

Aula de Laboratório 01

Amostragem (Aliasing em senoides)

A teoria da amostragem fornece uma compreensão matemática clara da interface entre o mundo analógico e o mundo discreto dos computadores digitais. O objetivo deste laboratório é colocarmos em prática essa teoria, apresentando alguns exemplos simples no MATLAB.

Projeto 1: Gere uma sinusóide analógica e em seguida, amostre esse sinal. Plote os sinais envolvidos e salve os gráficos.

Observe que o sinal inicial criado usando a função `makecos` (função em anexo) já é amostrado, mas com uma taxa de amostragem muito alta. Consideramos isso praticamente um sinal analógico, já que não queremos usar variáveis simbólicas em para criar sinais de tempo contínuo.

```
% Faça o sinal analógico inicial
% (frequency 20 Hz)
[m,t] = makecos(20);
```

Agora vamos amostrar este sinal com uma frequência de amostragem adequada. Nós sabemos que nosso sinal inicial é um cosseno com frequência de 20 Hz. Isso significa que a taxa Nyquist para este sinal é $2 * 20 = 40\text{Hz}$. No primeiro cenário de amostragem iremos usar uma taxa de amostragem de 50Hz que é mais que suficiente para uma reconstrução perfeita. Este é o caso da sobreamostragem, assim temos:

```
%Vamos fazer um trem de impulso para amostrar nosso sinal

% (frequência de amostragem de 50 Hz)
[it1,ts1] = makeimp(50);

% Agora amostre o sinal original
ms1 = sampleit1(t,m,ts1);

% Plote todos os sinais para visualizar o processo de amostragem

c1 = 'r'; % cor para o primeiro cenário
smp1_plot(t,m,ts1,it1,ms1,c1);
```

Observe que os gráficos discretos com setas denotam a integral da função dirac delta. Agora vamos amostrar o mesmo sinal analógico, mas com uma taxa de amostragem menor que a taxa de Nyquist. Este é o caso da subamostragem e escolhemos 30Hz.

```
% Agora faça um segundo trem de impulsos
% (frequência de amostragem de 30 Hz menor que a taxa de Nyquist)
[it2,ts2] = makeimp(30);
% Agora amostra o sinal original com a nova taxa de amostragem
ms2 = sampleit1(t,m,ts2);
% Plote todos os sinais para visualizar o processo de amostragem
c2 = 'g'; % cor para o Segundo cenário
smp1_plot(t,m,ts2,it2,ms2,c2);
```

De acordo com a teoria, para o primeiro sinal amostrado ms1, devemos ser capazes de reconstruir o original, enquanto para o segundo ms2 o aliasing deve ocorrer. Vamos agora reconstruir usando a função `interpsinc` (anexo) e plotar as formas de onda no tempo e na frequência para verificar a teoria da amostragem.

```
% Agora reconstrua as duas versões amostradas
mr1 = interpsinc(ms1,ts1,t);
mr2 = interpsinc(ms2,ts2,t);
% Plote o sinal original e o sinal reconstruído para comparar
recon_plot(t,m,ts1,ms1,mr1,c1);
recon_plot(t,m,ts2,ms2,mr2,c2);
```

Observe como o primeiro cosseno reconstruído tem a mesma frequência que o original, enquanto segundo tem uma frequência diferente. Claramente, no segundo sinal reconstruído, não temos reconstrução perfeita. Seguindo a teoria, podemos calcular que a frequência aparente é de 10Hz. Como estamos examinando frequências, podemos avaliar diretamente nossos cálculos na frequência domínio.

```
% Agora vamos obter o espectro para examinar o aliasing
% Faça o índice de frequência para plotagem
f = (-5000/2):(1/2):(5000/2);
```

```

% Use a função do anexo para calcular o espectro
M = am_spectrum(m);
MR1 = am_spectrum(mr1);
MR2 = am_spectrum(mr2);
% Plote o espectro para comparar
am_plot(f,M,MR1,MR2,0.02);

```

Verifique a frequência do sinal original e dos dois sinais reconstruídos, observe o aliasing.

TAREFAS :

Seguindo o procedimento descrito acima, verifique o aliasing e calcule a frequência dos co-senos reconstruídos nos seguintes casos:

- Cosseno: 30Hz, Amostragem: 50Hz
- Cosseno: 40Hz, Amostragem: 15Hz
- Cosseno: 10Hz, Amostragem: 50Hz
- Cosseno: 20Hz, Amostragem: 40Hz