Laboratório de Sinais e Sistemas de Tempo Discreto

Exemplo 01

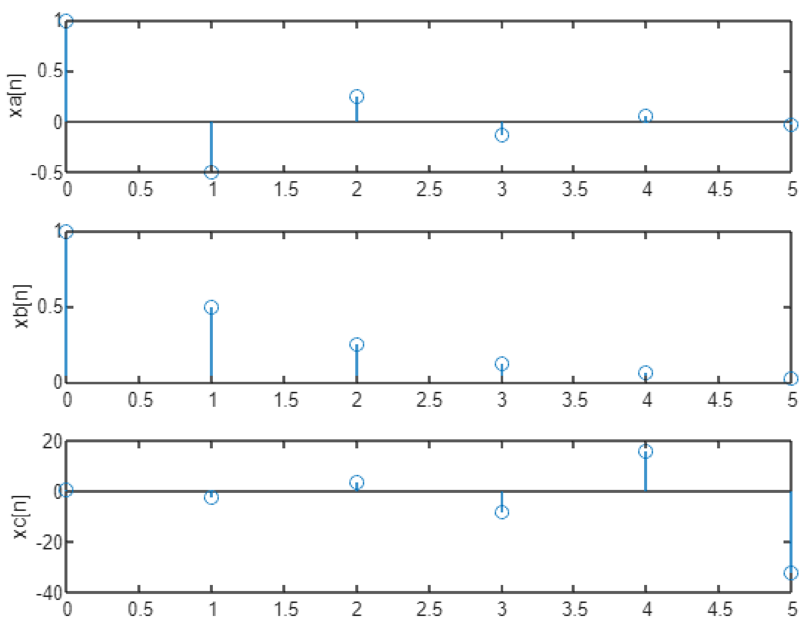
Trace os seguintes sinais discretos no tempo:

a) $x_a = (-0.5)^n$

b) $x_a = (2)^{-n}$

c) $x_a = (-2)^n$

```
n = 0:5;
%xa = (-0.5).^n;
xa=@(n) (-0.5).^n;
xb = (2).^(-n);
xc = (-2).^(n);
figure(1)
subplot(3,1,1);
%stem(n,xa);
stem(n,xa(n));
ylabel('xa[n]')
subplot(3,1,2);
stem(n,xb);
ylabel('xb[n]')
subplot(3,1,3);
stem(n,xc);
ylabel('xc[n]')
```

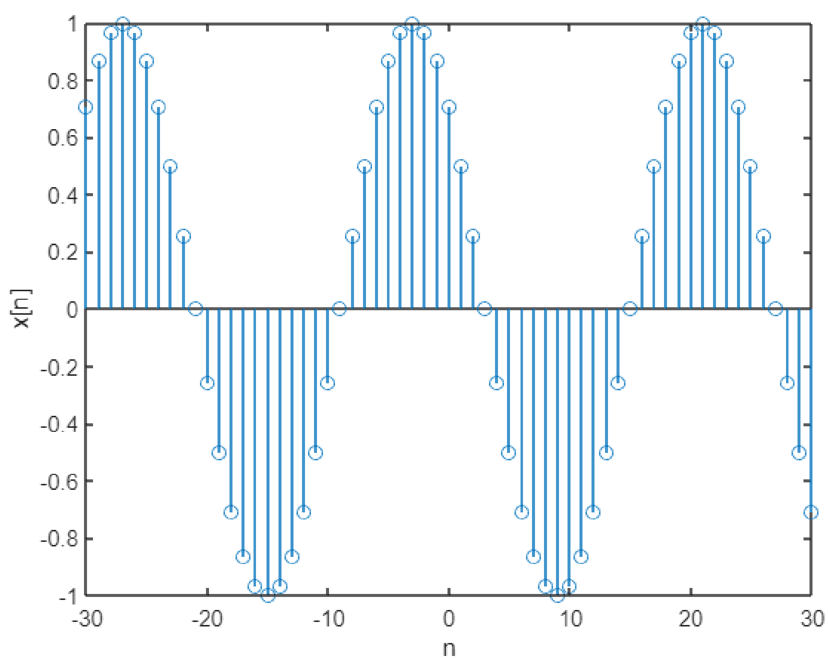


Exemplo 2

Trace a seguinte senoide discreta no tempo

$$x[n] = \cos\left(\frac{\pi}{12}n + \frac{\pi}{4}\right)$$

```
clear all
n=-30:30;
x = cos(n*pi/12+pi/4);
figure(2)
stem(n,x);
xlabel('n');
ylabel('x[n]');
```



