
Laboratório 4 - Revisão do Conceito de Série de Fourier

Table of Contents

Inicialização	1
Parâmetros Gerais	1
Função Auxiliar para Calcular Espectro	1
(a) Síntese das Formas de Onda Periódicas	2
Onda Quadrada	2
Onda Triangular	2
Onda Dente de Serra (Sawtooth)	3
(b) Impacto do Número de Componentes da Série de Fourier	4
Comparação - Onda Quadrada	5
Comparação - Onda Triangular	5
Comparação - Onda Dente de Serra	5
(c) Espectro de Frequências das Formas de Onda	7
Espectro da Onda Quadrada	7
Espectro da Onda Triangular	7
Espectro da Onda Dente de Serra	7
(d) Sinal Composto Específico	9
Parâmetros para o sinal composto	9
Definição do sinal $x(t)$	9
Gráfico temporal do sinal $x(t)$	9
Espectro de frequências do sinal $x(t)$	10
Análise dos Resultados	11

Disciplina: Processamento Digital de Sinais **Professor:** Marcelo Eduardo Pellenz **Alunos:** Francisco Bley e Stefan Rodrigues -----

Inicialização

```
clc; clear; close all;
```

Parâmetros Gerais

```
F = 20;           % Frequência fundamental (Hz)
Fs = 16000;       % Frequência de amostragem (Hz)
T = 1/F;          % Período fundamental
num_periods = 5;  % Número de períodos a simular
total_time = num_periods * T; % Tempo total de simulação
Ts = 1/Fs;        % Período de amostragem
t = 0:Ts:total_time; % Vetor de tempo
```

Função Auxiliar para Calcular Espectro

Função para calcular e plotar o espectro de frequências

```
function plot_spectrum(signal, Fs, title_str, fig_num)
    L = length(signal); % número de pontos
    Y = fft(signal); % FFT do sinal
    P2 = abs(Y/L); % espectro bilateral normalizado
    P1 = P2(1:floor(L/2)+1); % metade positiva
    P1(2:end-1) = 2*P1(2:end-1); % converter para espectro unilateral

    f = Fs*(0:floor(L/2))/L; % eixo de frequências

    figure(fig_num);
    stem(f, P1, 'Marker', 'none'); % usar stem para destacar harmônicos
    title(['Espectro de Frequências - ' title_str]);
    xlabel('Frequência (Hz)');
    ylabel('Amplitude');
    grid on;
    xlim([0 500]); % limitar a 500 Hz (suficiente para ver harmônicos)
end
```

(a) Síntese das Formas de Onda Periódicas

Utilizando o conceito da representação de formas de onda periódicas usando a Série de Fourier para traçar os gráficos das formas de onda quadrada, triangular e dente de serra.

```
N = 50; % Número de harmônicos para a síntese principal
```

Onda Quadrada

Fórmula da Série de Fourier: $x(t) = (4/\pi) * \sum [\sin(2\pi(2k-1)Ft)/(2k-1)]$

```
func_square = zeros(size(t));
for k = 1:2:(2*N-1)
    harmonic = (4/pi) * (1/k) * sin(2*pi*k*F*t);
    func_square = func_square + harmonic;
end

figure(1);
plot(t, func_square, 'LineWidth', 1.5);
title(['Onda Quadrada - Síntese com N = ' num2str(N) ' harmônicos']);
xlabel('Tempo (s)'); ylabel('Amplitude'); grid on;
xlim([0 total_time]);
```

Onda Triangular

Fórmula da Série de Fourier: $x(t) = (8/\pi^2) * \sum [(-1)^{(k-1)/2} * \sin(2\pi kFt)/k^2]$

```
func_triangle = zeros(size(t));
for k = 1:2:(2*N-1)
    harmonic = (8/pi^2) * ((-1)^((k-1)/2)) * (sin(2*pi*k*F*t)/(k^2));
    func_triangle = func_triangle + harmonic;
end

figure(2);
```

