Relatório do Laboratório 2 - Teorema da Amostragem EET-01

Igor Magalhães
igorcmag@gmail.com

Rafael Gonçalves
rafael.goncalves@ga.ita.br

23 de maio de 2020

1 Aliasing de um Senoidal

1.1 a)

O gráfico solicitado se encontra na **figura 1**.

Listing 1: Código em MATLAB que cria a função senoidal e gera o gráfico.

```
set(0, 'defaulttextinterpreter', 'Latex');

% sin signal

x = @(n, f_s, f_0, phi) sin(2*pi*(f_0/f_s)*n + phi);

f_s = 8*10^3;
f_0 = 300;
phi = 0;
t = 10*10^(-3);

n_samples = round(t*f_s);

n = 0:1:n_samples;

figure;
stem(n, x(n, f_s, f_0, phi), 'filled', 'Color', 'blue', 'LineStyle', '--');
title("Onda senoidal amostrada");
legend("x[n] = sin(75\pi\cdot n)");
xlabel("n");
ylabel("x[n]");
```

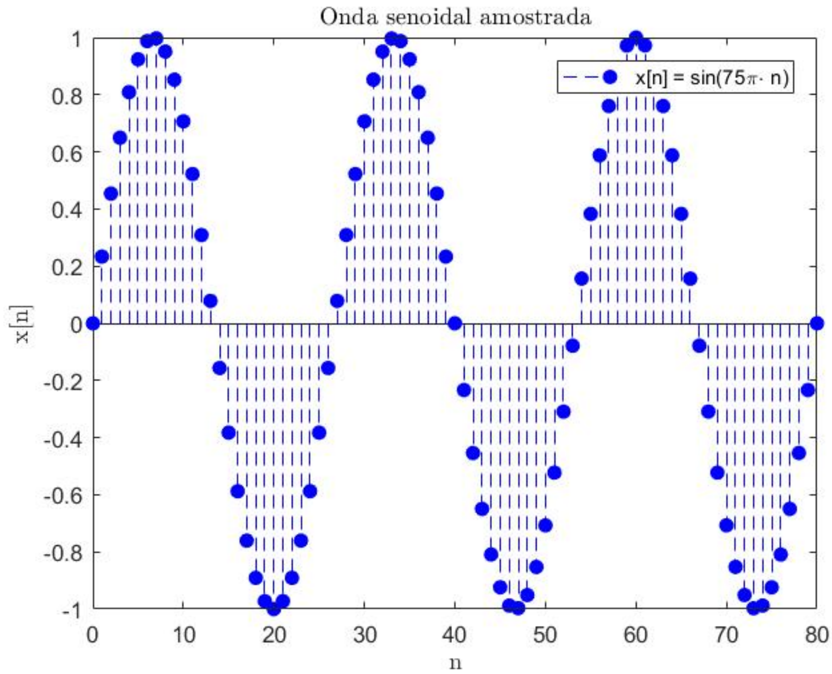

Figura 1: Onda senoidal amostrada

1.2 b)

O gráfico com plot se encontra na **figura 2**.

Listing 2: Código em MATLAB que cria a função senoidal e gera o gráfico.

```
set(0, 'defaulttextinterpreter', 'Latex');

% sin signal

x = @(n, f_s, f_0, phi) sin(2*pi*(f_0/f_s)*n + phi);

f_s = 8*10^3;
f_0 = 300;
phi = 0;
t = 10*10^(-3);

n_samples = round(t*f_s);

n = 0:1:n_samples;

figure;
plot(n, x(n, f_s, f_0, phi), 'blue');
title("Onda senoidal amostrada");
legend("x[n] = sin(75\pi\cdot n)");
xlabel("n");
ylabel("x[n]");
```

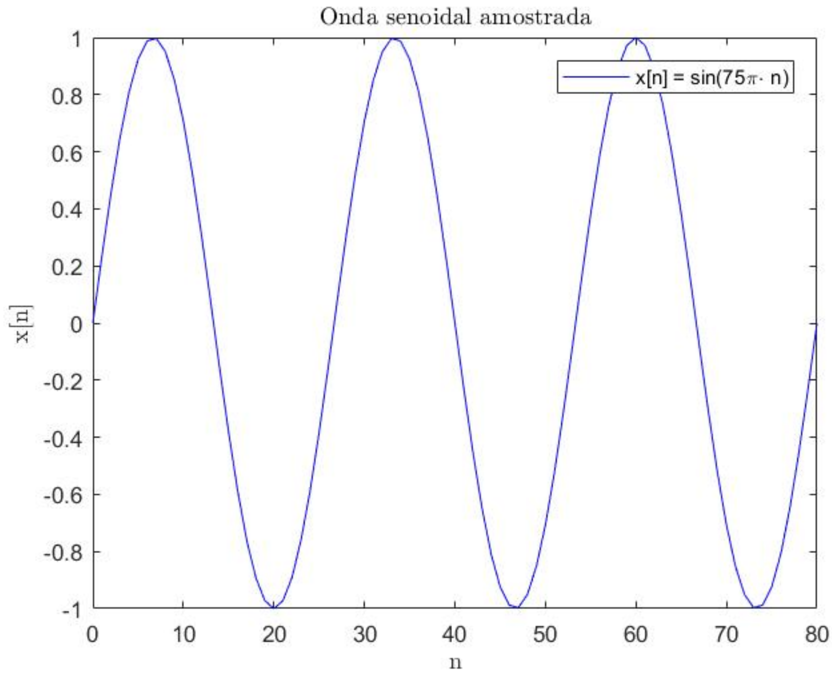

Figura 2: Onda senoidal amostrada, agora interpolada linearmente.

