

1º Semestre 16/17 SI

Comunicação e Processamento de Sinais

**Relatório do 4º Trabalho Prático**

Eng. Pedro Fazenda

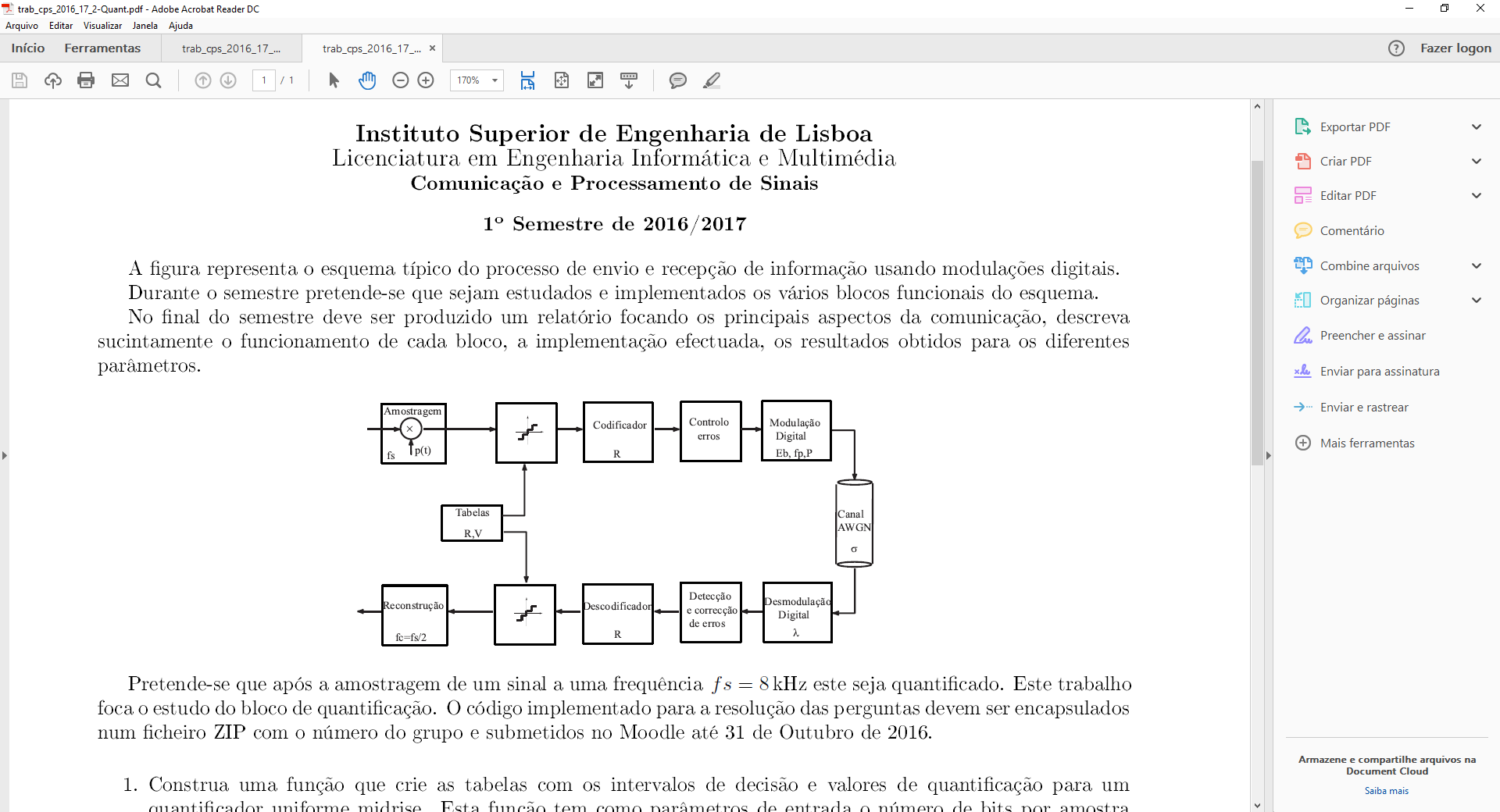
16/12/2016

**Trabalho realizado por:**

Diogo Fernandes nº39205

Rui Santos nº39286

**Introdução**



Este quarto trabalho foca-se nos blocos de Modulação digital e suas operações inversas, mas neste mesmo iremos utilizar Código Linha PRZ, iremos também utilizar código desenvolvido nos trabalhos anteriores para executar testes que pretendemos neste trabalho.

Exercícios

1. Construa uma função, para o emissor, em que dado um array de bits retorne um array com o código PRZ da mesma sequência. Esta função ainda deve receber como parâmetros de entrada o número par de pontos (amostras) por cada bit (P) e a amplitude do código.

