Soma de convolução

Alfredo Savi, *RA: 1882872 e* Rafael Dalmolin, *RA: 1436287*

***Abstract*—Este artigo descreve, através de um exemplo simulado em um ambiente matlab, a resolução e a discussão de um problema proposto na disciplina de controle digital, onde aplicamos a soma de convolução afim de determinar o comportamento de uma saída, analisando em diferentes tipos de sinais de entrada. Tendo em posse um sinal de como o sistema se comporta quando excitado com um sinal degrau unitário.**

***Index Terms*—linear time-invariant system (LTI); Convolução Discreta; Soma de Convolução**

I. Introdução

NA matemática, particularmente na área de processamento de sinal, convolução é um operador linear que, a partir de duas funções dadas, resulta numa terceira que mede a soma do produto dessas funções ao longo da região subentendida pela superposição delas em função do deslocamento existente entre elas.

A concepção de convolução está coesa com a análise e o processamento de sinais que são aplicados em várias áreas, dentre elas análise de imagens, como digitalização, alisamento, embasamento e aberração cromática, estatística, criptografia, acústica, oceanografia, sismologia, óptica, geometria, etc (ZILL, 2003; YNOGUTI, 2017).[3], [2]

A operação de convolução pode ser utilizada para encontrar a resposta de um sistema LTI (Linear Independente no tempo). A saída de um sistema também pode ser dada pela convolução da entrada pela sua resposta ao impulso.

O sinal impulso unitário discreto ou o sinal amostra unitária define-se como:

$$\delta(n) = \begin{cases} 1, & \text{se} \quad n = 0, \\ 0, & \text{se} \quad n \neq 0. \end{cases} \tag{1}$$

Ele determina um único valor no instante $n = 0$. Para deslocar o impulso unitário no tempo é preciso mudar o argumento da função $\delta(n)$ da seguinte forma; Por exemplo, se $k = 10$, a função desloca-se dez lugares à direita,

$$\delta(n-k) = \begin{cases} 1, & \text{se} \quad n = k, \\ 0, & \text{se} \quad n \neq k. \end{cases} \tag{2}$$

Um sinal discreto qualquer, pode ser representado como uma combinação linear dos impulsos unitários discretos deslocados no tempo(CARVALHO;VELOSO:GURJÃO,2015)[1]

A. Savi Acadêmico de engenharia de computação ,Universidade tecnológica federal do paraná, Pato Branco, PR.

Dalmolin. Rafael Acadêmico de engenharia de computação ,Universidade tecnológica federal do paraná, Pato Branco, PR.

A Definição de convolução para funções de domínio discreto, é dada por:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[n-k] \tag{3}$$

Onde o sinal $y[n]$ é a resposta do sistema, $x[k]$ é a entrada do sistema e $\delta[n-k]$ é um impulso deslocado no tempo. Com isso o sinal $x[n]$ é obtido através do somatório das amostras de $x[n]$ e com deslocamentos iguais ao período de amostragem.

A. Descrição da atividade

Nesta atividade abordaremos um método de convolução discreta para determinar o comportamento de um sistema de acionamento a uma onda triangular, sabendo que, a resposta ao degrau desse sistema é dada pela equação abaixo:

$$g[n] = 1 - e^{-\alpha \cdot T \cdot n} \cdot cos(\omega \cdot T \cdot n) \tag{4}$$

em que $T = 0.5$, $\alpha = 0.2$ e $\omega = 1$.