1.3 c)

A sequência de gráficos se encontra na **figura 3**. Fica claro que a frequência aparente da senóide aumenta.

Listing 3: Código em MATLAB que cria a função senoidal e gera a sequência de gráficos para diferentes frequências.

```
set(0, 'defaulttextinterpreter', 'Latex');

% sin signal

x = @(n, f_s, f_0, phi) sin(2*pi*(f_0/f_s)*n + phi);

f_s = 8*10^3;
f_0 = [100, 225, 350, 475];
phi = 0;
t = 10*10^(-3);

n_samples = round(t*f_s);

n = 0:1:n_samples;

subplot(2,2,1);
plot(n, x(n, f_s, f_0(1), phi), 'blue');
title("$f_0$ = 100Hz");
xlabel("n");
ylabel("x[n]");

subplot(2,2,2);
plot(n, x(n, f_s, f_0(2), phi), 'blue');
title("$f_0$ = 225Hz");
xlabel("n");
ylabel("x[n]");

subplot(2,2,3);
plot(n, x(n, f_s, f_0(3), phi), 'blue');
title("$f_0$ = 350Hz");
xlabel("n");
ylabel("x[n]");

subplot(2,2,4);
```

```
plot(n, x(n, f_s, f_0(4), phi), 'blue');
title("$f_0$ = 475Hz");
xlabel("n");
ylabel("x[n]");
```

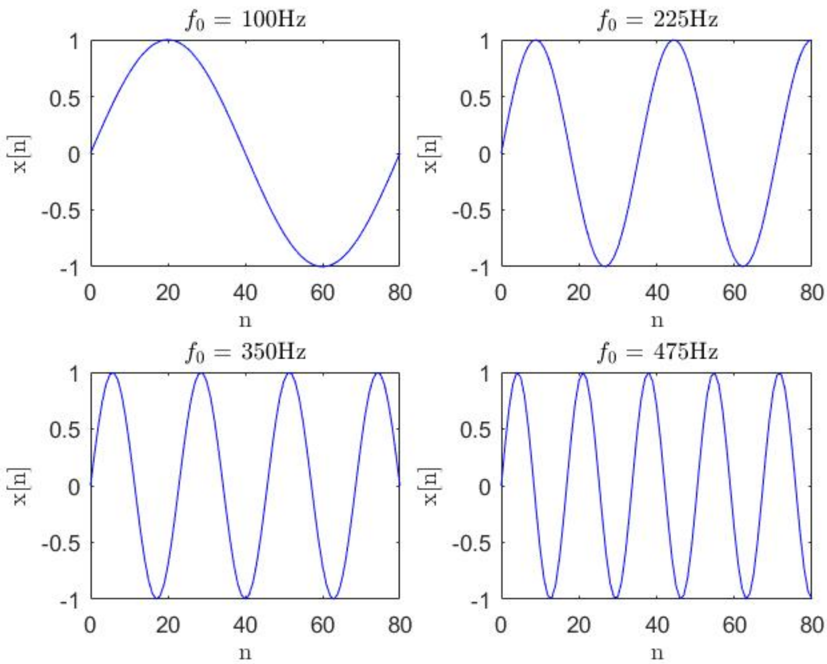

Figura 3: Série de gráficos com variação na frequência.

1.4 d)

A sequência de gráficos se encontra na **figura 4**. Fica claro que a frequência aparente da senóide diminui. Isto se deve ao fato de estarmos trabalhando com uma função 2π-periódica, ou seja, devemos olhar para o resto da divisão da frequência pelo maior múltiplo de 2π ainda menor que a frequência.

Listing 4: Código em MATLAB que cria a função senoidal e gera a sequência de gráficos para diferentes frequências.

```
set(0, 'defaulttextinterpreter', 'Latex');

% sin signal

x = @(n, f_s, f_0, phi) sin(2*pi*(f_0/f_s)*n + phi);

f_s = 8*10^3;
f_0 = [7525, 7650, 7775, 7900];
phi = 0;
t = 10*10^(-3);

n_samples = round(t*f_s);

n = 0:1:n_samples;

subplot(2,2,1);
plot(n, x(n, f_s, f_0(1), phi), 'blue');
title("$f_0$ = 7525Hz");
xlabel("n");
ylabel("x[n]");

subplot(2,2,2);
plot(n, x(n, f_s, f_0(2), phi), 'blue');
title("$f_0$ = 7650Hz");
```

```
xlabel("n");
ylabel("x[n]");

subplot(2,2,3);
plot(n, x(n, f_s, f_0(3), phi), 'blue');
title("$f_0$ = 7775Hz");
xlabel("n");
ylabel("x[n]");

subplot(2,2,4);
plot(n, x(n, f_s, f_0(4), phi), 'blue');
title("$f_0$ = 7901Hz");
xlabel("n");
ylabel("x[n]");
```

