INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGICA DE SANTA CATARINA
CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR DE ENGENHARIA ELETRÔNICA

ELVIS ROBERTO DE JESUS AVILA CARVALHO FERNANDES

ATIVIDADE AVALIATIVA #02 SÉRIE DE FOURIER

Relatório Técnico - Científico submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, disciplina de Sinais e Sistemas.

Professor Orientador: Robinson Pizzio, Dr Eng.

FLORIANÓPOLIS, OUTUBRO DE 2020.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Este relatório descreve o desenvolvimento de um *software* desenvolvido em *MATLAB* que permite ao usuário entrar com a quantidade de coeficientes a serem utilizados na visualização do sinal original juntamente com sua versão reconstruída através dos coeficientes da Série de Fourier.

Para dois sinais periódicos, os quais são decompostos em Série exponencial de Fourier, cujas equações são desenvolvidas utilizando o *software* *WolframAlpha*, serão apresentados, gráficos de magnitude e de fase dos coeficientes da Série, bem como a análise da síntese do sinal reconstruído devido ao erro da aproximação por série de Fourier devido às descontinuidades do sinal (Fenômeno de Gibbs) e, a avaliação da distorção harmônica do sinal na reconstrução com uma quantidade limitada de harmônicos, a chamada DHT (Distorção Harmônica Total).

1.1 OBJETIVO GERAL

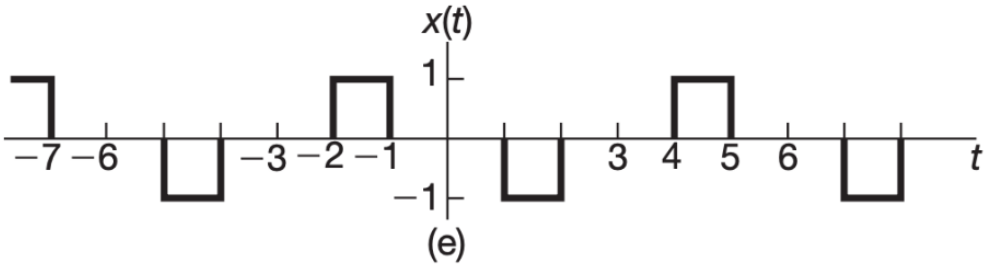
O objetivo geral deste relatório consiste em desenvolver criar um software em MATLAB para visualização da Série de Fourier e suas componentes dos sinais periódicos das figuras (e) e (g).

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desde relatório são:

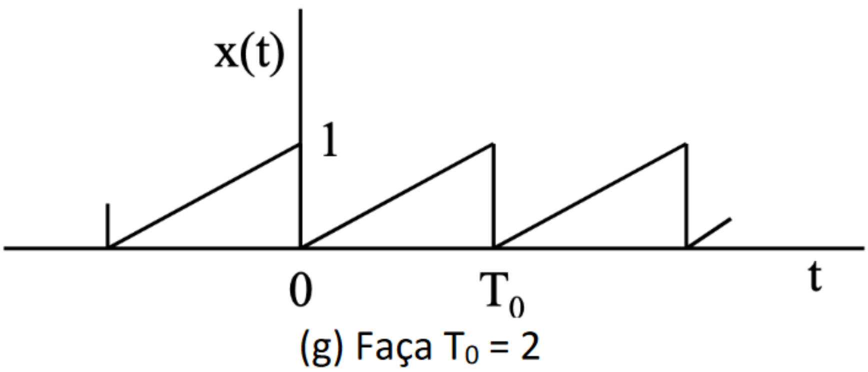
a) verificar o comportamento da representação de sinais periódicos por Série de Fourier;
b) constatar a relação entre o período fundamental e o espaçamento das componentes no eixo de freqüência;
c) visualizar o erro da aproximação por série de Fourier devido às descontinuidades do sinal; e
d) entender o conceito de Distorção Harmônica.

Figura 1 – Sinal periódico representado por (e)



Fonte: (Sinal fornecido pelo professor).

Figura 2 – Sinal periódico representado por (g)



Fonte: (Sinal fornecido pelo professor).

2 REVISÃO DA LITERATURA

Neste capítulo, serão discutidos os conceitos relativos ao comportamento ao comportamento da representação de sinais periódicos por Série de Fourier.

2.1 SÉRIE EXPONENCIAL DE FOURIER

2.1.1 Coeficientes D_n

Usando a igualdade de Euler, podemos expressar $cos(n\omega_0 t)$ e $sen(n\omega_0 t)$ em termos de exponenciais $e^{jk\omega_0 t}$ e $e^{-jk\omega_0 t}$. Claramente, somos capazes de expressar a série trigonométrica de Fourier representada pela equação 1 em termos de exponenciais na forma $e^{jk\omega_0 t}$ com o índice k assumindo todos os valores inteiros de $-\infty$ a ∞, incluindo zero:

$$x(t) = a_0 + \sum_{n=1}^{\infty} a_n cos(n\omega_0 t) + b_n sen(n\omega_0 t) \qquad (Equação\ 1)$$

A determinação da série exponencial de Fourier a partir dos resultados já obtidos da série trigonométrica de Fourier é direta, envolvendo a conversão de senóides em exponenciais. A série exponencial de Fourier para um sinal periódico x(t) pode ser escrita pela equação 2:

$$x(t) = \sum_{k=-\infty}^{\infty} D_n\, e^{jk\omega_0 t} \qquad (Equação\ 2)$$

na qual

$$D_n = \frac{1}{T_0} \int_{T_0} x(t)\, e^{-jk\omega_0 t} dt \qquad (Equação\ 3)$$

A série é bem compacta, bem como a expressão matemática para a obtenção dos coeficientes da série também é compacta. É muito mais conveniente trabalhar com série exponencial do que com a trigonométrica.

Um sinal periódico que possui energia finita em um período garante a convergência na média da sua série de Fourier. Se um sinal periódico x(t) satisfaz as condições de Dirichlet, tal sinal x(t) deve ser integrável, além de possuir um número finito de descontinuidades finitas em um período, bem como deve conter apenas um número finito de máximos ou mínimos em um período.

2.2 SÍNTESE DE FOURIER DE FUNCOES DESCONTÍNUAS – O FENÔMENO DE GIBBS

O gráfico de uma série truncada é muito próximo da função x(t) quando aumentamos o número de harmônicas (n), assim espera-se que a série convirja exatamente para x(t). Para (n) grande, a série exibe um comportamento oscilatório e um sobre-sinal aproximadamente de 9% na proximidade da descontinuidade no pico mais próximo da oscilação. Independentemente do valor de (n), o sobre-sinal permanece em aproximadamente 9%. Josiah Willard Gibbs, um matemático físico eminente, inventor da análise vetorial, forneceu uma explicação matemática para esse comportamento.

Podemos reconciliar a aparente aberração do comportamento da série de Fourier. A freqüência de oscilação (F_s) do sinal sintetizado é $Fs = nf_0$, tal que a largura do pico com sobre-sinal de 9% é aproximadamente $Largura_{PICO} = \frac{1}{2}nf_0$. Quando aumentamos (n), a freqüência de oscilação aumenta, mas a largura do pico com sobre-sinal diminui. Quando $n \to \infty$, a potência do erro $\to \infty$ porque o erro é constituído principalmente de picos, com larguras $\to \infty$.

Portanto, quando $n \to \infty$, a série de Fourier correspondente difere de x(t) por aproximadamente 9% imediatamente à esquerda e à direita do ponto de descontinuidade e, mesmo assim, a potência do erro $\to \infty$. A razão para essa confusão é que, neste caso, a série de Fourier converge para a média. Quando isso acontece, tudo o que prometemos é que a energia do erro (em um período) $\to \infty$ quando $n \to \infty$. Portanto, a série pode diferir de x(t) em alguns pontos e mesmo assim ter a potência do sinal de erro igual a zero. É precisamente nas descontinuidades que a série difere de x(t) por 9%. Quando utilizamos apenas os primeiros (n), termos da série de Fourier para sintetizar um sinal, estamos terminando bruscamente a série, dando um peso unitário para as primeiras (n), harmônicas e peso zero para todas as harmônicas restantes após (n). Esse truncamento abrupto da série causa o fenômeno Gibbs na síntese de funções descontínuas. O fenômeno Gibbs está presente apenas quando existe um salto de descontinuidade em x(t).

3 METODOLOGIA

No primeiro momento foi feita uma revisão bibliográfica com objetivo de analisar de forma teórica as diretrizes obrigatórias referentes ao relatório, a fim de, decompor os sinais em Série Exponencial de Fourier utilizando o *software WolframAlpha*.

Com as equações desenvolvidas serão gerados *scripts* no *software MATLAB,* tais que, serão visualizados o sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier, que permitirá constatar a relação entre o período fundamental e o espaçamento das componentes no eixo de freqüência, visualizar o erro da aproximação por série de Fourier devido às descontinuidades do sinal, bem como a Distorção Harmônica dos sinais envolvidos.

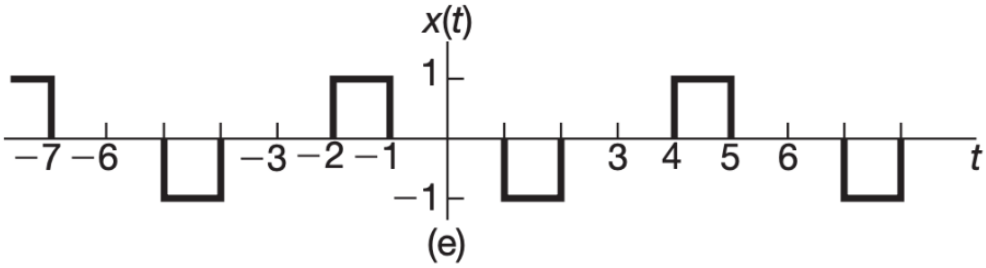
4 DESENVOLVIMENTO

Nesta seção serão apresentados as funções de cada sinal, bem como período, freqüência, freqüência fundamental, a série compacta de Fourier, a integral que representa a série de Fourier no *software WolframAlpha*, a expressão geral para o cálculo dos coeficientes da série Exponencial de Fourier, o Espectro de Fourier, **c**ódigos para calcular o somatório dos Coeficientes $\mathrm{D_n}$ da Série de Fourier na forma exponencial no Matlab, plotagem dos gráficos truncados da série de Fourier junto com o sinal original e o cálculo da distorção harmônica de cada sinal.

4.1 SINAL DA FIGURA (E)

4.1.1 Série Exponencial de Fourier da figura (e)

Figura 3 – Sinal periódico representado por (e)



Fonte: (Sinal fornecido pelo professor).

