

Universidade do Minho - Dep.^{to} Informática
Mestrado em Engenharia de Telecomunicações & Informática
Serviços de Rede & Aplicações Multimédia

Ano Letivo 2023/2024 • 1º Teste Escrito • março 2024

Duração Total: 110 Minutos

Escreva as suas respostas num editor de texto à sua escolha e envie no formato PDF (ou formato de texto) para **bruno.dias@di.uminho.pt** e **bafdias@gmail.com**



Figura 1: Sistema Multimédia.

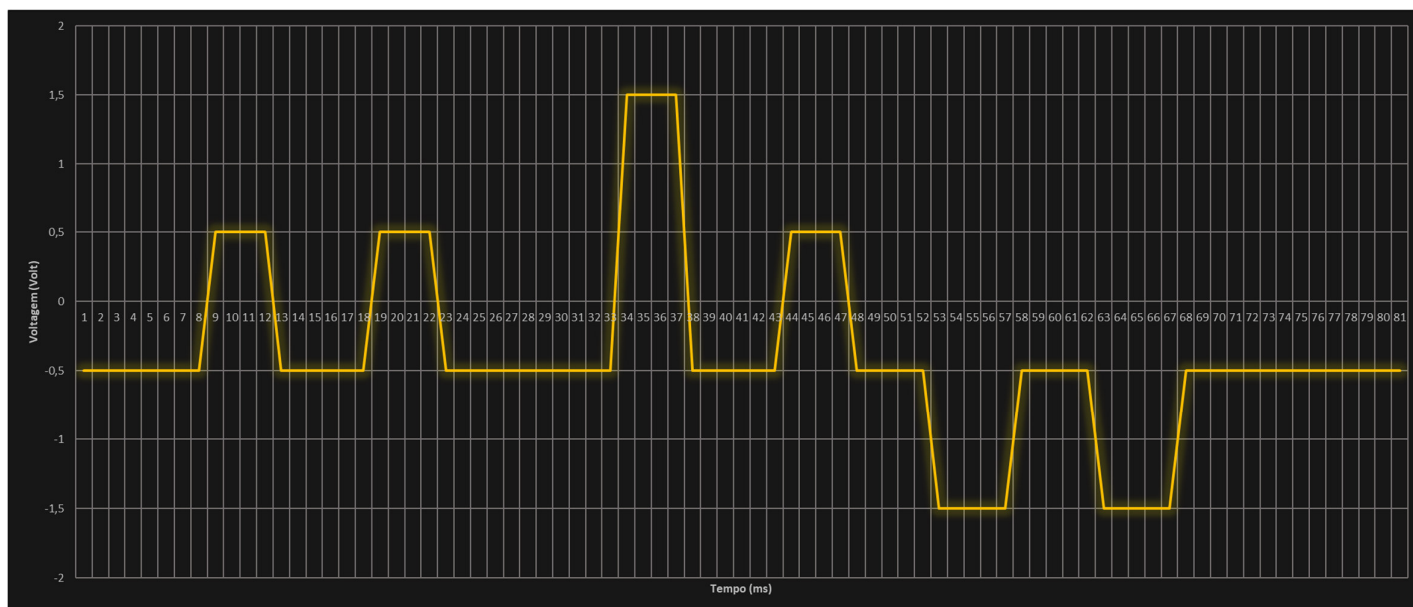


Figura 2: Um bloco exemplificativo de 80 milissegundos do sinal da fonte analógica.

Na figura 1 está esquematizado um sistema de digitalização, transmissão e gravação em ficheiro digital dum sinal analógico. A figura 2 é uma ilustração deste sinal num determinado intervalo de tempo de 80 milissegundos (ms). O sinal é captado por um sensor que digitaliza voltagens entre -2 e +2 Volt. A largura de banda B do sinal analógico é de 100 Hz.

O sistema digitaliza o sinal de voltagem do sensor através dum conversor AD com quantização uniforme e com a menor frequência de amostragem possível, gerando uma sequência PCM binária que é depois codificada num módulo com compressão por padrões LZWdR (algoritmo dado nas aulas teóricas e conforme anexado a este enunciado). Cada sequência binária resultante duma amostra (K bits/amostra) é considerada um símbolo de entrada no compressor LZWdR.

No destino do sistema de transmissão existe um descompressor LZWdR que realiza o trabalho inverso do compressor: a cada 80 ms analisa descomprime a sequência binária recebida, obtendo uma outra sequência binária que corresponde à sequência PCM original de símbolos binários (K Bits/amostra) resultante da digitalização na origem.

Por fim, esta sequência binária correspondente à digitalização de 80 ms é depois codificada/comprimida para um ficheiro de *output* através dum codificador estatístico Shannon-Fano, em que cada grupo de K bits é considerado um símbolo. A codificação Shannon-Fano dos dados

resultantes da digitalização de cada bloco de 80 ms origina um cabeçalho de metadados (informação necessária para que um descompressor Shannon-Fano possa decodificar/descomprimir o ficheiro de *output*) e uma sequência de códigos binários Shannon-Fano.