Exemplo 3

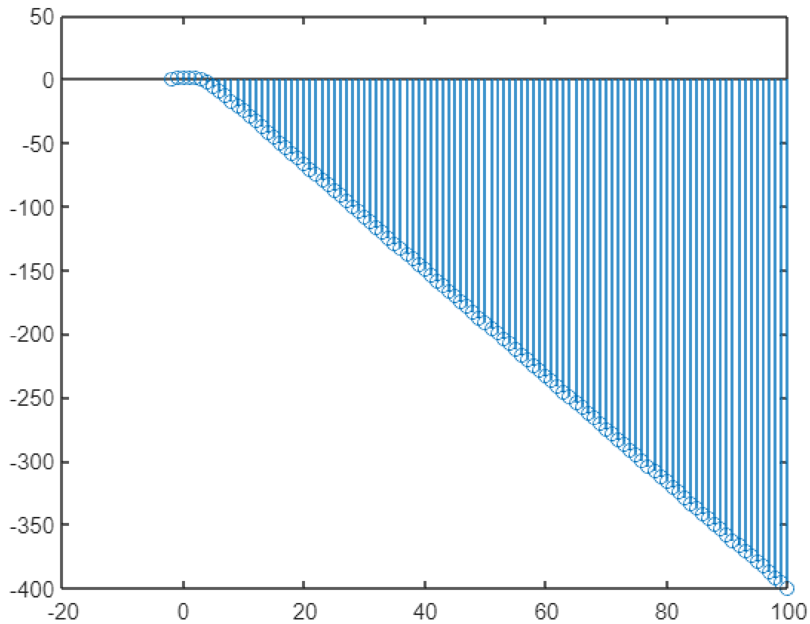
Resolva interativamente

$$y[n+2] - y[n+1] + 0.24y[n] = x[n+2] - 2x[n+1]$$

com condições iniciais $y[-1] = 2$ e $y[-2] = 1$ e entrada causal $x[n] = n$ (começando em $n = 0$).

```
n = [-2:100]';
```

```
y = [1; 2; zeros(length(n)-2,1)];
x = [0;0; n(3:end)];
for k=1:length(n)-2,
 y(k+2) = y(k+1) - 0.24*y(k) + x(k+2) - 2*x(k+1);
end
figure(3)
stem(n,y)
```

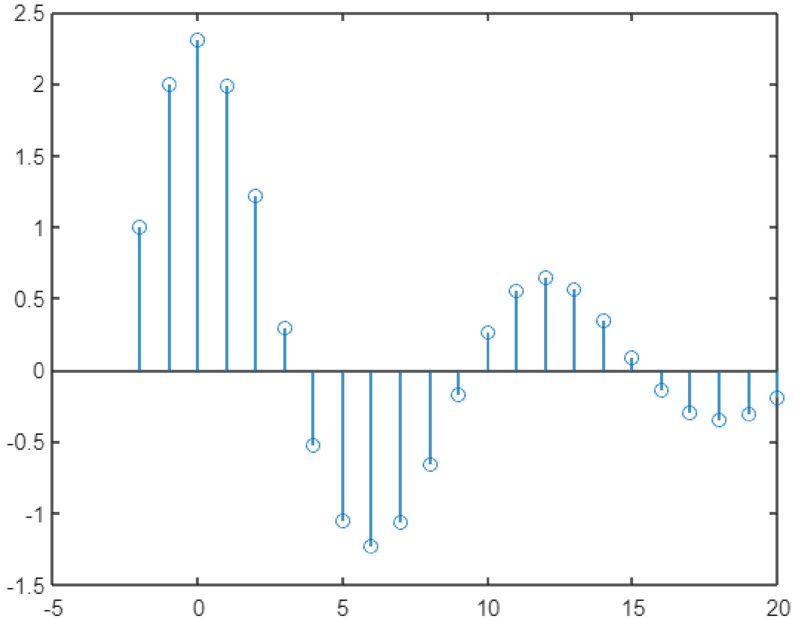


Exemplo 4

Usando as condições iniciais $y[-1]=2$ e $y[-2]=1$, determine e trace a resposta de entrada nula para o sistema descrito por

$$(E^2 - 1.56E + 0.81)y[n] = (E + 3)x[n].$$

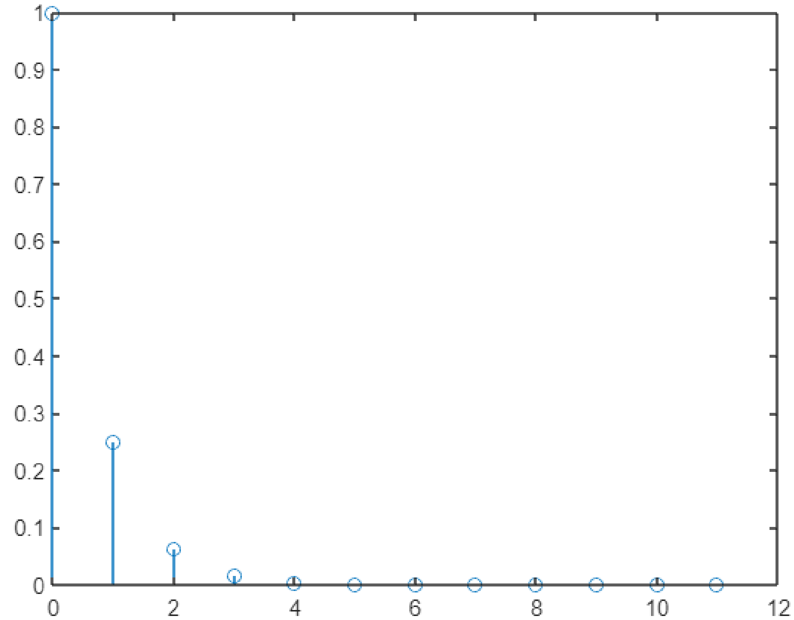
```
n = [-2:20]';
y = [1; 2; zeros(length(n)-2,1)];
for k=1:length(n)-2,
 y(k+2) = 1.56*y(k+1) - 0.81*y(k);
end
figure(4);
stem(n,y)
```



