Relatório do Laboratório 1 - Sinais
EET-01

Igor Magalhães
igorcmag@gmail.com

Rafael Gonçalves
rafael.goncalves@ga.ita.br

23 de março de 2020

1 Impulsos

1.1 a)

Para gerar e plotar as quatro sequências, usamos o código 1 apresentado abaixo.

Listing 1: Código em MATLAB para gerar as quatro sequências do item (a) do exercício 1, bem como a plotagem dos seus respectivos gráficos.

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

%sequencia 1
L = 20;
nn = 1:20;
x1 = zeros(L,1);x1(5) = 0.9;
subplot(2,2,1);stem(nn, x1);title('Sequencia 1');xlabel('$n$');ylabel('$x_1[n]$');

%sequencia 2
L = 31;
nn = -15:15;
x2 = zeros(L, 1);x2(16) = 0.8;
subplot(2,2,2);stem(nn, x2);title('Sequencia 2');xlabel('$n$');ylabel('$x_2[n]$');

%sequencia 3
L = 51;
nn = 300:350;
x3 = zeros(L, 1);x3(34) = 1.5;
subplot(2,2,3);stem(nn, x3);title('Sequencia 3');xlabel('$n$');ylabel('$x_3[n]$');

%sequencia 4
L = 11;
nn = -10:0;
x4 = zeros(L, 1);x4(4) = 4.5;
subplot(2,2,4);stem(nn, x4);title('Sequencia 4');xlabel('$n$');ylabel('$x_4[n]$');
```

Os gráficos das quatro sequências estão exibidos na figura 1, apresentada abaixo.

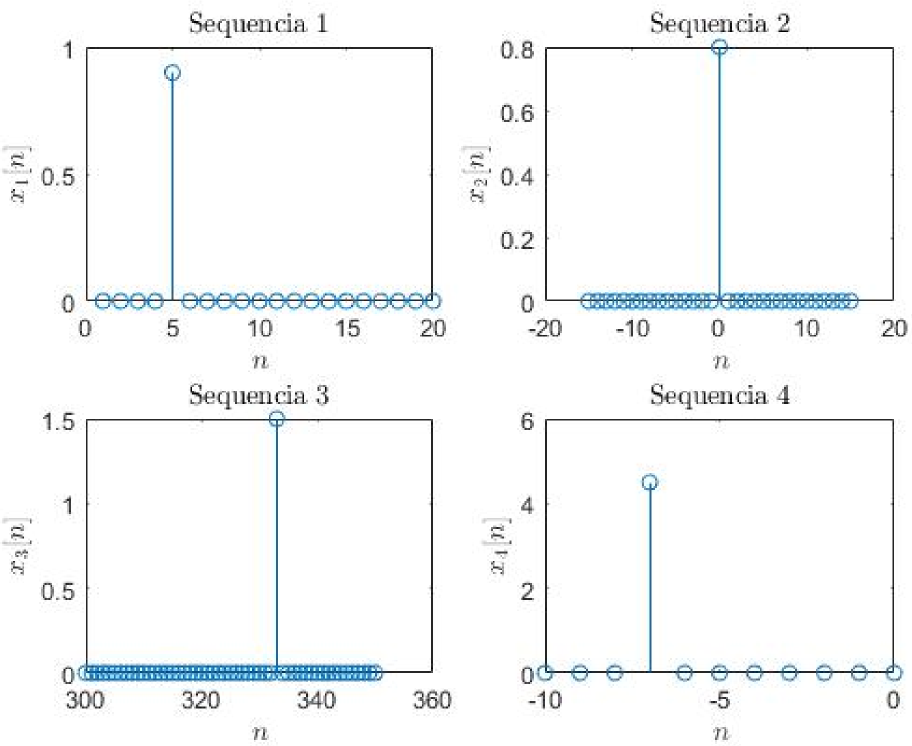

Figura 1: Gráficos das sequências em seus respectivos domínios

1.2 b)

Para gerar e plotar o trem de impulsos, usamos o código 2 apresentado abaixo.

Listing 2: Código em MATLAB para gerar e plotar o trem de impulsos.

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

%sinal s
nn = 0:49;
s = zeros(50, 1);s(1:5:50) = 1;

%grafico
stem(nn, s);title('Sinal s');xlabel('$n$');ylabel('$s[n]$');yticks([0 1]);
```

O gráfico obtido está na figura 2, exposta abaixo.

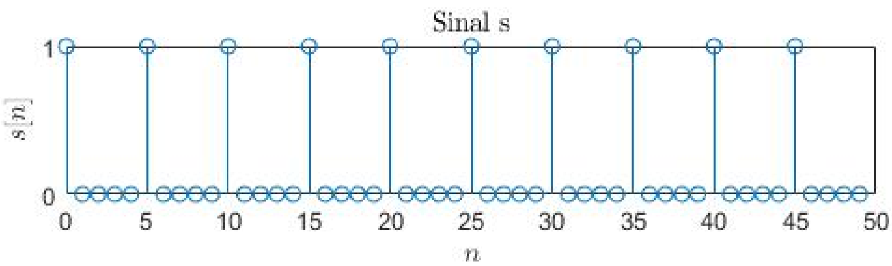

Figura 2: Gráfico do trem de impulsos

1.3 c)

Usamos o código 3, apresentado abaixo, para gerar e plotar o vetor x.

