Atividade Prática: Sinais e Sistemas

Muriel rodrigues **RU.: 2959981**

ATIVIDADE 1: OPERAÇÕES BÁSICAS

Algoritmo desenvolvido no Scinotes:

RESUMO

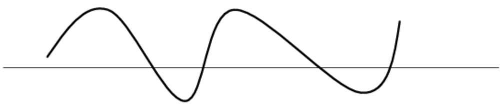
Um sinal pode ser descrito como um conjunto de dados ou informações. Podendo ser representado por um sinal em função do tempo ou do espaço.

Pontualmente, quando nos referimos a um sistema estamos dizendo que se trata de um grupo que pode processar um ou diversos sinais (sistema de entrada) e reproduzir um ou diversos sinais (sistema de saída), onde estes sistemas frequentemente ou convencionalmente extraem ou modificam informações compostas em um sinal.

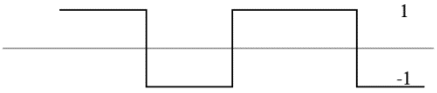
Os sinais se subdividem em contínuos e discretos; quando nos referimos a uma função de argumento real, estamos citando sinais contínuos. Em contrapartida quando nos referimos a funções de um argumento, que só podem assumir valores pertencente a um conjunto de maneira discreta, estamos nos referindo aos chamados sinais discretos no tempo.

Para ambos os valores adotados de ex.; (x), podem ser discretos ou reais.

Em sinais contínuos a amplitude pode assumir valores reais ou complexos em um instante de tempo ou para cada amostra.



Em sinais discretos a amplitude pode assumir valores reais ou complexos pertencente a um conjunto discreto.



Um dos teoremas imprescindíveis na análise de sinais e sistemas é o teorema da convolução, que é composto por duas partes, sendo a primeira abordando a transformada de Fourier da convolução e, a segunda, a parte da multiplicação de sinais.

Ex.; x1(t) * x2(t) ↔ x1(f) * x2(f)

Desta maneira os teoremas estabelecem que, a convolução no domínio do tempo transforma-se em multiplicação no domínio da frequência, enquanto a multiplicação no domínio do tempo transforma-se em convolução no domínio da frequência.

Quando abordamos sinais e sistemas não podemos deixar passar em branco o tópico de funções descontinuadas, como por ex.; função sinal, função constante, função impulso e o degrau.

Comumente, quando citamos funções nos referimos a estas como uma regra de associação de elementos de um domínio específico a um domínio de imagem.

Baseando-se na descrição anterior funções vetoriais, nada mais são que funções matemáticas de uma ou mais variáveis, cuja imagem das mesmas é um conjunto de vetores multidimensionais, enquanto o domínio é um conjunto de números reais.

Nas funções do tipo degrau a um tipo relevante de descontinuidade que surge, por exemplo na análise de circuitos elétricos, este é o de primeira espécie, ou seja, quando a função é continuada exceto por um número finito de “saltos”.

Podemos operar de maneira eficiente com esse tipo de descontinuidade usando a função degrau unitário $u \equiv u_0$

Em muitas circunstancias, a função de entrada possui um elevado valor, que atua em um curto intervalo de tempo. Matematicamente, o termo da direita da equação diferencial se forma em função de um impulso unitário, também conhecido como ‘’delta de dirac’’, retratada por δ(t).

ENUNCIADO

Criar a função impulso unitário. Criar a função degrau unitário. Gerar um vetor n de -20 até 20 com intervalo de 1.

- a = $RU1$
- b = $RU2$
- c = $RU3$/10, se $RU3$ = 0 adotar c = 0,3
- d = $RU4$/10, se $RU4$ = 0 adotar d = 0,4
- e = número e, sintaxe no Scinotes %e
- m = $RU5$, se $RU5$ = 0 adotar m = 5
- f = $RU6$, se $RU6$ = 0 adotar f = 6
- π = número π, sintaxe no Scinotes %pi
- g = $RU7$, se $RU7$ = 0 adotar g = 7