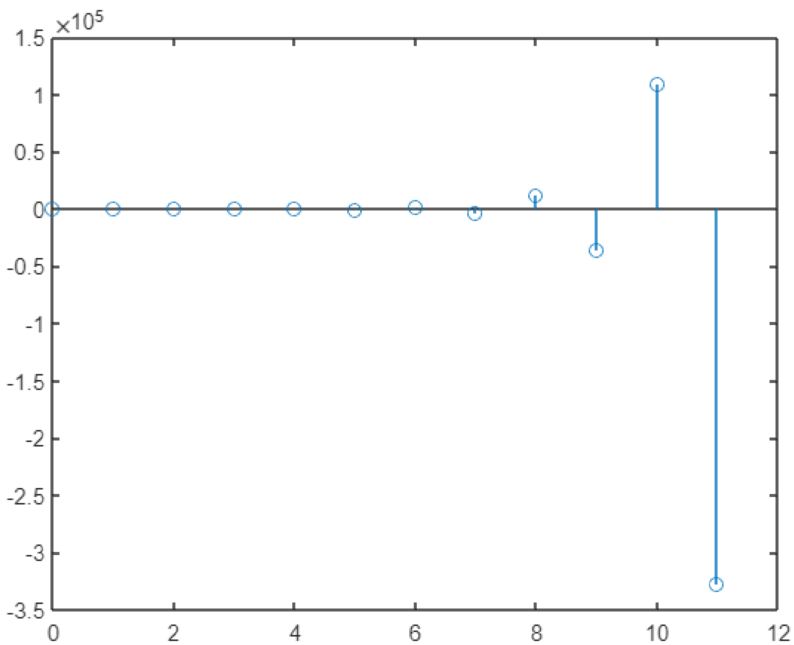
Exemplo 05

Determine e trace a resposta de estado nulo para o sistema descrito por $(E^2 + 6E + 9)y[n] = (2E^2 + 6E)x[n]$ para a entrada $x[n] = 4^{-n}u[n]$.

```
n = 0:11;
x = @(n) 4.^(-n).*(n>=0);
a = [1 6 9];
b = [2 6 0];
y = filter(b,a,x(n));
figure(5)
stem(n,x(n));
```

