CADERNO DE PRÁTICAS EM MATLAB
IF65F - Transmissão de Dados

Prof. Dr. Bruno A. Angélico

2012

Prática 1: PCM - *Pulse Code Modulation*

Na modulação PCM, as amostras são convertidas para códigos digitais antes de serem transmitidas. A transmissão digital é muito mais inume a ruídos do que a analógica. O processo de digitalização usando PCM segue três etapas:

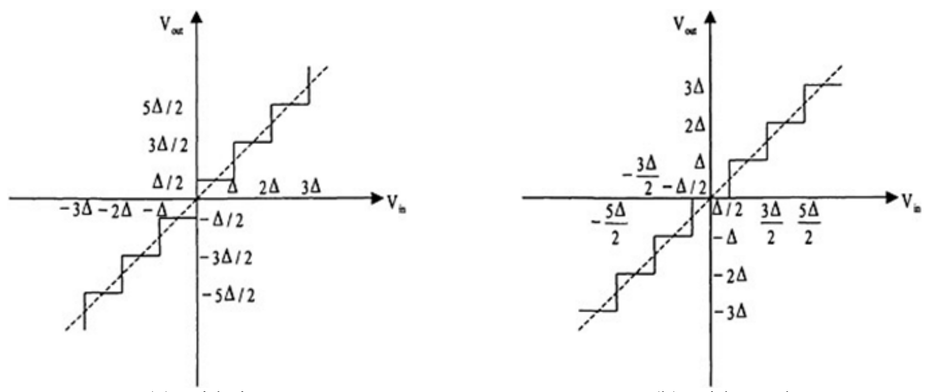
1. Amostragem do sinal analógico;

2. Quantização dos valores;

3. Codificação dos valores em binário.

A técnica PCM atribui bits de maneira uniforme para todas as amplitudes do sinal que se digitaliza. Há basicamente duas formas de quantizadores uniformes

$$
\begin{aligned}
midrise: \ q(x) &= \Delta \cdot \left(\left\lfloor \frac{x}{\Delta} \right\rfloor + \frac{1}{2} \right) \\
midtread: \ q(x) &= \mathrm{sgn}(x) \cdot \Delta \cdot \left\lfloor \frac{|x|}{\Delta} + \frac{1}{2} \right\rfloor
\end{aligned}
$$



(a) mid-rise (b) mid-tread

Figura 1: Tipos de quantizadores uniformes.

Uma forma de amenizar os efeitos do ruído de quantização para baixas amplitudes consiste na utilização de técnicas de Compansão.

Compansão é uma técnica de compressão e expansão de sinais para efeitos de redução de ruídos e quantização.

O processo de compressão é logarítmico. A compressão aumenta de forma logarítmica a medida em que as amostras do sinal aumentam. Quanto maior o valor da amostra, maior o grau de compressão. Isto causa um ruído de quantização que cresce a medida em que a amostra do sinal também cresce, sendo mais baixo para baixas amplitudes, se comparado à quantização uniforme.

Foram propostas duas maneiras de se alocar bits de forma não uniforme para representar sinais digitais: a Lei μ (EUA) e a Lei A (Europa - adotada no Brasil). A formulação destas leis se dá conforme apresentado a seguir:

- Lei μ

$$
V_o = \frac{\log_{10}(1 + \mu V_i)}{\log_{10}(1 + \mu)}, \ \ 0 < V_i, \ V_o \leq 1 \tag{1}
$$

O valor de compressão pode variar conforme μ, que é normalmente entre 100 e 255,

- Lei A

$$\begin{cases} V_o = \frac{AV_i}{1+\ln(A)}, & 0 < V_i \leq 1/A \\ V_o = \frac{1+\ln(AV_i)}{1+\ln(A)}, & 1/A \leq V_i \leq 1 \end{cases} \tag{2}$$

O valor $A = 87,6$ é geralmente utilizado (recomendado pelo CCITT).