1. Gerar as seguintes funções (para $y[n]$ manter os números do RU):

   $x[n] = tg\ (an + c\pi\ 3\ )\ .e\ -dn\ -b \le n \le f$

   $y[n] = [(-RU1)\ RU4\ RU3\ RU7\ RU5\ (-RU6)\ ]$

   $z[n] = 0{,}7\ n(x[-n - RU3]) + 0{,}3n(u[n + f] - u[n - g])$

2. Calcular:

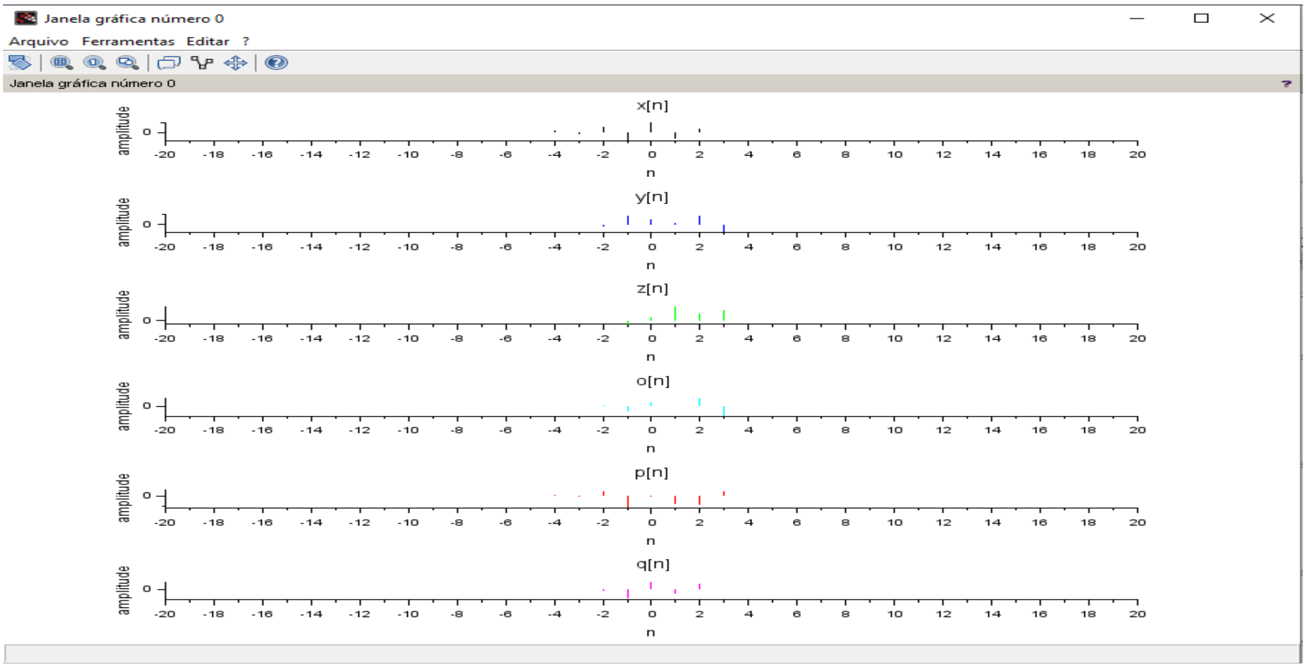
$o[n] = x[n] + y[n].\ z[n]$
$p[n] = x[n] - y[n] - z[n]$
$q[n] = x[n].\ (y[n] + z[n])$

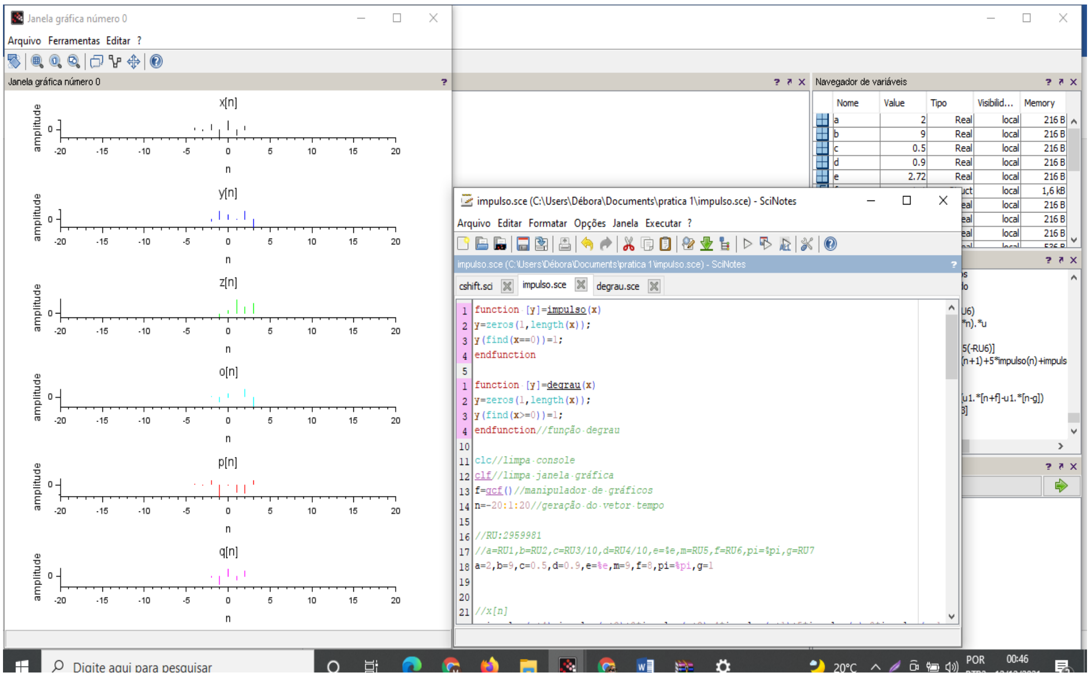
3. Plotar todos os gráficos ($x[n]$, $y[n]$, $z[n]$. $o[n]$, $p[n]$ e $q[n]$) como sinal discreto na mesma figura usando o comando subplot. Colocar os nomes nos eixos e o título de cada figura como no exemplo a seguir. Será tirada nota se a imagem não cumprir com o solicitado. Usar o comando plot2d3 para melhor visualização.