Função figura (e) $(\,\boldsymbol{x(t)}\,)$

$$\boldsymbol{x(t) = u(t-2) - u(t+2)}$$

$$x(t) = \begin{cases} +1 & para\ (-2) < t < (-1) \\ -1 & para\ (+1) < t < (+2) \end{cases}$$

Período ($\boldsymbol{T_0}$) para a figura (e)

$$\boldsymbol{T_0 = 6s}$$

Freqüência ($\boldsymbol{f_0}$) para a figura (e)

$$\boldsymbol{f_0 = \frac{1}{T_0} = \frac{1}{6} Hz}$$

Freqüência Fundamental ($\boldsymbol{\omega_0}$) para a figura (e)

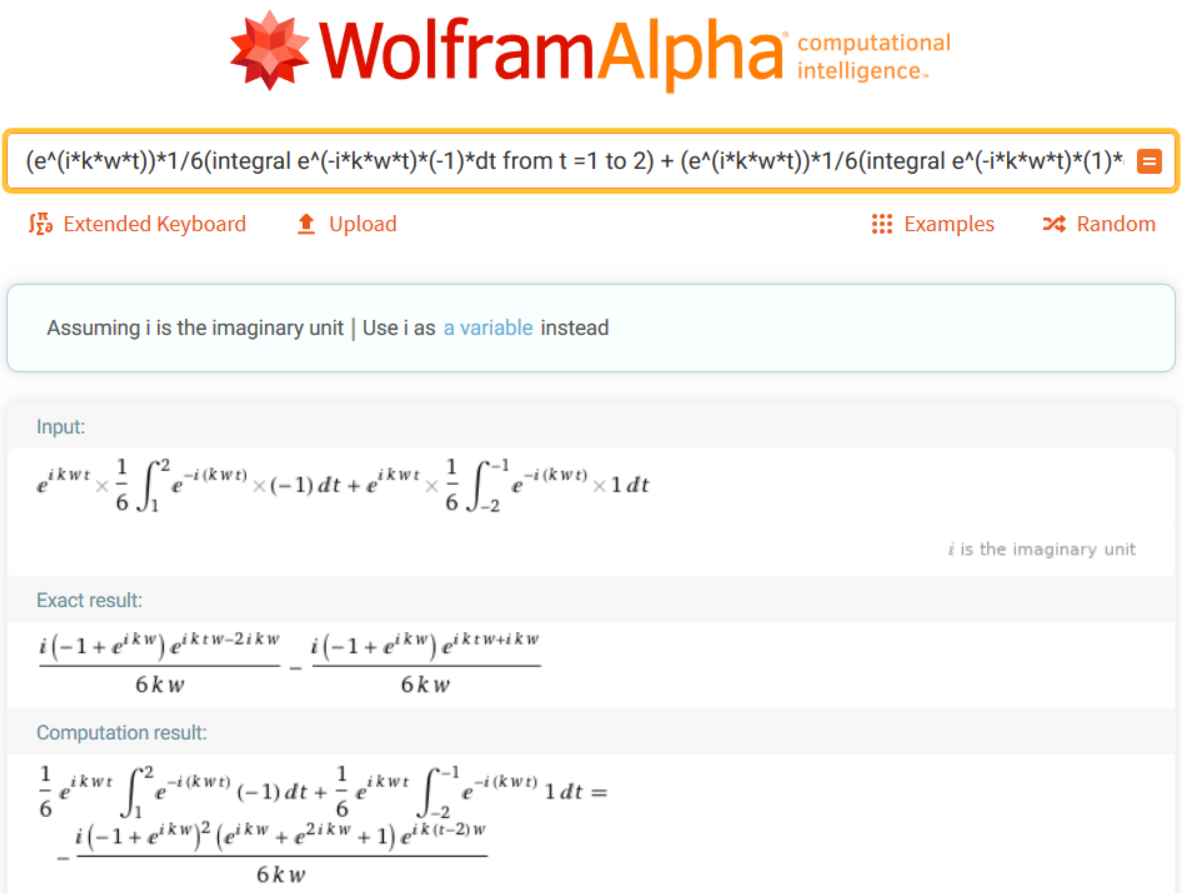
$$\boldsymbol{\omega_0 = \frac{2 * \pi}{T_0} = \frac{2 * \pi}{6} = \frac{\pi}{3}\ rad/s}$$

Para a função (e) temos que:

$$\boxed{D_n = \left(e^{-ik\omega_0 t}\right)\left(\frac{1}{6}\right)\int_{-2}^{-1}\left(e^{-jk\omega_0 t}\right)(1)dt + \left(e^{-ik\omega_0 t}\right)\left(\frac{1}{6}\right)\int_{1}^{2}\left(e^{-jk\omega_0 t}\right)(-1)dt}$$

4.1.2 Integral no software WolframAlpha para a figura (e)

(e^(i*k*w*t))*1/6(integral e^(-i*k*w*t)*(-1)*dt from t =1 to 2) + (e^(i*k*w*t))*1/6(integral e^(-i*k*w*t)*(1)*dt from t =-2 to -1)



https://www.wolframalpha.com/input/?i=+%28e%5E%28i*k*w*t%29%29*1%2F6%28integral+e%5E%28-i*k*w*t%29*%28-1%29*dt+from+t+%3D1+to+2%29+%2B++%28e%5E%28i*k*w*t%29%29*1%2F6%28integral+e%5E%28-i*k*w*t%29*%281%29*dt+from+t+%3D-2+to+-1%29

4.1.3 Expressão Geral para o cálculo dos coeficientes da série Exponencial de Fourier para a figura (e)

$$D_n = -\frac{i\left(-1+e^{ik\omega}\right)^2\left(e^{ik\omega}+e^{2ik\omega}+1\right)\left(e^{ik(t-2)\omega}\right)}{6k\omega}$$

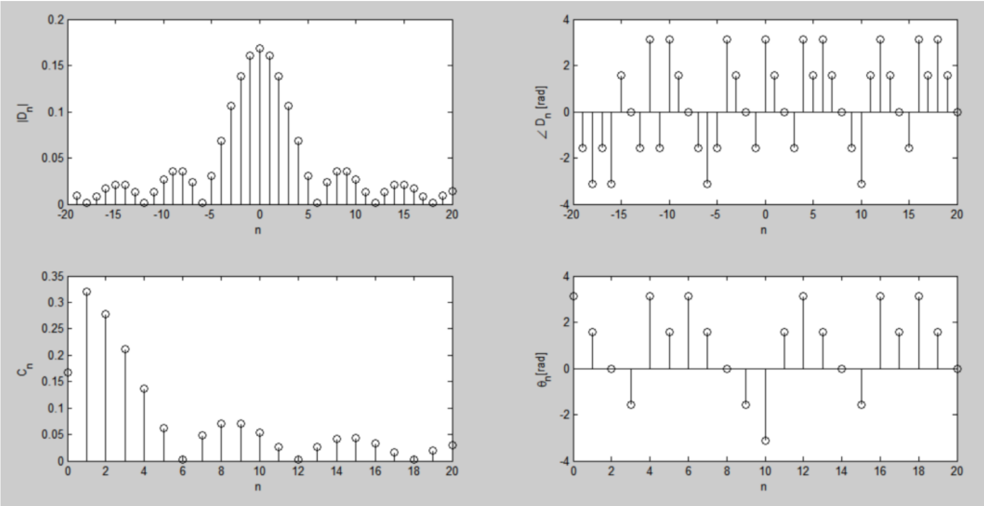
4.1.4 Código para visualização do Espectro de Fourier no *MATLAB* para a figura (e)

```
close all
clear all

clc
clear

%Estabelecendo os parâmetros básicos para plotagem do Espectro Exponencial
de Fourier
%Parametros basicos

%Período de 2 segundos
T_0 = 6;

%Como x(t) possui descontinuidade, Dn
%cai lentamente, em função de 1/n.
%Representa aproximadamente 1% da freqüência fundamental (100)= 2^8
N_0 = 256;

%Período de amostragem
T = T_0/N_0;

%intervalo de amostragem
t = (0:T:T*(N_0-1))';

%número finito de coeficientes para a aproximação da fft
M=20;

%funcao(e)
x=(-(t>=1).*(t<2))+((t>=-2).*(t<-1));

%Plotando na mesma figura todos os gráficos
figure(1)

%Algoritmo para calcular a Transformada rápida de Fourier da função rampa
e
%aproximar o espectro exponencial de Fourier %para -M<=N<=M
D_n = fft(x)/N_0;

%valores das amostras como sendo a média dos valores da função
%nos dois lados %da descontinuidade
n = [-N_0/2:N_0/2-1]';

%Limpar janela de figura atual
clf;

%Linha 1 Coluna 1 da figura
subplot (2,2,1);

%Plotando o módulo de D_n em função de n
%usando abs para calcular o módulo de D_n e para calcular a transformada
%de %Fourier usar fftshift para reorganizar deslocando o componente
%de freqüência zero para o centro da matriz.
stem(n,abs(fftshift(D_n)),'k');
```

```
%definindo os limites dos eixos do gráfico da Linha 1 Coluna 1 da figura 1
axis ([-M M 0 0.2]);

%titulo para o elemento do eixo x
xlabel ('n');

%titulo para o elemento do eixo y
ylabel('|D_n|');

%Linha 1 Coluna 2 da figura 1
subplot (2,2,2);

%Plotando o angulo de D_n em função de n usando angle para calcular o
%ângulo e %usando fftshift para reorganizar deslocando o componente de
%freqüência zero para o centro da matriz.
stem(n,angle(fftshift(D_n)),'k');

%definindo os limites dos eixos do gráfico da Linha 2 Coluna 2 da figura 1
axis ([-M M -4 4]);

%titulo para o elemento do eixo x
xlabel ('n');

%titulo para o elemento do eixo y
ylabel('\angle D_n [rad]');


%definindo o intervalodo espectro trigonométrico de Fourier aproximado
n = [0:M];

%Definindo o coeficiente C_n(1) do espectro trigonométrico de Fourier como
%sendo o módulo de D_n da transformada de Fourier X usando fftshift para
%reorganizar deslocando o componente de freqüência zero para o centro da
%matriz.
C_n(1)= abs(D_n(1));

%Definindo o coeficiente dos  demais C_n do espectro trigonométrico de
%Fourier %como sendo o módulo de D_n da transformada de Fourier X
%usando fftshift para %reorganizar deslocando o componente de frequência
%zero para o centro da %matriz.
C_n(2:M+1) = 2*abs(D_n(2:M+1));

%usando a função angle para calcular o ângulo repassando o valor de D_n(1)
%para theta_n(1)
theta_n(1) = angle (D_n(1));

%usando a função angle para calcular o ângulo repassando o valor de
%(D_n(2:M+1)) para theta_n(2:M+1)
theta_n(2:M+1) = angle(D_n(2:M+1));

%Linha 2 Coluna 1 da figura 1
subplot (2,2,3);

%Plotando os coeficientes trigonométricos de fourier em função de n
stem(n,C_n,'k');
```

```
%titulo para o elemento do eixo x
xlabel('n');

%titulo para o elemento do eixo y
ylabel('C_n');

%Linha 2 Coluna 2 da figura 1
subplot (2,2,4);

%Plotando o ângulo dos coeficientes trigonométricos de fourier
%em função de n
stem(n,theta_n,'k');

%titulo para o elemento do eixo x
xlabel('n');

%titulo para o elemento do eixo y
ylabel('\theta_n[rad]');
```

Figura 4 – Espectro de Fourier do Sinal periódico representado por (e)



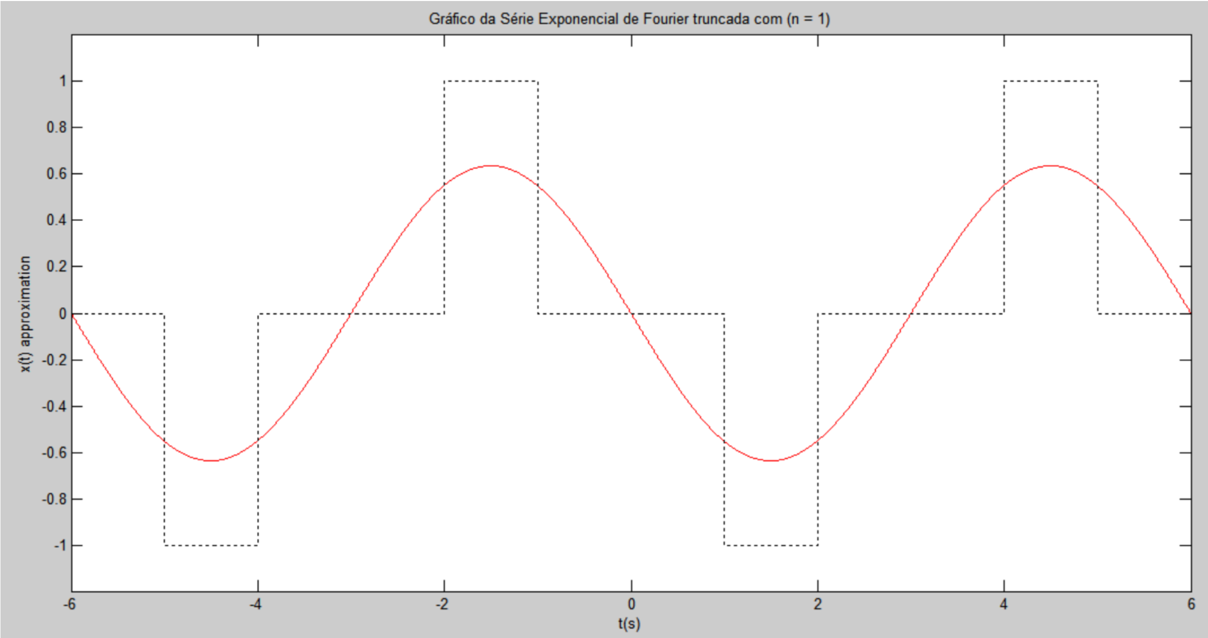
Fonte: (Elaboração própria).