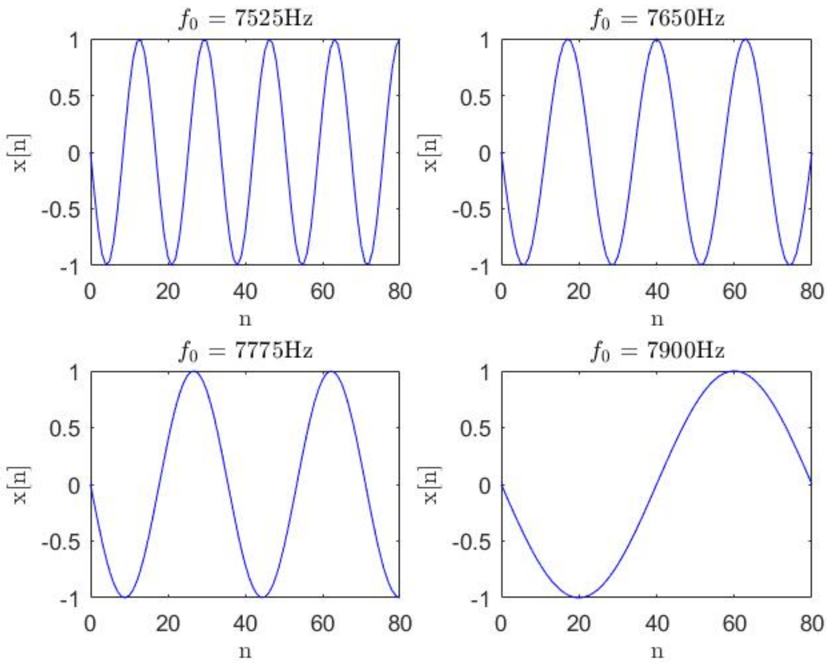

Figura 4: Série de gráficos com variação na frequência.

1.5 e)

A sequência de gráficos se encontra na **figura 5**. Pelo que foi explicado no item anterior, a frequência aparente deve aumentar no intervalo proposto, exatamente o que fora observado.

Listing 5: Código em MATLAB que cria a função senoidal e gera a sequência de gráficos para diferentes frequências.

```
set(0, 'defaulttextinterpreter', 'Latex');

% sin signal

x = @(n, f_s, f_0, phi) sin(2*pi*(f_0/f_s)*n + phi);

f_s = 8*10^3;
f_0 = [32100, 32225, 32350, 32475];
phi = 0;
t = 10*10^(-3);

n_samples = round(t*f_s);

n = 0:1:n_samples;

subplot(2,2,1);
plot(n, x(n, f_s, f_0(1), phi), 'blue');
```

```
title("$f_0$ = 32100Hz");
xlabel("n");
ylabel("x[n]");

subplot(2,2,2);
plot(n, x(n, f_s, f_0(2), phi), 'blue');
title("$f_0$ = 32225Hz");
xlabel("n");
ylabel("x[n]");

subplot(2,2,3);
plot(n, x(n, f_s, f_0(3), phi), 'blue');
title("$f_0$ = 32350Hz");
xlabel("n");
ylabel("x[n]");

subplot(2,2,4);
plot(n, x(n, f_s, f_0(4), phi), 'blue');
title("$f_0$ = 32475Hz");
xlabel("n");
ylabel("x[n]");
```

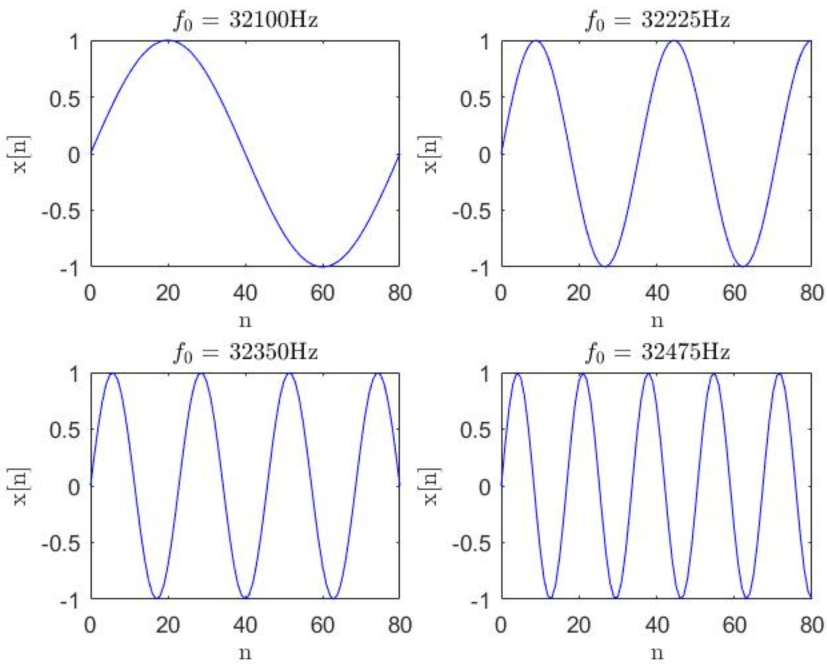

Figura 5: Série de gráficos com variação na frequência.

2 Aliasing de um Chirp

2.1 a)

$$f_i(t) = \mu t + f_l = 600t + 4 \therefore f_i(0) = 4kHz \therefore f_i(50ms) = 600 \cdot 0.05 + 4 = 34kHz$$

Assim, a faixa de frequência é de 0 a $34kHz$.

2.2 b)

$$c[n] = c(nT) = c(\frac{n}{f_s}) = cos(\frac{\pi\mu}{{f_s}^2}n^2 + \frac{2\pi f_l}{f_s}n + \psi)$$

Fazendo $f_s = 8kHz$ e $\psi = 0$:

$$c[n] = cos(\frac{3}{320}\pi n^2 + \pi n)$$

Perceba que a função é par. O sinal de amostragem foi gerado pelo código abaixo e seu gráfico está exposto na **figura 6**.