Tendo em atenção os dados do sistema multimédia da figura 1, responda às seguintes perguntas, justificando as respostas com todos os cálculos e todos os dados relevantes:

1. Qual o número mínimo de níveis quânticos que o conversor AD tem de implementar para que a qualidade de digitalização seja superior a 16 dB e qual o ritmo binário PCM correspondente à saída do conversor AD? **(10%)**

Quantização uniforme, PCM binário, $\left(\frac{S}{N}\right)_{dB} > 16 \text{ dB}$

$$\left(\frac{S}{N}\right) \leq 4.8 + 6K \Leftrightarrow 16 < 4.8 + 6K \Leftrightarrow K > 1,87 \geq 2 \text{ bits/amostra}$$

Se $f_a \geq 2B \geq 2 * 100\text{Hz} \geq 200\text{Hz}$, então, considerando a menor frequência de amostragem possível e o menor número de bits/amostra possível:

$$r_b = f_a * K = 200 * 2 = 400 \text{ bps}$$

2. Tendo em atenção os requisitos da questão anterior, qual a sequência PCM binária resultante da digitalização do bloco de 80 ms da figura 2 (assuma que a digitalização começa exatamente no momento 1 ms). **(10%)**

Sendo que $K=2$, logo $q=2^2=4$ níveis quânticos e considerando que as sequências de símbolos binários para cada nível quântico das amostras está definido como 00, 01, 10 e 11 para os intervalos de voltagem $[-2,-1[$, $[-1,0[$, $[0,1[$ e $[1,2[$, respetivamente, então a sequência PCM binária resultante é:
01 01 10 01 10 01 01 11 01 10 01 00 01 00 01 01

3. Tendo em consideração a sequência PCM calculada na questão anterior, qual a sequência de códigos de *output* (valores inteiros dos índices dos padrões) resultante da sua codificação utilizando o algoritmo LZWD? **(30%)**

Sequência a codificar ($k=4$ símbolos): 01 01 10 01 10 01 01 11 01 10 01 00 01 00 01 01

Dicionário Inicial: D.1=A (00), D.2=B (01), D.3=C (10) e D.4=D (11)

Sequência equivalente a codificar ($k=4$ símbolos): B B C B C B B D B C B A B A B B

Passo 1a: $P_a=B=D.2$ e $P_b=B=D.2$

Passo 1b: Novos padrões D.5=BB

Passo 1c: -

Passo 1d: Enviar para a saída código/índice de $P_a=2$; $P_a=P_b$

Passo 2a: $P_a=B=D.2$ e $P_b=C=D.3$

Passo 2b: Novos padrões D.6=BC

Passo 2c: Novos padrões D.7=CB

Passo 2d: Enviar para a saída código/índice de $P_a=2$; $P_a=P_b$

Passo 3a: $P_a=C=D.3$ e $P_b=BC=D.6$

Passo 3b: Novos padrões D.8=CBC

Passo 3c: -

Passo 3d: Enviar para a saída código/índice de $P_a=3$; $P_a=P_b$

Passo 4a: $P_a=BC=D.6$ e $P_b=BB=D.5$

Passo 4b: Novos padrões D.9=BCB, D.10=BCBB

Passo 4c: Novos padrões D.11=BBCB

Passo 4d: Enviar para a saída código/índice de $P_a=6$; $P_a=P_b$

Passo 5a: $P_a=BB=D.5$ e $P_b=D=D.4$
Passo 5b: Novos padrões $D.12=BBD$
Passo 5c: Novos padrões $D.13=DBB$
Passo 5d: Enviar para a saída código/índice de $P_a=5$; $P_a=P_b$

Passo 6a: $P_a=D=D.4$ e $P_b=BCB=D.9$
Passo 6b: Novos padrões $D.14=DB$, $D.15=DBC$, $D.16=DBC B$
Passo 6c: Novos padrões $D.17=BD$, $D.18=CBD$, $D.19=BCBD$
Passo 6d: Enviar para a saída código/índice de $P_a=4$; $P_a=P_b$

Passo 7a: $P_a=BCB=D.9$ e $P_b=A=D.1$
Passo 7b: Novos padrões $D.20=BCBA$
Passo 7c: Novos padrões $D.21=ABCB$
Passo 7d: Enviar para a saída código/índice de $P_a=9$; $P_a=P_b$

Passo 8a: $P_a=A=D.1$ e $P_b=B=D.2$
Passo 8b: Novos padrões $D.22=AB$
Passo 8c: Novos padrões $D.23=BA$
Passo 8d: Enviar para a saída código/índice de $P_a=1$; $P_a=P_b$

Passo 9a: $P_a=B=D.2$ e $P_b=AB=D.22$
Passo 9b: Novos padrões $D.24=BAB$
Passo 9c: -
Passo 9d: Enviar para a saída código/índice de $P_a=2$; $P_a=P_b$