Listing 3: Criação e plotagem do vetor x

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

%vetor x
x=[0;1;1;0;0;0]*ones(1,7);
x= x(:);
nn = 0:size(x)-1;

%grafico
stem(nn, x);title('Sinal x');xlabel('$n$');ylabel('$x[n]$');yticks([0 1]);
```

Obtivemos, então, o gráfico da figura 3, mostrado abaixo.

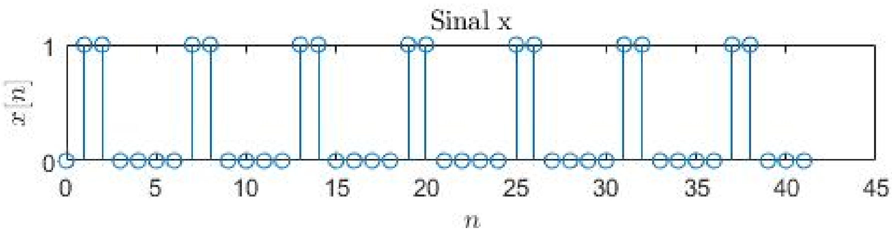

Figura 3: Gráfico do vetor x

O vetor x pode ser escrito na forma

$$x[n] = \sum_{k=0}^{P-1} \sum_{m=0}^{M-1} A_k \delta[n - k - mP]$$

com $P = 6$, $M = 7$ e

$$\begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

resultando em

$$x[n] = \sum_{m=0}^{6} \{\delta[n - 1 - 6m] + \delta[n - 2 - 6m]\}$$

2 Sinusoides

2.1 a)

Para gerar e plotar as quatro sequências, usamos o código 4 apresentado abaixo.

Listing 4: Código em MATLAB que gera as quatro sequências e plota seus respectivos gráficos

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

%sequencia 1
L = 21;
nn = 0:20;
x1 = sin((pi/17)*nn);
subplot(2,2,1);stem(nn, x1);title('Sequencia 1');xlabel('$n$');ylabel('$x_1[n]$');

%sequencia 2
L = 41;
nn = -15:25;
```

```
x2 = sin((pi/17)*nn);
subplot(2,2,2);stem(nn, x2);title('Sequencia 2');xlabel('$n$');ylabel('$x_2[n]$');

%sequencia 3
L = 21;
nn = -10:10;
x3 = sin((pi/17)*nn + (pi/2)*ones(L,1));
subplot(2,2,3);stem(nn, x3);title('Sequencia 3');xlabel('$n$');ylabel('$x_3[n]$');

%sequencia 4
L = 51;
nn = 0:50;
x4 = cos((pi/sqrt(23))*nn);
subplot(2,2,4);stem(nn, x4);title('Sequencia 4');xlabel('$n$');ylabel('$x_4[n]$');
```

Os gráficos das sequências estão na figura 4 exposta abaixo.

