

# Aula 03: Amostragem de Sinais Digitais.

## 1. Introdução

A amostragem de sinais digitais tem como objetivo recuperar os dígitos representados por cada um dos possíveis símbolos que compõe o sinal digital.

Como exemplo, podemos citar o sinal binário on-off onde se verifica a existência de dois símbolos:

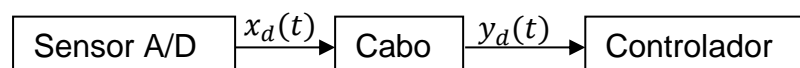
- $p(t)$  para representar o bit 1
- $q(t) = 0$  para representar o bit 0

Assim, a transmissão deste tipo de sinal se difere da transmissão de sinais analógicos no sentido de que tem como objetivo recuperar a sequência de bits transmitida, e não mais a forma, em si, do sinal transmitido.

Além disso, por apresentar uma menor quantidade de níveis de quantização, torna-se possível prever a taxa de erro de bit, conforme o nível do ruído do canal.

## 2. O Canal de transmissão

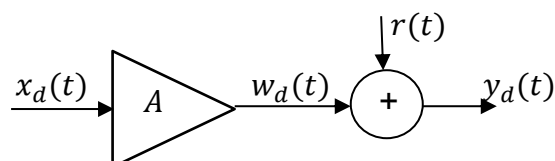
Para ilustrar o processo de amostragem, vamos utilizar como exemplo o sistema representado pelo seguinte diagrama de blocos:



Neste sistema, um dado sensor recebe as informações e as converte, através dos processos estudados na aula anterior, em um sinal digital e contínuo no tempo, nomeado de  $x_d(t)$ . Este sinal é, então, transmitido através de um cabo até alcançar o controlador.

Entretanto, devido as características não ideais do cabo, o sinal transmitido  $x_d(t)$  fica sujeito a atenuações e ruídos que acabam por fornecer ao controlador, uma versão distorcida do sinal original,  $y_d(t)$ .

No caso deste exemplo, a influência do cabo no sinal transmitido pode ser esquematizada de acordo com diagrama de blocos:



### 3. Atividades

- I. **SIMULINK:** O arquivo *Aula03\_ex01.mdl* fornece o diagrama de um circuito utilizado para amostrar e reconstruir o sinal a partir de suas amostras. Rode o arquivo e responda:
- Qual é o tempo de bit e a taxa de transmissão de bits por segundos deste gerador?
  - Qual é a frequência de amostragem utilizada para recuperar o sinal e qual é a sua relação com a taxa de bits verificada no item anterior?
  - Compare o sinal original com o sinal recuperado e verifique que este último encontra-se atrasado. De quantos tempo é este atraso? Qual é a relação deste atraso com o período de amostragem?
  - Após a execução da simulação, dois vetores (*original* e *recuperado*) são gerados no workspace. Estes vetores possuem a sequência de bits original e recuperada. Compare os dados destes vetores e verifique o efeito do atraso observado no item anterior.
  - Varie o período de amostragem, verifique os efeitos da amostragem e responda: qual deve ser a escolha de  $T_s$  para que a sequência de bits recuperada seja igual à sequência de bits original?
    - $T_s = 0,5 \cdot t_b$
    - $T_s = t_b$
    - $T_s = 0,7 \cdot t_b$
    - $T_s = 1,2 \cdot t_b$

Obs: Para modificar a frequência de amostragem, o tempo de amostragem ( $T_s = 1/f_s$ ) deve ser atualizado em três campos do bloco *amostrador do controlador*:

- no bloco *gerador de impulsos* campo *Period*
- no bloco *gerador de impulsos* campo *Phase Delay*, onde deve ser inserido o atraso verificado no item **c** ( $0,5 \cdot T_s$ )
- no bloco *to workspace* campo *Sample time*

- Utilizando a frequência de amostragem obtida no item anterior, varie a amplitude do ruído. Verifique sua influência no sinal recebido, assim como na sequência de bits recuperada, e apresente a taxa de erro de bits (TEB).

Ruído	TEB(%)	Ruído	TEB(%)
0,001		0,05	
0,01		0,1	

Obs:

A amplitude do ruído pode ser modificada campo *Variance* do ruído, pertencente ao bloco que representa o *cabo*.

Para contar a quantidade de erros, compare os vetores criados na *Workspace* através dos comandos *xor* e *sum*. **Atenção no atraso comentado nos itens c e d**

II. **MATLAB:** Suponha que o cabo por onde o sinal é transmitido gera uma perda de 30% (ou seja, ganho de 0,7). Assim, crie um programa chamado **Aula03\_ex02.m**, que:

Encontre e plote  $y_d[n.T_s]$  para  $0:T_s:5T_s$  considerando que para este mesmo intervalo

$$x_d[n.T_s] = [1 \ 0 \ 1 \ 1 \ 0 \ 1]$$

$$r[n.T_s] = [0.1, 0.2, -0.6, -0.1, 0.15, 0.7]$$

Compare  $x_d[n.T_s]$  transmitido com  $y_{rec}[n.T_s]$  recebido e responda: Qual sequência o receptor interpreta ter sido transmitida? Houve erro na recepção? Onde? Quantos?

Obs:

Para desenvolver este algoritmo, devem-se seguir os passos abaixo:

- Defina o vetor  $x_d$  original
- Defina o vetor  $r$  do ruído
- Encontre  $y_d$  através das operações apresentadas no segundo diagrama de blocos apresentado na primeira página desta experiência
- Crie uma sequência de comandos que verifique se o sinal recebido é maior ou menor que o limiar de decisão,  $a_0$ , definido por

$$a_0 = \frac{V_{bit1} + V_{bit0}}{2}$$

- Crie uma sequência de comandos que obtenha  $y_{rec}$  através da quantização das amostras maiores que  $a_0$  para  $V_{bit1}$  e as amostras menores que  $a_0$  para  $V_{bit0}$ .
- Crie uma sequência de comandos que compare  $y_{rec}[n]$  com  $x_d[n]$  e conte a quantidade de erros. (Dica: utilize a rotina utilizada no exercício anterior)

**Note que nesta simulação não estamos considerando o atraso verificado na etapa anterior, por isso não é necessário desprezar nenhum bit para fazer a comparação.**

## Exercícios de fixação\*

- I. A partir dos comandos do item anterior, substitua a sequência  $x_d[n]$  por uma sequência de 50 números zeros seguidos de 50 números uns [0 0 0 ...1 1 1 1] e a sequência  $r[n]$  por uma sequência de valores aleatórios, obtidos através da função *randn*. Gere os gráficos de  $x_d[n]$ ,  $r[n]$  e  $y_d[n]$  na mesma figura. Utilize para isso o comando *subplot*, com três linhas de gráficos.
  
- II. Para gerar um ruído gaussiano de potência  $P$  pode-se usar o comando  $r = P * \text{randn}(1, N)$ , onde  $N$  é a quantidade de amostras do ruído. Considere agora que  $x[n]$  é uma sequência aleatória de 1000 bits (0 ou 1). Faça um programa que varie o valor de  $P$  e gere um gráfico que relacione a potência do ruído e a taxa de erro de bit. (***Aula03\_fix01.m***)

---

\*Exercícios extraídos da apostila do Laboratório de Comunicações Digitais -março/2007, Márcio Eisencraft e Marco Antônio Assis.