A resposta ao degrau, da equação 4, possui o comportamento apresentado na figura 2 e a onda triangular possui o comportamento de acordo com a figura abaixo:

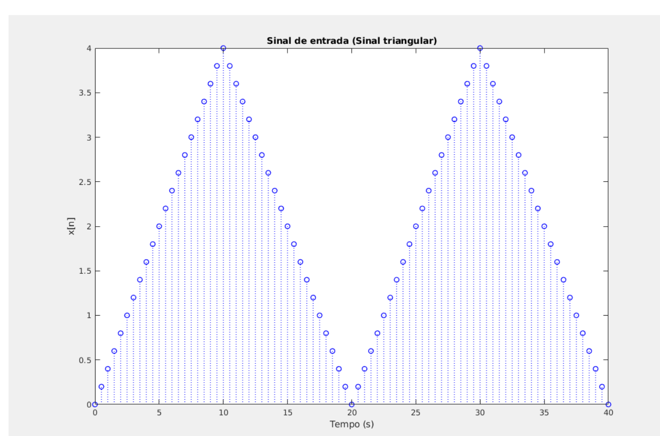

Fig. 1. Sinal Triangular

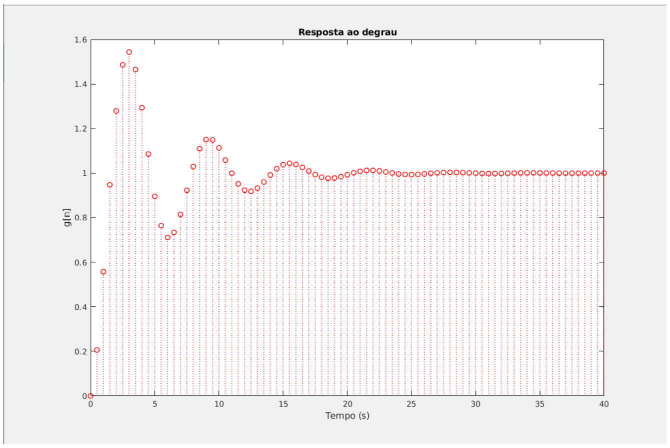

Fig. 2. Resposta ao degrau

Com esses dados e utilizando uma ferramenta computacional (Matlab) foi desenvolvido um programa (script) aplicando o algoritmo de convolução para apresentar os seguintes resultados:

- A resposta ao degrau e a resposta ao impulso;
- A onda triangular da entrada e a resposta a esta onda;
- A resposta a onda triangular com um ruído randômico com distribuição normal e desvio padrão de 0.5.

B. *Modelagem do algoritmo (script)*

Primeira sessão do script é responsável por limpar as variáveis e fechar as figuras que já estão abertas.

```
%% RESET
close all; % Fechar figuras
clear all; % Limpar variaveis
clc; % Limpar Command Window
```

Fig. 3. sessão - Reset

Segunda parte do script é responsável por criar a função g[n] (eq. 4) com os dados fornecidos pelo problema.

```
%% Function g[n]

% DADOS DA ATIVIDADE
Tg = 0.5;    % Periodo
alpha = 0.2; % Alpha
omega = 1;   % Omega

for n = 1:81 % Iniciando vetor da função g[n] com 80 pontos
    g(1,n) = 1 - exp(-alpha*Tg*(n-1)) .* cos(omega * Tg * (n-1));
end
```

Fig. 4. sessão - Criação g[n]

A próxima sessão é responsável por criar a onda triangular, com os dados fornecidos pelo gráfico da fig. 1.

```
%% Function x[n] (Onda triangular)

% DADOS DA ATIVIDADE ATRAVÉS DO GRÁFICO
A = 4;        % Amplitude
T = 20;       % Periodo
Ts = 0.5;     % Tempo de amostragem
NC = 2;       % Número de ciclos
NA = 40;      % Número de amostras por ciclo
NT = NA * NC; % Número total de pontos
TT = NT * Ts; % Tempo total de pontos

% Inicializacao de um vetor para criação da onda triangular
x = zeros(1, NT+1);
```

Fig. 5. sessão criação x[n] - Variáveis

```
for k=1:NA+1 % Gerando sinal triangular
    if(k >= 1 && k<=21) % SUBIDA
        x(1,k) = 0.2*(k-1);
    end

    if(k > 21 && k<=41) % DESCIDA
        x(1,k) = 0.2*(40-(k-1));
    end
end

% Preenchendo o vetor x com 80 pontos de um sinal triangular
for j=NA+1:NT+1
  x(1,j) = x(1,j-NA);
end
```