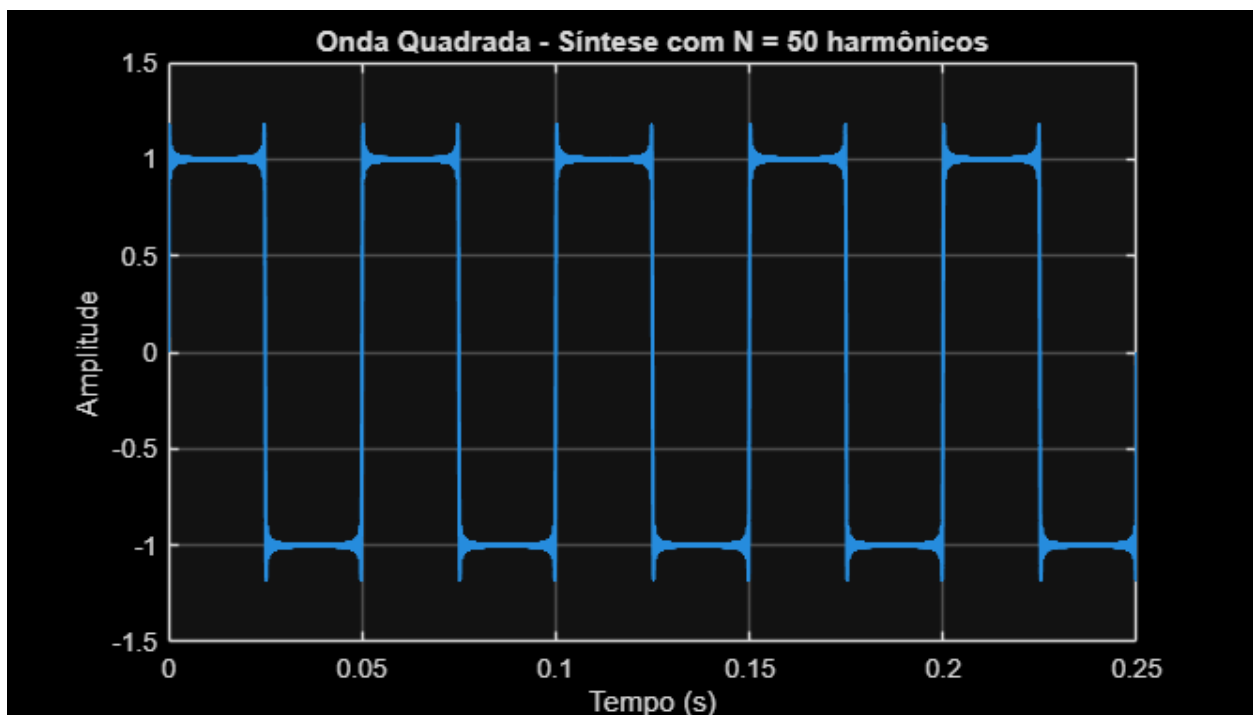
```
plot(t, func_triangle, 'LineWidth', 1.5);  
title(['Onda Triangular - Síntese com N = ' num2str(N) ' harmônicos']);  
xlabel('Tempo (s)'); ylabel('Amplitude'); grid on;  
xlim([0 total_time]);
```