4.1.5 Código para calcular o somatório dos Coeficientes D_n da Série de Fourier na forma exponencial no *MATLAB* da figura (e)

```
close all
clear all

clc
clear

%Número de harmônicas(n)
n=50;

%Valor do limite de tempo
M=6;

%Definindo o intervalo de tempo
intervalo = -M:0.001:M;

%Freqüência de amostragem
Fs = 0.001;

%Inicio do indice
indice = 1;

%Período de 6 segundos para a figura (e)
T= 6;

%Freqüência Fundamental
w=2.0*pi/T;

%Inicio do loop
%repassando os parâmetros de intervalo para a variável t
for t= intervalo

    %a variável da Série Exponencial de Fourier da função (g) com
%valor 0
     valor = 0.0;

         %a variável k [e limitada pelos valores de k para contemplar
%os dois lados da descontinuidade
     for k = -n:n;

        %tratando exceção para o caso de uma divisão por zero
        if (k ~=0.0)

      %Série Exponencial de Fourier da função (g) calulada no software
      %WolframAlpha
           termo = (i*(-1+ exp(i*k*w))*(-1+ exp(i*k*w)));
           termo2= (exp(i*k*w)+ exp(2*i*k*w)+1);
           termo3 = exp((i*k*w*(t-2)));
           valor = (valor +( termo*termo2*termo3 )*(-1) / (6*k*w));

        %senão
        else

            %incrementa o valor da função (e)
```

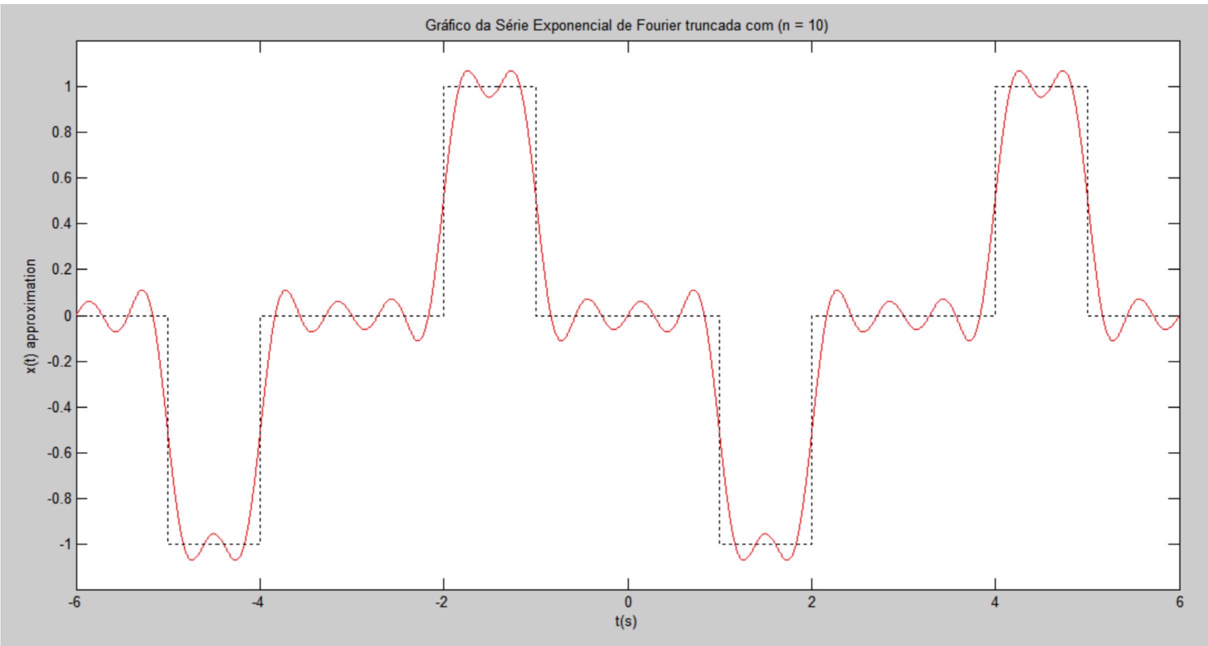
```
            valor = valor +1;

        %fim do tratamento da função (g)
        end

    %fim do repasse dos valores de n para k
     end

    %passando os parâmetros para a variável res seguindo o índice
    res (indice) = (valor-1);

    %incrementa o valor dda vari[avel índice
    indice = indice +1;

%fim do loop
end

%Plotando a série de Fourier Truncada com n aproximações
%Definindo o intervalo de tempo para a fun;áo da figura (e)
teste=-M:0.001:M;

%Criando o sinal original (traçejado e preto) da figura (e)
x = -(teste>=1).*(teste<2)+(teste>=4).*(teste<5)-((teste>=-
5).*(teste<-4))+((teste>=-2).*(teste<-1));

%Criando a figura 1
figure(1)

%Plotando a Série Exponencial de Fourier da função (g) calulada no
software em função do intervalo (em vermelho)
plot (intervalo, res,'r');

%Mantém  no mesmo gráfico a próxima plotagem
hold;
%Plotando o sinal original (traçejado e preto)
plot(teste,x,'k:');

%titulo para o elemento do eixo x
xlabel('t');

%titulo para o elemento do eixo y
ylabel(['x(t) approximation']);

%titulo para a figura 1
title(['Gráfico da Série Exponencial de Fourier truncada com (n =
{',num2str(n),'})']);

%definindo os limites dos eixos do gráfico da figura 1
axis ([-M M -1.2 1.2]);
```

Figura 5 – Gráfico do sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier utilizando 1 amostra



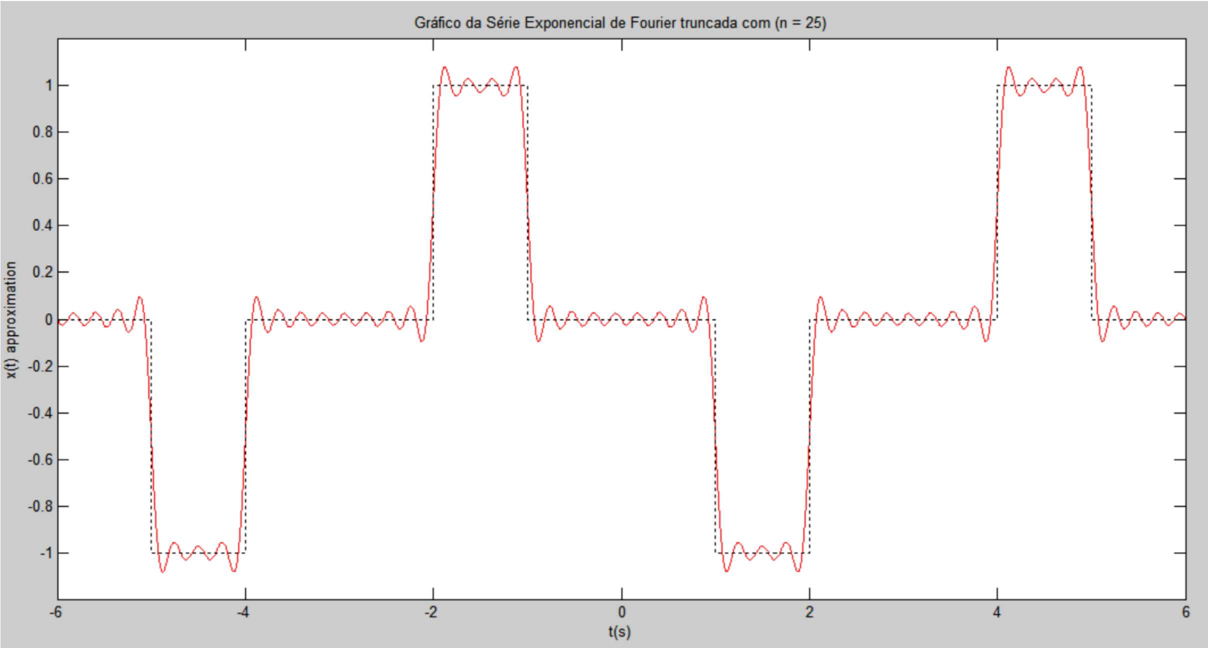
Fonte: (Elaboração própria).

Figura 6 – Gráfico do sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier utilizando 10 amostras



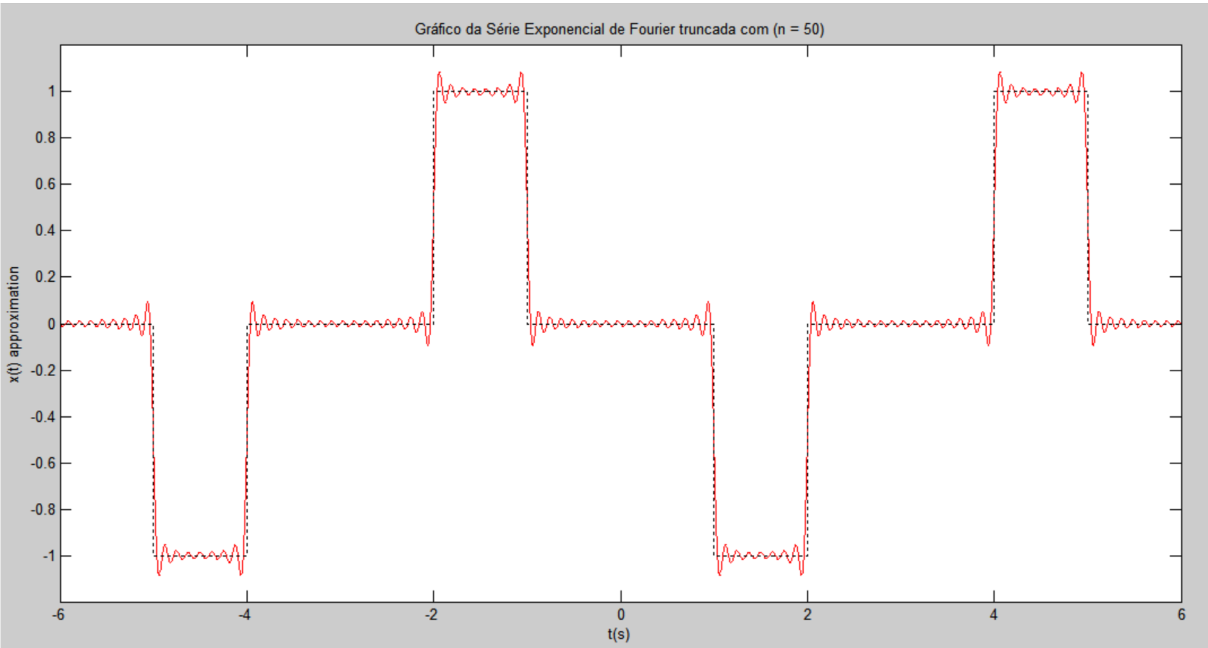
Fonte: (Elaboração própria).

