ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

ADEETC

Licenciatura em Engenharia

Informática e Multimédia

1º Semestre 15/16 SI

Comunicação e Processamento de Sinais

Relatório do Trabalho Final

Eng. José Nascimento

Trabalho realizado por:

Hugo Safara	nº40614
Rita Coelho	nº41154
Michael Madeira	nº41032

sábado, 16 de janeiro de 2016

Índice

Introdução

Neste trabalho final iremos concluir o processo da conversão de um sinal analógico para um sinal digital. O primeiro passo para a conversão começa com a amostragem, que tem como função transformar o sinal continuo num sinal discreto no domínio do tempo. De seguida, recorre-se à quantificação que basicamente torna o sinal discreto na amplitude transformando um número infinito num número finito de níveis (este processo introduz alguma distorção e perda de informação, o que é inevitável). O terceiro passo trata-se então da codificação, que atribui a cada amplitude discreta um código composto por um conjunto de bits. Estes são os três processos chave para garantir uma boa conversão de um sinal analógico para digital.

Desenvolvimento

O processo tem início na amostragem, como referido anteriormente, que se encarrega de transformar o sinal continuo num sinal descontínuo ao longo do tempo. No nosso projeto realizámos a amostragem de uma sinusoide, que fora amostrada a uma frequência de amostragem igual a 8000Hz e com uma amplitude igual a 20000, de banda limitada W igual a 3014Hz e com um passo de 1/8000 segundos (Período de Amostragem).

Verificámos que ao diminuir a frequência de amostragem para 4000Hz, ao fazer a reconstrução do sinal aplicando um filtro passa baixo, a frequência de corte (fc = fs/2) não irá respeitar o Teorema da Amostragem de Nyquist-Shannon, em que W <= fs/2, acontecerá o chamado Aliasing. Evitamos então a sobreposição espectral (Aliasing), tendo sempre a nossa frequência de amostragem maior ou igual à nossa frequência máxima do sinal.

A quantificação é o passo que faz com que se limite o número de amostras por período para que seja mais fácil quantificar cada amostra em L intervalos, de dimensão deltaQ, por aproximação a um valor de quantificação que torna o sinal discreto na amplitude.

A conversão analógico-digital provoca distorção por quantificação (ruído de quantificação). Para menor ruído de quantificação, os valores de decisão devem estar equidistantes dos valores de quantificação (Quantificação Uniforme). A relação entre a potência do sinal original e a potência do ruído de quantificação é uma das medidas de qualidade mais utilizadas (SNR = 6.02R + 10 log10(3P/V^2)). Para ter uma melhor relação sinal-ruído, podemos aumentar o número de bits por amostra R ou diminuir a tensão máxima de quantificação, sendo a opção mais adequada diminuir a tensão máxima V, mas continuando a respeitar a condição de ser sempre maior ou igual à amplitude máxima do sinal (V>=mmáx), devido à saturação que se poderia produzir.

Neste ponto utilizámos um quantificador uniforme *Midrise*, para quantificar cada amostra do processo anterior, sendo que para este tipo de quantificador, os valores de quantificação não são coincidentes com os extremos e o 0 conta como valor de decisão. Como o nosso o número de bits por amostra R é igual a 3, o quantificador tem 8 intervalos, e uma tensão máxima de quantificação de 20005 para não haver saturação de amplitudes.

A codificação é a representação binária da sequência de valores de um sinal após a quantificação, sendo que o número de bits de codificação de cada segundo do sinal, débito binário (Rb), é o produto entre o número de bits por amostra R e a frequência de amostragem. A codificação amostra-a-amostra que utilizamos no nosso sinal denomina-se PCM(Pulse Code Modulation), em que cada valor de quantificação foi representado por 3 bits por amostra, de acordo com a ordem de valores de quantificação na Tabela.

O código de Hamming é um código linear utilizado para a deteção de erros na transmissão de dados binários e é também capaz de corrigir esses mesmos erros. Este código adiciona um bloco de paridade a um bloco de dados, para que, caso ocorram erros de transmissão, seja possível detetar e/ou corrigir os erros do bloco transmitido.

Na modulação digital, nomeadamente na QAM, o processo em si consiste numa operação realizada sobre o sinal ou dados a transmitir e que produz um sinal apropriado para a transmissão sobre o meio de transmissão em causa. A escolha da técnica de modulação permite "moldar" as características do sinal a transmitir e adaptá-lo às características do canal.