Fig. 6. sessão criação x[n] - Gerando sinal triangular

Essa sessão é responsável por criar a convolução do sinal g[n] (resposta ao degrau) com x[n] (sinal triangular). O script para a convolução é apresentado nas figuras abaixo podendo ser usado a função nativa do matlab ou uma implementada manualmente.

```
%% CONVOLUÇÃO - CRIANDO FUNÇÃO - g[n] * x[n]
for n=1:NT+1
    for k=1:NT+1 % Convolucao
        if (k<n)
            y(n)=y(n)+x(k)*g(n-k);
        end
    end
end
```

Fig. 7. sessão convolução - Implementação manual

```
%% CONVOLUÇÃO - USANDO FUNÇÃO MATLAB - g[n] * x[n]
y1 = conv(g, x); % Convolução
% Como a função conv() retorna mais que 80 pontos
% vamos pegar somente os 80 primeiros pontos (NT)
y = y1(1:1:NT+1);
```

Fig. 8. sessão convolução - Usando função nativa matlab

Neste trecho, é plotado a função g[n], x[n] e a convolução feita na figura 8 entre as funções de g[n] e x[n].

```
%% A ONDA TRIANGULAR DA ENTRADA E A RESPOSTA A ESTA ONDA - PLOT
t=0:Ts:TT; % Criando vetor de tempo com 80 pontos

% Criando uma figura
figure
% Criando um subplot na figura
subplot(311)
% Plot do gráfico da resposta g[n]
stem(t, g, 'r:', 'Linewidth',1);
% Título do gráfico
title('Resposta ao degrau')
% Nome e unidade do eixo X
xlabel('Tempo (s)')
% Nome e unidade do eixo Y
ylabel('g[n]')
```

Fig. 9. sessão plot - Resposta ao degrau g[n]

```
% Plot do sinal de entrada triangular
subplot(312)
stem(t, x, 'b:','Linewidth',1);
title('Sinal de entrada (Sinal triangular)')
xlabel('Tempo (s)')
ylabel('x[n]')

% Plot do gráfico do sinal de saída (resultado da convolução)
subplot(313)
stem(t, y,'g:','Linewidth',1);
title('Sinal de saída')
xlabel('Tempo (s)')
ylabel('y[n]')
```

Fig. 10. sessão plot - Sinal triangular x[n] e sinal de saída y[n] convolado

Essa sessão faz a convolução g[n] com um degrau unitário fig. 11 e com um impulso unitário fig. 12, seguido do plot da sua saída respectiva obtida da convolução.

```
%% A RESPOSTA AO DEGRAU E A RESPOSTA AO IMPULSO

% DEGRAU
x1 = ones(1, 81); % Criando um vetor de um com tamanho de 80
w= conv(g, x1);   % Convolução de g[n] com o degrau (x1)

% Pegando somente os 80 primeiros pontos (NT)
result = w(1:1:NT+1);

figure % Criando uma nova figura
stem(t, result) % Plot da resposta ao degrau (convolução)
title('Resposta ao degrau') % Título do gráfico
xlabel('Tempo (s)') % Nome e unidade do eixo X
ylabel('y[n]') % Nome e unidade do eixo Y
```

Fig. 11. sessão convolução - Convolução g[n] com um degrau unitário

```
% IMPULSO
x2 = zeros(1,81); % Criando um vetor de zero com tamanho de 80
x2(1,1) = 1; % Definindo o primeiro indice igual a 1
w2 = conv(g, x2); % Convolução de g[n] com impulso unitário (x2)

% Pegando somente os 80 primeiros pontos (NT)
result2 = w2(1:1:NT+1);

figure % Criando uma nova figura
stem(t, result2) % Plot da resposta ao impulso (convolução)
title('Resposta ao impulso') % Título do gráfico
xlabel('Tempo (s)') % Nome e unidade do eixo X
ylabel('y[n]') % Nome e unidade do eixo Y
```

