

UNIVERSIDADE DE BRASÍLIA - BRASIL

FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ENGENHARIA ELÉTRICA



Universidade de Brasília



---

## LISTA DE EXERCÍCIO 1

---

***Alunos:***

Lucas Trigueiro Santana - 17/0121852

Vinícius Fernandes Valadares - 17/0157903

Li Wing Kee Ramos - 17/0108350

***Turma:*** A

***Professores:*** Flávia M. G. S Oliveira

***Disciplina:*** Tópicos em Engenharia - Processamento de Sinais  
Biomédicos

Brasília - DF  
2021

## Questão 1.1

Dado que a frequência de corte do filtro desejado é de 10 Hz, e que em 100 Hz, ou seja, uma década, o sinal deve ser atenuado em pelo menos 78 vezes, um filtro de quarta ordem com decaimento de 80 dB/decada satisfaz a condição do problema.

## Questão 1.5

```
Tp = 1;
f = 2;

Ts_005 = 0.05
t_005 = 0:Ts_005:1;
x_005 = sin(2*pi*f*t_005);
subplot(3,1,1)
plot(t_005,x_005,'k.-');
xlabel('Tempo (seg)');
ylabel('x(t)');

Ts_001 = 0.01;
t_001 = 0:Ts_001:1;
x_001 = sin(2*pi*f*t_001);

subplot(3,1,2)
plot(t_001,x_001,'k.-');
xlabel('Tempo (seg)');
ylabel('x(t)');

Ts_0001 = 0.001;
t_0001 = 0:Ts_0001:1;
x_0001 = sin(2*pi*f*t_0001);
subplot(3,1,3)
plot(t_0001,x_0001,'ko-');
xlabel('Tempo (seg)');
ylabel('x(t)');
```

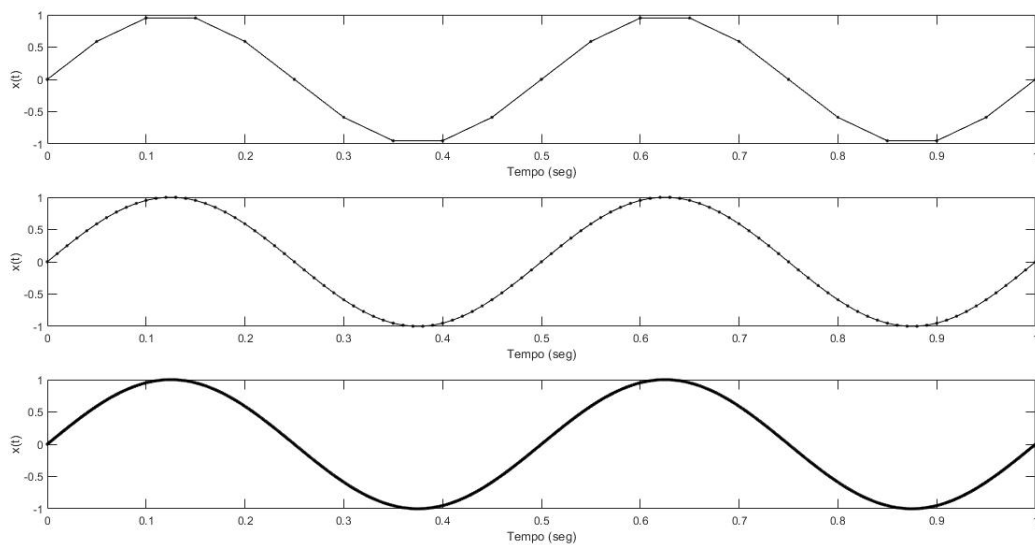


Figura 1: Gráficos da questão 1.5

## Questão 1.6

```
f = 4;
N = 1000;
Ts = 0.002;
t = (0:N-1)*Ts;
Signal_in = sin(2*pi*f*t);
```

```

for bits = [4,8,12,16]
    Signal_out = quantization(Signal_in,bits);
    Noise_signal = Signal_out - Signal_in;
    M = max(Noise_signal)-min(Noise_signal);
    q = max(Signal_in)/(2^bits-1);
    out = sprintf('%2d %5e %5e', bits, M, q);
    disp(out)
end

```

| Bits | Amplitude do Erro | Nível de quantização |
|------|-------------------|----------------------|
| 4    | 6.120598e-02      | 6.666140e-02         |
| 8    | 3.825425e-03      | 3.921259e-03         |
| 12   | 2.370130e-04      | 2.441810e-04         |
| 16   | 1.511926e-05      | 1.525782e-05         |

Tabela 1: Resultados da questão 1.6

## Questão 1.7

```

f = 4;
N = 1000;
Ts = 0.002;
t = (0:N-1)*Ts;
Signal_in = sin(2*pi*f*t);

for bits = [4,8,10,12]
    Signal_out = quantization(Signal_in,bits);
    Noise_signal = Signal_out - Signal_in;
    M = max(Noise_signal) - min(Noise_signal);
    q = max(Signal_in)/(2^bits-1);
    out = sprintf('%2d %5e %5e', bits, M, q);
    disp(out)
end

