com a alimentação da rede. Esta interface é um voltímetro digital, que pode usar-se para estudar a digitalização dum sinal variável no tempo.

A *Science Workshop 500* tem um conversor analógico-digital (ADC) de $N_b = 12$ bits e aceita tensões de entrada com valores extremos de $\pm A_{max} = \pm 10$ V. Assim, o valor mínimo capaz de ser discriminado é de

$$\Delta V_{SW} = \frac{2 \cdot 10 \text{ V}}{2^{12} - 2} = 4,885 \text{ mV} \quad (14)$$

Na fig. 6 na pág. seguinte está o resultado da digitalização dum sinal sinusoidal com $A_s = 35$ mV e $f_s = 1,94$ kHz. Notam-se bem os valores de tensão possíveis de obter, que são todos múltiplos de $\pm \Delta V_{SW}$ (eq. (14)).

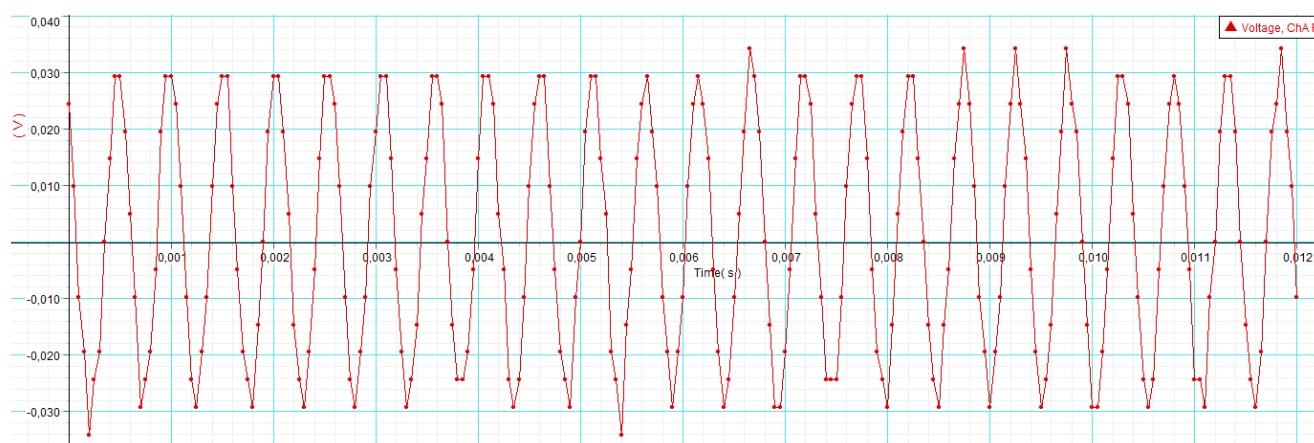


Figura 6. Exemplo de digitalização a $f_a = 20$ kHz obtido com a Science Workshop 500, dum sinal com $A_s = 35$ mV.

3.3 Armazenamento dos Dados Digitais

Como se explicou na sec. 3.1 (pág. 6), os valores efetivamente necessários para descrever o sinal original, são os valores binários inteiros (com sinal) que estão na coluna $k_{a,bin} \equiv k_a$, tab. 3c. Estes permitem recuperar as características de variação temporal e de tensão do sinal original (eq. (13)), sabendo quais foram os parâmetros da digitalização: a resolução ΔV_{rsl} e a taxa de amostragem f_a .

A questão do armazenamento tem diversas soluções, dependendo do formato escolhido para a file com estes dados. Repare que uma file é uma sequência de bytes que terminam com o byte de controlo EOF (End Of File). Quando a leitura da file é sequencial e ininterrupta no tempo esta designa-se por um *stream*, que é caso de música ou vídeo. Mas a designação pode ser usada para qualquer file.

3.3.1 Files de Texto

Se se quiser uma file de texto simples com as duas colunas $t_{a,i}$ (segundos) e $V_{i,digit}$ (volts), então é necessário converter todos os valores numéricos em letras (cada letra é 1 byte), separá-los entre si (espaço, vírgula, etc.) e separar as linhas umas das outras, o que exige a introdução de dois bytes de controlo para EOL (End Of Line) e CR (Carriage Return) no final de cada linha.

A vantagem deste formato é que qualquer programa lê facilmente os *streams* do tipo "file.txt": todos os bytes são letras e quando aparecem os caracteres de controlo EOL e CR criam-se novas linhas.