Figura 7 – Gráfico do sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier utilizando 25 amostras



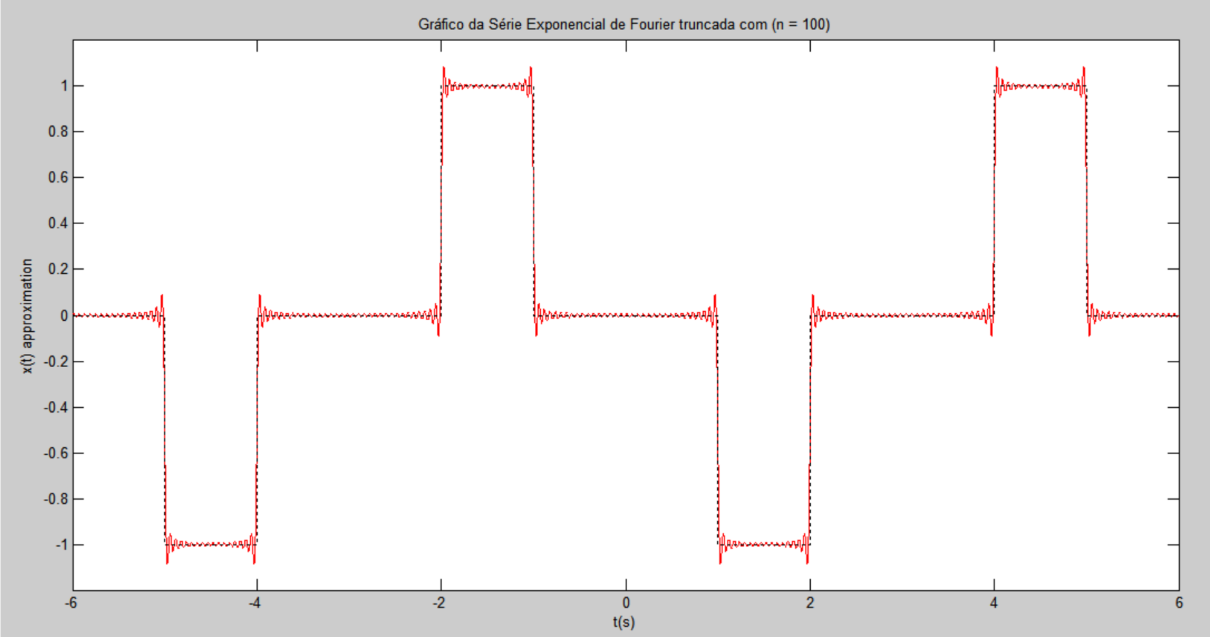
Fonte: (Elaboração própria).

Figura 8 – Gráfico do sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier utilizando 50 amostras



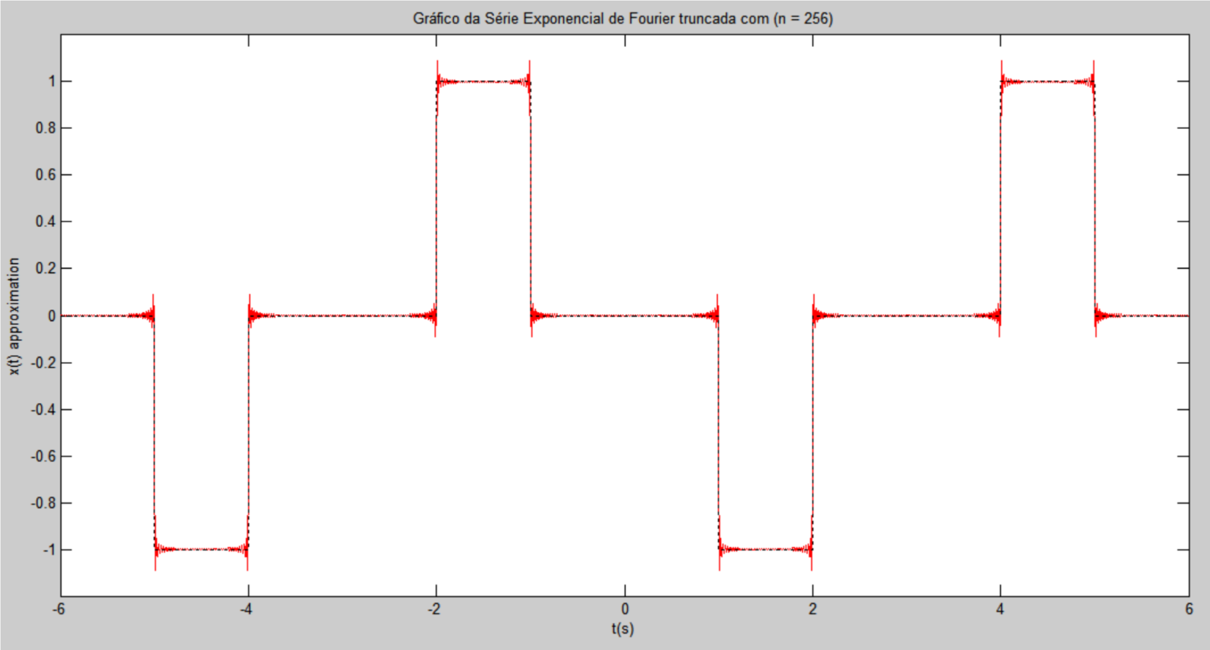
Fonte: (Elaboração própria).

Figura 9 – Gráfico do sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier utilizando 100 amostras



Fonte: (Elaboração própria).

Figura 10 – Gráfico do sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier utilizando 256 amostras



Fonte: (Elaboração própria).

4.1.6 Cálculo da distorção harmônica para a figura (e)

$$\boldsymbol{Potencia_e = \frac{4}{T_0}\int_0^{\frac{T_0}{4}} x(t)dt = \frac{4}{6}\int_0^{\frac{6}{4}} (1)^2 dt = \frac{2}{3}\int_0^{\frac{3}{2}} dt}$$

$$\boldsymbol{Potencia_e = \frac{2}{3}\left(\frac{3}{2}\right) = 1}$$

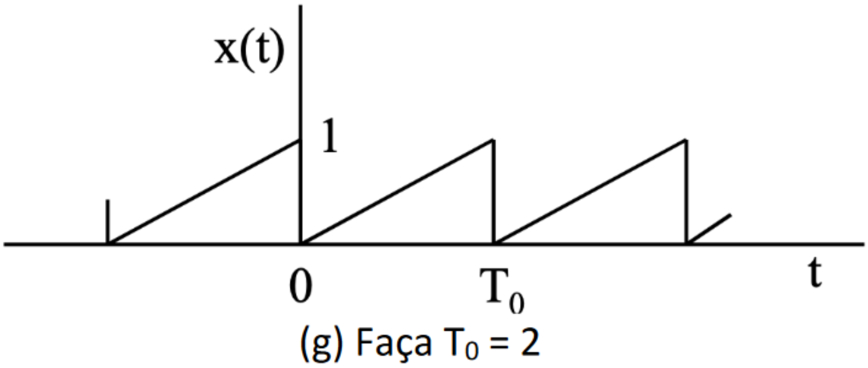
A potencia do sinal desejado é: $\boldsymbol{\frac{1^2}{2} = 0,5}$

$$\boldsymbol{D_{total} = \frac{1}{0,5} * 100 = 200\%}$$

4.2 SINAL DA FIGURA (G)

4.2.1 Série Exponencial de Fourier da figura (g)

Figura 11 – Sinal periódico representado por (g)



Fonte: (Sinal fornecido pelo professor).

Função figura (g) ($x(t)$)

$$x(t) = \frac{t}{2}[u(t) - u(t-2)]$$

$$x(t) = \begin{cases} \frac{t}{\tau} & , 0 < t < \tau \\ 0 & , C.C \end{cases}$$

Período (T_0) para a figura (g)

$$T_0 = 2s$$

Freqüência (f_0) para a figura (g)

$$f_0 = \frac{1}{T_0} = \frac{1}{2}Hz$$

Freqüência Fundamental (ω_0) para a figura (g)

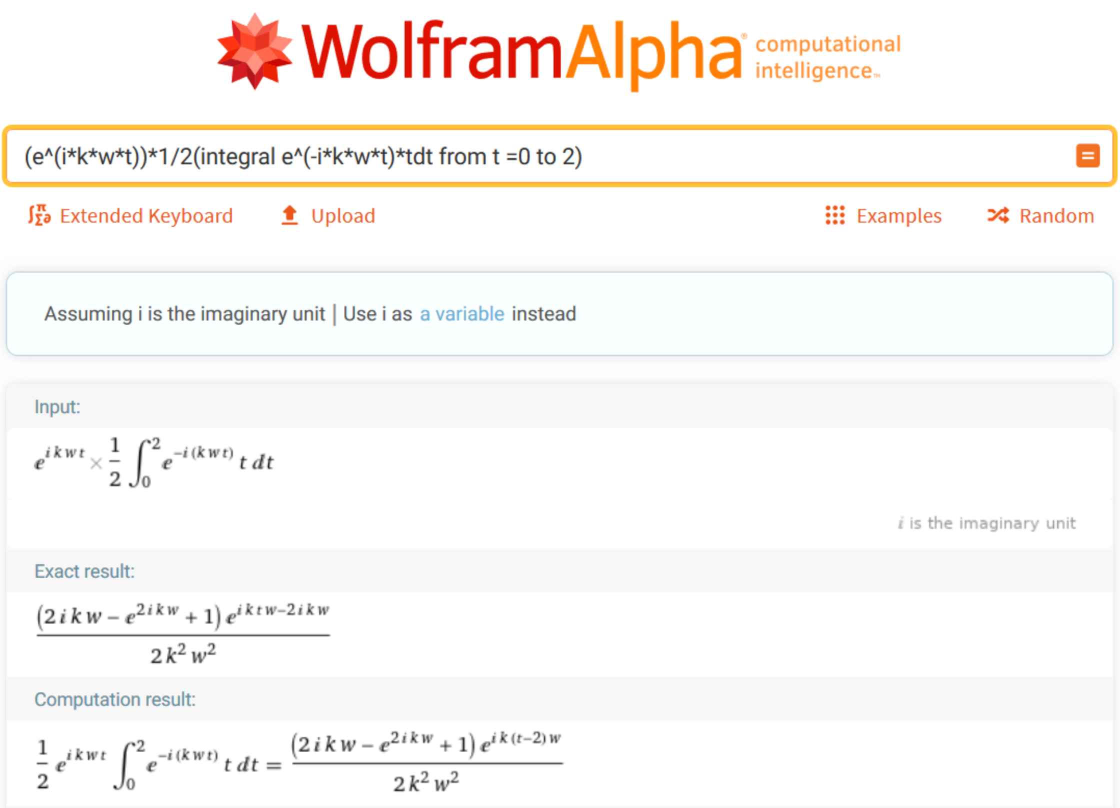
$$\omega_0 = \frac{2 * \pi}{T_0} = \frac{2 * \pi}{2} = \pi\, rad/s$$