Listing 6: Código em MATLAB para gerar o sinal amostrado do item (b) do exercício 2, bem como seus gráficos 'plot' e 'stem'.

```
set(0, 'defaulttextinterpreter', 'Latex');

n = 0:100;
c = cos((3/320)*pi*(n.^2)' + pi*n');
subplot(1,2,1);plot(n, c);title('$c[n]$ plot');xlabel('$n$');ylabel('$c[n]$');
subplot(1,2,2);stem(n, c);title('$c[n]$ stem');xlabel('$n$');ylabel('$c[n]$');
```

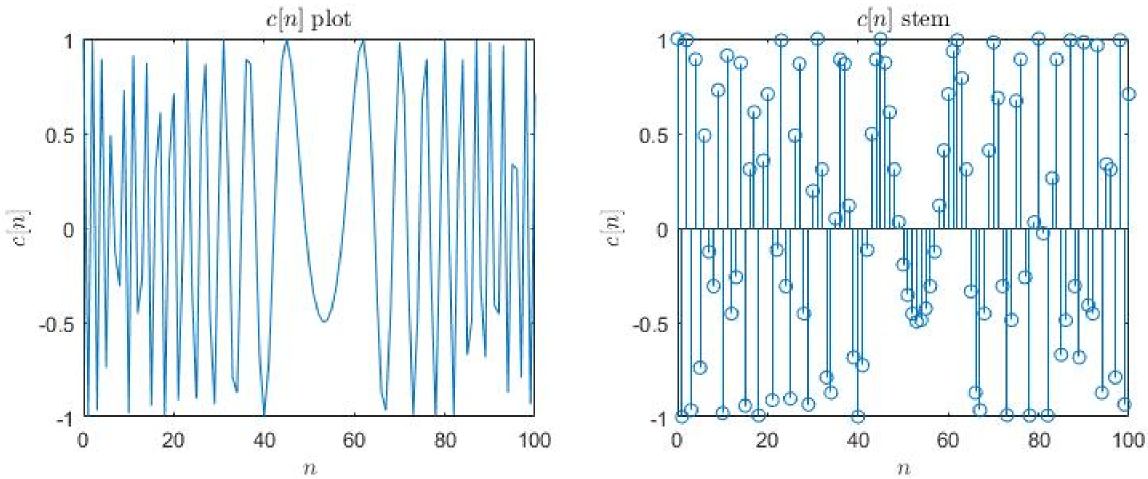

Figura 6: Gráficos 'plot' e 'stem', respectivamente, de $c[n]$

2.3 c)

$$f_i(t) = 0 \therefore \frac{\mu}{{f_s}^2}n + \frac{f_l}{f_s} = 0 \therefore n \approx -53$$

.

Isso é confirmado pelo gráfico, pois a função é par, isto é, $f(-n) = f(n)$, podemos olhar em $n = 53$. No tempo, isso equivale a $t = \frac{n}{f_s} = 6.67ms$. No código abaixo, plotamos $c[n]$ para n de 0 a 1000, como mostra a **figura 7**. Vemos que as regiões de baixa frequência estão regularmente espaçadas.

Listing 7: Código em MATLAB para gerar o sinal amostrado do item (b) do exercício 2, bem como seu gráfico 'plot' para n variando de 0 a 1000.

```
set(0, 'defaulttextinterpreter', 'Latex');

n = 0:1000;
c1 = cos((3/320)*pi*(n.^2)' + pi*n');
plot(n, c1);title('$c[n]$ plot');xlabel('$n$');ylabel('$c[n]$');
```

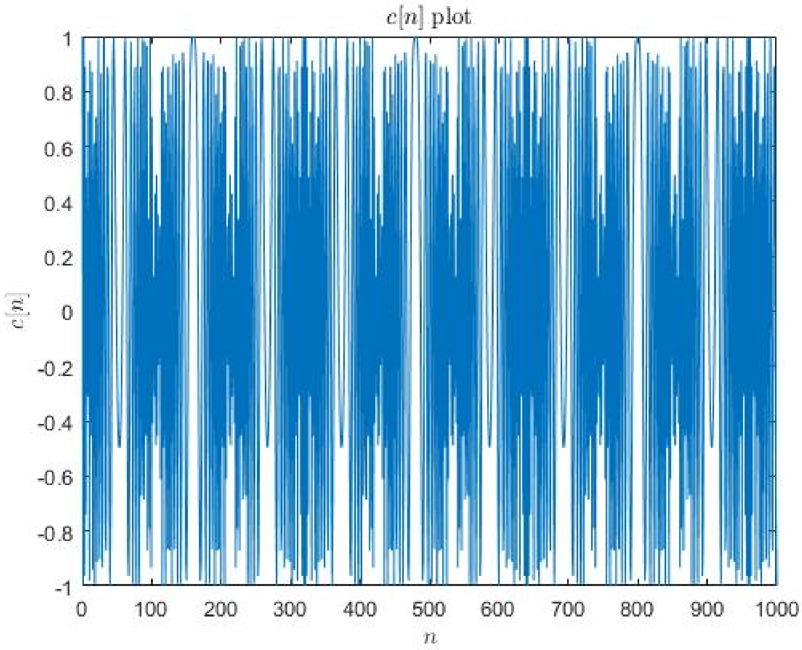

Figura 7: Gráfico 'plot' de $c[n]$ com n de 0 a 1000

.

3 Escutando o Aliasing

3.1 a)

O esboço dos 3 sinais em um subplot do MATLAB segue na **figura 8**.

Analisando os sons reproduzidos, é notável que quanto maior a frequência, mais agudo é seu som, além disso, o som parece ser o mesmo durante os 5 segundos.