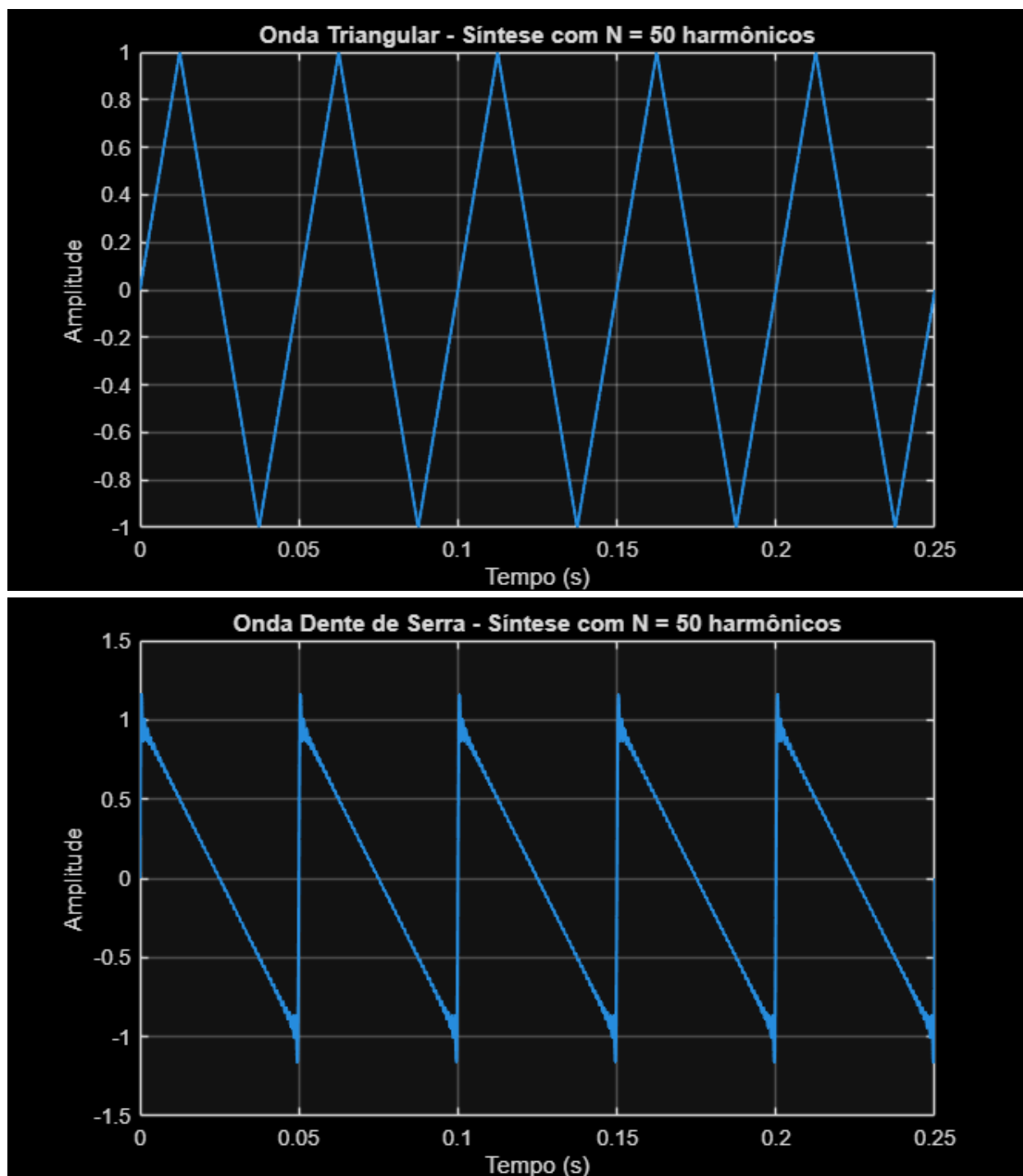
Onda Dente de Serra (Sawtooth)

Fórmula da Série de Fourier: $x(t) = (2/\pi) * \sum [\sin(2\pi k F t)/k]$

```
func_sawtooth = zeros(size(t));  
for k = 1:N  
    harmonic = (2/pi) * (sin(2*pi*k*F*t)/k);  
    func_sawtooth = func_sawtooth + harmonic;  
end
```

```
figure(3);  
plot(t, func_sawtooth, 'LineWidth', 1.5);  
title(['Onda Dente de Serra - Síntese com N = ' num2str(N) ' harmônicos']);  
xlabel('Tempo (s)'); ylabel('Amplitude'); grid on;  
xlim([0 total_time]);
```





(b) Impacto do Número de Componentes da Série de Fourier

Avaliação gráfica do impacto da composição dos termos da série na representação final da forma de onda periódica.

```
Ns = [1, 3, 5, 15, 50]; % valores de N a comparar
```