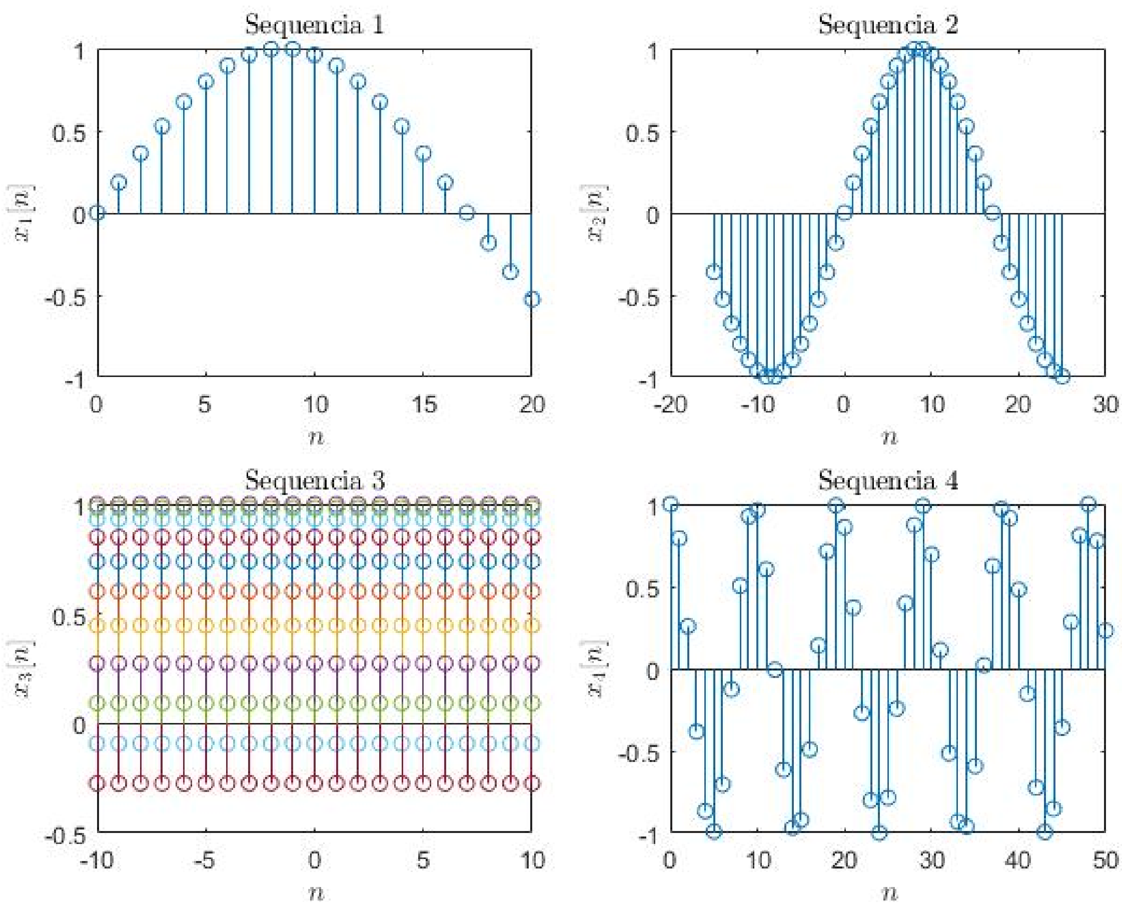

Figura 4: Gráficos das quatro sequências nos seus domínios correspondentes

Uma forma simplificada de escrever $x_3[n]$ é

$$x_3[n] = sin(3\pi n + \frac{\pi}{2}) = sin(3\pi n)cos(\frac{\pi}{2}) + cos(3\pi n)sen(\frac{\pi}{2}) = cos(3\pi n) \therefore$$

$$\therefore x[n] = \begin{cases} 1 & \text{se } n \text{ é par} \\ -1 & \text{se } n \text{ é impar} \end{cases}$$

Além disso, a sequência x_4 não é periódica pois o período fundamental P de um cosseno da forma $Acos(\omega t + \phi)$ é dado por

$$P = \frac{2\pi}{\omega}$$

mas para o caso da sequência, isso resultaria num período múltiplo de $\sqrt{23}$, que nunca é inteiro.

2.2 b)

A função *gerarSinusoideFinita* foi implementada com o código 5 apresentado abaixo

Listing 5: Função geradora de sinusoides finitos

```
function x = gerarSinusoideFinita(A, w, phi, ni, nf)
% gerarSinusoideFinita(A, w, phi, ni, nf): gera um vetor coluna da forma
% x[n,1] = A*cos(w*n + phi) variando de n = ni ate n = nf
if ni > nf
    error('ni deve ser menor ou igual a nf');
else
    nn = (ni : nf)';
    x = A*cos(w*nn + phi*ones(nf-ni+1, 1));
end
end
```

Testamos a função para alguns conjuntos de parâmetros de entrada, bem como aquele pedido pelo enunciado (sequência 6). Isto é feito pelo código 6 mostrado abaixo.

Listing 6: Teste da função *gerarSinusoideFinita* para alguns conjuntos de parâmetros de entrada

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

%vetor tempo
nn = -20:20

x1 = gerarSinusoideFinita(1, pi/20, 0, -20, 20);
x2 = gerarSinusoideFinita(1, 0.125, pi/4, -20, 20);
x3 = gerarSinusoideFinita(1.5, 0.25, pi/2, -20, 20);
x4 = gerarSinusoideFinita(1.5, 0.25, pi, -20, 20);
x5 = gerarSinusoideFinita(2, 0.5, pi, -20, 20);
x6 = gerarSinusoideFinita(2, pi/11, 3*pi/2, -20, 20);