Listing 8: Código em MATLAB que cria o sinal sinusoidal e gera a sequência de gráficos para diferentes frequências, além de reproduzir os sons de cada um dos sinais.

```
set(0, 'defaulttextinterpreter', 'Latex');

% sin signal

x = @(t, f_0, phi) sin(2*pi*f_0*t + phi);

f_s = [20*10^3, 4.5*10^3, 1.4*10^3];
f_0 = 2*10^3;
phi = 0;
t_total = 5;

t_1 = 0:1/f_s(1):t_total;
t_2 = 0:1/f_s(2):t_total;
t_3 = 0:1/f_s(3):t_total;

figure;
subplot(2,2,1);
plot(t_1, x(t_1, f_0, phi), 'blue');
title("$f_s$ = 20kHz");
xlabel("t(s)");
ylabel("x(t)");
xlim([0, 0.01]);

subplot(2,2,2);
plot(t_2, x(t_2, f_0, phi), 'blue');
title("$f_s$ = 4.5kHz");
xlabel("t(s)");
ylabel("x(t)");
xlim([0, 0.01]);
```

```
subplot(2,2,3);
plot(t_3, x(t_3, f_0, phi), 'blue');
title("$f_s$ = 1.4kHz");
xlabel("t(s)");
ylabel("x(t)");
xlim([0, 0.01]);

%soundsc(x(t_1, f_0, phi), f_s(1));
%soundsc(x(t_2, f_0, phi), f_s(2));
%soundsc(x(t_3, f_0, phi), f_s(3));
```

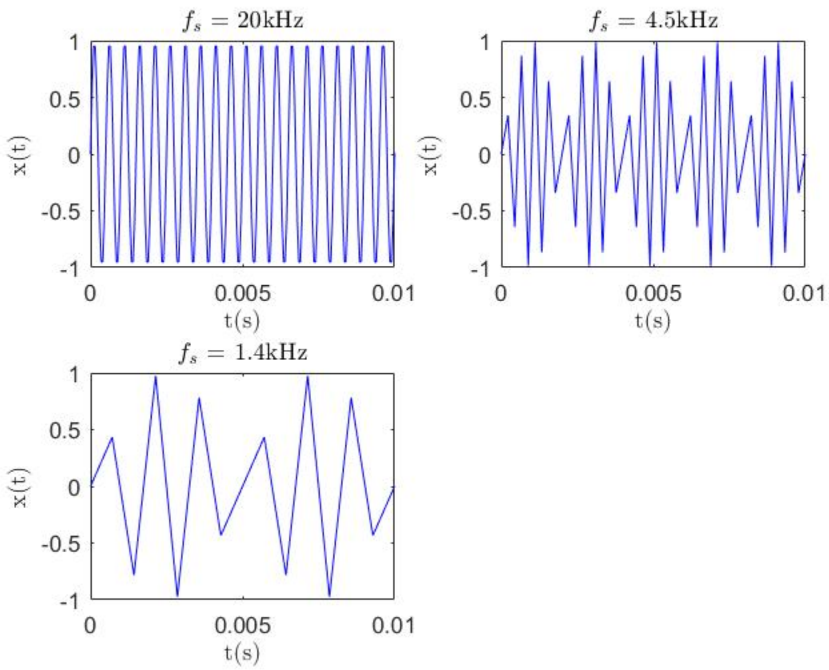

Figura 8: Série de gráficos com variação na frequência.

3.2 b)

O esboço dos 3 sinais em um subplot do MATLAB segue na **figura 9**.

Analisando os sons reproduzidos, temos:

20kHz: A frequência parece aumentar gradativamente.

8.5kHz: A frequência, que parece menor do que o áudio anterior, aumenta gradativamente e no final dos 5 segundos diminui.

3.4kHz: A frequência começa diminuindo, o som pára momentaneamente e a frequência volta a aumentar.

Listing 9: Código em MATLAB que cria o sinal "chirp" e gera a sequência de gráficos para diferentes frequências, além de reproduzir os sons de cada um dos sinais.

```
set(0, 'defaulttextinterpreter', 'Latex');

% sin signal

x = @(t, mu, f_l, phi) cos(pi*mu*t.^2 + 2*pi*f_l*t + phi);

f_s = [20*10^3, 8.5*10^3, 3.4*10^3];
f_l = 2*10^3;
mu = 0.5*10^3;
phi = 0;
t_total = 5;

```

```
t_1 = 0:1/f_s(1):t_total;
t_2 = 0:1/f_s(2):t_total;
t_3 = 0:1/f_s(3):t_total;

figure;
subplot(2,2,1);
plot(t_1, x(t_1, mu, f_l, phi), 'blue');
title("$f_s$ = 20kHz");
xlabel("t(s)");
ylabel("x(t)");
xlim([0, 0.01]);

subplot(2,2,2);
plot(t_2, x(t_2, mu, f_l, phi), 'blue');
title("$f_s$ = 8.5kHz");
xlabel("t(s)");
ylabel("x(t)");
xlim([0, 0.01]);

subplot(2,2,3);
plot(t_3, x(t_3, mu, f_l, phi), 'blue');
title("$f_s$ = 3.4kHz");
xlabel("t(s)");
ylabel("x(t)");
xlim([0, 0.01]);

%soundsc(x(t_1, mu, f_l, phi), f_s(1));
%soundsc(x(t_2, mu, f_l, phi), f_s(2));
soundsc(x(t_3, mu, f_l, phi), f_s(3));
```