Comparação - Onda Quadrada

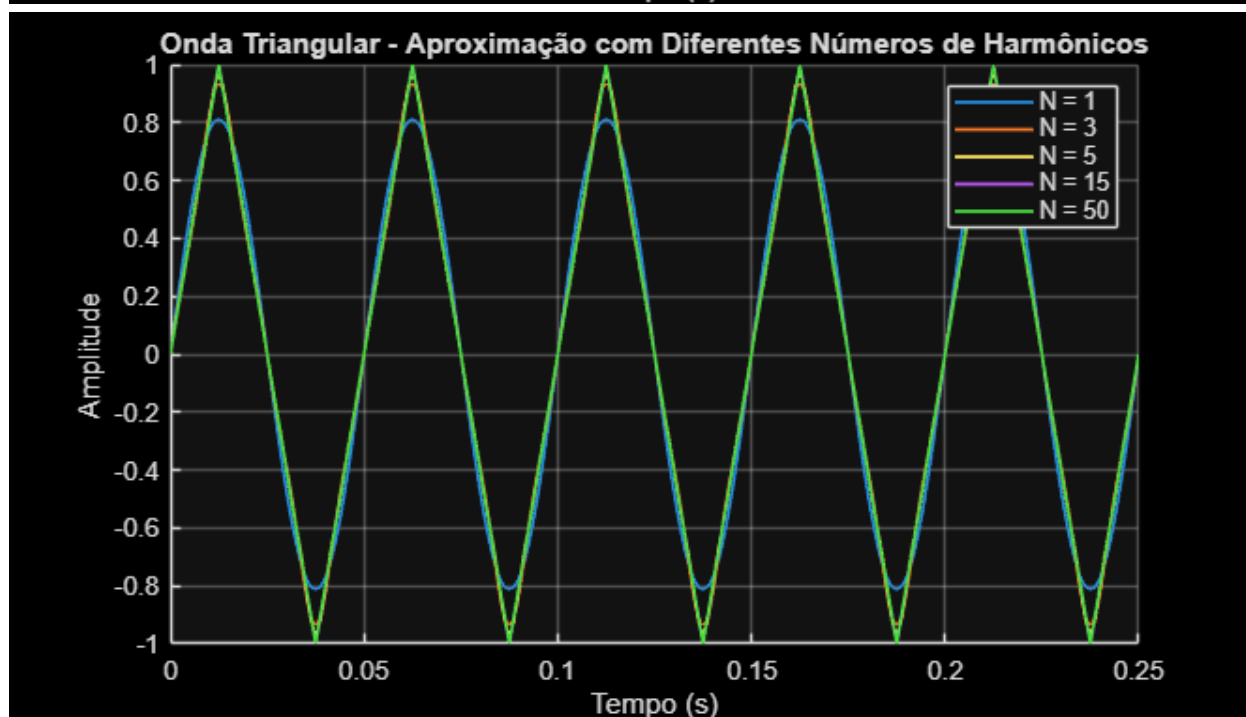
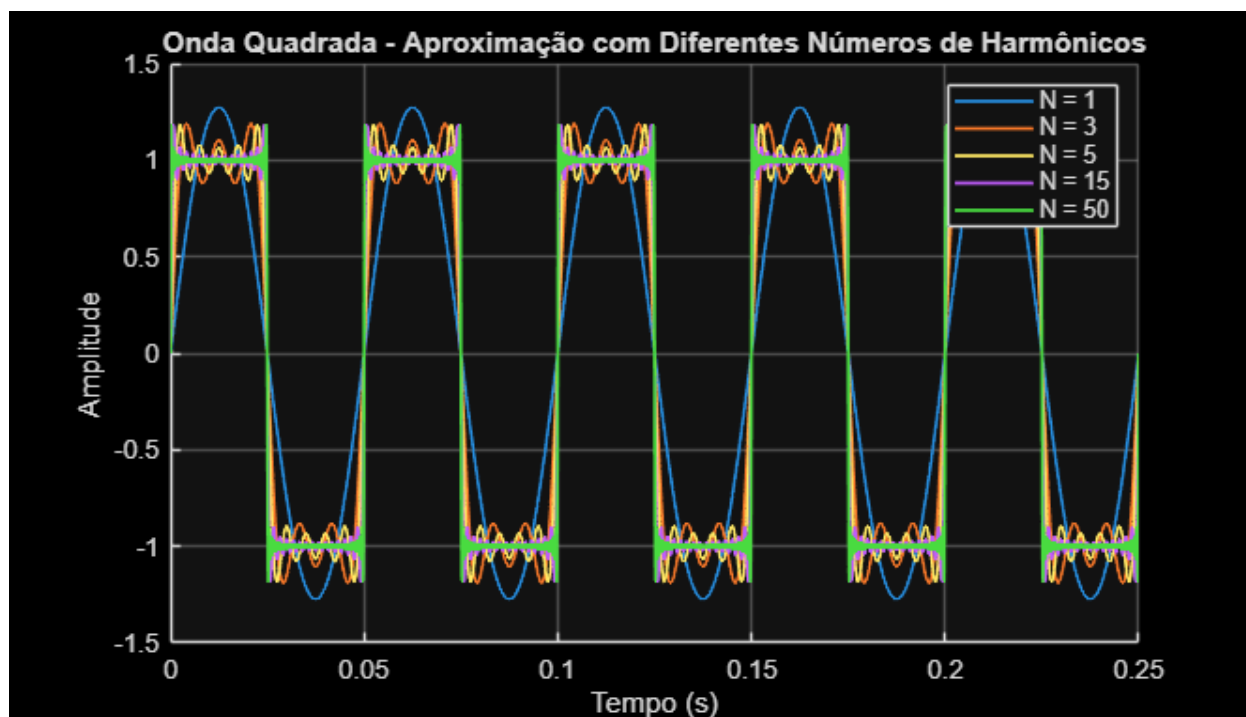
```
figure(4); hold on;
colors = lines(length(Ns));
for i = 1:length(Ns)
    N = Ns(i);
    y = zeros(size(t));
    for k = 1:2:(2*N-1)
        y = y + (4/pi) * (1/k) * sin(2*pi*k*F*t);
    end
    plot(t, y, 'Color', colors(i,:), 'LineWidth', 1.2, ...
        'DisplayName', ['N = ' num2str(N)]);
end
title('Onda Quadrada - Aproximação com Diferentes Números de Harmônicos');
xlabel('Tempo (s)'); ylabel('Amplitude'); legend show; grid on;
xlim([0 total_time]);
```