Para avaliar o efeito da quantização, o arquivo `MA.WAV` será considerado, que representa uma voz masculina para a frase *Why were you away a year Roy*?, frequentemente utilizada como *benchmark* na área de processamento de sinais de voz. O arquivo `Teste_PCM_lin.m` efetua a quantização uniforme *midrise* e gera os gráficos do sinal original, do sinal quantizado, assim como do erro de quantização.

Atividades:

1. Teste o programa, varie o número de bits de quantização e verifique o que acontece com a qualidade do sinal;

2. Implemente a quantização linear *midtread*;

3. Implemente a lei-μ de compansão com $\mu = 255$ e compare com o sinal quantizado de forma linear *midrise*. O que acontece com o ruído de quantização?

4. Faça uma pesquisa sobre a relação sinal ruído (SNR) do sinal quantizado utilizando quantizador linear e a lei-μ de compansão com $\mu = 255$.

Prática 2: Formatação de Pulso e Filtro Casado

Em canais limitados em banda é preciso ter uma formatação de pulsos de tal forma que a largura banda do pulso transmitido esteja de acordo com a bana limitada do canal. Para canais sem distorção dentro de uma banda finita, o pulso sinc é ótimo para evitar interferência intersimbólica (ISI). No entanto, tal pulso não é fisicamente realizável. Como alternativa, tem-se o pulso cosseno levantado (*raised cosine*) com fator de *rollof* α, cuja resposta em frequência é dado por

$$G_{rc}(f) = \begin{cases} T, & 0 \leq f \leq \frac{(1-\alpha)}{2T} \\ \frac{T}{2}\left[1+\cos\left(\frac{\pi T}{\alpha}\left\{f - \frac{1-\alpha}{2T}\right\}\right)\right], & \frac{(1-\alpha)}{2T} \leq f \leq \frac{(1+\alpha)}{2T} \\ 0, & f > \frac{(1+\alpha)}{2T} \end{cases}$$

No domínio do tempo, tem-se a seguinte resposta impulsiva:

$$g_{rc}(t) = \mathrm{sinc}(t/T)\frac{\cos(\pi\alpha t/T)}{1-4\alpha^2 t^2/T^2}$$

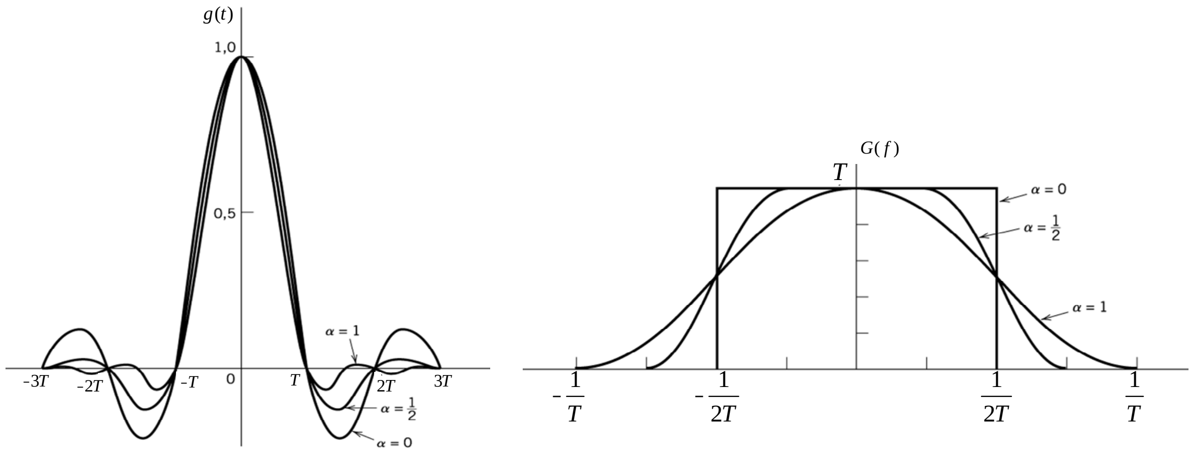


Figura 2: Pulso cosseno levantado e espectro de amplitude.

