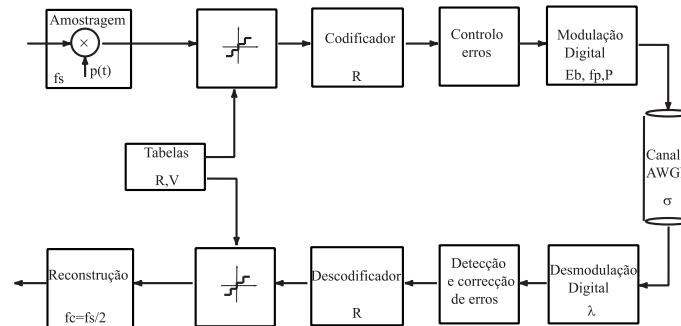


Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e Multimédia
Comunicação e Processamento de Sinais

1º Semestre de 2019/2020

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 primeiro guia de trabalho foca-se os blocos - **Amostragem, Reconstrução, Quantificação e Desquantificação**.

O código implementado para a resolução das perguntas práticas e um relatório que sumarie a implementação deve ser encapsulados num ficheiro ZIP com o número do grupo e submetidos no Moodle.

1. Admita que tem o sinal $20000\cos(2\pi 5050t) + 10000\sin(2\pi 2502t)$. Represente o seu espectro Amplitude e Fase.
2. Admitindo que faz a amostragem do sinal com uma frequência $f_s = 8\text{kHz}$, represente o espectro do sinal amostrado.
3. Gere o sinal em Python considerando que tem duração de um segundo.
 - a) Faça a amostragem do sinal com uma frequência $f_s = 44.1\text{kHz}$. Grave o sinal e reproduza o som. Apresente o seu espectro. Confirme os resultados teóricos da questão 1.
 - b) Faça a amostragem do sinal com uma frequência $f_s = 8\text{kHz}$. Grave o sinal e reproduza o som. Apresente o seu espectro. Confirme os resultados teóricos da questão 2. Tire conclusões.
4. Construa uma função em Python que crie as tabelas com os intervalos de decisão e valores de quantificação para um quantificador uniforme. Esta função tem como parâmetros de entrada o número de bits por amostra (R), o valor máximo a quantificar (Vmax) e o tipo de quantificador (midrise ou midtread). Como parâmetros de saída tem dois *Numpy arrays* com valores de quantificação e os intervalos de decisão.
5. Construa uma função em Python que dado um *Numpy array* com as amplitudes de um sinal amostrado, retorne um *Numpy array* com o sinal quantificado e um *Numpy array* com o índice dos valores de quantificação usados (vq).
6. Construa uma função que implemente a lei- μ .

7. Com as funções criadas nos pontos anteriores, e assumindo que o número de bits é $R = 3$:
- Quantifique um sinal de rampa (com e sem lei- μ).
Represente as amostras do sinal original e do sinal quantificado em função do tempo.
 - Represente o erro de quantificação em função do tempo e o seu histograma.
 - Para cada valor de $R=\{3,4,5,6,7,8\}$ meça a SNR e compare com o valor teórico. Construa um gráfico com os valores da SNR teórica e medida em função do número de bits R .
8. Grave um sinal de voz através do microfone com aproximadamente um segundo de duração. Faça a leitura do ficheiro de áudio previamente gravado para um *Numpy array*.
- Represente o histograma do sinal de áudio.
 - Represente cada amostra do sinal em função da amostra anterior. Comente o gráfico.
 - Para cada valor de $R=\{3,4,5,6,7,8\}$ Quantifique o sinal e meça a SNR (com e sem lei- μ). Ouça o sinal quantificado para os diferentes valores de R . Construa um gráfico com os valores da SNR em função do número de bits R . Analise o gráfico e tire conclusões.
9. Faça a leitura do ficheiro de imagem. Teste o código a baixo com vários valores de Q . Descreva o que produz o código.

```
from PIL import Image
Q = 5
im = Image.open("lena_color.tif")
im = im.quantize(Q)
im.show()
```