```

| Bits | Amplitude do Erro | Nível de quantização |
|------|-------------------|----------------------|
| 4    | 6.120598e-02      | 6.666140e-02         |
| 8    | 3.825425e-03      | 3.921259e-03         |
| 10   | 9.390508e-04      | 9.774399e-04         |
| 12   | 2.370130e-04      | 2.441810e-04         |

Tabela 2: Resultados da questão 1.7

## Questão 1.8

```

%% dados do sinal
f = 5;%Freq entrada Hz
fs = 7;% Frequencia de amostragem Hz

%% gerar sinal
tempo = linspace(0, 1, 1000); %t amostragem
sinal = sin(2*pi*f*tempo); % Sinal senoidal

```

```

%% plotar sinal
subplot(2,1,1)
plot(tempo,sinal,'g-')
hold;

%% sinal amostrado
Ts = 1/fs;
n = [0:0.001:1];
t_sample = [0 : Ts : 1];
sinal_sample = sin(2*pi*f*t_sample);

plot(t_sample, sinal_sample,'r*');
xlabel('Tempo (s)')
ylabel('Sinal (t)')
legend('Sinal 5Hz', 'Amostragem 7Hz')

%% Aliasing
subplot(2,1,2);
%seno defasado em 180°
Sinal_Ali = cos(2*pi*2*tempo);
plot(tempo,Sinal_Ali);
hold;

%seno defasado em 180°
sinal_sample = cos(2*pi*f*t_sample);
plot(t_sample, sinal_sample,'r*');
xlabel('Tempo (s)')
ylabel('Sinal (t)')
legend('Sinal 2Hz', 'Amostragem 7Hz')