No caso especial de canal sem distorção, tem-se que

$$G_{rc}(f) = G_T(f)G_R(f)$$

, ou seja,

$$G_T(f) = \sqrt{X_{rc}(f)}e^{-j2\pi f t_0}$$

onde t_0 é o atraso necessário para tornar o filtro fisicamente realizável (nesse caso, para deixar o filtro causal).

A resposta impulsiva do filtro transmissor (RRC - *Root Raised Cosine*) pode ser escrita como

$$g_T(t) = \begin{cases} \frac{1}{\sqrt{T}}\frac{\sin[\pi(1-\alpha)t/T]+(4\alpha t/T)\cos[\pi(1+\alpha)t/T]}{(\pi t/T)\left[1-(4\alpha t/T)^2\right]}, & t \neq 0, t \neq \frac{T}{4\alpha} \\ \frac{1}{\sqrt{T}}\left[1-\alpha+\frac{4\alpha}{\pi}\right], & t = 0 \\ \frac{\alpha}{\sqrt{2T}}\left[\left(1+\frac{2}{\pi}\right)\sin\left(\frac{\pi}{4\alpha}\right)+\left(1-\frac{2}{\pi}\right)\cos\left(\frac{\pi}{4\alpha}\right)\right], & t = \pm\frac{T}{4\alpha} \end{cases}$$

Um filtro que é casado (*matched filter*) com o pulso $s(t)$, $0 \leq t \leq T$, possui resposta impulsiva dada por

$$h(t) = g(T-t), \quad 0 \leq t \leq T \tag{3}$$

Dessa forma, a saída do filtro casado a $g(t)$, quando a entrada é $g(t)$, é dada por

$$y(t) = \int_0^t s(\tau)h(t-\tau)\,d\tau = \int_0^t g(\tau)g(T-t+\tau)\,d\tau \tag{4}$$

Se tal saída for amostrada em $t = T$, tem-se

$$y(T) = \int_0^T g^2(\tau)\,d\tau = \mathcal{E}, \tag{5}$$

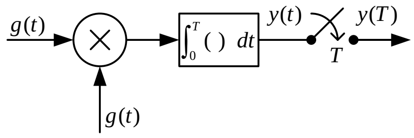


Figura 3: Correlacionador e amostrador.

onde $\mathcal{E}$ é a energia do sinal. Considere agora a Figura seguinte,

Note que

$$y(T) = \int_0^T g^2(t)\, dt = \mathcal{E}, \tag{6}$$

Suponha que $g(t)$ seja o sinal recebido e seja dado por

$$r(t) = g(t) + \eta(t), \tag{7}$$

onde $\eta(t)$ é o ruído AWGN. A saída do filtro casado é dada por

$$y = \mathcal{E} + \nu \tag{8}$$

onde

$$\nu = \int_0^T \eta(t) g(t)\, dt \tag{9}$$

representa o ruído na saída do filtro casado. Como $\eta(t)$ tem media zero, ν também possui média zero. Pode-se mostrar que a variância de ν é dada por

$$\sigma^2 = \frac{N_0 \mathcal{E}}{2} \tag{10}$$

No arquivo `rtrcpulse.m` é implementada uma função para gerar o pulso RRC. O arquivo `MF_teste.m` traz um exemplo de implementação de um sistema de comunicação com filtro RRC no transmissor e no receptor, sujeito a canal AWGN. O filtro casado é implementado por meio de um correlacionador. A partir desse exemplo, a seguinte análise é inicialmente feita.

Uma sequência de 10 bits é gerada de forma equiprovável. Para uma execução particular, a seguinte sequência é gerada: b = [1 1 0 1 1 0 1 1 0 1].