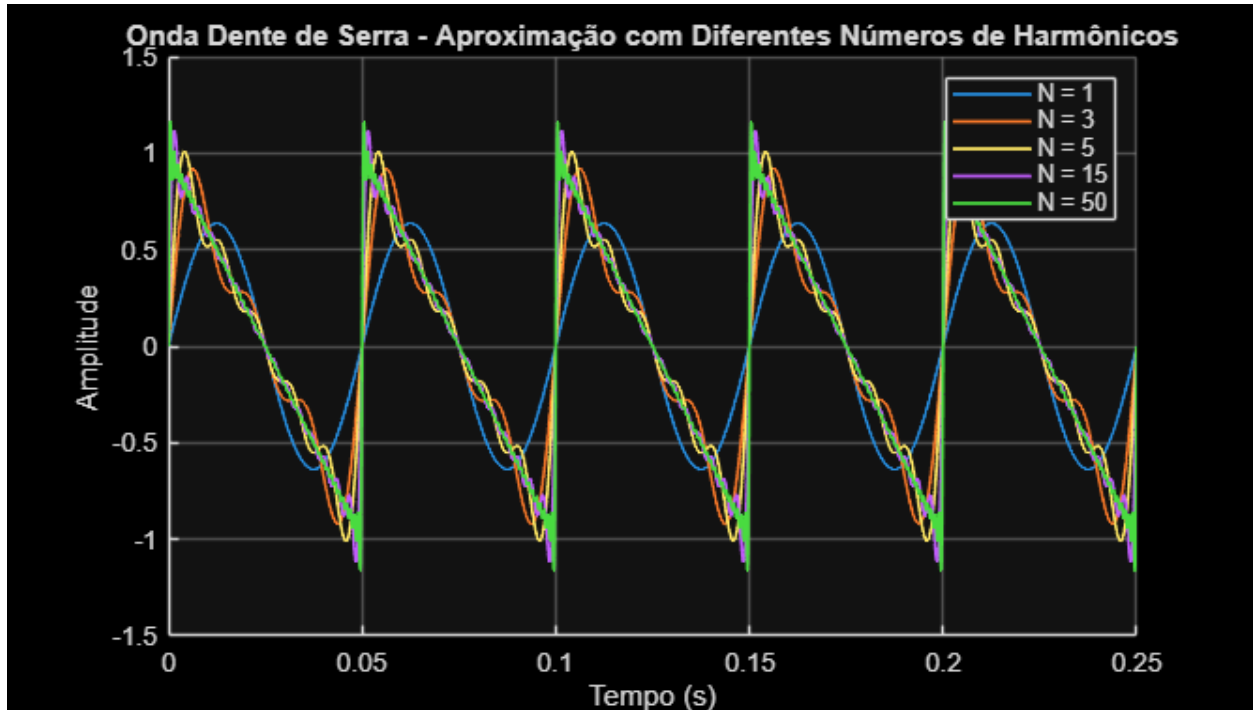
Comparação - Onda Triangular

```
figure(5); hold on;
for i = 1:length(Ns)
    N = Ns(i);
    y = zeros(size(t));
    for k = 1:2:(2*N-1)
        y = y + (8/pi^2) * ((-1)^((k-1)/2)) * (sin(2*pi*k*F*t)/(k^2));
    end
    plot(t, y, 'Color', colors(i,:), 'LineWidth', 1.2, ...
        'DisplayName', ['N = ' num2str(N)]);
end
title('Onda Triangular - Aproximação com Diferentes Números de Harmônicos');
xlabel('Tempo (s)'); ylabel('Amplitude'); legend show; grid on;
xlim([0 total_time]);
```

Comparação - Onda Dente de Serra

```
figure(6); hold on;
for i = 1:length(Ns)
    N = Ns(i);
    y = zeros(size(t));
    for k = 1:N
        y = y + (2/pi) * (sin(2*pi*k*F*t)/k);
    end
    plot(t, y, 'Color', colors(i,:), 'LineWidth', 1.2, ...
        'DisplayName', ['N = ' num2str(N)]);
end
title('Onda Dente de Serra - Aproximação com Diferentes Números de Harmônicos');
xlabel('Tempo (s)'); ylabel('Amplitude'); legend show; grid on;
xlim([0 total_time]);
```





(c) Espectro de Frequências das Formas de Onda

Implementação de rotina para traçar o gráfico do espectro de frequências de cada sequência periódica, indicando amplitude e componentes de frequência.