%graficos
plot(nn, x1);hold on;plot(nn, x2);plot(nn, x3);plot(nn, x4);plot(nn, x5);plot(nn, x6);
title('gerarSinusoideFinita');xlabel('$n$');ylabel('$x_k[n]$');
legend('A=1, w=pi/20, phi=0', 'A=1, w=0.125, phi=pi/4', 'A=1.5, w=0.25, phi=pi/2', 'A=1.5, w=0.25, phi=pi', 'A=2, w
     =0.5, phi=pi', 'A=2, w=pi/11, phi=3*pi/2');
```

Os gráficos obtidos estão expostos na figura 5.

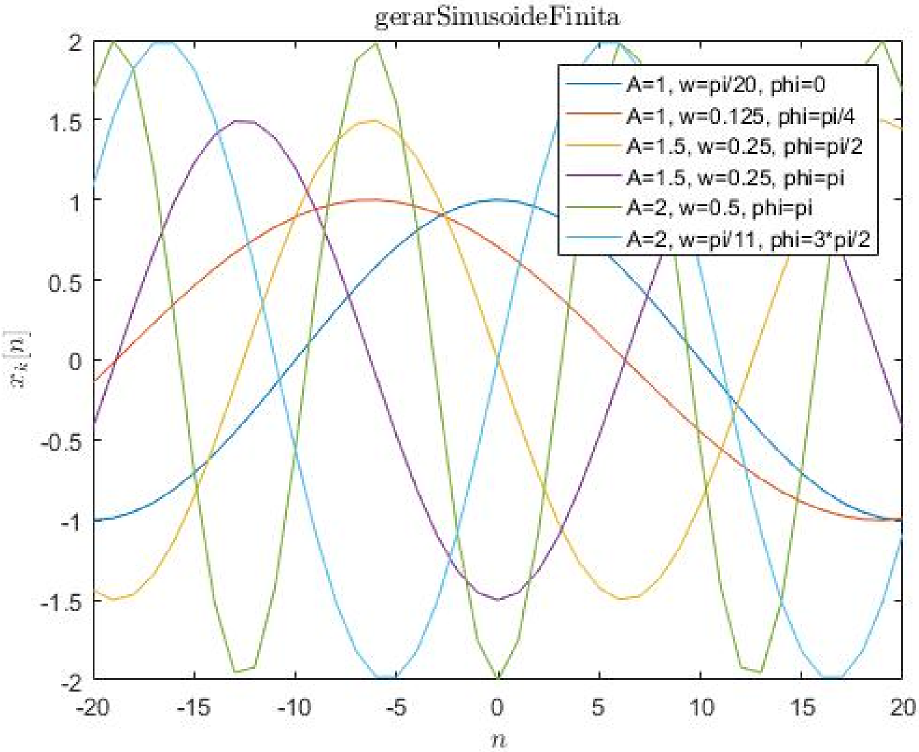

Figura 5: Gráficos das sequências geradas pela função geradora de sinusoide

Neste caso, preferimos utilizar a função *plot* do MATLAB para facilitar a visualização das ondas.

2.3 c)

Reescrevemos a função *gerarSinusoideFinita* para que ela também devolvesse um vetor de índices sobre o intervalo de n. O código é apresentado abaixo.

Listing 7: Função *gerarSinusoideFinita2*, que agora também devolve um vetor dos índices n

```
function [nn, x] = gerarSinusoideFinita2(A, w, phi, ni, nf)
% [nn, x] = gerarSinusoideFinita2(A, w, phi, ni, nf): gera um vetor linha nn com
% os valores de n, indo de ni ate nf, e um vetor coluna
% x[n] = A*cos(w*n + phi)

if ni > nf
    error('ni deve ser menor ou igual a nf');
else
    nn = ni : nf;
    x = A*cos(w*nn' + phi*ones(nf-ni+1, 1));
end
end
```

3 Sinusoides amostrados

3.1 a)

Utilizamos o código abaixo para criar a função *sinusoidesAmostradas*, que amostra um sinal sinusoidal

Listing 8: Função *sinusoidesAmostradas*

```
function s = sinusoidesAmostradas(A, f0, phi, ti, tf, fs)
% sinusoidesAmostradas(A, f0, phi, ti, tf, fs): amostra (a uma taxa fs) uma
% sinusoide da foram A*cos(2*pi*f0*t + phi), com t (em segundos) variando de ti a
% tf

if ti > tf
    error('ti deve ser menor ou igual a tf');
else
    N = (tf - ti)*fs;
    tt = ti : 1/fs : tf;
    s = A*cos(2*pi*f0*tt' + phi*ones(N+1, 1));
end
end
```

Utilizamos a função *sinusoidesAmostradas* para amostrar o sinal descrito pelo enunciado, conforme o código abaixo.

Listing 9: Plotagem dos gráficos em função do tempo e do índice

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

s = sinusoidesAmostradas(50, 1200, pi/4, 0, 0.007, 8000);

%grafico em funcao do tempo
tt = 0 : 1/8000 : 0.007;
figure (1);plot(tt, s, 'b*-');title('Amostra em funcao do tempo');xlabel('$t (segundos)$');ylabel('$s(t)$');

%grafico em funcao do indice
nn = 1 : (0.007*8000+1)
figure (2);stem(nn, s);title('Amostra em funcao do indice');xlabel('$n$');ylabel('$s[n]$');
```