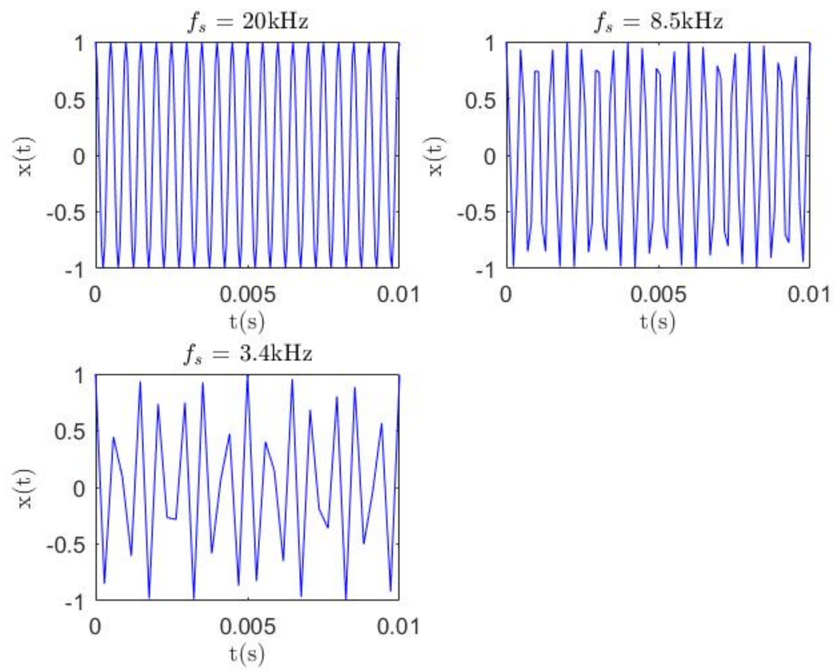

Figura 9: Série de gráficos com variação na frequência.

4 Montagem de uma Onda Sinoidal

As equações são:

$$2 = Acos(\phi)$$

$$1 = Acos(\omega + \phi)$$

$$-1 = Acos(2\omega + \phi)$$

Somando a segunda com a terceira, tem-se:

$$Acos(\omega + \phi) + Acos(2\omega + \phi) = 0 \therefore cos(\frac{3\omega}{2} + \phi)cos(\frac{\omega}{2}) = 0$$

$$cos(\frac{\omega}{2}) = 0 \therefore \omega = (2k+1)\pi,$$

ou

$$cos(\frac{3\omega}{2} + \phi) = 0 \therefore \omega = \frac{(2k+1)\pi}{3} - \frac{2\phi}{3}.$$

Substituindo A da primeira equação na segunda, tem-se

$$\frac{1}{2}cos(\phi) = cos(\omega + \phi).$$

No primeiro caso, $cos(\omega + \phi) = cos((2k+1)\pi + \phi) = -cos(\phi)$, logo $\frac{1}{2} = -1$, absurdo. No segundo caso

$$\frac{1}{2}cos(\phi) = cos(\frac{2(k+1)\pi}{3} + \frac{\phi}{3})$$

A resolução analítica desta equação é complicada e depende do resto que k deixa na divisão por 3, isto é, é preciso separar em três casos. Para mostrar que não há uma única solução, é suficiente darmos um contra-exemplo.

Para $k = 1$, temos a solução $\phi = \pi$ no intervalo $[0, 2\pi)$. A solução final fica

$$x_1(t) = -2cos(\frac{\pi}{3}t + \pi) = 2cos(\frac{\pi}{3}t)$$

Para $k = 4$, temos a mesma equação em ϕ e portanto a mesma solução $\phi = \pi$. No entanto, dessa vez a solução final fica

$$x_2(t) = -2cos(\frac{7\pi}{3}t + \pi) = 2cos(\frac{7\pi}{3}t)$$

Portanto, a informação não é suficiente para determinar $x(t)$.

5 Interpolação Polinomial Linear

5.1 a)

A função 'grid minor' foi utilizada para dispor a grade fina, que tem espaçamento automático de 0.05 e o gráfico segue na **figura 10**.

Listing 10: Código em MATLAB que cria o sinal e gera seu gráfico.

```
set(0, 'defaulttextinterpreter', 'Latex');

figure;
plot([0, 1, 2], [2, 1, -1], '-o');
grid on;
grid minor;
```

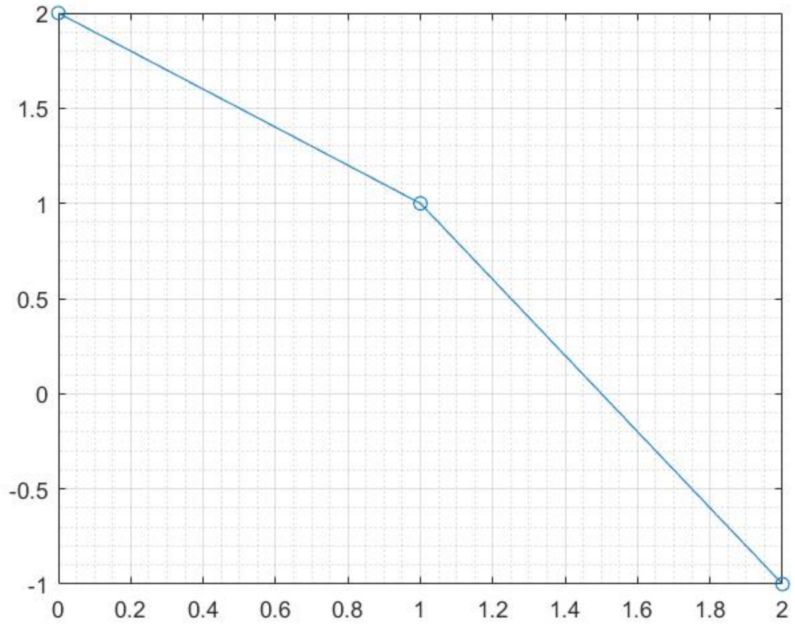

Figura 10: Gráfico do sinal solicitado.