Espectro da Onda Quadrada

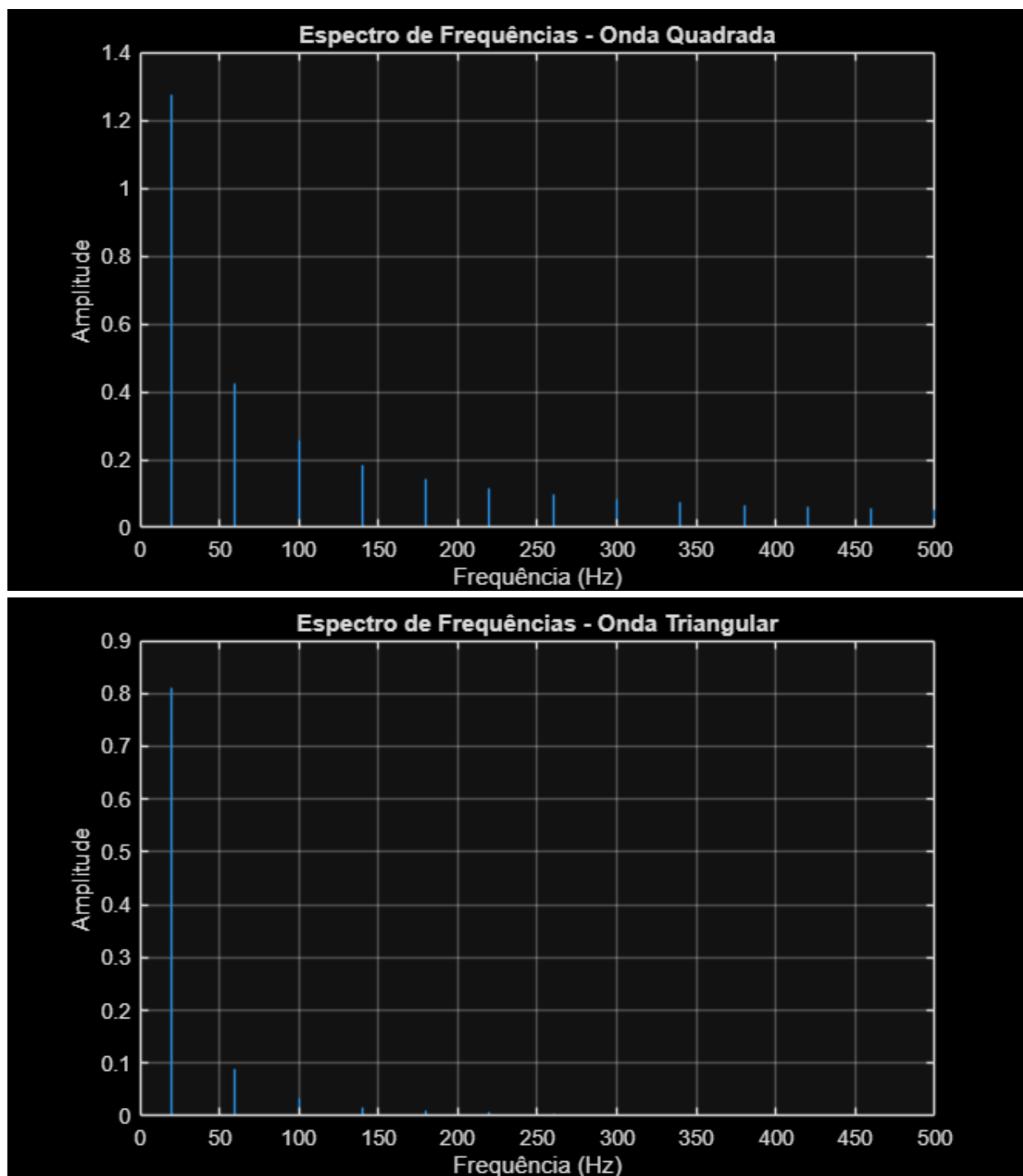
```
plot_spectrum(func_square, Fs, 'Onda Quadrada', 7);
```