Para a função (g) temos que:

$$D_n = \left(e^{-ik\omega_0 t}\right)\left(\frac{1}{2}\right)\int_0^2 \left(e^{-jk\omega_0 t}\right)(t)dt$$

4.2.2 Integral no software WolframAlpha para a figura (g)

(e^(i*k*w*t))*1/2(integral e^(-i*k*w*t)*tdt from t =0 to 2)



https://www.wolframalpha.com/input/?i=%28e%5E%28i*k*w*t%29%29*1%2F2%28integral+e%5E%28-i*k*w*t%29*tdt+from+t+%3D0+to+2%29

https://www.youtube.com/watch?v=EKoo0gWHFiY

4.2.3 Expressão Geral para o cálculo dos coeficientes da série Exponencial de Fourier para a figura (g)

$$\boxed{D_n = \frac{\left(2ik\omega - e^{2ik\omega} + 1\right)\left(e^{ik(t-2)\omega}\right)}{2k^2\omega^2}}$$

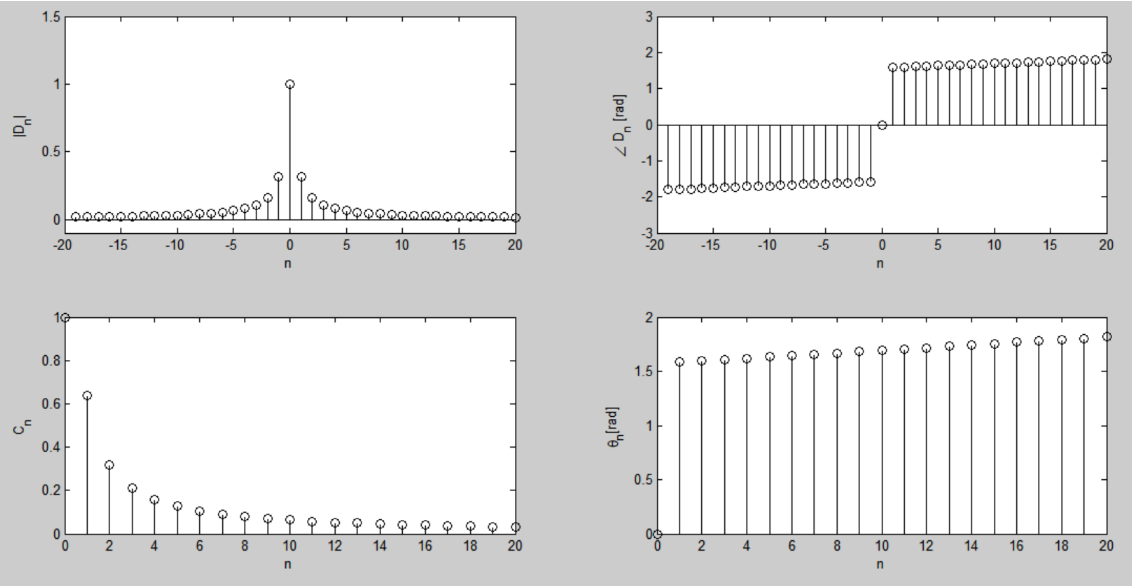
4.2.4 Código para visualização do Espectro de Fourier no *MATLAB* para a figura (g)

```
close all
clear all

clc
clear


%Estabelecendo os parâmetros básicos para plotagem do Espectro Exponencial de
Fourier

%Período de 2 segundos
T_0 = 2;

%Como x(t) é continuo, Dn
%cai lentamente, em função de 1/n².
%Representa aproximadamente 1% da freqüência fundamental (100)= 2^8
N_0 = 256;

%Período de amostragem
T = T_0/N_0;

%intervalo de amostragem
t = (0:T:T*(N_0-1))';

%número finito de coeficientes para a aproximação da fft
M=20;

%função (g)rampa com t0 =2
x = (t.*(t>=0).*(t<2));

%Plotando na mesma figura todos os gráficos
figure(1)

%Algoritmo para calcular a Transformada rápida de Fourier da função rampa e
%aproximar o espectro exponencial de Fourier %para -M<=N<=M
D_n = fft(x)/N_0;

%valores das amostras como sendo a média dos valores da função nos dois lados
%da descontinuidade
n = [-N_0/2:N_0/2-1]';

%Limpar janela de figura atual
clf;

%Linha 1 Coluna 1 da figura 1
subplot (2,2,1);

%Plotando o módulo de D_n em função de n
%usando abs para calcular o módulo de D_n e para calcular a transformada de
Fourier usar fftshift para reorganizar deslocando o componente de freqüência
zero para o centro da matriz.
stem(n,abs(fftshift(D_n)),'k');

%definindo os limites dos eixos do gráfico da Linha 1 Coluna 1 da figura 1
axis ([-M M -.1 1.5]);

%titulo para o elemento do eixo x
```

```
xlabel ('n');

%titulo para o elemento do eixo y
ylabel('|D_n|');

%Linha 1 Coluna 2 da figura 1
subplot (2,2,2);
%Plotando o angulo de D_n em função de n usando angle para calcular o ângulo e
%usando fftshift para reorganizar deslocando o componente de freqüência zero
para o centro da matriz.
stem(n,angle(fftshift(D_n)),'k');

%definindo os limites dos eixos do gráfico da Linha 2 Coluna 2 da figura 1
axis ([-M M -3 3]);

%titulo para o elemento do eixo x
xlabel ('n');

%titulo para o elemento do eixo y
ylabel('\angle D_n [rad]');

%definindo o intervalodo espectro trigonométrico de Fourier aproximado
n = [0:M];

%Definindo o coeficiente C_n(1) do espectro trigonométrico de Fourier como
%sendo o módulo de D_n da transformada de Fourier X usando fftshift para
%reorganizar deslocando o componente de freqüência zero para o centro da
%matriz.
C_n(1)= abs(D_n(1));

%Definindo o coeficiente dos  demais C_n do espectro trigonométrico de Fourier
%como sendo o módulo de D_n da transformada de Fourier X usando fftshift para
%reorganizar deslocando o componente de frequência zero para o centro da
%matriz.
C_n(2:M+1) = 2*abs(D_n(2:M+1));

%usando a função angle para calcular o ângulo repassando o valor de D_n(1)
%para theta_n(1)
theta_n(1) = angle (D_n(1));

%usando a função angle para calcular o ângulo repassando o valor de
%(D_n(2:M+1)) para theta_n(2:M+1)
theta_n(2:M+1) = angle(D_n(2:M+1));

%Linha 2 Coluna 1 da figura 1
subplot (2,2,3);

%Plotando os coeficientes trigonométricos de fourier em função de n
stem(n,C_n,'k');

%titulo para o elemento do eixo x
xlabel('n');

%titulo para o elemento do eixo y
ylabel('C_n');

subplot (2,2,4);
%Linha 2 Coluna 2 da figura 1

%Plotando o ângulo dos coeficientes trigonométricos de fourier em função de n
```

```
stem(n,theta_n,'k');

%titulo para o elemento do eixo x
xlabel('n');

%titulo para o elemento do eixo y
ylabel('\theta_n[rad]');
```

Figura 12 – Espectro de Fourier do Sinal periódico representado por (g)



Fonte: (Elaboração própria).

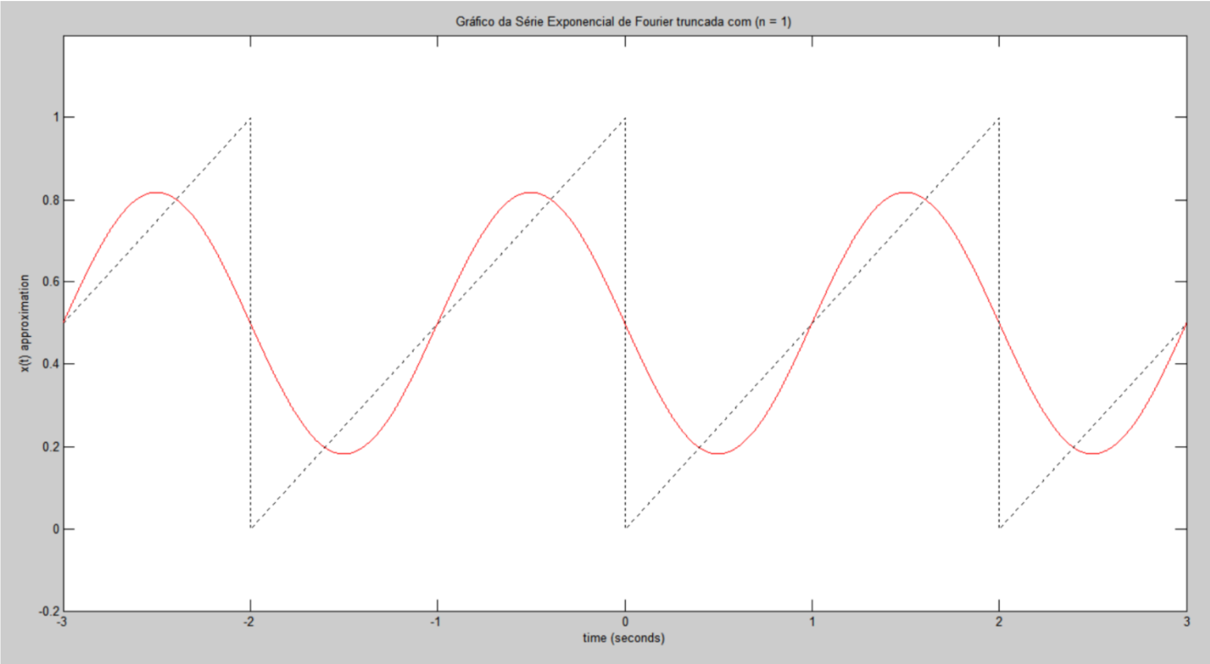
4.2.5 Código para calcular o somatório dos Coeficientes D_n da Série de Fourier na forma exponencial no *MATLAB* da figura (g)

```
close all
clear all

clc
clear

%CONSTANTES%

% 1)Definir parâmetros para plotar a onda dente de serra/rampa
tr = 2*[-2 -1 -1 0 0 1 1 2 2],...
yr = [0 1 0 1 0 1 0 1 0],...

%Período de 2 segundos para a figura (g)
T= 2;

%Número de harmônicas(n)
n=9;

%Valor do limite de tempo
```

```
M=3;

%Freqüência de amostragem
Fs = 0.001;

%Definindo o intervalo de tempo
intervalo = -M:Fs:M;

%Valor de ajuste de amplitude da função a ser amostrada
mult = 2;


%Inicio do indice
indice = 1;

%Freqüência Fundamental
w=2.0*pi/T;

%Inicio do loop
%repassando os parâmetros de intervalo para a variável t
for t= intervalo

    %a variável da Série Exponencial de Fourier da função (g) com
valor 0
    valor = 0.0;

    %a variável k [e limitada pelos valores de k para contemplar os
dois lados da descontinuidade
    for k = -n:n

        %tratando exceção para o caso de uma divisão por zero
        if (k ~=0.0)

        %Série Exponencial de Fourier da função (g) calulada no
software
        %WolframAlpha
        valor = valor +( ( (2*i*k*w - exp(2*i*k*w)+1) * (exp(i*k*t*w -
2*i*w*k)) ) / (2*(k^2)*w^2));


        %senão
        else

            %incrementa o valor da função (g)
            valor = valor +1;

        %fim do tratamento da função (g)
        end
     %fim do repasse dos valores de n para k
    end

    %passando os parâmetros para a variável res seguindo o índice
    res (indice) = (valor/mult);

    %incrementa o valor dda vari[avel índice
    indice = indice +1;
end
%fim do loop
```