Passo 10a: $P_a=AB=D.22$ e $P_b=B=D.2$
Passo 10b: Novos padrões $D.25=ABB$
Passo 10c: Novos padrões $D.26=BBA$
Passo 10d: Enviar para a saída código/índice de $P_a=22$; $P_a=P_b$

Passo 11a: $P_a=B=D.2$ e $P_b=*\text{NULL}/\text{END}*$
Passo 11b: -
Passo 11c: -
Passo 11d: Enviar para a saída código/índice de $P_a=2$; enviar código/índice de $*\text{END}*=*27*$

Sequência final de inteiros da codificação LZWdR: 2,2,3,6,5,4,9,1,2,22,*27*

4. Se codificar a sequência de códigos de *output* (valores inteiros dos índices dos padrões calculados na alínea anterior) numa sequência binária otimizada (menor número de bits possível), qual a sequência de bits que seria transmitida na linha, correspondente à digitalização e compressão LZWdR do bloco de 80 ms da figura 2? (10%)

Sequência final binária da codificação LZWdR:
01,01,010,101,100,0011,1000,00000,00001,10101,11010
ou, sem as vírgulas: 01010101011000011100000000000011010111010

Nota (i): na sequência binária utilizam-se valores a partir do zero, por isso os códigos/índices inteiros do dicionário serão enviados para o *output* como o valor binário equivalente menos 1.
Nota (ii): o número de bits mínimo para representar cada código/índice em cada passo é dado pelo número mínimo de bits necessários para representar o valor de código/índice-1 dois passos antes, sendo que o dicionário começa com 4 elementos, logo começa-se com um mínimo de dois bits.

5. Depois dos dados passarem pelo descompressor LZWdR, voltamos a obter uma sequência PCM igual à resultante da digitalização referida na questão 2. Utilizando codificação estatística Shannon-Fano para este bloco de dados e considerando que cada símbolo é um

grupo de K bits, qual a sequência binária resultante (sem considerar os metadados necessários)? (30%)

Sequência equivalente a codificar (alfabeto de 4 símbolos, sem blocos): BBCBCBDBCBABABB

i	S _i	Frequência*	Código	N _i
2	B	10	0	1
3	C	3	10	2
1	A	2	110	3
4	D	1	111	3

*Ou Probabilidade...

Sequência binária final: 0,0,10,0,10,0,0,111,0,10,0,110,0,110,0,0
ou, sem as vírgulas: 0010010001110100110011000

6. Tendo em consideração a codificação definida na questão anterior, qual a sequência binária necessária para representar os metadados (explique que informação contém os metadados e como os codificou em binário)? (10%)

Uma forma simples e eficiente seria codificar os metadados da forma canónica, i.e., indicar quantos bits (N_i) tem o código de cada símbolo S_i (ordenados por ordem de código i). O número símbolos dá-nos o valor máximo que o N_i pode assumir. No pior dos casos é o número de símbolos menos 1, ou seja, neste caso é 3. Então, podemos usar com segurança dois bits para representar o valor de N_i do código de cada símbolo S_i. Como um código tem, no mínimo, um bit, o valor binário que vamos enviar para o output é igual a N_i-1. Assim, a sequência canónica (N₁, N₂, N₃ e N₄), em binário, seria então: 10,00,01,10, ou, sem as vírgulas, 10000110.

Além disso, também se deveria acrescentar o número de símbolos codificados (para o descompressor saber quando parar). Se considerarmos que, neste caso, o bloco máximo poderia ser de 16 símbolos, então usaríamos 4 bits para representar esse valor: 1111.

Assim, a sequência binária final completa dos metadados poderia ser: 100001101111.

Algoritmo de Codificação LZWdR

Considere-se a sequência de símbolos $S=S_1S_2...S_N$ na entrada de dados, em que cada símbolo S_i pode assumir um de K valores possíveis dum alfabeto $A=\{X_1, X_2, ..., X_K\}$. Por conveniência, $S[i]=S_i$ e $A[i]=X_i$.
Defina-se um dicionário com um máximo de T padrões tal que $D=\{P_1, P_2, ...\}$ em que $D[i]=P_i$ é o padrão identificado pelo código/índice i. P_i^* representa a sequência de símbolos invertida de P_i , da direita para a esquerda. Valores típicos: $K=2^8$, $T=2^{20}$.
Inicia-se o dicionário de padrões D com os K padrões de símbolos individuais de A.
Processar S em pares consecutivos de padrões conhecidos $P_a|P_b$ (os maiores já existentes em D), acrescentando ao dicionário: i) os padrões novos formados por concatenação de P_a com todos os padrões que estão em P_b e ii) todos os padrões novos formados ao inverter os padrões obtidos em i).
Enviar para a saída o código/índice de P_a .
Voltar a processar S em pares consecutivos de padrões a partir do primeiro símbolo de P_b , i.e., $P_a=P_b$, repetindo os passos anteriores até não haver mais símbolos em S para processar depois de P_b .
Terminar enviando para a saída o código/índice de P_b .