5.2 b)

O resultado deve ser o mesmo da interpolação linear considerando $x(-1) = t(3) = 0$, uma vez que essa situação coincide com a operação feita considerando o pulso como não-nulo apenas nos pontos $t(0)$, $t(1)$ e $t(2)$.

Listing 11: Código em MATLAB que cria o sinal, o impulso, realiza a convolução entre eles e gera o gráfico da figura 11.

```
set(0, 'defaulttextinterpreter', 'Latex');

x = [2, 0, 0, 0, 0, 1, 0, 0, 0, 0, -1];
resposta = [0.2, 0.4, 0.6, 0.8, 1.0, 0.8, 0.6, 0.4, 0.2];
y = conv(x, resposta);

figure;
plot(0:2/18:2, y, 'blue');
title("Resposta ao impulso triangular");
xlabel("x");
ylabel("y");
grid on;
grid minor;
```

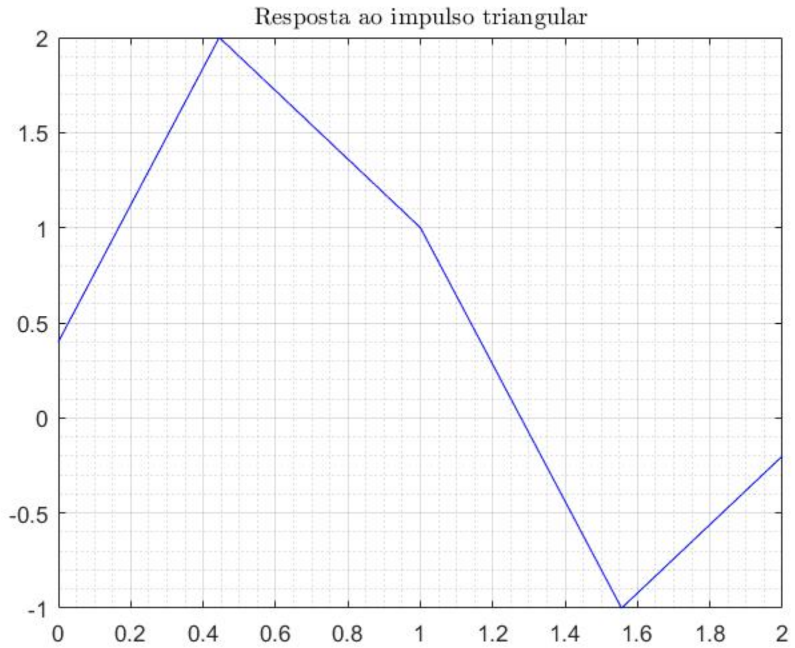

Figura 11: Série de gráficos com variação na frequência.

5.3 c)

Essa curva não é realista em um sentido prático, uma vez que se a estendermos para valores além do intervalo, ela diverge, e produz resultados distantes da realidade, além de apresentar uma única concavidade.

Listing 12: Código em MATLAB que cria o sinal, realiza sua interpolação quadrática e gera o gráfico da figura 12

```
set(0, 'defaulttextinterpreter', 'Latex');

x = [0, 1, 2];
y = [2, 1, -1];
p = polyfit(x, y, 2);
t = 0:0.1:3;
y_fit = polyval(p, t);

figure;
plot(t, y_fit, 'blue');
title("Polinomio interpolar de grau 2");
xlabel("x");
ylabel("y");
grid on;
grid minor;
```

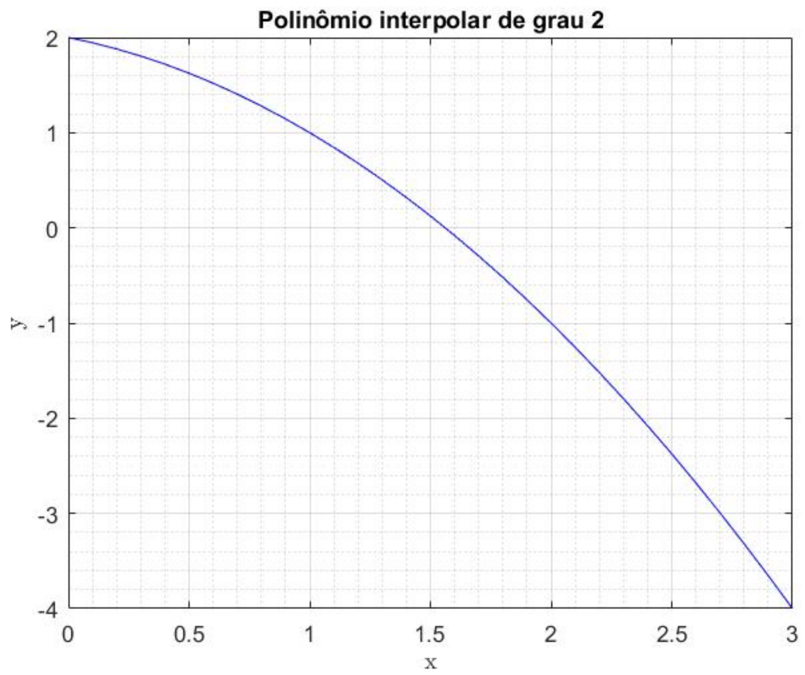

Figura 12: Série de gráficos com variação na frequência.