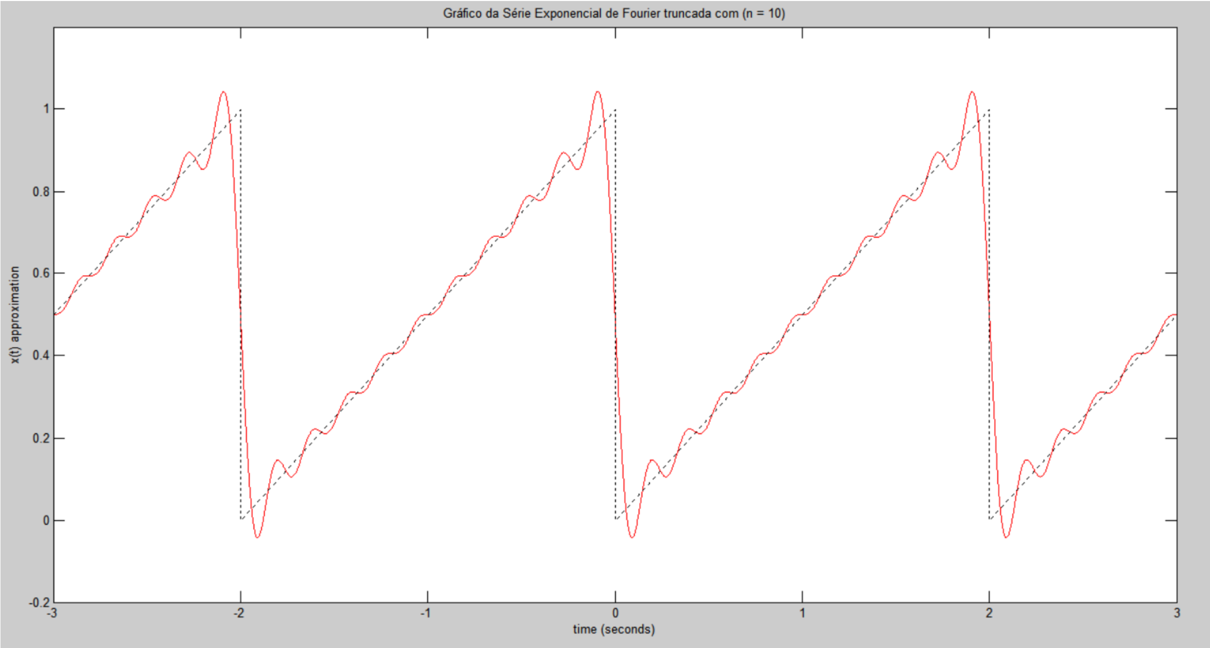
```
%Plotando a série de Fourier Truncada com n aproximações
%Criando a figura 1
figure(1);

%Plotando a Série Exponencial de Fourier da função (g) calulada no
software
%em função do intervalo (em vermelho)
plot (intervalo, res,'r');

%Mantém  no mesmo gráfico a próxima plotagem
hold;
%Plotando o sinal original (traçejado e preto)
plot(tr,yr,'k:');

%titulo para o elemento do eixo x
xlabel('time (seconds)');

%titulo para o elemento do eixo y
ylabel(['x(t) approximation']);

%titulo para a figura 1
title(['Gráfico da Série Exponencial de Fourier truncada com (n =
{',num2str(n),'})']);

%definindo os limites dos eixos do gráfico da figura 1
axis ([-M M -0.2 1.2]);
```

Figura 13 – Gráfico do sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier utilizando 1 harmônica



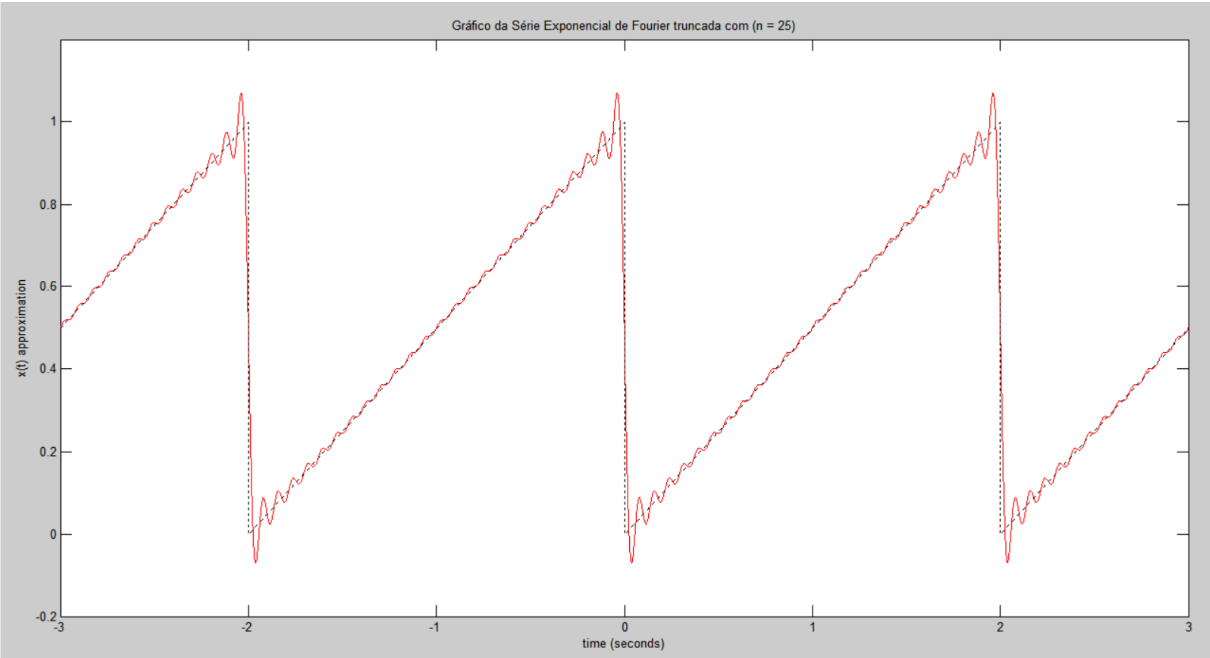
Fonte: (Elaboração própria).

Figura 14 – Gráfico do sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier utilizando 10 harmônicas



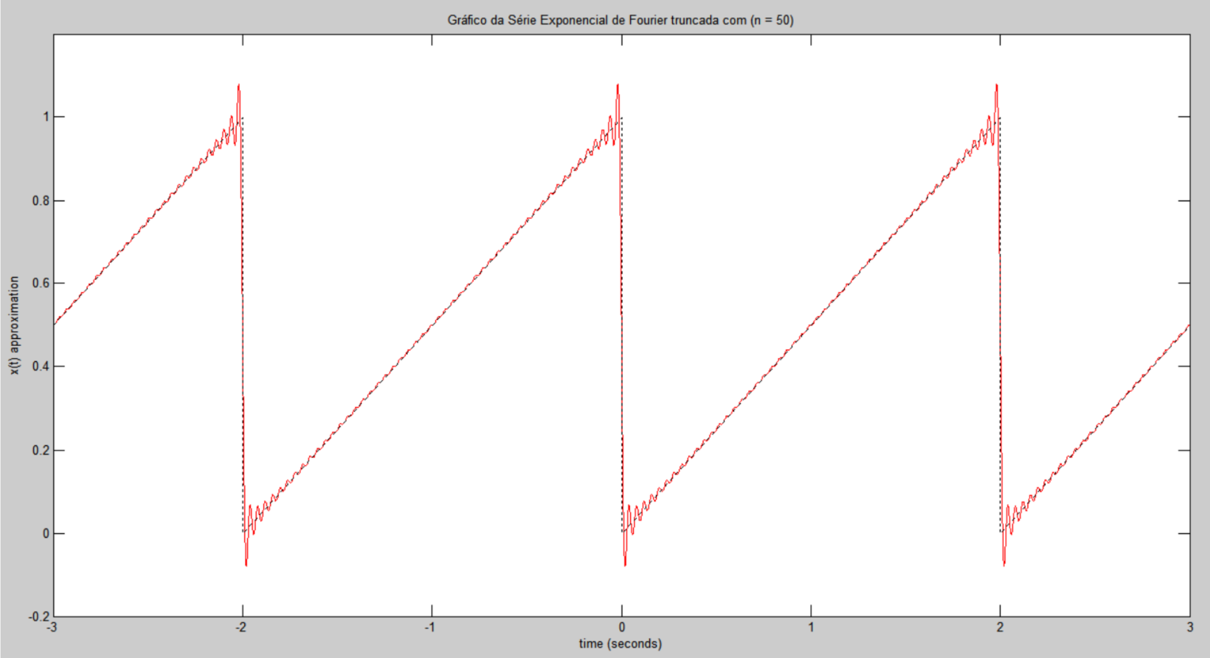
Fonte: (Elaboração própria).

Figura 15 – Gráfico do sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier utilizando 25 harmônicas



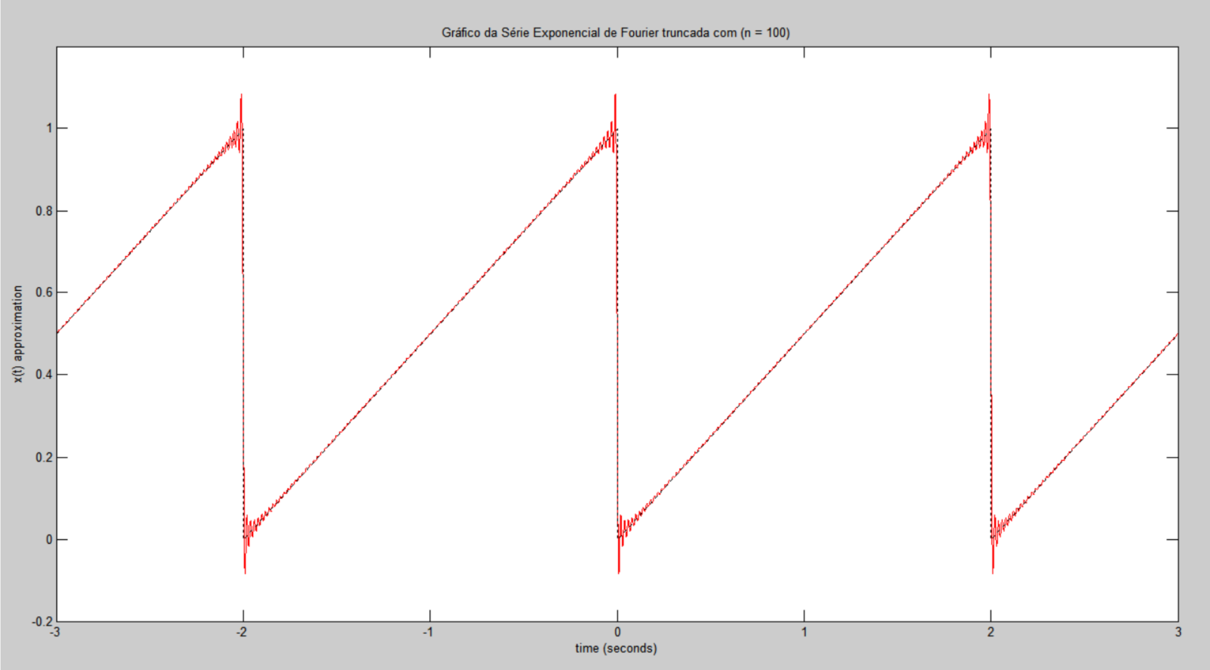
Fonte: (Elaboração própria).