O sinal é posteriormente passado por um filtro formatador RRC (com energia normalizada), com sinalização antipodal (b_{pol} = [-1 -1 1 -1 -1 1 -1 -1 1 -1]), gerando o sinal apresentado na Figura seguinte

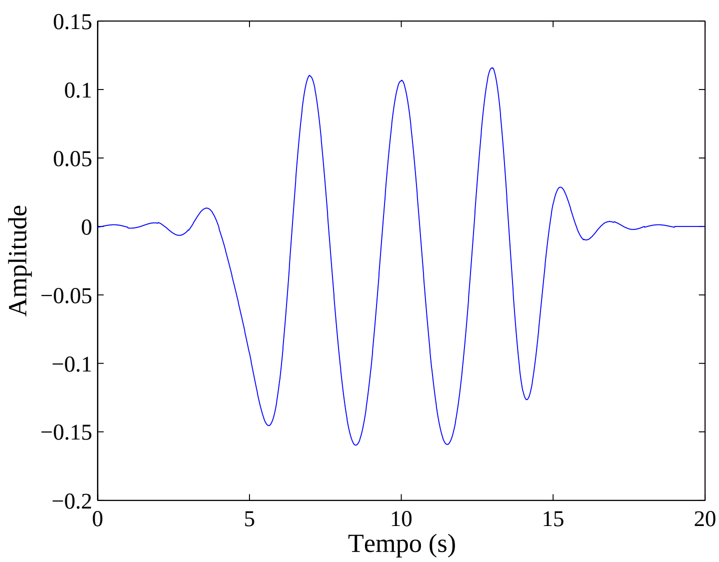


Figura 4: Sinal gerado com formatação de pulso RRC, sinalização polar NRZ, para a sequência de bits b = [1 1 0 1 1 0 1 1 0 1].

Se for assumido canal AWGN e $E_b/N_0 = 3$ dB, tem-se o seguinte sinal recebido

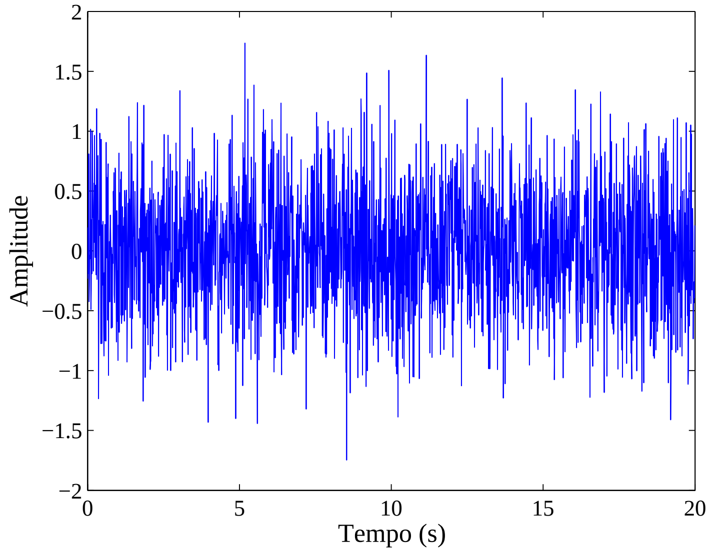


Figura 5: Sinal recebido corrompido por ruído AWGN com $E_b/N_0 = 3$ dB.

Pergunta: é possível detectar o sinal transmitido a partir desse sinal recedido?

Após a implementação do filtro casado no receptor por meio do correlacionador, e a amostragem na taxa de símbolos, veja o que ocorre

