

Sinais e Sistemas

Laboratório - Convolução de Sinais

I – OBJETIVO

Empregar o Método Gráfico para a convolução de sinais no Octave (Scilab)

II - INTRODUÇÃO

A convolução de sinais ocorre em vários contextos no processamento de sinais. A característica de entrada e saída dos sistemas lineares invariantes no tempo (LTI), o tipo de sistema mais amplamente usado no processamento de sinais, é descrita inteiramente em termos de resposta ao impulso do sistema. A resposta ao impulso é a saída do sistema devido a um sinal de entrada de impulso. Dado o sinal de entrada, a saída de um sistema LTI é a convolução do sinal de entrada com a resposta ao impulso do sistema. Portanto, a convolução desempenha um papel fundamental no relacionamento dos sinais de entrada e saída de um sistema LTI.

Uma vez que se está trabalhando com um computador, versões de tempo discreto de sinais de tempo contínuo serão considerados. É importante notar que a convolução em sistemas de tempo contínuo não podem ser replicados exatamente em um sistema de tempo discreto. Contudo, pode-se explorar os efeitos básicos e obter alguma compreensão da convolução usando função `conv` da Linguagem de Programação Científica Octave (ou Scilab).

Neste laboratório, será enfatizado o Método Gráfico para a convolução de sinais no Octave (ou Scilab). Ao escrever e modificar os programas correspondentes executados no ambiente Octave (ou Scilab), os alunos irão dominar cada etapa do processo de convolução.

III – PROCEDIMENTOS

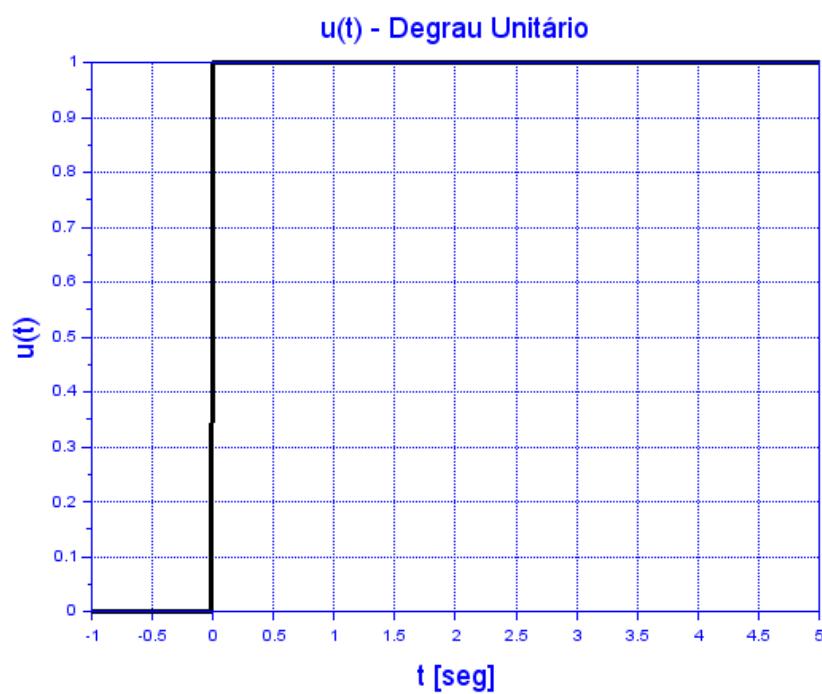
Manipulação e convolução de função contínua

Nesta parte, os sinais contínuos são tratados internamente pelo Octave (ou Scilab) como sinais discretos. O aluno irá operar com os *scripts* (códigos) fornecidos como se fossem sinais contínuos, visto que o Octave (ou Scilab) funcionará com versões de amostra dos sinais contínuos.