Figura 16 – Gráfico do sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier utilizando 50 harmônicas



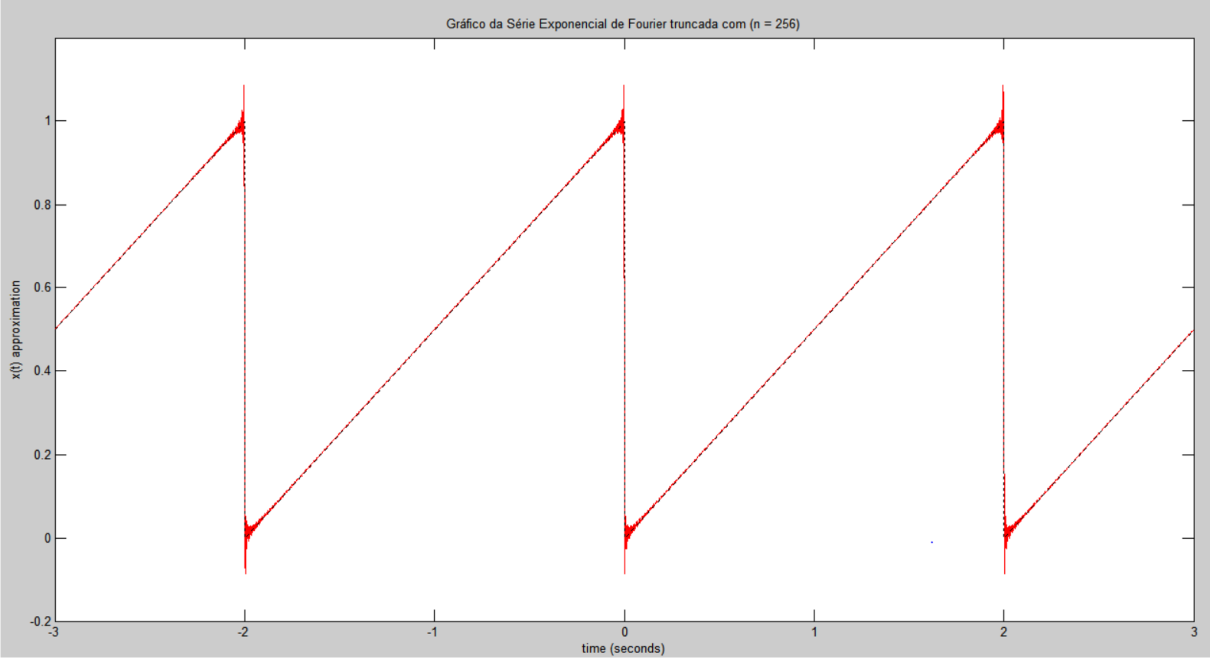
Fonte: (Elaboração própria).

Figura 17 – Gráfico do sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier utilizando 100 harmônicas



Fonte: (Elaboração própria).

Figura 18 – Gráfico do sinal original juntamente com sua versão sintetizada através dos coeficientes da Série Exponencial de Fourier utilizando 256 harmônicas



Fonte: (Elaboração própria).

4.2.6 Cálculo da distorção harmônica para a figura (g)

$$Potencia_g = \frac{4}{T_0}\int_0^{\frac{T_0}{4}} x(t)dt = \frac{4}{T_0}\int_0^{\frac{T_0}{4}} (t)^2 dt = \frac{4}{2}\int_0^{\frac{2}{4}} \left(\frac{t}{2}\right)^2 dt$$

$$Potencia_g = \frac{2}{4}\frac{t^3}{3}\Big|_0^{1/2} = \frac{1}{2}\left(\frac{1}{2}\right)^3 = 0,0625$$

A potencia do sinal desejado é: $\frac{1^2}{2} = 0,5$

$$D_{total} = \frac{0,0625}{0,5} * 100 = 12,50\%$$

5 CONCLUSÃO

Qualquer sinal periódico pode ser expresso por uma série de Fourier. Neste relatório, a partir dos sinais (e) (g) que foram decompostos por uma série Exponencial de Fourier, ou seja, a igualdade dos dois lados da equação é feita no sentido da média quadrática.

O *script* desenvolvido é capaz de usar apenas um número finito (n) de termos para a série, que se mostrou altamente desejável para todo valor de t garantindo a convergência com um erro arbitrário pequeno. Portanto, para os sinais apresentados as séries exponenciais decompostas podem ser chamadas de séries uniformemente convergentes.

Os gráficos de série truncada se mostraram muito próximos da função x(t) quando aumentamos o número de harmônicas (n), convergindo aproximadamente para x(t). Para (n) grande, as série exibiram um comportamento oscilatório e um sobre-sinal aproximadamente de 9% na proximidade da descontinuidade no pico mais próximo da oscilação para os sinais sintetizados das figuras (e) e (g). O salto de descontinuidade que causa o efeito de Gibbs.

Foi necessário ajustar a freqüência de amostragem porque freqüências baixas afetam o comportamento em grande escala de x(t), enquanto que em altas freqüências determinam a estrutura fina, tal como uma rápida variação. Portanto, mudanças bruscas em x(t), necessitam de altas freqüências, porque quanto mais brusca a variação (maior a derivada temporal de x(t)) maiores as freqüências necessárias na série.

Um sinal suave tem uma variação da derivada temporal menos rápida, logo a síntese desse sinal requer senóides de freqüência predominantemente baixas e uma pequena quantidade de senóides que variam rapidamente (alta freqüência). Logo, o espectro de amplitude de tal função decai rapidamente com a freqüência e poucos termos da série de Fourier foram necessários para uma boa aproximação.

Um sinal com mudanças muito bruscas, tais como saltos de descontinuidades, contendo variações muito rápidas tem uma variação da derivada temporal mais rápida, logo a síntese desse sinal requer na sua síntese uma quantidade relativamente grande de componentes de alta freqüência. Logo, o espectro de amplitude de tal função decai lentamente com a freqüência e muitos termos da série de Fourier são necessários para uma boa aproximação.

Em resumo, o sinal da figura (e) é uma onda quadrada descontínua com saltos de descontinuidade, e por isso seu espectro de fase de amplitude que indica o total (amplitudes) das várias componentes de freqüência de x(t), decaiu lentamente, com $\frac{1}{n}$. Em contrapartida o sinal da figura (g) é uma onda dente-de-serra é mais suave, pois é uma função contínua (sem saltos de descontinuidade e seu espectro decaiu rapidamente com a freqüência, com $\frac{1}{n^2}$.

Para constatar a relação entre o período fundamental e o espaçamento das componentes no eixo de freqüência o papel do espectro de fase é crucial para que se possa conseguir uma mudança rápida na forma de onda, fazendo com que a fase de todas as infinitas componentes são positivas exatamente antes de t = 1, tornando-se negativas após t = 1. Portanto, para sintetizar uma mudança instantânea em um salto de descontinuidade, as fases das várias componentes senoidais do espectro do sinal x(t) devem ser tais que todas as componentes harmônicas tenham um sinal antes da descontinuidade e o sinal oposto após a descontinuidade, resultando numa mudança brusca em x(t) no ponto de descontinuidade.

Combinando as amplitudes e as fases das várias senoides resultaram no espectro de Fourier de x(t), sendo assim os sinais originais puderam ser sintetizados.

Os valores dos cálculos da distorção harmônica não saíram como desejados.

O anexo 1 mostra o código final para a figura (e) .

O anexo 2 mostra o código final para a figura (g) .

Para acessar o repositório com o desenvolvimento dessa atividade e outros, acesse:

https://github.com/engineerIOT/Sinais-e-Sistemas

Para a figura (e):

https://github.com/engineerIOT/Sinais-e-Sistemas/blob/master/IntegracaoExponencialFourierQuadrada.m

Para a figura (g):

https://github.com/engineerIOT/Sinais-e-Sistemas/blob/master/IntegracaoExponencialFourierRampa.m