Resultado 01.:

```
function [y]=impulso(x)
y=zeros(1,length(x));
y(find(x==0))=1;
endfunction

function [y]=degrau(x)
y=zeros(1,length(x));
y(find(x>=0))=1;
endfunction//função degrau

clc//limpa console
clf//limpa janela gráfica
f=gcf()//manipulador de gráficos
n=-20:1:20//geração do vetor tempo

//RU:2959981
//a=RU1,b=RU2,c=RU3/10,d=RU4/10,e=%e,m=RU5,f=RU6,pi=%pi,g=RU7
a=2,b=9,c=0.5,d=0.9,e=%e,m=9,f=8,pi=%pi,g=1

//x[n]
x=impulso(n+4)-impulso(n+3)+3*impulso(n+2)-4*impulso(n+1)+5*impulso(n)-3*impulso(n-
1)+2*impulso(n-2)

//y[n]
y=-2*impulso(n+2)+9*impulso(n+1)+5*impulso(n)+impulso(n-1)+9*impulso(n-2)-8*impulso(n-3)

//z[n]
z=-impulso(n+1)+impulso(n)+4*impulso(n-1)+2*impulso(n-2)+3*impulso(n-3)
//z=0.7^n*(x*[-n-0.5])+0.3.*n(u2.*[n+8]-u2.*[n-1])

//x1[n]
u1=degrau(n+9)-degrau(n-8)
x1=(tan(a*n+5*%pi/3).*%e^(-0.9*n)).*u1

//z2[n]
u2=degrau(n+2)-degrau(n-6)
xa=cshift(x,[0,(0,5)]);//x[-n-0.5]
for i=-20:1:20
 z2(i+21)=(xa(-i+21)+sin(i*%pi/5+0.5))*u2(i+21)
end

//OPERAÇÕES
//o[n]
o=x+y.*z
//p[n]
p=x-y-z
//q[n]
q=x.*(y+z)
```

```
//GRÁFICOS
subplot(611)
plot2d3(n,x,style=1)//x[n]
f.children.children(1).children.thickness=2;//controla a grossura da linha
title('x[n]')
xlabel('n')
ylabel('amplitude')

subplot(612)
plot2d3(n,y,style=2)//y[n]
f.children.children(1).children.thickness=2;//controla a grossura da linha
title('y[n]')
xlabel('n')
ylabel('amplitude')

subplot(613)
plot2d3(n,z,style=3)//z[n]
f.children.children(1).children.thickness=2;//controla a grossura da linha
title('z[n]')
xlabel('n')
ylabel('amplitude')

subplot(614)
plot2d3(n,o,style=4)//o[n]
f.children.children(1).children.thickness=2;//controla a grossura da linha
title('o[n]')
xlabel('n')
ylabel('amplitude')

subplot(615)
plot2d3(n,p,style=5)//p[n]
f.children.children(1).children.thickness=2;//controla a grossura da linha
title('p[n]')
xlabel('n')
ylabel('amplitude')

subplot(616)
plot2d3(n,q,style=6)//q[n]
f.children.children(1).children.thickness=2;//controla a grossura da linha
title('q[n]')
xlabel('n')
ylabel('amplitude')
```





ATIVIDADE 2: SISTEMAS LINEARES – CONVOLUÇÃO

Algoritmo desenvolvido no Scinotes:

ENUNCIADO

- Para $x[n]$, $h_1[n]$ e $h_2[n]$ gere um vetor n de -5 até 5 com intervalo de 1
- Para $y[n]$ gere um vetor $n1$ de -10 até 10 com intervalo de 1

1. Sendo as funções:

$$x[n] = e^{-dn}\cos\left(m.n+\frac{\pi}{3}\right) \quad -4 \le n < RU1$$

$$h_1[n] = [RU1\ RU2\ RU3\ ]$$

$$h_2[n] = sen(c\pi n) \ -2 \le n < RU1$$

a. Calcule $y[n]$ como indicado na equação a seguir.

$$y[n] = x[n] * (h_1[n] - h_2[n])$$

i. Resolução matemática (pode ser gráfica)

ii. Algoritmo.

iii. Plote $x[n]$, $y[n]$, $h_1[n]$ e $h_2[n]$ no mesmo gráfico usando o comando *subplot*. Use o comando plot2d3 para melhor visualização. Não se esqueça de colocar os nomes nos eixos das figuras (será descontada nota).

b. Calcule $y[n]$ como indicado na equação a seguir.

$$y[n] = x[n] * h_1[n] - x[n] * h_2[n]$$

i. Resolução matemática (pode ser gráfica)

ii. Algoritmo.

iii. Plote $x[n]$, $y[n]$, $h_1[n]$ e $h_2[n]$ no mesmo gráfico usando o comando *subplot*. Use o comando plot2d3 para melhor visualização. Não se esqueça de colocar os nomes nos eixos das figuras (será descontada nota).

2. Compare os sistemas dos pontos 1(a) e 1(b) e explique os resultados baseado nas propriedades da convolução se necessário.

Comparando os pontos da integral de definição, deixa claro que convolução é uma medida da área, subentendida pelo produto de determinada função y(n) com uma cópia invertida no tempo de x(n), em função da distância relativa entre ambas. Interpretando de maneira singela torna-se possível adquirir uma ampla visão intuitiva da convolução, chegando a uma conclusão que ela consiste no cálculo do valor médio da função y(n). Tratando-se de uma média ponderada com pesos dados por x(n). A inversão de y(x) não altera em nada na interpretação, pois y não tem significado físico. Inverte-se y(n) apenas por conveniência, uma vez que sem isso a operação não seria comutativa.

Resultado 02.:

```
function [y]=impulso(x)
y = zeros(1, length(x));
y(find(x==0)) = 1;
endfunction//função impulso

function [y]=degrau(x)
y=zeros(1,length(x));
y(find(x>=0))=1;
endfunction//função degrau

//RU:2959981
//a=RU1,b=RU2,c=RU3/10,d=RU4/10,e=%e,m=RU5,f=RU6,pi=%pi,g=RU7
a=2,b=9,c=0.5,d=0.9,e=%e,m=9,f=8,pi=%pi,g=1

clc//limpa console
clf//limpa janela gráfica
f=gcf()//manipulador de gráficos

//vetor n para x[n],h1[n],h2[n]
n=-5:1:5

//vetor n1 para y[n]
n1=-10:1:10

//x[n]
u1=degrau(n+2)-degrau(n-2)
x=(%e^(-0.9*n)+cos(9*n+%pi/3)).*u1

//h1[n]
h1=2*impulso(n)+9*impulso(n-1)+0.5*impulso(n-2)

//h2[n]
u2=degrau(n+2)-degrau(n-2)
h2=sin(0.5*%pi*n).*u2

//ya[n]

ya=(x.*(h1-h2))

//yb[n]
yb=(x.*(h1)-x.*(h2))

//GRÁFICOS
subplot(511)
plot2d3(n,x,style=1)//x[n]
f.children.children(1).children.thickness=2;//controla a grossura da linha
title('x[n]')
xlabel('n')
ylabel('amplitude')

subplot(512)
plot2d3(n,ya,style=2)//y[n]
f.children.children(1).children.thickness=2;//controla a grossura da linha
title('ya[n]')
xlabel('n')
ylabel('amplitude')

subplot(513)
plot2d3(n,yb,style=2)//y[n]
f.children.children(1).children.thickness=2;//controla a grossura da linha
title('yb[n]')
xlabel('n')
ylabel('amplitude')

subplot(514)
plot2d3(n,h1,style=3)//h1[n]
f.children.children(1).children.thickness=2;//controla a grossura da linha
```

```
title('h1[n]')
xlabel('n')
ylabel('amplitude')

subplot(515)
plot2d3(n,h2,style=4)//h2[n]
f.children.children(1).children.thickness=2;//controla a grossura da linha
title('h2[n]')
xlabel('n')
ylabel('amplitude')
```