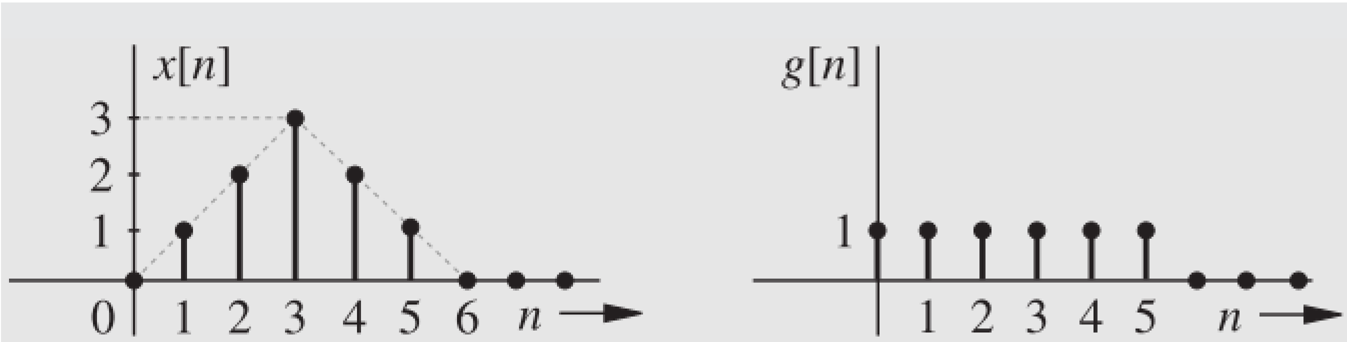



```
figure(6)
stem(n,y);
```



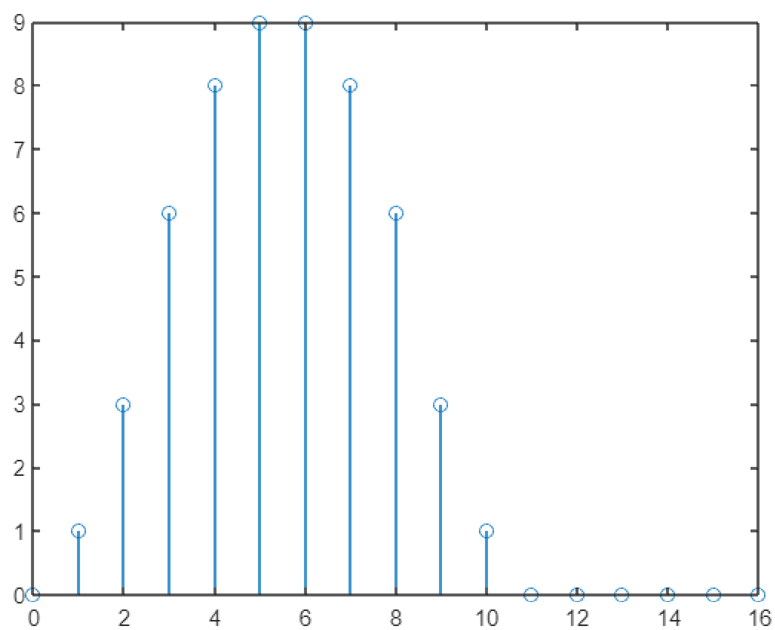
Exemplo 06

Para os sinais $x[n]$ e $g[n]$



Trace $c[n] = x[n] * g[n]$

```
x = [0 1 2 3 2 1 0 0 0];
g = [1 1 1 1 1 1 0 0 0];
n = (0:length(x)+length(g)-2);
c = conv(x,g);
figure(7)
stem(n,c);
```



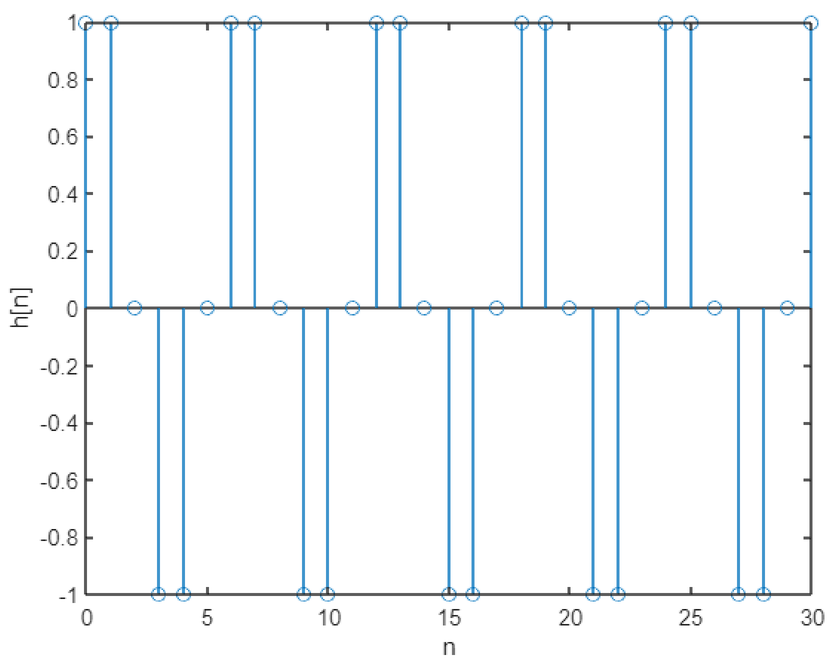
Exemplo 07

O comando filter do MATLAB é uma forma eficiente para calcular a resposta do sistema de uma equação diferença linear de coeficientes constantes representada na forma atraso como

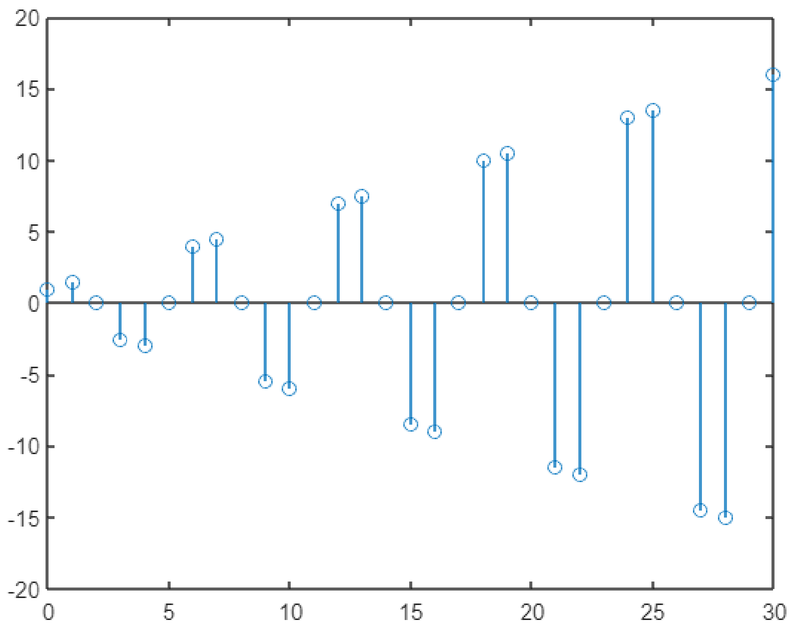
$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{N} b_k x[n-k]$$