Concluindo, o nosso sinal vai então passar por um canal, denominado de AWGN (Additive White Gaussian Noise). Este canal permite-nos simular um canal de comunicações e então permitir a propagação do sinal. No entanto, depois da realização de todos estes processos, é necessário que o recetor faça exatamente os processos contrários de modo a que consiga receber o sinal. Para isso terá que se desmodular o sinal, detetar e corrigir os erros nele existentes, descodificar o mesmo, transformando o mesmo de novo no conjunto de amostras que ele era, fazendo a quantificação inversa e a reconstrução do sinal. No entanto, e apesar de realizar todo este processo de maneira correta, o sinal chegará sempre ao recetor com erros, visto ser impossível de realizar o conversor analógico digital perfeito.

Implementação

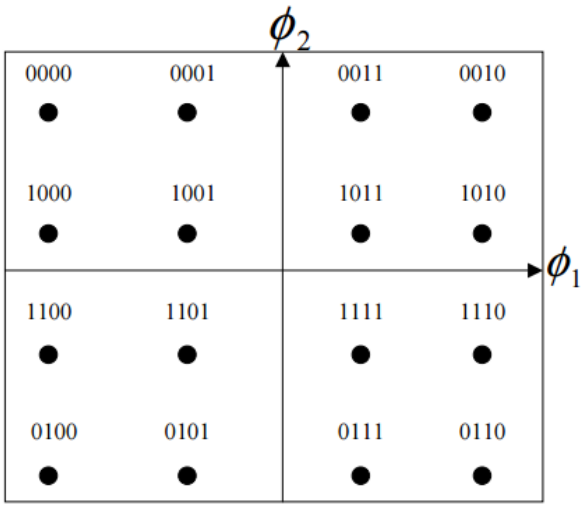
Para simularmos uma onda sinusoidal que irá ser usada para converter um sinal analógico para digital, é necessário recorrer-se ao uso do cosseno que se trata de uma função trigonométrica que permite gerar uma "onda". Essa onda terá uma amplitude (A), uma frequência (f) e um período (T). De seguida, faz-se a amostragem do sinal com uma respetiva frequência de amostragem (Fs) e período de amostragem (Ts). Este processo permite mostrar o sinal num número finito de amostras e representá-las ao longo do tempo, estando assim, a transformar o sinal contínuo em descontínuo ao longo do tempo.

Na etapa de quantificação, são gerados os intervalos de decisão (valD) e os valores de quantificação (valQ). Os valores de decisão e de quantificação variam de acordo com o número de intervalo de bits que estejamos fornecer (R). De acordo com os valores do intervalo de quantificação, para cada amostra do sinal, designaremos um novo valor, valor este, que definimos previamente no intervalo de quantificação. Ou seja, valores das amostras correspondentes ao sinal que sejam quase iguais aos valores no intervalo de quantificação, tomam o valor corresponde ao valor no intervalo. A função encarregue deste processo é a quantificação (sinal, valoresQ, intervalos). Esta função retorna os valores quantificados (xq) e os índices dos valores de quantificação (índices).

Na codificação, transforma-mos os índices para valores binários. Nesta conversão, recorre-se ao uso do *slice* para fazer desaparecer os caracteres 'Ob' que aparecem no início. O processo de descodificação baseia-se no processo inverso, transforma-mos de binário para inteiro.

O código de Hamming permite a deteção do erro. Percorre-se o sinal de 11 em 11 bits. A estes blocos iremos adicionar 4 bits de paridade. Bits de paridade esses que serão importantes para mais tarde conseguir-se corrigir os bits errados nos blocos. A função Hamming_correcao permite detetar e corrigir bits que se encontram ao longo desses blocos de 11 bits.

Na modulação QAM os símbolos são tratados separadamente por duas portadoras, os de índice par por uma portadora cosinusoidal e os de índice ímpar por uma portadora sinusoidal sendo que as duas componentes obtidas são definidas como a componente em fase e as de componente em quadratura. No espaço de sinal o mapeamento da combinação destas duas componentes é feito por intermédio de uma constelação como vemos na figura seguinte:



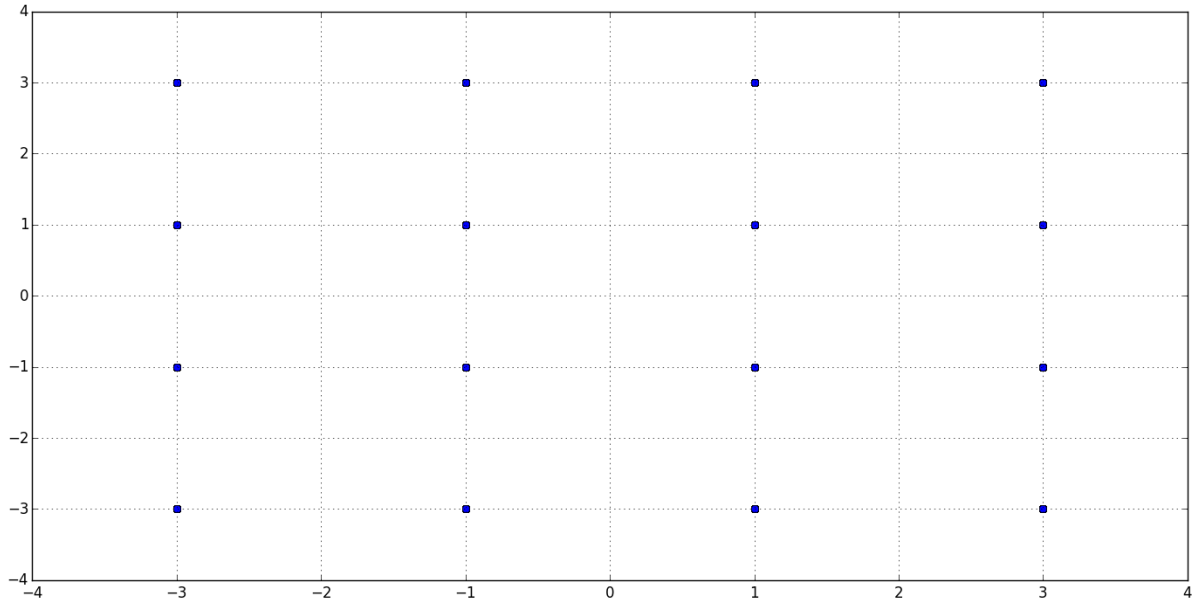
Segundo o eixo das abcissas serão mapeados os bits modulados no canal em fase e no eixo das ordenadas os bits modulados no canal em quadratura. O pormenor a notar é que os bits não estão ordenados de uma forma aleatória, eles estão mapeados de acordo com o código Gray por forma a minimizar o erro cometido pois assim a maior probabilidade de erro é de um bit pois as amplitudes "vizinhas" apenas diferem de um bit entre si, ou seja se se cometer um erro na sua deteção está-se apenas a errar um bit.

A função QAM_16 retorna então as amplitudes relativas dos símbolos de acordo com a constelação previamente representada. Podendo os símbolos tomar as posições de (-3,3), (-3,1), (-3,-3), (-3,-1), (3,3), (3,-3), (3,1), (3,-1), (1,3), (1,-1), (1,-3), (1,1), (-1,3), (-1,-3), (-1,1), (-1,-1).

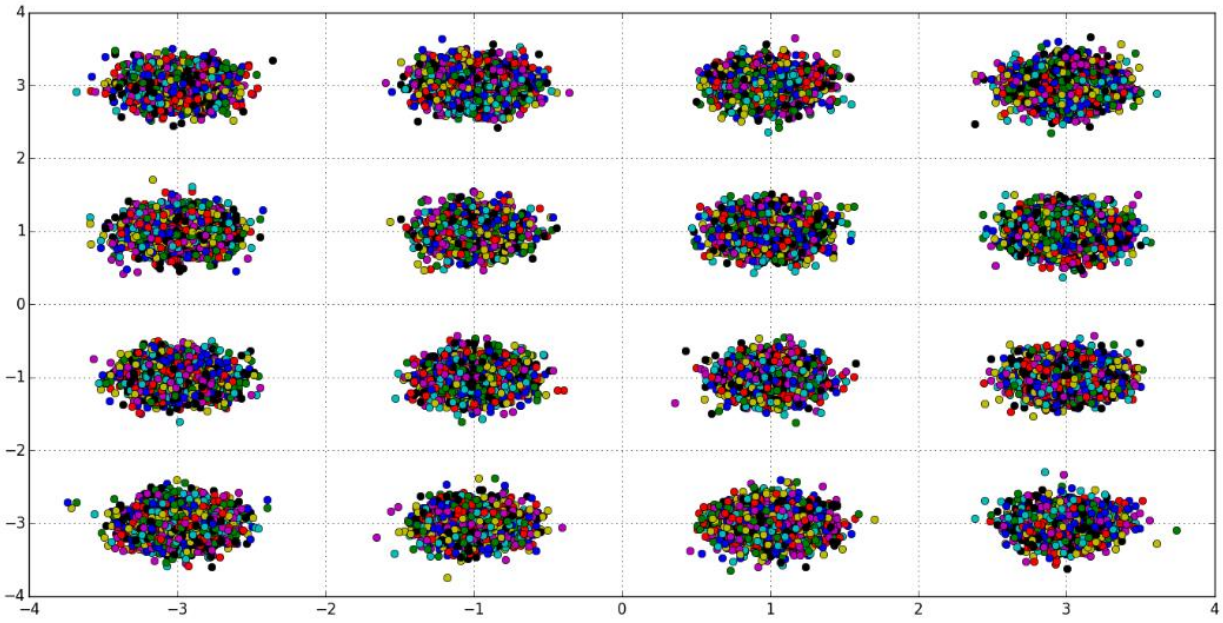
Na desmodulação, depois do sinal ruidoso sair do canal AWGN, multiplica-se o sinal ruidoso por duas ondas portadoras novamente, uma cosinusoidal e sinusoidal, para que seja possível fazer o processo inverso das amplitudes discretas dos símbolos para um conteúdo binário. A função des_QAM_16 basicamente trata de rearranjar as amplitudes do sinal ruidoso para amplitudes discretas corretas que pertencem à constelação. Com esses valores rearranjados, trata-se de retornar o valor binário correto. Esses valores de amplitude servirão também de índices para percorrer o ArrayBits que irá permitir retornar o valor binário.