O gráfico obtido em função do tempo foi

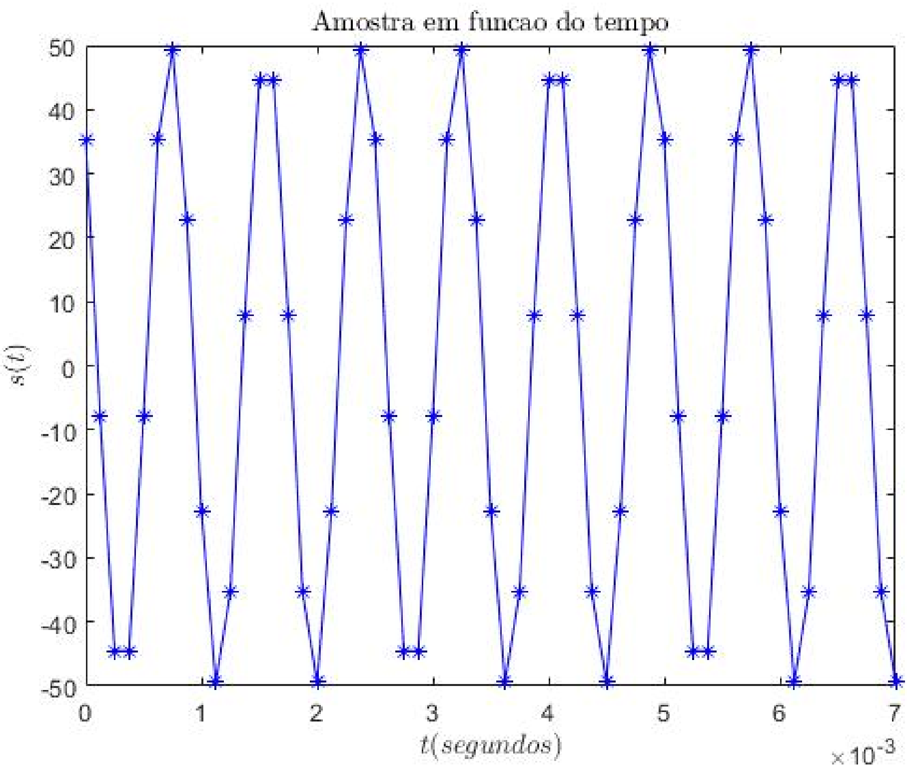

Figura 6: Gráfico da amostra em função do tempo

e o gráfico obtido em função do índice foi

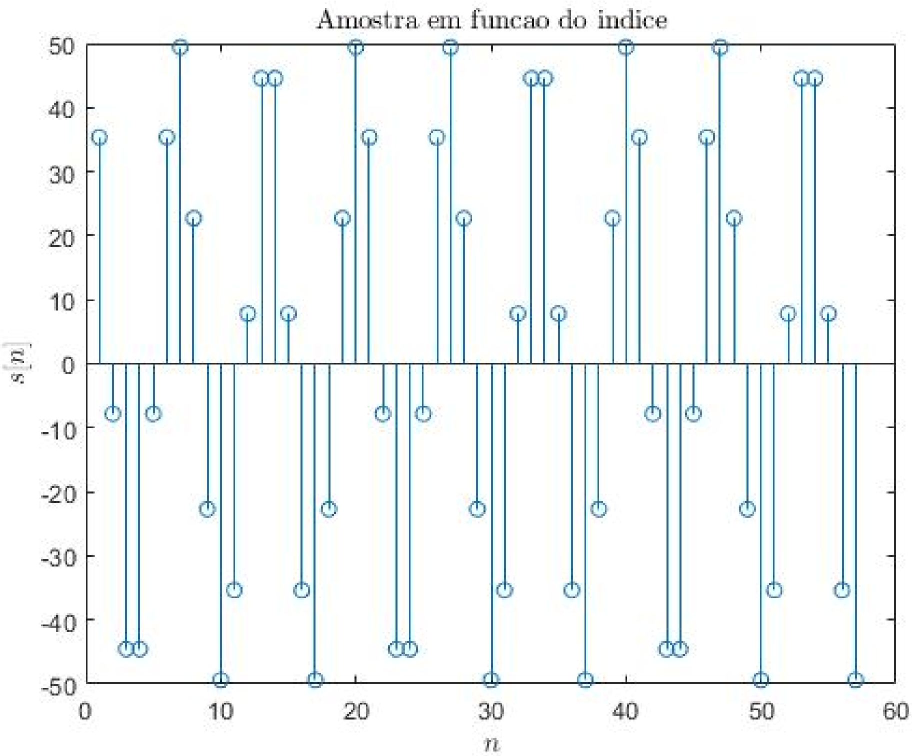

Figura 7: Gráfico da amostra em função do índice

O sinal discreto resultante tem comprimento 57 e possui 8 períodos incluídos.

3.2 b)

Se $t_n = nT = \frac{n}{f_0}$ e $\phi = \frac{3\pi}{2}$, então

$$s(t_n) = s[n] = Acos(2\pi f_0 t_n + \phi) = Acos(2\pi n + \phi) = Asen(2\pi n),$$

que é uma onda senoidal.

