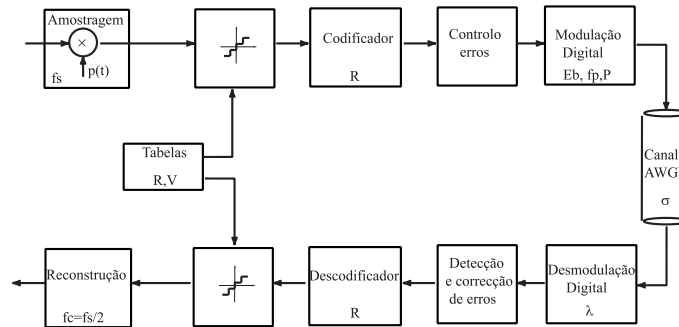


Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e Multimédia
Comunicação e Processamento de Sinais

1º Semestre de 2017/2018

A figura representa o esquema típico do processo de envio e recepção de informação usando modulações digitais. Durante o semestre pretende-se que sejam estudados e implementados os vários blocos funcionais do esquema.



Neste trabalho pretende-se que seja implementada a codificação e decodificação PCM uniforme (com R bits por amostra) e a codificação e decodificação de canal, usando códigos de Hamming. O código implementado para os pontos 1, 2, 3 e 4 e os cálculos para a resolução da pergunta 5 e 6 devem ser encapsulados num ficheiro ZIP com o número do grupo e submetidos no Moodle até 12 de Novembro de 2017.

1. Construa uma função que implemente a codificação PCM. Assuma que a entrada é um *Numpy array* (com dimensão N) com valores inteiros e o número de bits (R) a usar na conversão de cada inteiro. A saída deverá ser um *Numpy array* (com dimensão $N.R$) com os valores convertidos para binário, onde cada posição representa 1 bit representado no formato inteiro. Construa também a função que realiza a operação inversa, isto é, que dado um *Numpy array* com os bits, faça a conversão para inteiros. **OPCIONAL:** Implemente codificação de Gray, para tal adicione um argumento de entrada que defina o tipo de codificador.
2. Teste o codificador implementado usando o sinal de áudio previamente gravado. Verifique qual a SNR obtida e ouça o sinal decodificado. Use para o efeito $R=3, 5$ e 8 .
3. Construa uma função que implemente a codificação de canal usando um código de Hamming $H(15,11)$. A entrada deverá ser um *Numpy array* (com dimensão $N.R$). Construa também a função decodificadora que detecte e corrige possíveis erros.
4. Teste a codificação e decodificação de canal admitindo que o canal é simulado pela expressão " $y = (np.random.binomial(1, BER_t, len(x))) + x \% 2$ ", onde BER_t contém o valor pretendido para o bit error rate, x é *Numpy array* com os valores de entrada no canal e y os de saída. Meça SNR na recepção, o BER antes e após a correção de erros, para diferentes valores de BER_t .
5. Considere o sinal $y(t) = \sin(2\pi 3500t)$, amostrado a uma frequência de 8kHz.
 - a) Usando um quantificador midrise com $R=3$, faça a codificação PCM das 4 primeiras amostras do sinal.
 - b) Usando um quantificador não uniforme lei A , com $A = 87.56$ e $R=3$, faça a codificação das 4 primeiras amostras do sinal.
 - c) Faça a codificação DPCM ($R=3$), usando um quantificador midrise adequado, das 4 primeiras amostras do sinal.
6. Admita o bloco com a mensagem $[1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1]$.
 - a) Através do seguinte polinómio gerador, do código Hamming $H(15,11)$ ($x^4 + x + 1$), gere a mensagem com os bits de controlo de erros.
 - b) Admita que recebe a mensagem $[1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1]$. O que pode concluir?
 - c) Admitindo que o sistema transmissão funciona a 10Mbit/s com um BER inicial de 10^{-3} , qual o BER após a aplicação do controlo de erros?
 - d) Nas condições da alínea anterior, qual o tempo médio entre dois bits errados (com e sem detecção de erros)?