Fig. 12. sessão convolução - Convolução g[n] com um impulso unitário

Última parte do script é a criação de um ruído randômico com desvio padrão de 0.5 na onda triangular e a convolução entre o sinal g[n] e o sinal triangular com ruído. Em seguida, é plotado os gráficos da convolução dos sinais com e sem ruído.

```
%% A RESPOSTA A ONDA TRIANGULAR COM UM RUÍDO RANDÔMICO
%  COM DISTRIBUIÇÃO NORMAL E DESVIO PADRÃO DE 0.5.

% Definindo desvio padrão igual a 0.5 (conforme enunciado)
desvioPadrao = 0.5;
%Escolhendo um valor de média arbitrária
media = 0.5;

% Gerando um vetor randômico com desvio Padrão e média definida
% anteriormente
R = media + desvioPadrao .* (randn(1,81));

% Aplicando o ruído a onda triangular
Ruido = x + R;

figure % Criando uma nova figura
stem(t, Ruido) % Plot do gráfico da onda triangular com ruído
title('Sinal triangular com ruído')
xlabel('Tempo (s)')
ylabel('x[n]')
```

Fig. 13. sessão - Gerando e plotando sinal triangular com ruído

```
% CONVOLUÇAO COM RUIDO
w3 = conv(Ruido, g); % Convolução do ruído com o sinal g[n]
result3 = w3(1:1:NT+1);

% PLOT DO SINAL DE SAÍDA COM E SEM RUÍDO

% COM RUÍDO
figure
subplot(211)
stem(t, result3)% Plot do sinal triangular com ruído (convolução)
title('Resposta a onda triangular com ruído')
xlabel('Tempo (s)')
ylabel('y[n]')

% SEM RUÍDO
subplot(212)
stem(t, y,'b:','Linewidth',1);% Plot do sinal sem ruído
title('Sinal de saída sem ruido')
xlabel('Tempo (s)')
ylabel('y[n]')
```

Fig. 14. sessão - Plotando o sinal convolucionado com e sem ruído

C. *Análise e resultados obtidos*

No gráficos abaixo, encontram-se as figuras da resposta em degrau g[n](eq.4), sinal triangular x[n] (fig. 1) e o sinal de saída que seria o resultado da convolução entre g[n] e x[n].

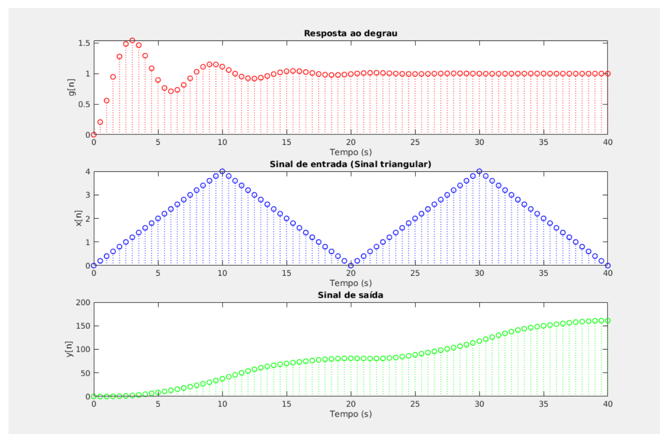

Fig. 15. Plot - Sinal g[n], x[n] e y[n]

A figura 16 representa a resposta da convolução entre o sinal g[n] e um degrau unitário, já a figura 17 representa a resposta da convolução entre o sinal g[n] e um impulso unitário.