Utilizamos o código 10 para gerar uma onda senoidal discreta, começando em $t = 5s$ e terminando em $t = 15s$. Para isso apenas precisamos inserir uma fase inicial $\phi = \frac{3\pi}{2}$. A onda é da forma $30sin(0.4\omega t)$.

Listing 10: Gerando uma onda senoidal discreta

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

s = sinusoidesAmostradas(30, 0.2, 3*pi/2, 5, 15, 10);
tt = 5 : 1/10 : 15
stem(tt, s);title('Amostragem de $30sin(0.4\pi t)$');xlabel('$t (segundos)$');ylabel('$s(t)$');
```

O gráfico obtido está apresentado na figura 8.

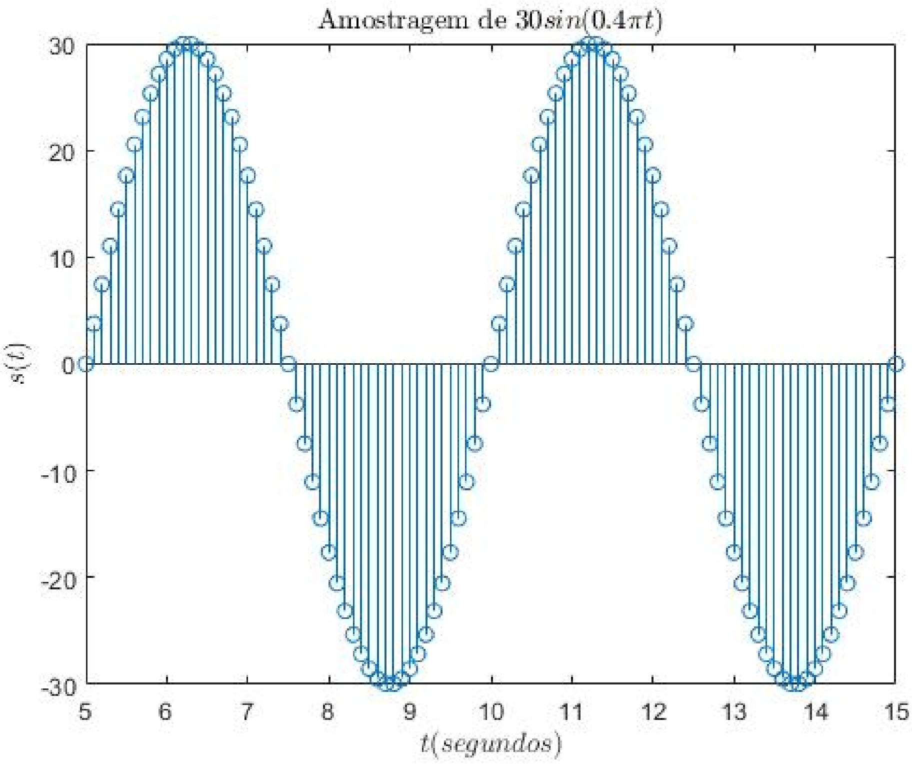

Figura 8: Gráfico de uma onda senoidal discreta

4 Exponenciais complexos

4.1 a)

$$z_0 = 0,9\angle 45^o = 0,9e^{j\frac{\pi}{4}} \therefore x[n] = z_0^n = 0,9^n e^{j\frac{\pi}{4}n} = 0,9^n(cos(\frac{\pi}{4}n) + jsin(\frac{\pi}{4}n))$$

$$Re(x[n]) = 0,9^n cos(\frac{\pi}{4}n) \text{ e } Im(x[n]) = 0,9^n sin(\frac{\pi}{4}n)$$

O código 11 foi utilizado para implementar a exponencial complexa, bem como os gráficos de sua parte real e imaginária.

Listing 11: Implementação e gráficos da exponencial complexa

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

%exponencial complexa
nn = 0 : 20;
xx= exp(nn).^(log(0.9)).*exp(j*(pi/4)*nn );

%grafico
subplot (2,1,1);stem(nn , real(xx));title('Real part');xlabel('$n$');
subplot(2,1,2);stem(nn , imag(xx));title('Imaginary part');xlabel('$n$');
```

Os gráficos obtidos estão na figura 9.