Neste contexto, a convolução de dois sinais podem ser calculados usando as seguintes etapas descritas no *scripts* a seguir:

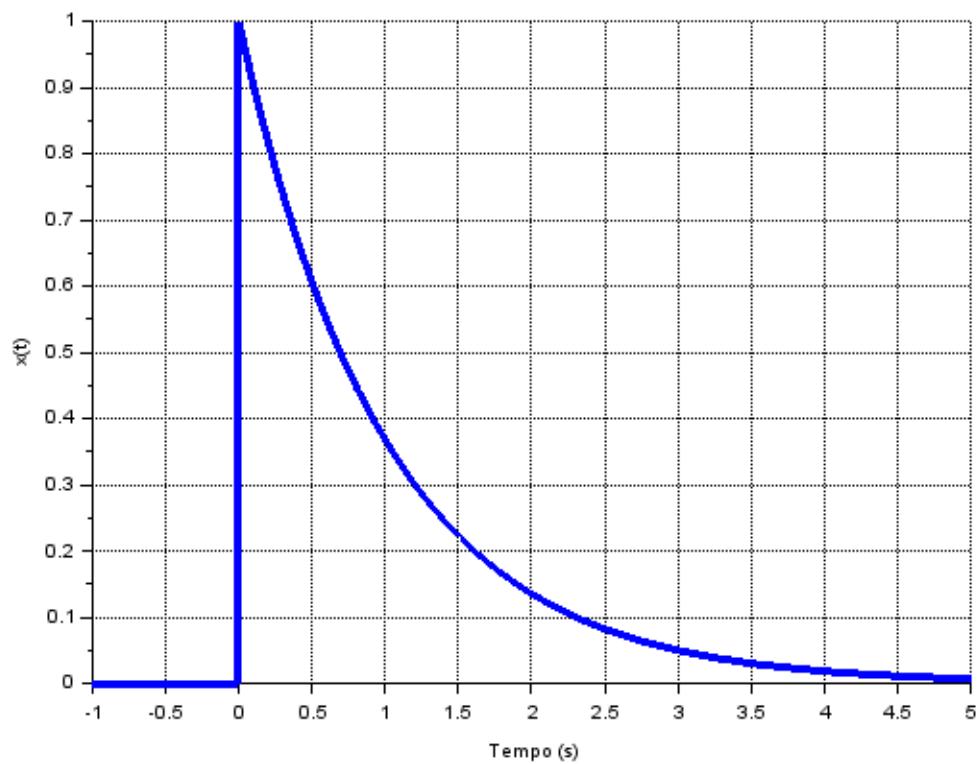
```
//  
// Inicialização - definindo a "função" degrau unitário - Figura 1  
//
```

```
t = -1:0.01:5;  
t1 = -1:0.01:-0.01;  
t2 = 0:0.01:5;  
u1(1,1:max(size(t1))) = 0;  
u2(1,1:max(size(t2))) = 1;  
u = [u1,u2];  
figure(0)  
plot2d(t,u,2);
```



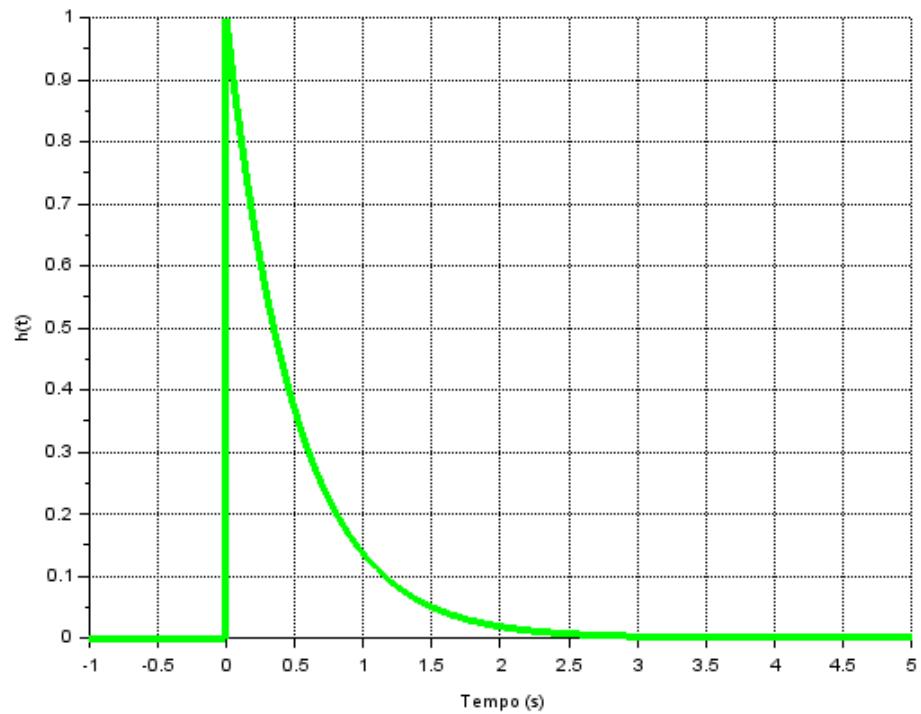
```
//Definindo a função x(t) e gerando gráfico - Figura 2
//-----
```

```
x = exp(-t) .*u;
figure(1);
plot2d(t,x,2);
```



```
//  
// Definindo a função h(t) e gerando gráfico - Figura 2  
//
```