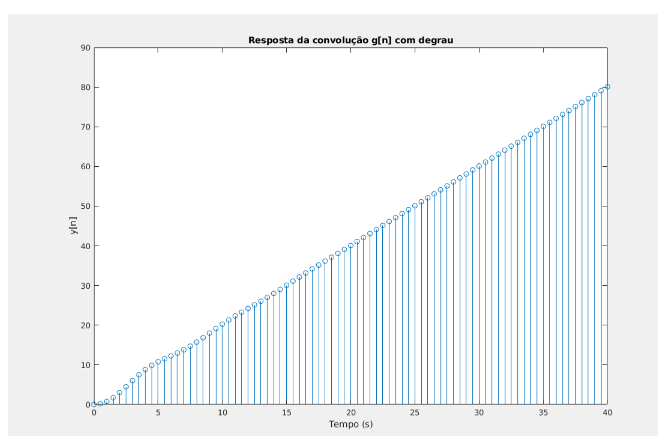

Fig. 16. Plot - Convolução do sinal g[n] com um degrau unitário

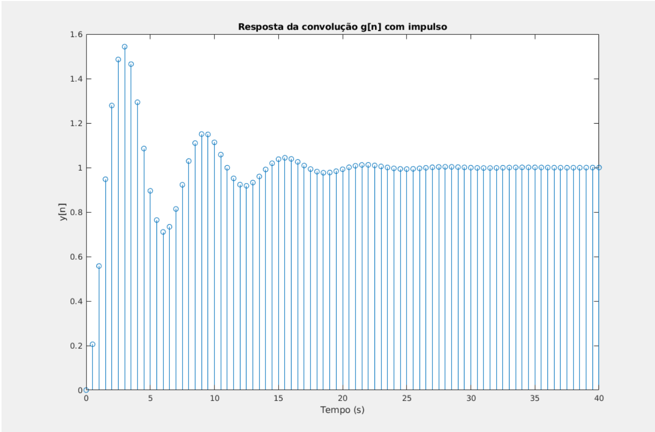

Fig. 17. Plot - Convolução do sinal g[n] com um impulso unitário

Nota-se que, na figura 17, o resultado é o próprio sinal de g[n]. Isto porque se aplicar uma convolução entre um impulso unitário e uma resposta ao degrau o sinal convolucionado será o próprio sinal.

A figura abaixo representa o sinal triangular com um ruído randômico com distribuição normal e desvio padrão de 0.5.

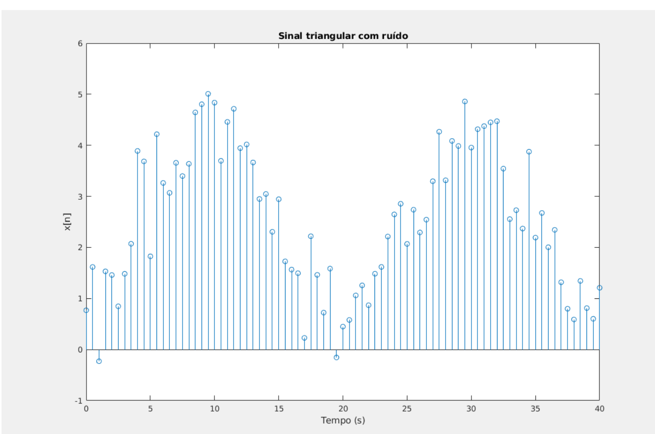

Fig. 18. Plot - Sinal triangular com ruído randômico

O gráfico abaixo, mostra uma figura com a resposta a onda triangular com o ruído randômico, isto é, a convolução entre a g[n] (eq. 4) e com o sinal triangular com o ruído x[n] (fig. 18). No mesmo gráfico a outra figura representa a mesma convolução, mas com uma onda triangular x[n] sem ruído (fig. 1).