Na forma mais simples, o comando filter necessita de três argumentos de entrada: um vetor de tamanho $N+1$ com os coeficientes de alimentação $[b_0, b_1, b_2, \ldots, b_N]$ um vetor de tamanho $N+1$ com os coeficientes de realimentação $[a_0, a_1, a_2, \ldots, a_N]$ e um vetor de entrada. Como nenhuma condição inicial foi especificada, a saída corresponde à resposta de estado nulo do sistema. A título de exemplo, considere um sistema descrito por $y[n] - y[n-1] + y[n-2] = x[n]$. Quando $x[n] = \delta[n]$, a resposta de estado nulo é igual a resposta $h[n]$ ao impulso, a qual nós calculamos para $(0 \le n \le 30)$.

```
b = [1 0 0];
a = [1 -1 1];
n = 0:30;
impulso = @(n) n==0;
h = filter(b,a,impulso(n));
figure(8)
stem(n,h);
xlabel('n');
ylabel('h[n]');
```

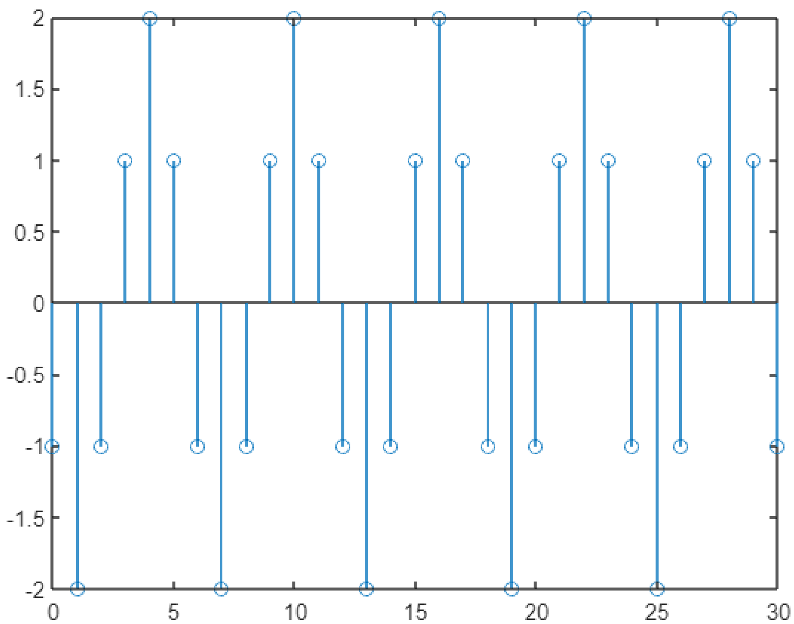



```
x = @(n) cos(2*pi*n/6).*(n>=0);
y = filter(b,a,x(n));
figure(9)
stem(n,y)
```