**def** codigoPRZ**(**arrayBits**,**P**,**A**):**

#sinal codificado, numero de amostras por bit, amplitude do codigo

arrayPRZ **=** np**.**zeros**(**len**(**arrayBits**)\***P**)**

aux **=** int**(**P**/**2**)**

**for** i **in** range**(**len**(**arrayBits**)):**

current **=** arrayPRZ**[**i**\***P**:(**i**+**1**)\***P**]**

**if(**arrayBits**[**i**]==**0**):**

current**[:**aux**]** **=** **-**A

current**[**aux**:]** **=** 0

**else:**

current**[:**aux**]** **=** A

current**[**aux**:]** **=** 0

**return** arrayPRZ**.**astype**(**"int16"**)**

**2.**Construa uma função que simule, no recetor, um filtro adaptado. Este deve receber um código de linha com ruído e deve retornar uma sequência binária. Esta função deve ter também como parâmetro de entrada o valor de limiar de decisão().

**def** adaptiveFilter**(**codLinha**,** lim**,** P**):**

filtered **=** np**.**zeros**(**int**(**len**(**codLinha**)/**P**))**

#vetor base

c **=** np**.**ones**(**P**)\*(**1**/**np**.**sqrt**(**P**))**

aux **=** int**(**P**/**2**)**

c**[**aux**:]\*=-**1

**for** i **in** range**(**len**(**filtered**)):**

value **=** sum**(**c**\***codLinha**[**i**\***P**:(**i**+**1**)\***P**])**

**if(**value **>** lim**):**

filtered**[**i**]=**1

**else:**

filtered**[**i**]=**0

**return** filtered**.**astype**(**"int16"**)**

**3.**Construa uma função que simule um canal AWGN. Esta função recebe ainda o código linha do emissor, adiciona ruído (com determinada potência) e devolve o código com ruído num array. A função deve ter como parâmetro de entrada a potência do ruído adicionar.

**def** canalAWGN**(**signal**,**noisePower**):**

signalOut **=** signal **+** np**.**sqrt**(**noisePower**)\***np**.**random**.**randn**(**len**(**signal**))**

**return** signalOut

**4.**Com base nas funções elaboradas, simule a emissão para o canal de uma sequência binária e a sua receção.

**a)** Admita que os bits à entrada no emissor são [0,1,1,0,0,1].  
Gere o código de linha correspondente (P=8,A=1), transmita o sinal pelo canal () e descodifique o código linha (). Represente o sinal à entrada e saída do canal. Compare a sequência de bits descodificada com a original.

array inicial: [0, 1, 1, 0, 0, 1]

array PRZ: [-1 -1 -1 -1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 -1

-1 -1 -1 0 0 0 0 -1 -1 -1 -1 0 0 0 0 1 1 1 1 0 0 0 0]

array Canal: [-2.6708868 -0.11399331 -3.28406686 -1.31132921 -0.16704936 0.46497778

1.22371501 -1.45119862 0.36795999 0.86464372 1.10813696 2.79101457

0.05715173 0.51747266 1.78392637 0.29145232 0.62995818 0.2599314

1.88021596 2.5705678 -0.746663 -0.01897815 -0.06393129 -0.70130314

-1.63838877 -0.40937385 -0.85851312 -0.03620316 2.03312267 0.05958375

0.51412244 0.16007635 -1.57294667 -1.5146073 -2.23791967 -1.90831945

2.35126813 -0.64532134 1.49620573 -0.24865796 0.88877418 1.50671961

0.99001112 1.36669012 1.74377898 -2.17902479 0.56846846 -0.50741368]

array Filtrado: [0 1 1 0 0 1]

Podemos observar pelos outputs do nosso código que com uma potência de ruído baixa, o nosso sinal à entrada do emissor e à saída, não apresenta alterações.  
Mas como o nosso o nosso canal está a ser simulado com um ruído dependente de uma função random, por vezes podemos obter diferenças nos outputs mesmo usando os valores iguais ao anterior.

Como podemos observar neste seguinte exemplo:

array inicial: [0, 1, 1, 0, 0, 1]

array PRZ: [-1 -1 -1 -1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 -1

-1 -1 -1 0 0 0 0 -1 -1 -1 -1 0 0 0 0 1 1 1 1 0 0 0 0]

array Canal: [-0.53749253 -0.44025818 -1.8642982 -1.50665253 1.23745658 2.14243317