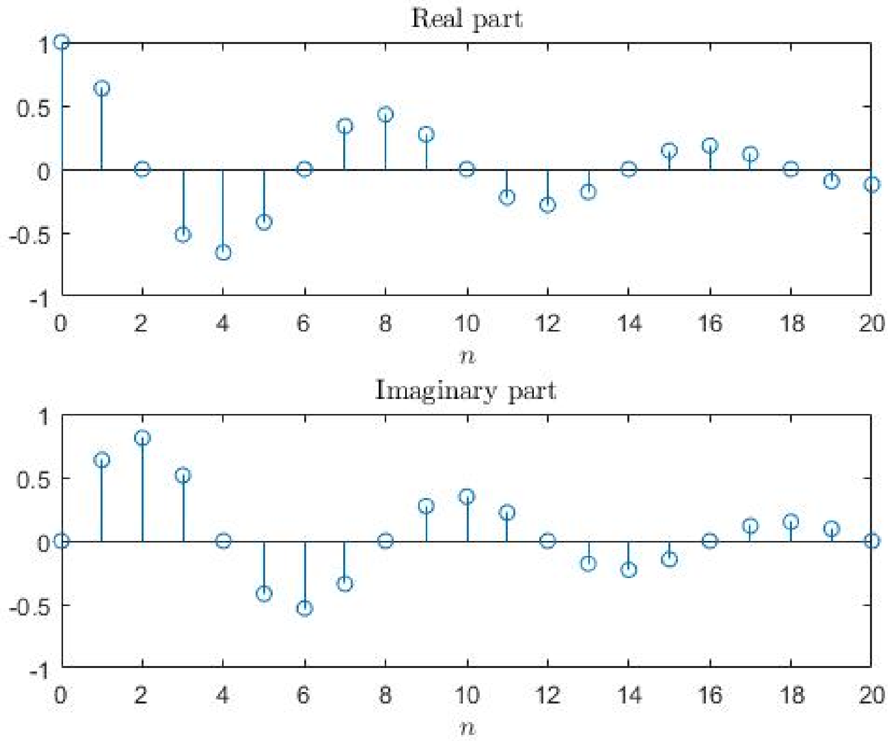

Figura 9: Parte real e parte imaginária da exponencial complexa

4.2 b)

No código 12, fizemos gráficos da parte imaginária pela parte real para alguns valores de θ. A sequência do item (a) é a terceira sequência ($\theta = \frac{\pi}{4}$).

Listing 12: teste

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

nn = 0 : 20;
%theta = pi/2
xx1= exp(nn).^(log(0.9)).*exp(j*(pi/2)*nn);
subplot(2,2,1);plot(imag(xx1), real(xx1));title('$\pi/2$');xlabel('Real part');ylabel('Imaginary part');

%theta = pi/3
xx2= exp(nn).^(log(0.9)).*exp(j*(pi/3)*nn);
subplot(2,2,2);plot(imag(xx2), real(xx2));title('$\pi/3$');xlabel('Real part');ylabel('Imaginary part');

%theta = pi/4
xx3= exp(nn).^(log(0.9)).*exp(j*(pi/4)*nn);
subplot(2,2,3);plot(imag(xx3), real(xx3));title('$\pi/4$');xlabel('Real part');ylabel('Imaginary part');