Resultados

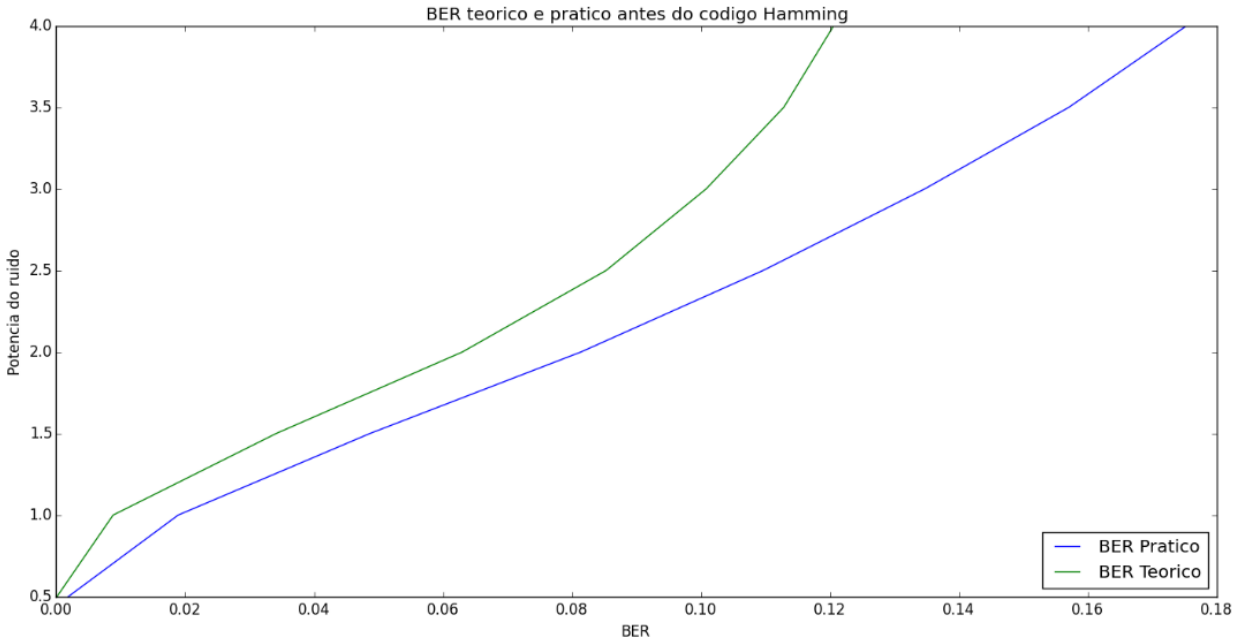
Constelação da sinusoide com a modulação 16-QAM (Potência de ruído = 0):