0.0823528 -0.6370765 -0.22814083 1.6869125 1.51699705 2.48862225

1.93427762 -1.99788497 1.36583109 -0.20763745 0.25236772 1.48955405

0.49373837 -0.16822409 0.00391469 0.57349331 -1.05959211 0.19415037

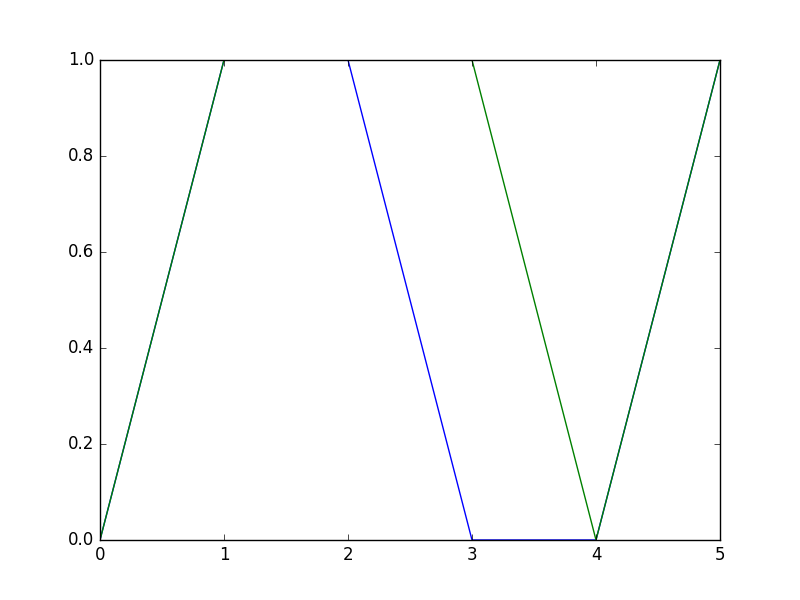
-1.23298789 0.64645049 -0.96571979 0.898754 0.10718433 -0.70238134

-0.49760381 0.01460266 -2.6976326 0.0041091 -1.20795624 -1.66585742

-1.25992643 -0.88568292 0.0104213 1.33736607 0.90598151 3.43239031

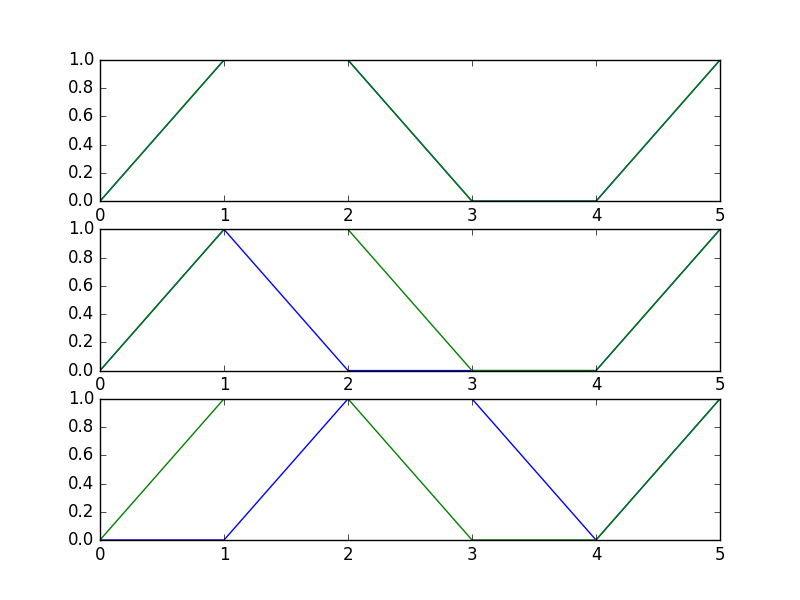
0.79352115 1.39509139 1.37223766 1.09157609 0.27694223 -0.06942042]

array Filtrado: [0 1 1 1 0 1]



**b)** Repita a alínea anterior para cada valor do parâmetro

Observando a imagem com os gráficos com as comparações do sinal de entrada e saída das potencias de ruido diferentes:



Concluímos que o aumento da potencia de ruido, provoca alterações no resultado final.

**5.**Considerando o sinal previamente gravado, gere o código linha para este (P=8,A=1), transmita o sinal pelo canal e descodifique-o. Considere diferentes valores de potência para o ruido e elabore um gráfico com o BER medido e o BER teórico. Comente os resultados. Calcule também a SNR obtida.

array ber teoricos: [ 0.0045271 0.04342696 0.1387408 0.24999563]

array ber pratico: [ 0.00429769 0.04297688 0.13916249 0.25071883]

array SNR Pratica: [ 12.9678531 2.85366244 -2.23449307 -4.75498453]

Os valores de output da nossa simulação, mostram que aumentando o ruido do canal, obtemos valores de BER mais elevados mesmo que valores baixos.  
Supostamente as SNR’s deviam ficar mais elevadas com atribuição de mais ruido, o primeiro valor de SNR é o considerado aceitável.

**6.**Considere um sistema de transmissão que usa o código de linha PRZ, com um factor de rol-off de 0.5, sabe-se que o canal de comunicação (AWGN) é do tipo passa-baixo, cujo ruido tem uma densidade espectral de potencia 0.5x10-6 W/Hz, uma atenuação de 5Db E UMA LARGURA DE BANDA DISPONIVEL DE 400KHZ. Pretende-se que o receptor tenha um BER igual ou inferior a 10-5.

**a)** Determine o débito binário máximo do sistema.

**b)** Determine a energia por bit no receptor que satisfaça as condições pretendidas.

Utilizando a expressão de calculo do BER do código linha PRZ:

Conseguimos deduzir:

**c)**Determine a potencia recebida (apos o canal) nas condições da alínea anterior

**d)**Determine a potencia mínima que deve ser emitida