6 Filtragem Passa-Baixo Ideal

6.1 a)

No **codigo**, implementamos a função *interpolador-seno*, que recebe o vetor x de amostras, o vetor n dos índices correspondentes, e o período de amostragem T_s. Ele então reconstrói o sinal contínuo x_r dado por

$$x_r(t) = \sum_{i=1}^{length(n)} x[i] \frac{sin(\pi(t - n[i]T_s)/T_s)}{\pi(t - n[i]T_s)/T_s}$$

Listing 13: Código em MATLAB que cria a função interpoladora senoidal.

```
function f = interpolador_seno(x, n, Ts)
syms t;
f(t) = 0*t;
for i=1:length(n);
   f(t) = f(t) + x(i)*sin(pi*(t - n(i)*Ts)/Ts)/(pi*(t - n(i)*Ts)/Ts);
end
```

6.2 b)

Usamos a função criada para interpolar uma amostra de um único ponto $x(0) = 1$, como descrito no **codigo** e exposto na **figura 13**.

Listing 14: Código em MATLAB interpolar a amostra de um único ponto x(0)=1, bem como plotar seu gráfico.

```
set(0, 'defaulttextinterpreter', 'Latex');

n = [0];
x = [1];
Ts = 1;
x_r = interpolador_seno(x, n, Ts);
```

```
fplot(x_r);title('x(0)=1');xlabel('$t$');ylabel('$x_r(t)$');
```

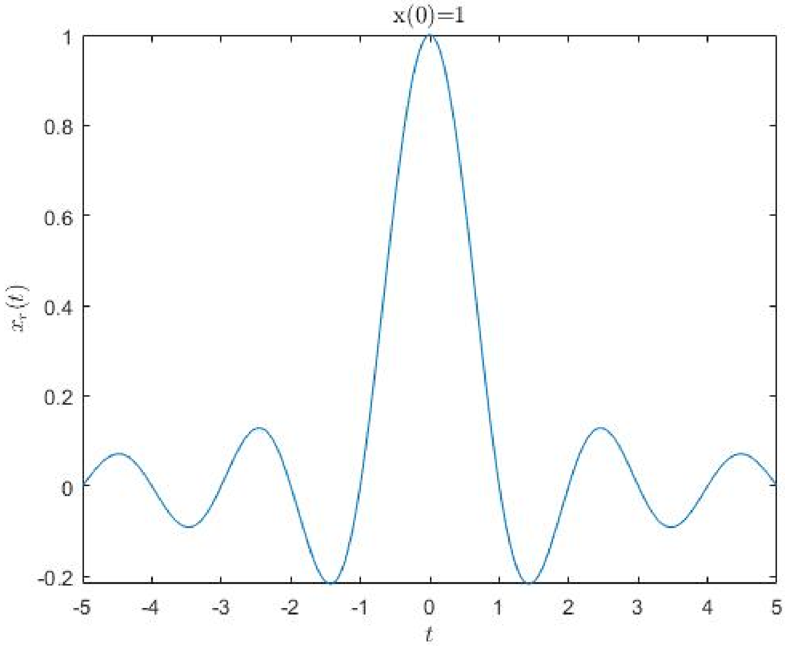

Figura 13: Gráfico da interpolação senoidal da amostra de um único ponto $x(0) = 1$
.

6.3 c)

Dessa vez usamos a função criada para interpolar uma amostra de três pontos $x(0) = 2$, $x(1) = 1$ e $x(2) = -1$ como descrito no **codigo** e exposto na **figura 14**.

Listing 15: Código em MATLAB interpolar a amostra de três pontos $x(0) = 2$, $x(1) = 1$ e $x(2) = -1$, bem como plotar seu gráfico.

```
set(0, 'defaulttextinterpreter', 'Latex');

n = [0, 1, 2];
x = [2, 1, -1];
Ts = 1;
x_r = interpolador_seno(x, n, Ts);
fplot(x_r);title('x(0)=2, x(1)=1 e x(2)=-1');xlabel('$t$');ylabel('$x_r(t)$');
```

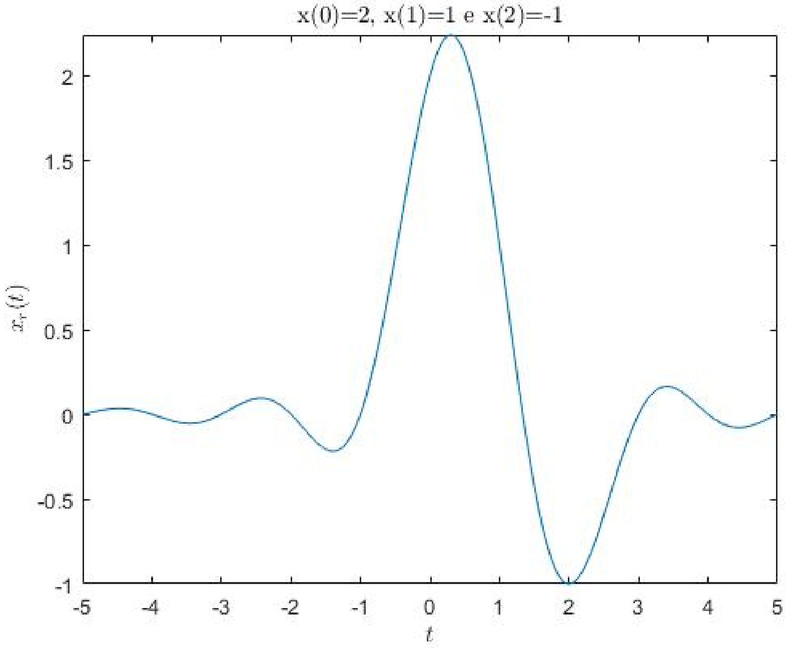

Figura 14: Gráfico da interpolação senoidal da amostra de três pontos $x(0) = 2$, $x(1) = 1$ e $x(2) = -1$

.

Perceba que o gráfico da **figura** é um ajuste do gráfico da **figura**, de modo que este passe pelos pontos amostrados.