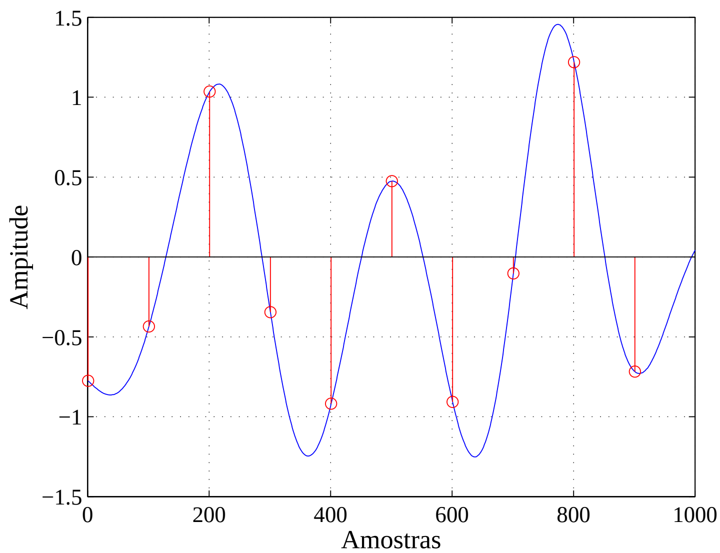


Figura 6: Saída do correlacionador (em azul) e saída do amostrador na taxa de símbolos (vermelho).

Se a saída do correlacionador for comparada com o limiar em zero o a seguinte informação é estimada: $\hat{\mathrm{b}}_{\mathrm{pol}}$ = [-1 -1 1 -1 -1 1 -1 -1 1 -1]. Verifica-se que, nesse caso em particular, a informação foi recuperada. Nem sempre todos serão perfeitamente estimados. Deve-se observar a probabilidade de erro em sinalização antipodal em canal AWGN, em função de E_b/N_0 em dB, dada por:

$$\begin{aligned} P_e = P(y<0) &= \int_{-\infty}^{0} p\,(y|0)\,dr = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{0} e^{-(y-\mathcal{E})^2/2\sigma^2}dy \\ &= \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\mathcal{E}/\sigma} e^{-y^2/2}\,dr = Q\left(\frac{\mathcal{E}}{\sigma}\right) = Q\left(\sqrt{\frac{2\mathcal{E}}{N_0}}\right) \end{aligned} \tag{11}$$

O Matlab nao implementa a função $Q(x)$, mas tem a função `erfc(`x`)`. A relação entre elas é dada por

$$Q(x) = 0,5 \cdot \texttt{erfc}\left(\frac{x}{\sqrt{2}}\right) \tag{12}$$

Atividades:

1. Crie uma função em Matlab para gerar a formatação de pulso cosseno levantado, com parâmetros de entrada t, T e α;
2. Simule o arquivo `Teste_MF.m` com uma sequência de 100 bits e $E_b/N_0 = 100$ dB. Apresente a curva dos sinais na saída do correlacionador e na saída do amostrador. Quantos erros foram obtidos?
3. Refaça o exemplo anterior para $E_b/N_0 = -20$ dB;
4. Crie uma função em Matlab, denominada `qf.m`, que implemente a função $Q(x)$;
5. Faça 10 simulações do arquivo `Teste_MF.m` com $E_b/N_0 = 3$ dB, com sequência de 500 bits em cada simulação. Faça uma tabela com o número da simulação e a quantidade de erros gerada em cada simulação. Calcule a probabilidade de erro, considerando todas as simulações, dividindo o número total de erros pelo número total de bits transmitidos. Compara com o valor teórico.

Prática 3: Probabilidade de Erro em Transmissão Antipodal em Canal AWGN
Realizada como exemplo pelo professor.

Prática 4: Probabilidade de Erro em Transmissão M-ASK em Canal AWGN

Nesta prática será efetuada a simulação de um sistema de comunicação M-ASK, com $M = 4$, sujeito a canal AWGN. A constelação a ser utilizada é representada na Figura 7.

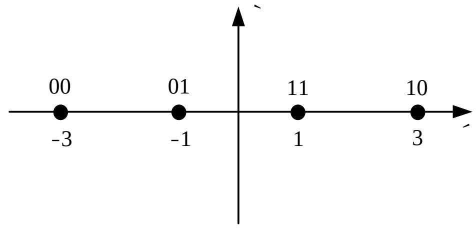


Figura 7: Constelação 4-ASK.

Note que a codificação Gray deve ser utilizada para mapear os bits na constelação.

Para calibração da relação E_b/N_0, deve-se considerar a energia média transmitida por bit, que é dada por:

$$E_b = \overline{E}_s/\log_2(M) = \frac{1/2\left[1^2 + 3^2\right]}{2} = \frac{5}{2} = 2,5 \tag{13}$$

Na recepção, o decisor deve utilizar como limite para decisão do símbolo estimado os valores -2, 0 e 2, como mostrado na Figura 8.