Por exemplo, ao guardar todos os valores na tab. 1 (pág. 3) em modo de texto, com espaços de separação (ou outros carateres) entre as variáveis ou de embelezamento e EOL+CR, serão usados:

$$fileSizeTxt = 16 \text{ linhas} \times \left(2 \frac{\text{vars}}{\text{linha}} \times 8 \frac{\text{bytes}}{\text{var}} + 4 \frac{\text{chars}}{\text{linha}} \times 1 \frac{\text{byte}}{\text{char}} + 2 \frac{\text{bytes}}{\text{linha}} \right) = 352 \text{ bytes} \quad (15)$$

3.3.2 Files Binárias

Uma variante da sequenciação anterior é manter os valores reais $t_{a,i}$ (segundos) e $V_{i,digit}$ (volts) em formato binário, o que permite reduzir bastante a quantidade de bytes usados. A desvantagem é que precisa de um programa adequado à leitura do *stream*, para interpretar corretamente a sequência dos bytes que estão guardados na file.

Os valores podem ser reais de precisão simples, o mesmo que *float* [5], ocupando 4 bytes cada, ou então de precisão dupla, *double* [6], que ocupam 8 bytes cada. A sua formatação interna é a seguinte:

Note-se que o 1º bit é o sinal da mantissa. O expoente é um número inteiro sem sinal mas o seu verdadeiro valor é obtido subtraindo 127 ao valor escrito (nos *floats*). Corretamente, a norma IEEE 754 põe o expoente a variar de -126 a $+127$, mas esse detalhe (entre outros) não será discutido aqui. Convertendo em base 10, o valor

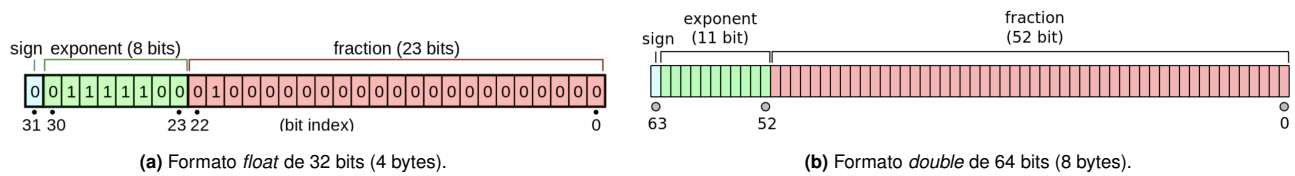


Figura 7. Distribuição dos bits dentro das variáveis do tipo *float* [5] e do tipo *double* [6]. Imagens do Wikipedia.

máximo equivale a $\log_{10}(2^{127}) = 38,23$, ou seja, na ordem de grandeza 10^{38} .

As mantissas são armazenadas na forma binária "1.*abcde*...", onde cada letra *abcde* é 0 ou 1. O "1." *não é escrito mas é subentendido*, ou seja, apenas se guarda o resto "*abcde*" nos bits de posição $p = 22, 21, 20, \dots, 0$, no caso dum *float*. O valor do bit b_p (0 ou 1) na posição p é de 2^{p-23} . Logo, o valor da mantissa obtém-se por

$$mant = 1 + \sum_{p=0}^{22} b_p 2^{p-23} = 1 + a 2^{-1} + b 2^{-2} + c 2^{-3} + d 2^{-4} + \dots \quad (16)$$

O máximo da mantissa é $abcde\dots = 1111\dots_2 \Rightarrow mant = 1,999999881$ que gera o valor máximo de $x_{max} = mant \times 2^{127} = 3,4028234664 \times 10^{+38}$. O valor mínimo positivo é $(1 + 2^{-23}) \times 2^{-126} = 1,1754944910 \times 10^{-38}$. Nos *floats* há 7 a 9 algarismos significativos.

Para os *double* o expoente toma valores entre $\pm(2^{10} - 1) = \pm 1023$. Convertendo em base 10 equivale a $\log_{10}(2^{1023}) = 307,954$. Pode calcular os valores máximo e mínimo de maneira análoga ao caso anterior:

$$x_{max} = 1,797693134862316 \times 10^{308} \quad \text{e} \quad x_{min} = 2,225073858507201 \times 10^{-308} > 0 \quad (17)$$

Uma variável *double* tem até 16 algarismos significativos na mantissa. Encontra uma boa descrição nas referências [5] e [6], com todos os detalhes e outras especificidades.

O que nos interessa aqui é que os valores das variáveis são guardados num formato independente do seu real valor, mas de comprimento fixo: 4 bytes para *floats* e 8 bytes para *doubles*.