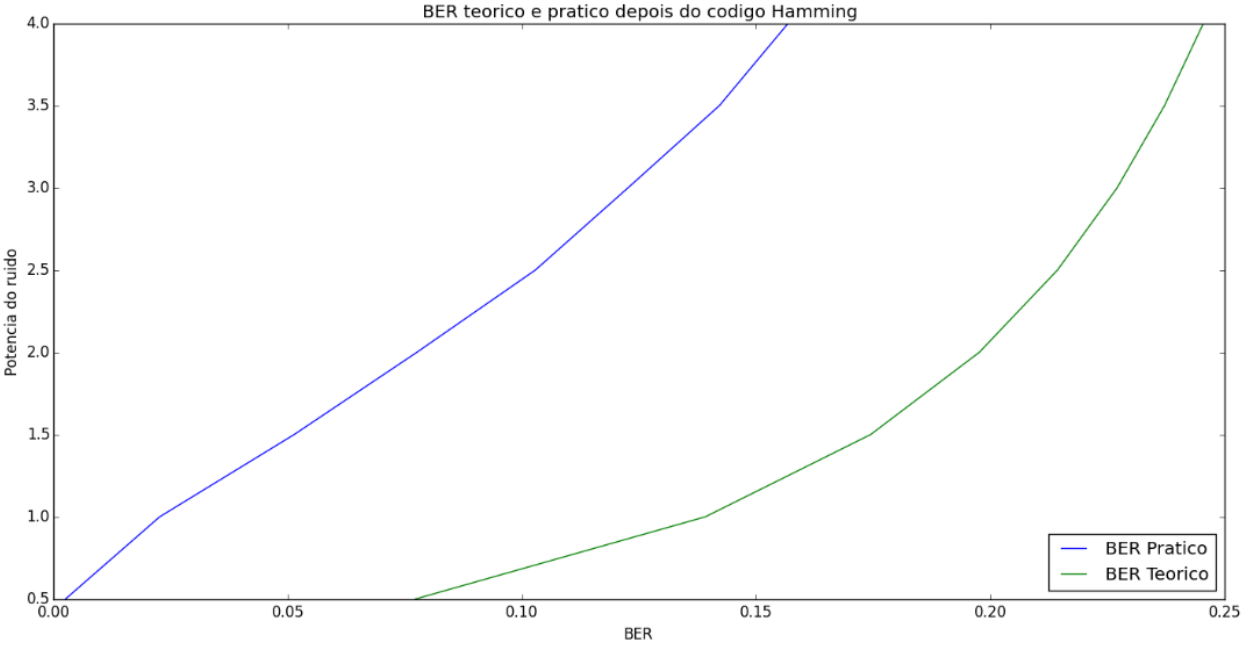


Constelação da sinusoide com a modulação 16-QAM (Potência de ruído = 0.03):



Com o sinal de áudio, apresenta-mos o BER teórico e prático antes e depois da correção:





Com potência 0 o canal não tem ruído, logo o ficheiro é ouvido tal e qual como é enviado. Para 0.5 há muito pouco ruído, o que ainda não compromete em massa a perceção do código. Para potência 3 o ruído já é considerável mas o som ainda é percetível. Para ruídos superiores o som passa a ter cada vez mais ruído, até que chega ao ruído de potência 10 e o som passa a não ser percetível.

Conclusão

Apesar de todo este processo, e seguindo minuciosamente todos os parâmetros estabelecidos nos trabalhos das aulas, passando o som por todo o conversor pode-se constatar que este não é exatamente igual ao som inicial, ou seja, existem erros que ou não são detetados ou não são corrigidos e que alteram a qualidade do som. Aliás, neste tipo de transmissão de sinal existirá sempre ruído agregado a este processo, o que também influenciará a pureza que com a qual o sinal é ouvido no final da conversão.

Tendo em conta estas condições, e agora referindo o código por nós implementado, podemos constatar que este ruído se encontra mesmo presente, aplicando ruído no canal e dando-nos uma ideia de que o nosso corretor não consegue detetar nem corrigir todos os erros.

Apesar de termos, com sucesso, conseguido fazer com que um sinal passasse pelo canal e no final conseguirmos comparar as principais diferenças entre o sinal de áudio original e o final, houve alguns aspetos que não conseguimos realizar neste trabalho. Nomeadamente, com a falta de realização das SNR's no canal e no recetor no código, não nos permite a 100% tirar conclusões relativamente a todo o processo deste trabalho final.

Existiram, também, algumas dificuldades na realização da constelação em Python. Criou-se uma função, embora não fosse a melhor função para a construção da constelação, que só permite construir uma constelação com a sinusóide. Com o sinal de voz, dá erro no programa. Colocámos um ruido de 0.03 para se ter uma boa imagem do ruido na constelação e como as amplitudes do sinal modulado com erros variam. Acima deste valor, denotam-se "nuvens" muito grandes que não nos permite ver com clareza a constelação.