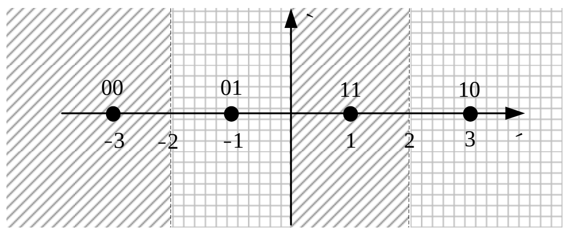


Figura 8: Constelação 4-ASK com regiões de decisão.

A probabilidade de erro de símbolo (SER) teórica para a modulação M-ASK é dada por:

$$\text{SER} = 2\frac{(M-1)}{M}Q\left(\sqrt{\frac{6\log_2(M)}{(M^2-1)}\frac{E_b}{N_0}}\right) \tag{14}$$

A probabilidade de erro de bit (BER) teórica para a modulação M-ASK é dada por:

$$\text{BER} = \frac{\text{SER}}{\log_2(M)} \tag{15}$$

Atividades:

1. Faça um *script* em MATLAB para simular um sistema de comunicação com modulação 4-ASK em canal AWGN utilizando o método de simulação computacional Monte Carlo;
2. Obtenha a curva de probabilidade de ero de bit (BER) via simulação e compare com a curva de desempenho teórica;
3. Obtenha a curva de probabilidade de ero de símbolo (SER) via simulação e compare com a curva de desempenho teórica;

Prática 5: Codificação de Huffman

1) Verifique no help do MATLAB a útilização das seguintes funções: `huffmandict`, `huffmanenco`, `huffmandeco`.

2) Execute o seguinte fragmento de código:

```
sinal = repmat([2 3 1 3 3 3 3 3 2 3],1,50); % Dados da Fonte
simb = [1 2 3]; % Símbolos distintos
p = [0.1 0.2 0.7]; % Probabilidade de cada símbolo
dict = huffmandict(simb,p); % Criação do dicionário

% Apresentação do dicionário na tela
temp = dict;
for i = 1:length(temp)
    temp{i,2} = num2str(temp{i,2});
end
disp('Dicionario'); disp(temp);

hcod = huffmanenco(sinal,dict); % Codificação do dado
hdec = huffmandeco(hcod,dict); % Decodificação do dado
```

Verifique e explique:

- A formação do dicionário. Qual função implemente isso?
- Se os dados foram corretamente codificados. Qual função implemente isso?
- Se os dados foram unicamente decodificados. Qual função implemente isso?

3) Execute o seguinte código extraído do *help* da função `huffmanenco.m`.

```
symbols = [1:6]; % Distinct symbols that data source can produce
p = [.5 .125 .125 .125 .0625 .0625]; % Probability distribution
[dict,avglen] = huffmandict(symbols,p); % Create dictionary.
actualsig = randsrc(100,1,[symbols; p]); % Create data using p.
comp = huffmanenco(actualsig,dict); % Encode the data.
```

Verifique e explique:

- O que são os dois argumentos de saída da função `huffmandict.m`?
- O que a função `randsrc.m` implemente? Verifique no help!
- Os dados foram corretamente codificados?

A partir desse exemplo pode-se determinar a taxa de compactação da codificação. Define-se:

$$R_c = \left(1 - \frac{\text{tamanho dados codificados}}{\text{tamanho dados originais}}\right) \cdot 100\%$$

Note que para os dados codificados devem ser considerados: número de *bytes* da sequência codificada + o número de *bytes* necessários para formar o dicionário. Nesse exemplo particular, pode-se assumir que o dicionário ocupa um espaço de 2 *bytes* por símbolo. Calcule a taxa de compressão para o item em questão. Gere agora 1000 símbolos ao invés de 100, calcule a taxa de compressão e responda: foi maior ou menor do que no caso com 100 símbolos? Por que?