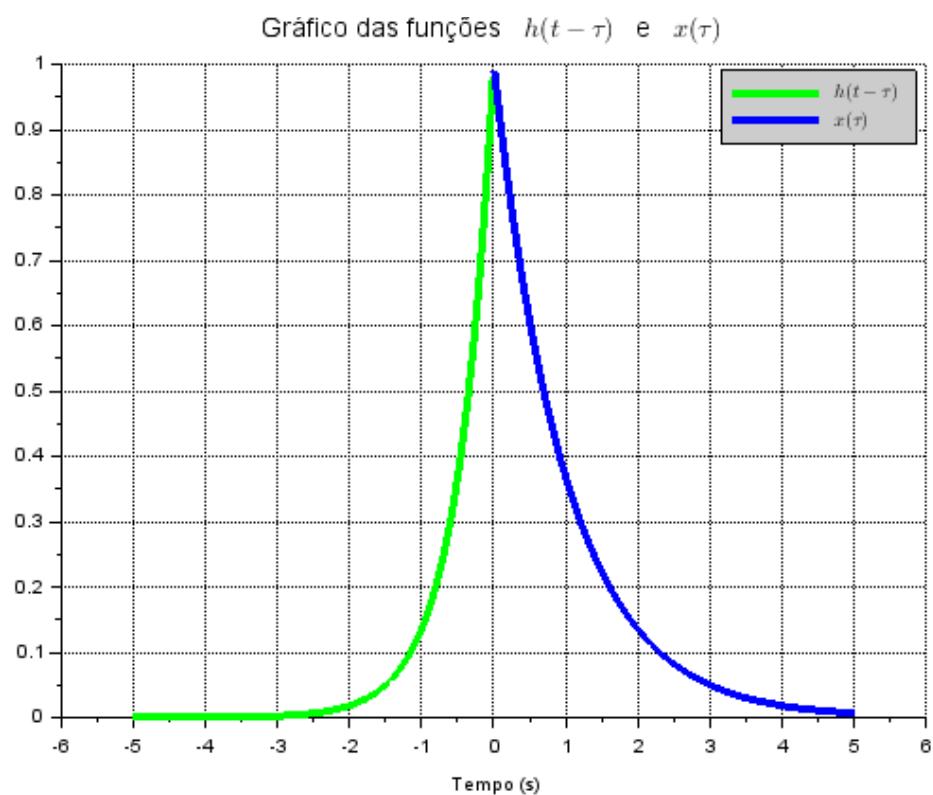
```
figure(2);  
h = exp(-2*t) .*u;  
plot2d(t,h,3);  
p = gce();
```



```

//-
// Definindo a função x(t) para t > 0
// Realizando a reflexão de h(t) --> h(-t) e gerando gráfico - Figura 3
//-
figure(3);
t = 0.01:0.01:5;
u1 = mtlb_i(u1,1:max(size(t)),1);
x = exp(-t) .*u1;
h = exp(-2*t) .*u1;
plot2d(-t,h,3);
plot2d(t,x,2);

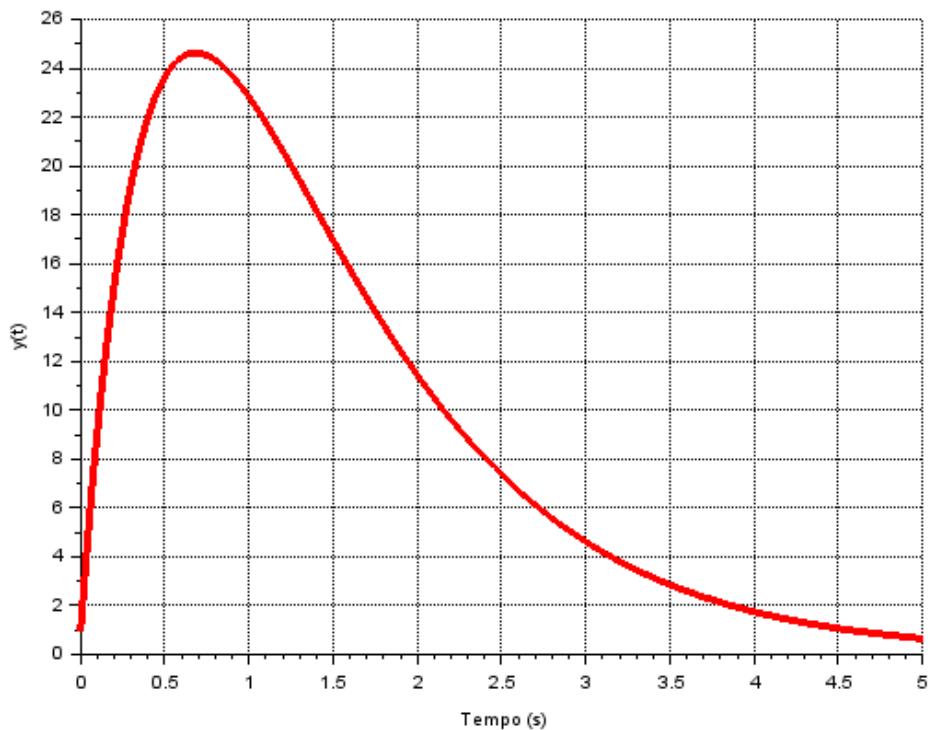
```



```

// Realizando a convolução y(t)=x(t)*h(t) para t > 0
// e gerando gráfico - Figura 4
//-
figure(4);
t = 0:0.01:5;
y = conv(h,x);
plot2d(t,y(1:max(size(t))),5);