%theta = pi/5
xx4= exp(nn).^(log(0.9)).*exp(j*(pi/5)*nn);
subplot(2,2,4);plot(imag(xx4), real(xx4));title('$\pi/5$');xlabel('Real part');ylabel('Imaginary part');
```

Os gráficos obtidos estão na figura 10.

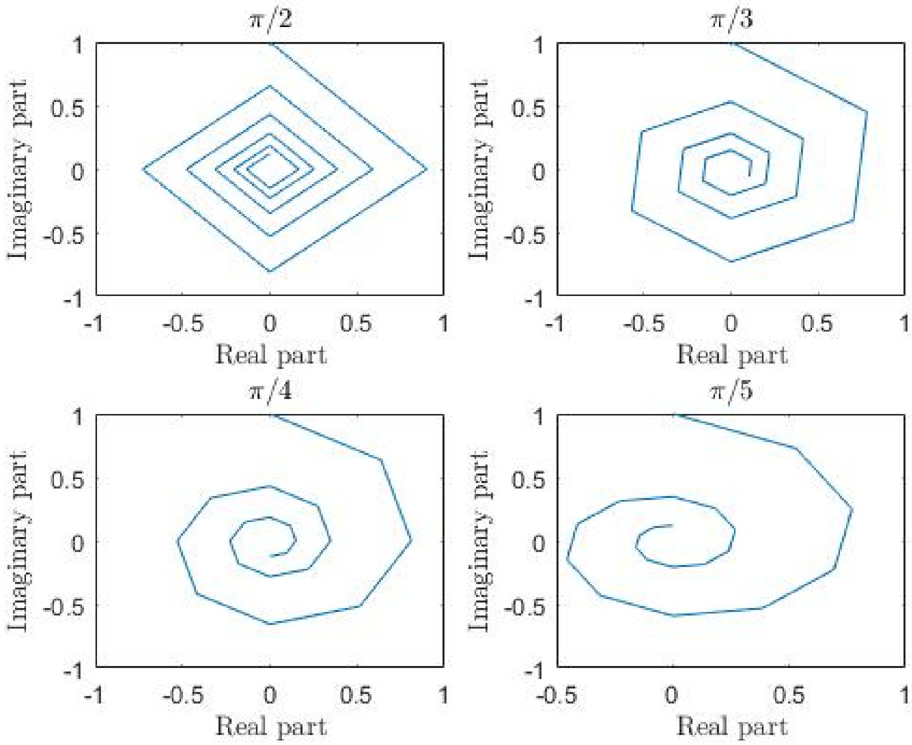

Figura 10: Parte imaginária versus parte real para alguns valores de θ

4.3 c)

Para $z_k[n] = A_k r_k^n cos(\theta_k n + \phi_k) + jsen(\theta_k n + \phi_k)$, temos

$$A = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \ r = \begin{bmatrix} 1 \\ 1 \\ 1,1 \\ 0,9 \end{bmatrix}, \ \theta = \begin{bmatrix} -\frac{\pi}{7} \\ -\frac{\pi}{17} \\ \frac{\pi}{11} \\ \frac{\pi}{11} \end{bmatrix} \text{ e } \phi = \begin{bmatrix} \frac{\pi}{2} \\ \frac{\pi}{2} \\ \frac{\pi}{4} \\ 0 \end{bmatrix}$$

Implementamos as sequência e obtivemos seus gráficos através do código 13.

Listing 13: Implementação e plotagem das quatro exponenciais complexas

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

%sequencia 1
n1 = 0 : 20;
x1 = 3*exp(j*((-pi/7)*n1 + (pi/2)*ones(1, length(n1))));
subplot(2,2,1);plot(real(x1), imag(x1));title('Sequencia 1');xlabel('Real part');ylabel('Imaginary part');

%sequencia 2
n2 = -15 : 25;
x2 = real(exp(j*((-pi/17)*n2 + (pi/2)*ones(1, length(n2)))));
subplot(2,2,2);stem(n2, x2);title('Sequencia 2');xlabel('$n$');ylabel('$x_2[n]$');

%sequencia 3
n3 = 0 : 50;
x3 = exp(n3).^(log(1.1)).*real(exp(j*((pi/11)*n3 + (pi/4)*ones(1, length(n3)))));
subplot(2,2,3);stem(n3, x3);title('Sequencia 3');xlabel('$n$');ylabel('$x_3[n]$');

%sequencia 4
n4 = -10 : 20;
x4 = exp(n4).^(log(0.9)).*real(exp((pi/11)*n4*j));
subplot(2,2,4);stem(n4, x4);title('Sequencia 4');xlabel('$n$');ylabel('$x_4[n]$');
```

Os gráficos obtidos estão expostos na figura 11

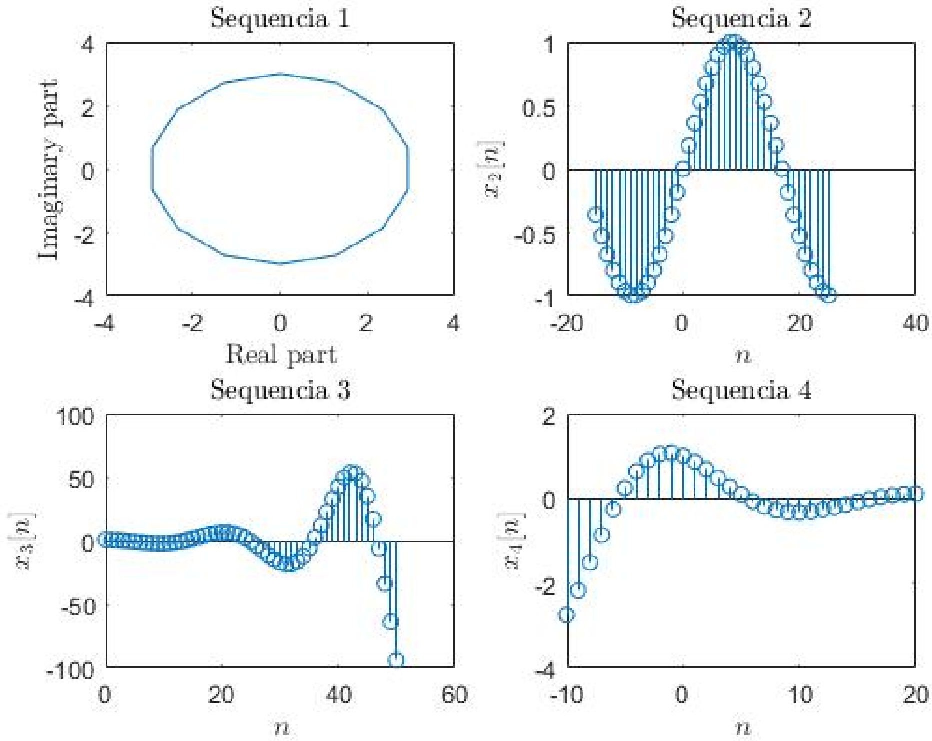

Figura 11: Gráficos das exponenciais complexas. Somente a sequência 1 possui parte real e parte imaginária. As demais possuem apenas parte real.