Universidade de Brasília

Departamento de Engenharia Elétrica

Tópicos em Engenharia - Processamento de Sinais Biomédicos

Lista de Exercícios 2

Autores:

Caio Luiz Candeias Flôres 190134283

Felipe Carneiro da Motta 200017616

João Pedro Daher Aranha 190109742

Brasília

01 de Julho de 2022

Conteúdo

1 Exercícios

1.1 Exercício 2.4:

Generate one cycle of the square wave similar to the one shown above in a 500-point MATLAB array. Determine the RMS value of this waveform using Equation 2.13. [Hint: When you take the square of the data array, be sure to use a period before the up arrow so that MATLAB does the squaring point-by-point (i.e., $x.^2$).]

O código no MATLAB:

```
clc; clear all; close all;

N = 500; % número de pontos
TT = 0.2; % tempo total do sinal (onda quadrada)
fT = 1/TT; % frequencia do sinal
Ts = TT/N; % periodo de amostragem
A = 1; % amplitude do sinal
t = (0:N-1)*Ts; % vetor de tempo

y = A*square(2*pi*fT*t); % onda quadrada

% valor RMS
RMS = sqrt(mean(y.^2)); % RMS
sprintf('O valor RMS é: %4f', RMS); % print

% visualizando a forma de onda
hold on;
plot(t, y, 'k', 'DisplayName', 'Onda quadrada'); grid;
xlabel('Tempo (s)');
ylabel('Amplitude (V)');
yline(RMS, 'b--', 'DisplayName', sprintf('Valor RMS: %.1f V', RMS)); % valor RMS no gráfico
ylim([-1.5*A, 1.5*A]);% limites do eixo y
title('Onda quadrada');
legend('Orientation','horizontal','Box','on','Location','southoutside');

saveas(gcf,'semmlow_2_4.png'); % salvar imagem
```

Desta forma, a onda quadrada de um ciclo gerada e o seu valor RMS são explicitadas no gráfico 2:

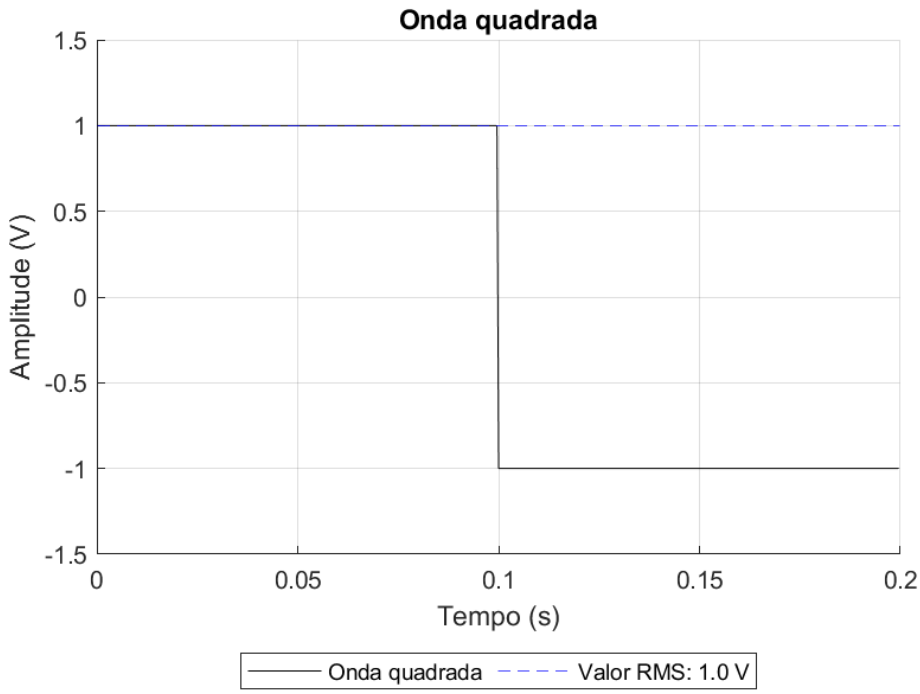

Figura 1: Gráfico do Exercício 2.4 e seu valor RMS

É sabido que o valor RMS de um sinal contínuo e de uma sinal discreto são dados por $x(t)_{RMS} = \sqrt{\frac{1}{T}\int_0^T x(t)^2 \ dt}$ e $\sqrt{\frac{1}{N}\sum_{n=1}^{N} x_n^2}$ respectivamente. Todavia, a nível de código no MATLAB, foi usada a seguinte equação pra obter o valor RMS, $y_{RMS} = \text{sqrt}(\text{mean}(\text{y.}^2))$, em que todos os pontos do vetor sinal são elevados à segunda potência e, por fim, tira-se a raiz quadrada da média desses pontos. Tratando-se de uma aproximação discreta do cálculo do valor RMS da onda quadrada de um ciclo.

1.2 Exercício 2.6:

Generate the waveform shown for Problem 2.5 above in MATLAB. Use 1000 points to produce one period. Take care to determine the appropriate time vector. (Constructing the function in MATLAB is more difficult than the square wave of Problem 2.3, but can still be done in one line of the code.) Calculate the RMS value of this waveform as in Problem 2.3. Plot this function to ensure you have constructed it correctly.

O código no MATLAB:

```
clc; close all; clear all;
N = 1e3; % numero de pontos
TT = 1; % tempo toal
Ts = TT/N; % periodo de amostragem
f = 1/0.5; % frequencia dos sinais
t = (0:N-1)*Ts; % vetor de tempo
A = 1; % amplitude do sinal
s = A*sawtooth(2*pi*f*t); % onda dente de serra
s_rms = sqrt(sum(s.^2)/N); % valor RMS do sinal s

% plot
plot(t, s, 'k', 'DisplayName', 'Sawtooth wave');
hold on;
yline(s_rms,'b--','DisplayName', sprintf('RMS = %.2f V'
    ,s_rms)); % valor RMS
ylabel('Amplitude (V)');
xlabel('Time (sec)');
ylim([-1.1*A,1.1*A]); % limites de visualizacao
legend('Orientation','horizontal','Box','on','Location'
    ,'southoutside');
title('RMS value of a sawtooth wave');
saveas(gcf, 'semmlow_2_6.png')
```