```



Nota: Em anexo, segue os códigos do programa sobre convolução para os ambientes de programação Octave e Scilab.

Anexo 1 – Código em Octave

Anexo 2 – Código em Scilab

IV – ATIVIDADE

- 1) Executar o código em Octave ou Scilab, considerando os seguintes sinais:

$$x(t) = e^{-at}u(t) \quad \text{e} \quad h(t) = \sin(at)u(t) \quad \text{com} \quad u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$$

onde a é o número da sua equipe.

- 2) Verificar as propriedades da comutatividade e da distributividade da convolução em tempo contínuo, utilizando um dos códigos (Octave ou Scilab) em anexo.

Propriedade da comutatividade: $x(t) * h(t) = h(t) * x(t)$

Propriedade da associatividade: $x(t) * (h(t) + g(t)) = x(t) * h(t) + x(t) * g(t)$

Considerar os sinais de tempo contínuo:

$$x(t) = p(t), \quad h(t) = [u(t) - u(t-5)] \quad \text{e} \quad g(t) = r(t)$$

onde:

$u(t)$ (degrau unitário)

$p(t)$ (pulso retangular unitário)

$r(t)$ (rampa unitária),

Comentar.

Anexo 1 – Código em Octave

```
%Convolução de dois sinais contínuos
```

```
close all; clc; clear;
%-----
% Inicialização - definindo a "função" degrau unitário
% -----
t = -1:0.01:5;
t1=-1:0.01:-0.01;
t2=0:0.01:5;
u1(1:max(size(t1)))=0;
u2(1:max(size(t2)))=1;
u=[u1 u2];

% -----
% Definindo a função x(t) e gerando gráfico - Figura 1
% -----
x = exp(-t).*u;
fig=figure();
p=plot(t,x,'k');
set(p,'LineWidth',2);
ylabel('x(t)')
xlabel('Tempo (s)');

% -----
% Definindo a função h(t) e gerando gráfico - Figura 2
% -----
fig=figure();
h = exp(-2*t).*u;
p=plot(t,h,'k');
set(p,'LineWidth',2);
ylabel('h(t)')
xlabel('Tempo (s)');

% -----
% Definindo a função x(t) para t > 0
% Realizando a reflexão de h(t) --> h(-t) e gerando gráfico - Figura 3
% -----
fig=figure();
t = 0.01:0.01:5;
u1(1:max(size(t)))=1;
x = exp(-t).*u1;
h = exp(-2*t).*u1;
p=plot(-t,h,'k',t,x);
set(p,'LineWidth',2);
legend('h(t-\tau)', 'x(\tau)')
xlabel('Tempo (s)');
```

```
% -----  
% Realizando a convolução  $y(t)=x(t)*h(t)$  para  $t > 0$   
% e gerando gráfico - Figura 4  
%-----  
t = 0:0.01:5;  
fig=figure();  
y=conv(h,x);  
p=plot(t,y(1:length(t)), 'k');  
set(p,'LineWidth',2);  
ylabel('y(t)');  
xlabel('Tempo (s)');
```