6 ANEXOS

6.1.1 ANEXO 1 - Código final para figura (e)

```
close all
clear all

clc
clear

%Criando comunicacao com o usuario
prompt = 'Usuário favor digitar o numero de harmônicas a serem utilizadas
na sintetização do sinal da figura (e): ';

%passando o valor de entrada para a variavel
valorDigitado = input(prompt)


%Número de harmônicas(n)
n=valorDigitado;

%Valor do limite de tempo
M=6;

%Definindo o intervalo de tempo
intervalo = -M:0.001:M;

%Freqüência de amostragem
Fs = 0.001;

%Inicio do indice
indice = 1;

%Período de 6 segundos para a figura (e)
T= 6;

%Freqüência Fundamental
w=2.0*pi/T;

%Inicio do loop
%repassando os parâmetros de intervalo para a variável t
for t= intervalo

    %a variável da Série Exponencial de Fourier da função (e) com valor 0
     valor = 0.0;

         %a variável k [e limitada pelos valores de k para contemplar os
         %dois lados da descontinuidade
     for k = -n:n;

        %tratando exceção para o caso de uma divisão por zero
        if (k ~=0.0)

      %Série Exponencial de Fourier da função (g) calulada no software
      %WolframAlpha
           termo = (i*(-1+ exp(i*k*w))*(-1+ exp(i*k*w)));
```

```
            termo2= (exp(i*k*w)+ exp(2*i*k*w)+1);
            termo3 = exp((i*k*w*(t-2)));
            valor = (valor +( termo*termo2*termo3 )*(-1) / (6*k*w));

         %senão
         else

             %incrementa o valor da função (e)
             valor = valor +1;

         %fim do tratamento da função (e)
         end

     %fim do repasse dos valores de n para k
      end

     %passando os parâmetros para a variável res seguindo o índice
     res (indice) = (valor-1);

     %incrementa o valor dda vari[avel índice
     indice = indice +1;

 %fim do loop
 end

 %Plotando a série de Fourier Truncada com n aproximações
 %Definindo o intervalo de tempo para a fun;ão da figura (e)
 teste=-M:0.001:M;

 %Criando o sinal original (traçejado e preto) da figura (e)
 fx = -(teste>=1).*(teste<2)+(teste>=4).*(teste<5)-((teste>=-5).*(teste<-
 4))+((teste>=-2).*(teste<-1));

 %Criando a figura 1
 figure(1)

 %Plotando a Série Exponencial de Fourier da função (e) calulada no
 software
 %em função do intervalo (em vermelho)
 plot (intervalo, res,'r');

 %Mantém  no mesmo gráfico a próxima plotagem
 hold;
 %Plotando o sinal original (traçejado e preto)
 plot(teste,fx,'k:');

 %titulo para o elemento do eixo x
 xlabel('t(s)');

 %titulo para o elemento do eixo y
 ylabel(['x(t) approximation']);

 %titulo para a figura 1
 title(['Gráfico da Série Exponencial de Fourier truncada com (n =
 {',num2str(n),'})']);

 %definindo os limites dos eixos do gráfico da figura 1
```

```
axis ([-M M -1.2 1.2]);

%Definir função da figura (e)
x = inline('(-(t>=1).*(t<2))+((t>=-2).*(t<-1))','t');

%Freqüência de amostragem
Fs = 0.001;

%intervalo de tempo por oscilação
t =(-6/4:Fs:6/4);

%plotar o gráfico
figure(2)

%Plotando a série função da figura (e) em função do tempo
%Criando a figura 2
plot(t,x(t))

%titulo para o elemento do eixo x
xlabel('t(s)');

%titulo para o elemento do eixo y
ylabel('x(t)');

%definindo os limites dos eixos do gráfico da figura 2
axis ([-3 3 -1.2 1.2]);

%titulo para a figura 2
title(['Gráfico da função da figura (e)  (n = {',num2str(n),'})']);

xlabel('t(s)');

%titulo para o elemento do eixo y
ylabel('x(t)');

%titulo para a figura 2
title(['Gráfico da função da figura (e)  (n = {',num2str(n),'})']);

%Calcula a integral
Potencia_g = (4/T)*sum((x(t).^2*Fs))

%Calcula a distorção harmônica
Dtotal = ((Potencia_g)/n*n)*100

%Px=abs(valor)

%Estabelecendo os parâmetros básicos para plotagem do Espectro Exponencial
de Fourier
%Parametros basicos

%Período de 2 segundos
T_0 = 6;

%Como x(t) possui descontinuidade, Dn
%cai lentamente, em função de 1/n.
%Representa aproximadamente 1% da freqüência fundamental (100)= 2^8
N_0 = 256;
```

```
%Período de amostragem
T = T_0/N_0;

%intervalo de amostragem
t = (0:T:T*(N_0-1))';

%número finito de coeficientes para a aproximação da fft
M=20;

%funcao(e)
x=(-(t>=1).*(t<2))+((t>=-2).*(t<-1));

%Plotando na mesma figura todos os gráficos
figure(3)

%Algoritmo para calcular a Transformada rápida de Fourier da função rampa
e
%aproximar o espectro exponencial de Fourier %para -M<=N<=M
D_n = fft(x)/N_0;

%valores das amostras como sendo a média dos valores da função
%nos dois lados %da descontinuidade
n = [-N_0/2:N_0/2-1]';

%Limpar janela de figura atual
clf;

%Linha 1 Coluna 1 da figura
subplot (2,2,1);

%Plotando o módulo de D_n em função de n
%usando abs para calcular o módulo de D_n e para calcular a transformada
%de %Fourier usar fftshift para reorganizar deslocando o componente
%de freqüência zero para o centro da matriz.
stem(n,abs(fftshift(D_n)),'k');

%definindo os limites dos eixos do gráfico da Linha 1 Coluna 1 da figura 1
axis ([-M M 0 0.2]);

%titulo para o elemento do eixo x
xlabel ('n');

%titulo para o elemento do eixo y
ylabel('|D_n|');

%Linha 1 Coluna 2 da figura 1
subplot (2,2,2);

%Plotando o angulo de D_n em função de n usando angle para calcular o
%ângulo e %usando fftshift para reorganizar deslocando o componente de
%freqüência zero para o centro da matriz.
stem(n,angle(fftshift(D_n)),'k');

%definindo os limites dos eixos do gráfico da Linha 2 Coluna 2 da figura 1
axis ([-M M -4 4]);
```

```
%titulo para o elemento do eixo x
xlabel ('n');

%titulo para o elemento do eixo y
ylabel('\angle D_n [rad]');


%definindo o intervalodo espectro trigonométrico de Fourier aproximado
n = [0:M];

%Definindo o coeficiente C_n(1) do espectro trigonométrico de Fourier como
%sendo o módulo de D_n da transformada de Fourier X usando fftshift para
%reorganizar deslocando o componente de freqüência zero para o centro da
%matriz.
C_n(1)= abs(D_n(1));

%Definindo o coeficiente dos  demais C_n do espectro trigonométrico de
%Fourier %como sendo o módulo de D_n da transformada de Fourier X
%usando fftshift para %reorganizar deslocando o componente de frequência
%zero para o centro da %matriz.
C_n(2:M+1) = 2*abs(D_n(2:M+1));

%usando a função angle para calcular o ângulo repassando o valor de D_n(1)
%para theta_n(1)
theta_n(1) = angle (D_n(1));

%usando a função angle para calcular o ângulo repassando o valor de
%(D_n(2:M+1)) para theta_n(2:M+1)
theta_n(2:M+1) = angle(D_n(2:M+1));

%Linha 2 Coluna 1 da figura 1
subplot (2,2,3);

%Plotando os coeficientes trigonométricos de fourier em função de n
stem(n,C_n,'k');

%titulo para o elemento do eixo x
xlabel('n');

%titulo para o elemento do eixo y
ylabel('C_n');

%Linha 2 Coluna 2 da figura 1
subplot (2,2,4);

%Plotando o ângulo dos coeficientes trigonométricos de fourier
%em função de n
stem(n,theta_n,'k');

%titulo para o elemento do eixo x
xlabel('n');

%titulo para o elemento do eixo y
ylabel('\theta_n[rad]');
```

6.1.2 ANEXO 2 - Código final para figura (g)

```
close all
clear all

clc
clear

%CONSTANTES%

% 1)Definir parametros para plotar a onda dente de serra/rampa
tr = 2*[-2 -1 -1 0 0 1 1 2 2],...
yr = [0 1 0 1 0 1 0 1 0],...

%Período
T= 2;

%Criando comunicacao com o usuario
prompt = 'Usuário favor digitar o numero de harmônicas a serem utilizadas
na sintetização do sinal da figura (g): ';

%passando o valor de entrada para a variavel
valorDigitado = input(prompt)


%Número de harmônicas(n)
n=valorDigitado;

%Valor do limite de tempo
M=3;

%Frequencia de amostragem
Fs = 0.001;

%Definindo o intervalo de tempo
intervalo = -M:Fs:M;

%Valor de ajuste de amplitude da função a ser amostrada
mult = 2;


%Inicio do indice
indice = 1;

%Freqüência Fundamental
w=2.0*pi/T;

%Inicio do loop
%repassando os parâmetros de intervalo para a variável t
for t= intervalo

    %a variável da Série Exponencial de Fourier da função (g) com valor 0
    valor = 0.0;

    %a variável k [e limitada pelos valores de k para contemplar os dois
lados da descontinuidade
    for k = -n:n
```

```
        %tratando exceção para o caso de uma divisão por zero
        if (k ~=0.0)

        %Série Exponencial de Fourier da função (g) calulada no software
        %WolframAlpha
        valor = valor +( ( (2*i*k*w - exp(2*i*k*w)+1) * (exp(i*k*t*w -
2*i*w*k)) ) / (2*(k^2)*w^2));

        %senão
        else

            %incrementa o valor da função (g)
            valor = valor +1;

        %fim do tratamento da função (g)
        end
     %fim do repasse dos valores de n para k
    end

    %passando os parâmetros para a variável res seguindo o índice
    res (indice) = (valor/mult);

    %incrementa o valor dda vari[avel índice
    indice = indice +1;
end
%fim do loop

%Plotando a série de Fourier Truncada com n aproximações
%Criando a figura 1
figure(1);

%Plotando a Série Exponencial de Fourier da função (g) calulada no
software
%em função do intervalo (em vermelho)
plot (intervalo, res,'r');
hold;

%Mantém  no mesmo gráfico a próxima plotagem

%Plotando o sinal original (traçejado e preto)
plot(tr,yr,'k:');
hold;
%titulo para o elemento do eixo x
xlabel('time (seconds)');

%titulo para o elemento do eixo y
ylabel(['x(t) approximation']);

%titulo para a figura 1
title(['Gráfico da Série Exponencial de Fourier truncada com (n =
{',num2str(n),'})']);

%definindo os limites dos eixos do gráfico da figura 1
axis ([-M M -0.2 1.2]);

%intervalo de tempo por oscilação
t =(0:Fs:T);
```

```
%Definir função da figura (g)
x = inline('(t.*(t>=0).*(t<2))','t');

%Freqüência de amostragem
Fs = 0.001;

%plotar o gráfico
figure(2)

%Plotando a função da figura (g) em função do tempo
%Criando a figura 1
plot(t,x(t))

%titulo para o elemento do eixo x
xlabel('t(s)');

%titulo para o elemento do eixo y
ylabel('x(t)');

axis ([-1 M -0.2 2.2]);

%titulo para a figura 2
title(['Gráfico da função da figura (g)  (n = {',num2str(n),'})']);

%Calcula a integral
Potencia_g = (4/T)*sum(((x(t)/T).^2)*Fs)/2

%Calcula a distorção harmônica
Dtotal = (Potencia_g/0.5)*100

%Potencia_x=(1/T)*sum((x(t).^2*Fs));



%Período de 2 segundos
T_0 = 2;

%Como x(t) é contínuoe, Dn
%cai lentamente, em função de 1/n².
%Representa aproximadamente 1% da freqüência fundamental (100)= 2^8
N_0 = 256;

%Período de amostragem
T = T_0/N_0;

%intervalo de amostragem
t = (0:T:T*(N_0-1))';

%número finito de coeficientes para a aproximação da fft
M=20;

%função (g)rampa com t0 =2
x = (t.*(t>=0).*(t<2));

%Plotando na mesma figura todos os gráficos
figure(3)
```

```
%Algoritmo para calcular a Transformada rápida de Fourier da função rampa
e
%aproximar o espectro exponencial de Fourier %para -M<=N<=M
D_n = fft(x)/N_0;

%valores das amostras como sendo a média dos valores da função nos dois
lados %da descontinuidade
n = [-N_0/2:N_0/2-1]';

%Limpar janela de figura atual
clf;

%Linha 1 Coluna 1 da figura 1
subplot (2,2,1);

%Plotando o módulo de D_n em função de n
%usando abs para calcular o módulo de D_n e para calcular a transformada
de Fourier usar fftshift para reorganizar deslocando o componente de
freqüência zero para o centro da matriz.
stem(n,abs(fftshift(D_n)),'k');

%definindo os limites dos eixos do gráfico da Linha 1 Coluna 1 da figura 1
axis ([-M M -.1 1.5]);

%titulo para o elemento do eixo x
xlabel ('n');

%titulo para o elemento do eixo y
ylabel('|D_n|');

%Linha 1 Coluna 2 da figura 1
subplot (2,2,2);
%Plotando o angulo de D_n em função de n usando angle para calcular o
ângulo e
%usando fftshift para reorganizar deslocando o componente de freqüência
zero para o centro da matriz.
stem(n,angle(fftshift(D_n)),'k');

%definindo os limites dos eixos do gráfico da Linha 2 Coluna 2 da figura 1
axis ([-M M -3 3]);

%titulo para o elemento do eixo x
xlabel ('n');

%titulo para o elemento do eixo y
ylabel('\angle D_n [rad]');

%definindo o intervalodo espectro trigonométrico de Fourier aproximado
n = [0:M];

%Definindo o coeficiente C_n(1) do espectro trigonométrico de Fourier como
%sendo o módulo de D_n da transformada de Fourier X usando fftshift para
%reorganizar deslocando o componente de freqüência zero para o centro da
%matriz.
C_n(1)= abs(D_n(1));

%Definindo o coeficiente dos  demais C_n do espectro trigonométrico de
Fourier %como sendo o módulo de D_n da transformada de Fourier X usando
```

```
fftshift para %reorganizar deslocando o componente de frequência zero para
o centro da %matriz.
C_n(2:M+1) = 2*abs(D_n(2:M+1));

%usando a função angle para calcular o ângulo repassando o valor de D_n(1)
%para theta_n(1)
theta_n(1) = angle (D_n(1));

%usando a função angle para calcular o ângulo repassando o valor de
%(D_n(2:M+1)) para theta_n(2:M+1)
theta_n(2:M+1) = angle(D_n(2:M+1));

%Linha 2 Coluna 1 da figura 1
subplot (2,2,3);

%Plotando os coeficientes trigonométricos de fourier em função de n
stem(n,C_n,'k');

%titulo para o elemento do eixo x
xlabel('n');

%titulo para o elemento do eixo y
ylabel('C_n');

subplot (2,2,4);
%Linha 2 Coluna 2 da figura 1

%Plotando o ângulo dos coeficientes trigonométricos de fourier em função
de n
stem(n,theta_n,'k');

%titulo para o elemento do eixo x
xlabel('n');

%titulo para o elemento do eixo y
ylabel('\theta_n[rad]');
```