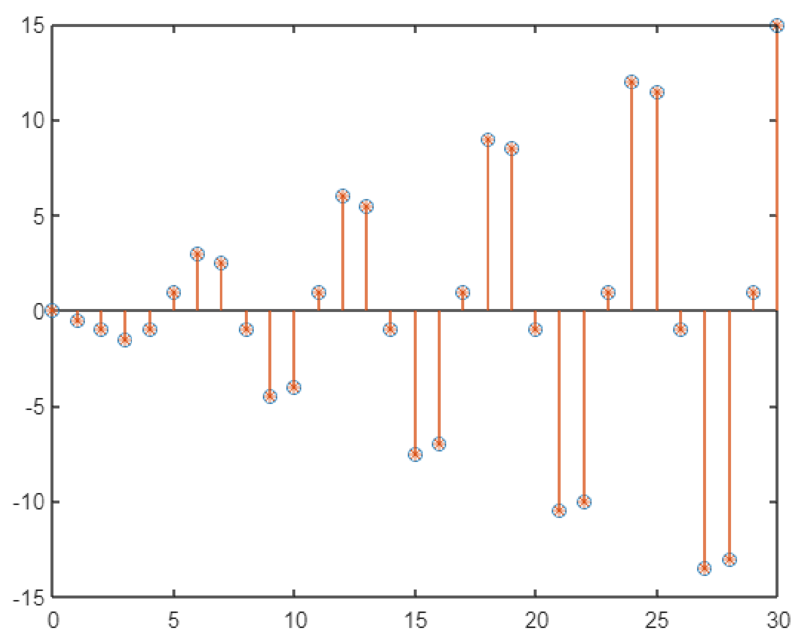


Adicionando as condições iniciais, o comando filter também pode calcular a resposta do sistema a entrada nula e a resposta total. Continuando no exemplo anterior, considere a determinação da resposta de entrada nula para $y[-1]=1$ e $y[-2]=2$ para $(0 \le n \le 30)$.

```
z_i = filtic(b,a,[1 2]);
y0 = filter(b,a,zeros(size(n)),z_i);
figure(3)
stem(n,y0)
```



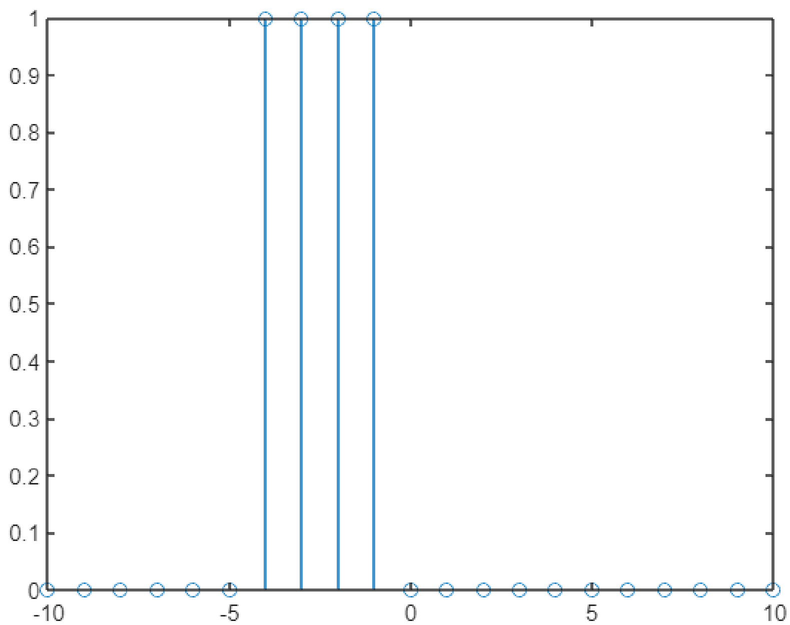

```
ytotal = filter(b,a,x(n),z_i);
figure(10)
stem(n,ytotal)
hold on
stem(n,y+y0,'*')
```