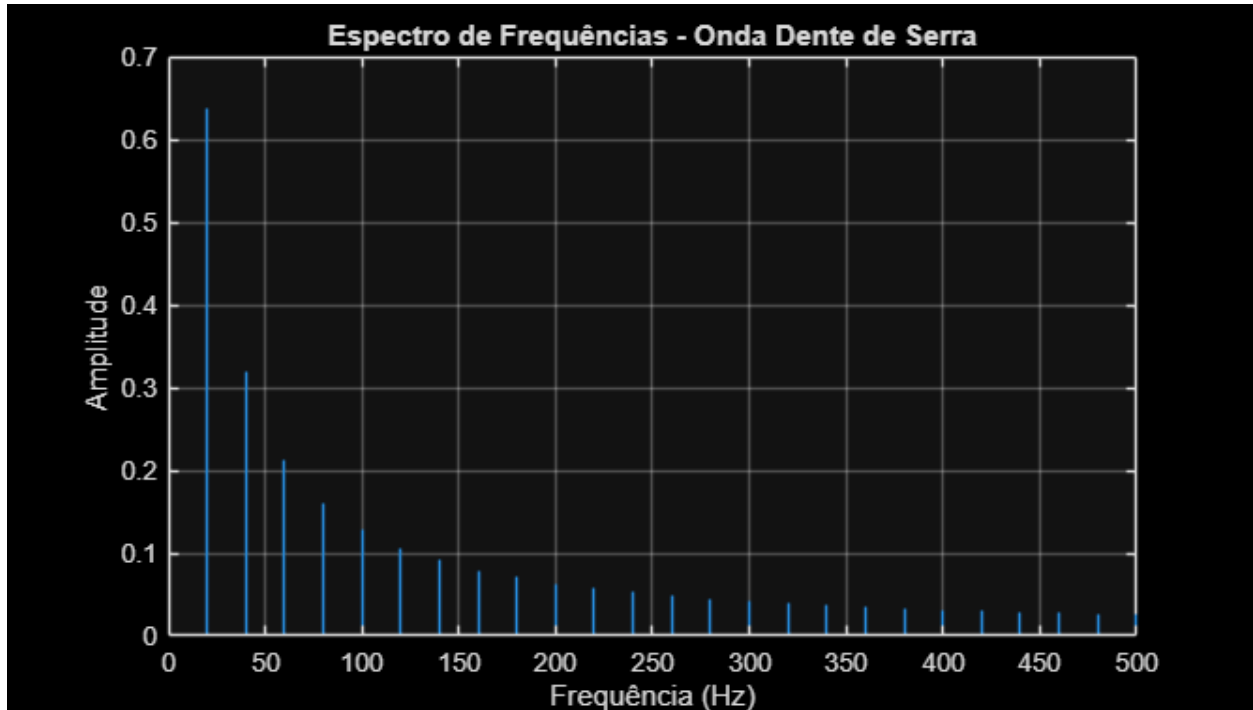
Espectro da Onda Triangular

```
plot_spectrum(func_triangle, Fs, 'Onda Triangular', 8);
```

Espectro da Onda Dente de Serra

```
plot_spectrum(func_sawtooth, Fs, 'Onda Dente de Serra', 9);
```





(d) Sinal Composto Específico

Traçar o gráfico do sinal $x(t) = \cos(2\pi 1000t) + (1/2)\cos(2\pi 3000t) + (1/4)\cos(2\pi 4000t) + \cos(2\pi 5000t)$ e seu espectro.

Parâmetros para o sinal composto

```
F0 = 1000;           % frequência fundamental
T0 = 1/F0;           % período fundamental
total_time_d = num_periods * T0;
t_d = 0:Ts:total_time_d; % vetor de tempo
```

Definição do sinal $x(t)$

```
x = cos(2*pi*1000*t_d) + ...
    0.5*cos(2*pi*3000*t_d) + ...
    0.25*cos(2*pi*4000*t_d) + ...
    cos(2*pi*5000*t_d);
```