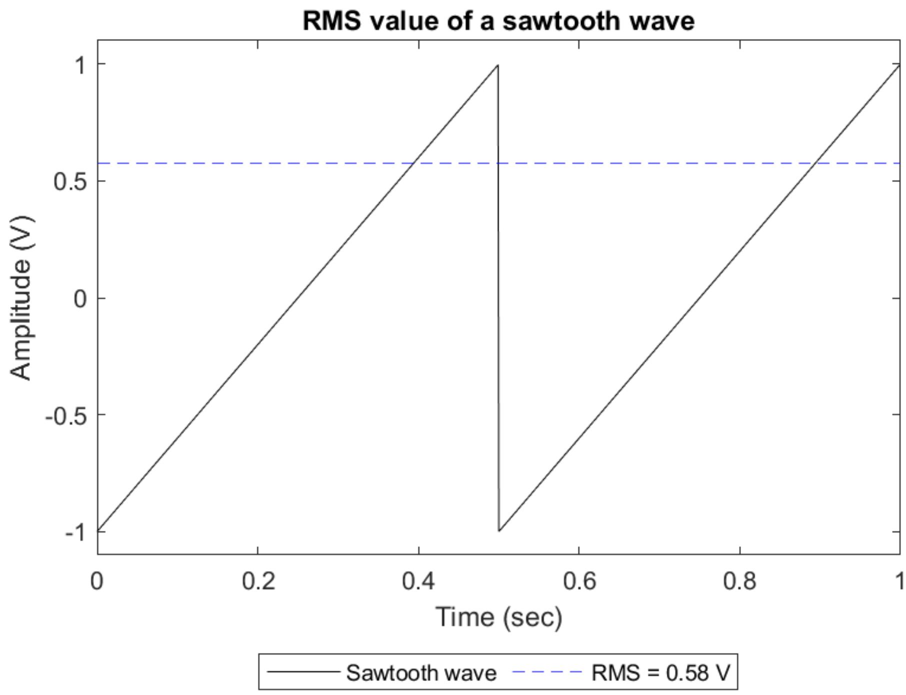

Figura 2: Gráfico do Exercício 2.6

O valor RMS da onda "Sawtooth"obtido por meio do MATLAB foi ≈ 0.58 V. Analiticamente, calcula-se o valor RMS da onda "Sawtooth"da seguinte forma:

$$x_{RMS}(t) = \sqrt{\frac{1}{T}\int_0^{T/2} x(t)^2\ dt} = \sqrt{\frac{1}{0.5}\int_0^{0.5} (2t-1)^2\ dt} = 0.57735 \approx 0.58 \tag{1}$$

Portanto, ao comparar o valor RMS obtido analiticamente com o valor obtido pelo MATLAB, obteve-se a mesma resposta.

1.3 Exercício 2.7:

For the waveform of Problem 2.6, calculate the standard deviation using MATLAB's std and compare it to the RMS value determined using Equation 2.13.

Sabe-se da teoria que, caso a média for zero, os valores do RMS e do desvio padrão serão diferentes apenas pelo fator de normalização.

Equação 2.13

$$x_{RMS} = \left[\frac{1}{N} \sum_{n=1}^{N} x_n^2 \right]^{1/2} \tag{2}$$

A fórmula do desvio padrão, conforme equação 2.17:

Equação 2.17

$$\sigma = \left[\frac{1}{N-1} \sum_{n=1}^{N} (x_n - \bar{x})^2 \right]^{1/2} \tag{3}$$

O código no MATLAB:

```
clear all
close all

periodo=0.5; %periodo de 0.5s
f=1/periodo; %frequencia da funcao
N=1000; %numero de pontos do vetor de tempo
Tt=periodo; %tempo total
Ts=Tt/N; %periodo de amostragem
t1=(0:N-1)*Ts; %vetor de tempo
y=sawtooth(2*pi*f*t1-pi); %funcao do matlab da funcao
    dente de serra

media = mean(y); %media do vetor y

rms = sqrt(mean(y.^2)); %valor rms do vetor y

standard_deviation = std(y); %desvio padrao do vetor y

%print dos valores
out = sprintf(" Média: %.6f\n RMS: %.6f\n Desvio Padrão
    : %.6f ", media, rms, standard_deviation);
disp(out);
```

```
Média: -0.001000
RMS: 0.577351
Desvio Padrão: 0.577639
```

Figura 3: Output do código do exercício 2.7

Os valores de RMS e desvio padrão diferem por duas razões:

Primeiramente, embora a média do sinal contínuo seja 0, a média do sinal amostrado é diferente de zero como pôde ser constatado.

Ainda que a média fosse zero, os valores não seriam iguais devido ao fator de normalização 1/(N-1) do desvio padrão, que difere do fator de normalização 1/N do cálculo do RMS. Essa sendo a segunda razão.

1.4 Exercício 2.8:

Load the signal used in Example 2.2 found in file data-c1.mat. This signal was shown to be nonstationary in that example. Apply MATLAB's detrend operator and evaluate the detrended signal by calculating the means and variances of signal segments as in Example 2.2. Is the modified signal now stationary?