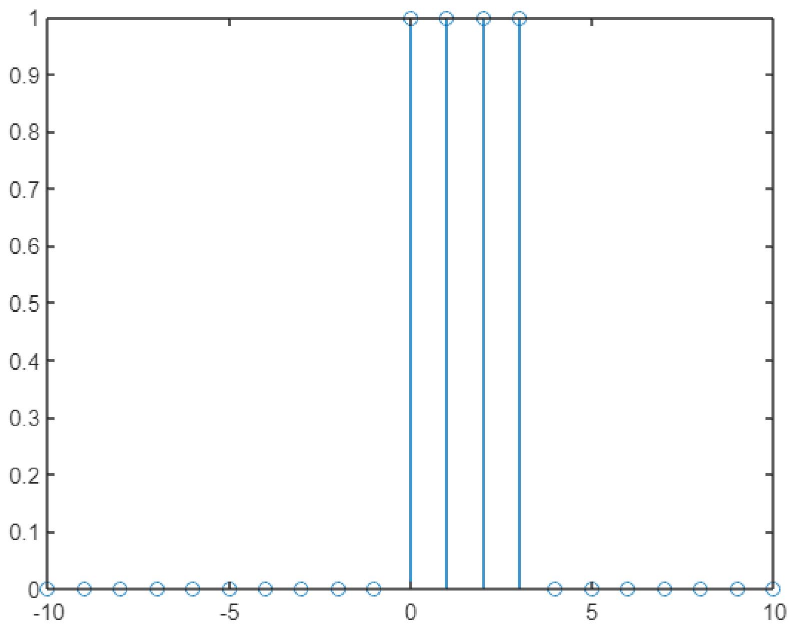
Exemplo 08

Trace a convolução de $u[n+4] - u[n]$ com $u[n] - u[n-4]$

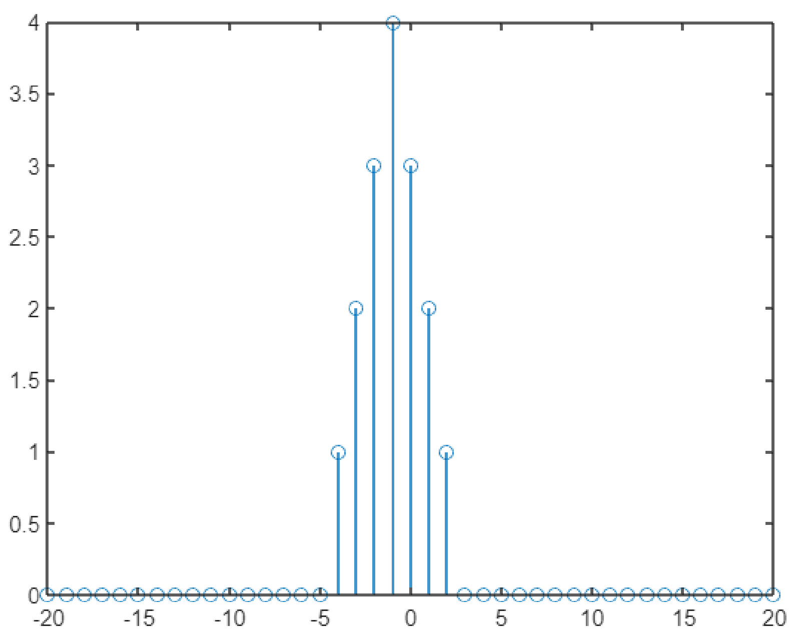
```
degrau =@(n) n>=0;
n = -10:10;
x1 = degrau(n+4) - degrau(n);
figure(11)
stem(n,x1);
```




```
x2 = degrau(n) - degrau(n-4);
figure(12)
stem(n,x2);
```



```
y = conv(x1,x2);
figure(13)
stem([-20:20],y);
```

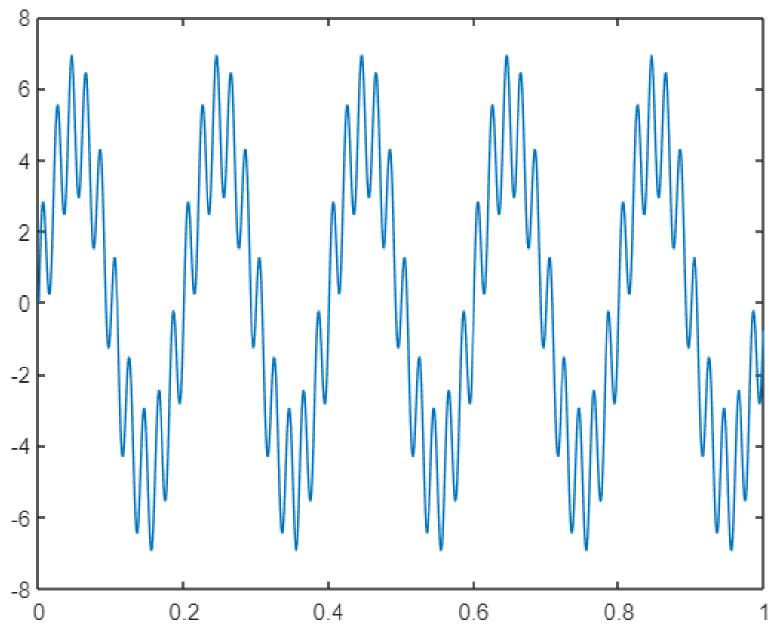


Exemplo 9

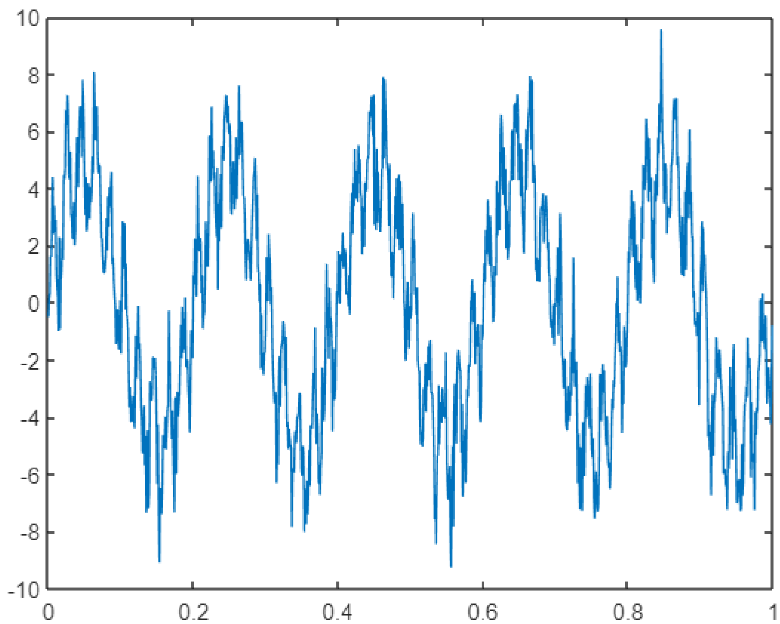
Suponha que o sinal $x(t) = \cos(2\pi 5t) + 2os(2\pi 50t)$, amostrado a $F_s = 1000\text{Hz}$ é corrompido por uma pequena quantidade de ruíıdo. Esse sinal é gerado pelos seguintes comandos:

```
amplitude_1 = 5;
```

```
freq_1 = 5;
amplitude_2 = 2;
freq_2 = 50;
Fs = 1000;
time = 0:1/Fs:(1-1/Fs);
sine_1 = amplitude_1*sin(2*pi*freq_1.*time);
sine_2 = amplitude_2*sin(2*pi*freq_2.*time);
noise = randn(1,length(time));
x_clean = sine_1 + sine_2;
x_noisy = x_clean + noise;
figure(14);
plot(time,x_clean);
```




```
figure(15);
plot(time,x_noisy);
```