Anexo 2 – Código em Scilab

```
// Modo de exibição
mode(0);

// Exibir aviso para a exceção de ponto flutuante
ieee(1);

//Convolução de dois sinais contínuos

xdel(winsid());clc;clear;

// -----
// setup das cores
// 2- blue, 3 - green, 4 - light blue,
// 5 - red, 6 - magenta, 7 - yellow
// ----

//-----
// Inicialização - definindo a "função" degrau unitário
// -----
t = -1:0.01:5;
t1 = -1:0.01:-0.01;
t2 = 0:0.01:5;
u1(1,1:max(size(t1))) = 0;
u2(1,1:max(size(t2))) = 1;
u = [u1,u2];
figure(0)
plot2d(t,u,2);
p = gce();
title('Gráfico da função degrau unitário','fontsize',3)
ylabel("u(t)")
xlabel("Tempo (s)");
xgrid
// Manipulação do gráfico
hf = gcf();
ha = gca();
// Ajustar Cor
hf.background = 8; // Área externa
ha.background = 8; // Área interna
ha.children.children.thickness = 4; // espessura da linha
//Definindo a função x(t) e gerando gráfico - Figura 1
//-----
x = exp(-t) .*u;
figure(1);
plot2d(t,x,2);
p = gce();

ylabel("x(t)")
```

```

xlabel("Tempo (s)");
xgrid
// Manipulação do gráfico
hf = gcf();
ha = gca();
// Ajustar Cor
hf.background = 8; // Área externa
ha.background = 8; // Área interna
ha.children.children.thickness = 4; // espessura da linha
// -----
// Definindo a função h(t) e gerando gráfico - Figura 2
// -----
figure(2);
h = exp(-2*t) .*u;
plot2d(t,h,3);
p = gce();
ylabel("h(t)");
xlabel("Tempo (s)");
xgrid
hf = gcf();
ha = gca();
// Ajustar Cor
hf.background = 8; // Área externa
ha.background = 8; // Área interna
ha.children.children.thickness = 4; // espessura da linha
// -----
// Definindo a função x(t) para t > 0
// Realizando a reflexão de h(t) --> h(-t) e gerando gráfico - Figura 3
// -----
figure(3);
t = 0.01:0.01:5;
u1 = mtlb_i(u1,1:max(size(t)),1);
x = exp(-t) .*u1;
h = exp(-2*t) .*u1;
plot2d(-t,h,3);
plot2d(t,x,2);
p = gce();
legend('$h(t-\tau)$','$x(\tau)$',1)
title(['Gráfico das funções '$h(t-\tau)$' e '$x(\tau)$'],'fontsize',3)
xlabel("Tempo (s)");
xgrid
// Manipulação do gráfico
hf = gcf();
ha = gca();
// Ajustar Cor
hf.background = 8; // Área externa
ha.background = 8; // Área interna
ha.children.children.thickness = 4; // espessura da linha
// -----
// Realizando a convolução y(t)=x(t)*h(t) para t > 0

```

```
// e gerando gráfico - Figura 4
//-----
figure(4);
t = 0:0.01:5;
y = conv(h,x);
plot2d(t,y(1:max(size(t))),5);
p = gce();
ylabel("y(t)");
xlabel("Tempo (s)");
xgrid
// Manipulação do gráfico
hf = gcf();
ha = gca();
// Ajustar Cor
hf.background = 8; // Área externa
ha.background = 8; // Área interna
ha.children.children.thickness = 4; // espessura da linha
```