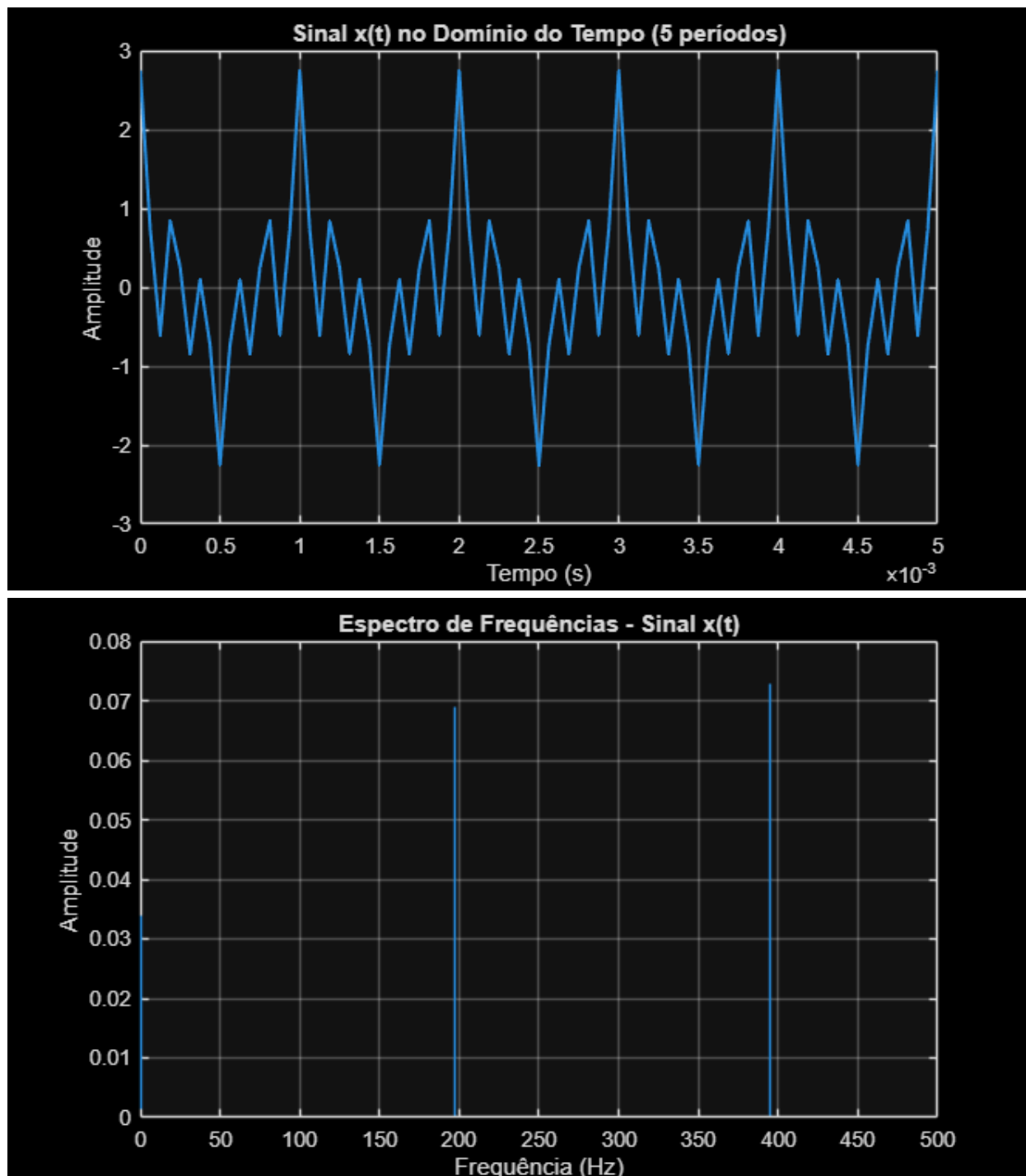
Gráfico temporal do sinal $x(t)$

```
figure(10);
plot(t_d, x, 'LineWidth', 1.5);
title('Sinal x(t) no Domínio do Tempo (5 períodos)');
xlabel('Tempo (s)');
ylabel('Amplitude');
```

```
grid on;  
xlim([0 total_time_d]);
```

Espectro de frequências do sinal $x(t)$

```
plot_spectrum(x, Fs, 'Sinal x(t)', 11);
```



Análise dos Resultados

Os gráficos demonstram:

- **Item (a):** Síntese correta das três formas de onda periódicas usando as fórmulas da Série de Fourier
- **Item (b):** O impacto do número de harmônicos na qualidade da aproximação, onde mais harmônicos resultam em melhor representação da forma de onda
- **Item (c):** Os espectros mostram as componentes harmônicas de cada forma de onda, evidenciando suas características espectrais distintas
- **Item (d):** O sinal composto e seu espectro, mostrando as quatro componentes de frequência especificadas (1kHz, 3kHz, 4kHz e 5kHz) com suas respectivas amplitudes

Published with MATLAB® R2025b