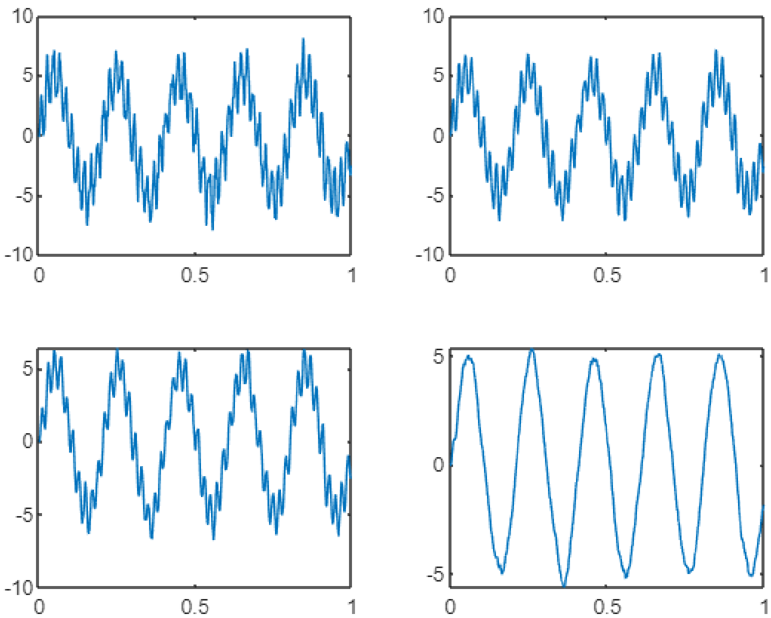
Em particular, o comando *randn* gera o número especificado de amostras de um sinal pseudo-aleatório com distribução gaussiana de média zero e variância unitária. Podemos minimizar o efeito do ruído fazendo a média de N amostras sucessivas de $x[n] = x\_\mathrm{noisy}$, implementando a seguinte equação de diferenças:

$$y[n] = \frac{x[n] + x[n-1] + x[n-2] + ... + x[n-N+1]}{N}$$

Podemos realizar esse processamento especificando o valor de N e usando os comandos abaixo:

```
N=3;
b = ones(1,N);
y = filter(b,N,x_noisy);
figure(16)
subplot(2,2,1)
plot(time,y)
N=6;
b = ones(1,N);
y = filter(b,N,x_noisy);
subplot(2,2,2)
plot(time,y)
N=10;
b = ones(1,N);
y = filter(b,N,x_noisy);
subplot(2,2,3)
plot(time,y)
N=20;
```

```
b = ones(1,N);
y = filter(b,N,x_noisy);
subplot(2,2,4)
plot(time,y)
```



Nesse caso, quanto maior o valor de N, maior a capacidade de remover a componente de ruído.

Exemplo 10

Plotar um sinal de ECG

```
Name = 'C:\Users\elenl\OneDrive\Documentos\Disciplinas\ECG\a01m';
infoName = strcat(Name, '.info');
matName = strcat(Name, '.mat');
load(matName);
fid = fopen(infoName, 'rt');
fgetl(fid);
fgetl(fid);
fgetl(fid);
[freqint] = sscanf(fgetl(fid), 'Sampling frequency: %f Hz  Sampling interval: %f sec');
interval = freqint(2);
fgetl(fid);

for i = 1:size(val, 1)
    [row(i), signal(i), gain(i), base(i), units(i)]=strread(fgetl(fid),'%d%s%f%f%s','delimiter
end
```

```
fclose(fid);
val(val==-32768) = NaN;

%% Normalizacao do sinal pelo ganho
for i = 1:size(val, 1)
    val(i, :) = (val(i, :) - base(i)) / gain(i);
end

%% Plotando sinal original
fval = 1./size(val, 1);
tam=length(val)/freqint(1,1);
t=0:tam/length(val):tam-(tam/length(val));
figure(17)
plot(t,val); grid on;
xlim([0 10])
title(['Sinal original '])
ylabel('Amplitude (Volt)')
xlabel('Tempo (s)')
```