```

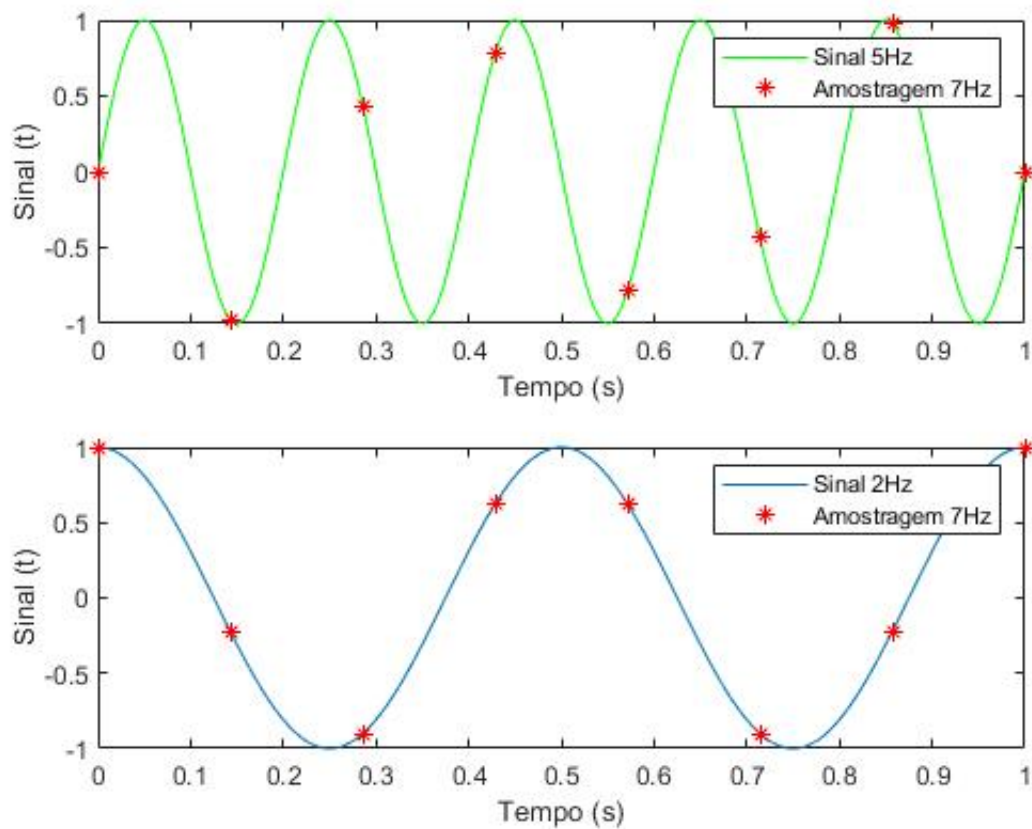


Figura 2: Gráficos da questão 1.8

## Questão 1.9

```
%% dados do sinal
f = 5;%Freq entrada Hz
fs = 9;% Frequencia de amostragem Hz

%% gerar sinal
tempo = linspace(0, 1, 1000); %t amostragem
sinal = sin(2*pi*f*tempo); % Sinal senoidal
%% plotar sinal
subplot(2,1,1)
plot(tempo,sinal,'g-')
hold;

%% sinal amostrado
Ts = 1/fs;
n = [0:0.001:1];
t_sample = [0 : Ts : 1];
sinal_sample = sin(2*pi*f*t_sample);

plot(t_sample, sinal_sample,'r*');
xlabel('Tempo (s)')
ylabel('Sinal (t)')
legend('Sinal 5Hz','Amostragem 9Hz')
```

```

%% Aliasing
subplot(2,1,2);
    %seno defasado em 180º
    Sinal_Ali = cos(2*pi*4*tempo);
    plot(tempo,Sinal_Ali);
    hold;

    %seno defasado em 180º
    sinal_sample = cos(2*pi*f*t_sample);
    plot(t_sample, sinal_sample,'r*');
    xlabel('Tempo (s)')
    ylabel('Sinal (t)')
    legend('Sinal 4Hz','Amostragem 9Hz')

```

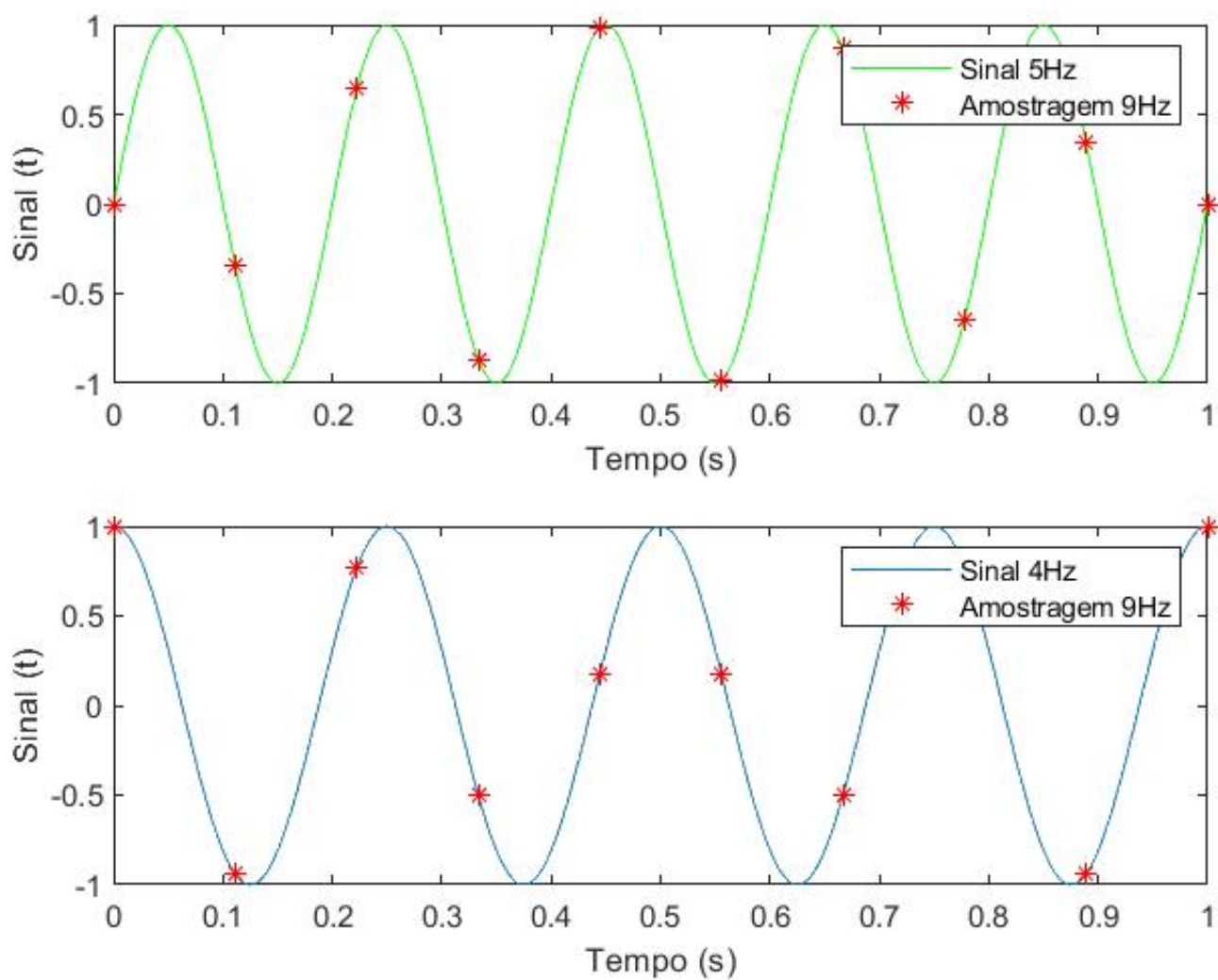


Figura 3: Gráficos da questão 1.9