A estrutura da file não precisa de EOL's e CR's pois não há linhas (com texto) para apresentar ao público. Porém, a sua leitura exige conhecer a organização das variáveis, a sua sequência, para que os bytes sejam lidos na quantidade correta e bem atribuídos às variáveis no programa que os lê.

Finalizando, ao guardar todos os valores na tab. 1 (pág. 3) em formato *float* (file binária), serão usados

$$fileSizeBin = 16 \text{ linhas} \times 2 \frac{vars}{linha} \times 4 \frac{bytes}{var} = 128 \text{ bytes} \quad (18)$$

que é muito menor que o caso duma file de texto (eq. (15) na pág. anterior).

3.3.3 Streams de Música

A norma CD-Audio digitaliza os sinais (som stereo em 2 canais) com uma taxa de amostragem de $f_a = 44,100$ kHz e os valores de tensão $V_g(t)$ são guardados em palavras de $N_b = 16$ bits. Esta definição impõe um $\delta t_a = 22,7 \mu s$ e uma resolução $\Delta V_{rsl,\pm} = 15,259 \mu V / (1V \text{ de sinal})$.

Estabelecida a norma, os aparelhos/software têm de antemão os valores de δt_a e de $\Delta V_{rsl,\pm}$. Isto permite que ao digitalizar a música, se guardem apenas os valores sequenciais dos $k_{i,bin}$ que são palavras de 16 bits (inteiros com sinal). É daí que se recupera o sinal original através da eq. (13) na pág. 8, com um Conversor Digital-Analógico (DAC, do inglês), que é enviado ao sistema amplificador de som.

A totalidade de valores guardados, indicada pelo contador $i = 0..i_{max}$, é determinado pelo tempo total da música Δt_{tot}

$$i_{max} = \Delta t_{tot} \times f_a \quad (19)$$

Por exemplo, para $\Delta t_{tot} = 1 \text{ min} \Rightarrow i_{max} = 2,64600 \times 10^6 = 2,646$ milhões de amostras por cada um dos canais (esquerdo, direito). Ao guardar apenas os valores dos $k_{i,bin}$ (ex: tab. 3c na pág. 8) em 2 bytes cada um e sequencialmente (sem bytes extra de separação), a file binária de música (*stream*) terá

$$2 \text{ canais} \cdot 2,64600 \times 10^6 \frac{\text{vals}}{\text{canal}} \times 2 \frac{\text{bytes}}{\text{val}} = 10,5840 \text{ MB} = 10,094 \text{ MiB} \quad (20)$$

onde Mi indica o uso do prefixo *mebi* para sistemas binários : $1 \text{ Mi} = 2^{20} = 1024 \times 1024 = 1048576$. Também se usa o prefixo *kibi*: $1 \text{ Ki} = 2^{10} = 1024$. Repare-se na letra 'K' maiúscula, em contraste com o prefixo *kilo* (1000) que tem um 'k' minúsculo.

Curiosidade: Norio Ohga que foi cantor de ópera e posteriormente vice-Presidente da Sony, na altura da criação do Disco Compacto CD-Audio [8] (entre a Sony e a Philips) exigiu que este novo suporte tivesse capacidade suficiente para conter a 9ª Sinfonia de Beethoven (em CD-Audio), que dura aproximadamente 70 minutos mas há interpretações com 74 minutos, $\approx 747 \text{ MiB}$. Consequentemente, assim escolheu-se o valor de 74 minutos como norma do Disco Compacto.

Referências

- [1] "Certain topics in telegraph transmission theory", abr. 1928, Harry Nyquist, *Transactions AIEE*, **47** (2), 617-644.
- [2] "Communication in the presence of noise", jan. 1949, Claude E. Shannon, *Proceedings of the Institute of Radio Engineers*, **37** (1), 10-21.
- [3] https://en.wikipedia.org/wiki/Nyquist-Shannon_sampling_theorem
- [4] https://en.wikipedia.org/wiki/Moiré_pattern
- [5] https://en.wikipedia.org/wiki/Single-precision_floating-point_format
- [6] https://en.wikipedia.org/wiki/Double-precision_floating-point_format
- [7] <https://en.wikipedia.org/wiki/Mebibyte>
- [8] https://en.wikipedia.org/wiki/Compact_Disc_Digital_Audio
- [9] Rui Jorge Agostinho, *Protocolo do Trabalho Lab #6- Digitalização de Sinais Analógicos*, FCUL.