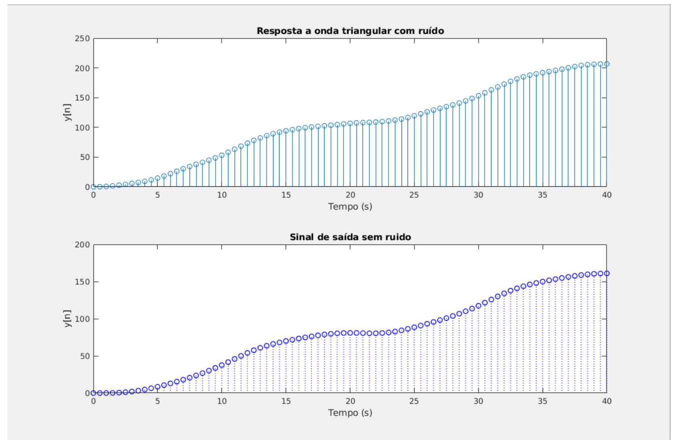

Fig. 19. Plot - Resposta a onda triangular com e sem ruído

Podemos notar que: mesmo aplicando um ruído na onda triangular x[n] o resultado final é parecido com o sinal de saída sem esse ruído, mudando, perceptivelmente, somente sua amplitude.

Na figura de todas as convoluções foi definido que: seu tempo de amostragem seria de 0 a 40 segundos respeitando o tempo de inicio e fim dos sinais que foram cedidos (x[n] e g[n]). Isto porque, como não foi definido o comportamento desses sinais depois dos 40 segundos não teria utilidade mostrar a convolução completa.

Porém, a fim de curiosidade a figura abaixo mostra a convolução completa, isto é, com a soma dos comprimentos dos sinais x[n] e g[n], totalizando as 160 amostras.

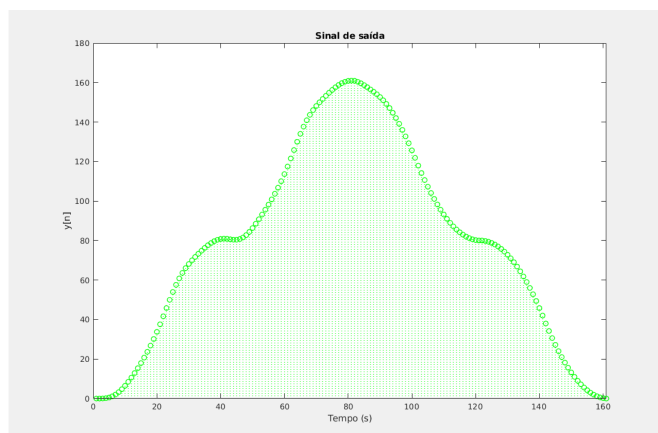

Fig. 20. Plot - Convolução completa do sinal de saída

II. Conclusão

Com o presente artigo foi possível reafirmar que o assunto de convolução está diretamente ligado a filtros de sinais, de processamento de áudio onde está presente nas técnicas de reverberação simulada digitalmente, podendo assim melhorar muito a qualidade do áudio. No processamento de imagem ou seja, é utilizada para suavização, onde a imagem resultante da convolução será menos ruidosa do que se comparado com a original.

Como visto anteriormente, aplicando a convolução com um sinal com e sem ruído sua diferença é perceptível somente em sua amplitude. Com essa característica, podemos aplicar em filtros digitais, como o filtro FIR que é um filtro de resposta finita ao impulso para filtragem de sinais de áudio. Entao como proposto pelo problema podemos comprovar que se conhecemos a resposta a impulso de um sistema, temos condições de calcular a resposta deste a qualquer impulso na entrada.

REFERENCES

[1] CARVALHO, J. M.; VELOSO, L.; GURJÃO, E. C., *Análise de Sinais e Sistemas*, Rio de Janeiro: Elsevier, 2015.

[2] YNOGUTI, C. A.*Processamento Digital de Sinais.*Campus em Santa Rita do Sapucaí: Instituto Nacional de Telecomunicações, 2017. Disponível em:http://www.inatel.br/docentes/ynoguti/graduacao-sp-2113502489/52-convolucao. Acesso em: 27 Jul. 2021.

[3] ZILL, D. G. *Equações Diferenciais com Aplicações em Modelagem.* São Paulo, SP: Thomson, 2003. xiv 492 p.