O código no MATLAB:

```
clc; clear all; close all;

% info sobre o arquivo:
% N = 1000
% Ts = 1e-3
% variavel x

load("data_c1.mat"); % carregar arquico 'data_c1.mat'

% sinal nao-estacionario e aplicando o detrend

for j=1:4                        % dividir o sinal em 4
    partes
    idx = 250*(j-1)+1;           % indices dos segmentos
    seg_ns = x(idx:idx+249);     % extrair segmento
    avg_ns(j) = mean(seg_ns);    % media do segmento
    var_ns(j) = var(seg_ns);     % variancia do segmento
```

```
    % aplicando o detrend
    detrend_x = detrend(x);
    seg_s = detrend_x(idx:idx+249);
    avg_s(j) = mean(seg_s);
    var_s(j) = var(seg_s);

end

% print do sinal nao-estacionario
disp("Sinal não-estacionário");
avg_ns_out = sprintf("mean_seg1: %.4f | mean_seg2: %.4f | mean_seg3: %.4f | mean_seg4: %.4f", avg_ns(1), avg_ns(2), avg_ns(3), avg_ns(4));
var_ns_out = sprintf("var_seg1: %.4f  | var_seg2: %.4f | var_seg3: %.4f  | var_seg4: %.4f\n", var_ns(1), var_ns(2), var_ns(3), var_ns(4));
disp(avg_ns_out);
disp(var_ns_out);

% print do sinal com o detrend aplicado
disp("Sinal modificado");
avg_s_out = sprintf("mean_seg1: %.4f | mean_seg2: %.4f | mean_seg3: %.4f | mean_seg4: %.4f", avg_s(1), avg_s(2), avg_s(3), avg_s(4));
var_s_out = sprintf("var_seg1: %.4f  | var_seg2: %.4f | var_seg3: %.4f  | var_seg4: %.4f", var_s(1), var_s(2), var_s(3), var_s(4));
disp(avg_s_out);
disp(var_s_out);
```

Após aplicar o detrend ao sinal não-estacionário do arquivo **data-c1.mat**, o novo sinal gerado teve, de um modo geral, melhoria nos valores de média e variância, como pode ser visto na Figura 4.

```
Sinal não-estacionário
mean_seg1: 0.0215 | mean_seg2: 0.3500 | mean_seg3: 0.5803 | mean_seg4: 0.9321
var_seg1: 0.0804  | var_seg2: 0.0696  | var_seg3: 0.0937  | var_seg4: 0.1030

Sinal modificado
mean_seg1: -0.0282 | mean_seg2: 0.0194 | mean_seg3: -0.0311 | mean_seg4: 0.0398
var_seg1: 0.1019  | var_seg2: 0.0789  | var_seg3: 0.0800  | var_seg4: 0.1028
```

Figura 4: Métricas do Exercício 2.8

Por definição, um sinal estacionário possui média e desvio padrão não-variantes no tempo, sendo considerado, a nível de código, valores aproximadamente invariantes, visto que o sinal é amostrado (apenas partes do sinal contínuo são coletas). Entretanto, a Figura 4 demonstra valores de variância em torno de ≈ 0.1, o que não é aproximadamente 0. Deste modo, o sinal modificado não é estacionário.

1.5 Exercício 2.9:

If a signal is measured as 2.5 V and the noise is 28 mV (28×10^{-3} V), what is the SNR in dB?

O SNR pode ser calculado usando a seguinte equação:

$$SNR = 20 \cdot log(\frac{Signal}{Noise})\ dB \tag{4}$$

Desse modo, considerando o valor do sinal como 2.5 V e do ruído 28 mV, o valor SNR será:

$$SNR = 20 \cdot log(\frac{2.5}{0.028}) \approx 39.01\ dB \tag{5}$$

1.6 Exercício 2.12:

An 8-bit ADC converter that has an input range of ± 5 V is used to convert a signal that ranges between ± 2 V. What is the SNR of the input if the input noise equals the quantization noise of the converter? [Hint: Refer back to Equation 1.8 to find the quantization noise.]

Equação 1.8 resumida

$$ruido\ de\ quantizacao = \sigma^2 = \frac{q^2}{12} \tag{6}$$

Equação 1.6

$$q = passo\ de\ quantizacao = \frac{V_{MAX}}{2^b - 1} \tag{7}$$

Essa fórmula assume que o menor nível é zero. Como os valores do conversor variam de -5V a 5V, a extensão dessa faixa é de 5-(-5)=10.

$$q = \frac{10}{2^8 - 1} = 0.03921568627 \tag{8}$$

$$ruido = \frac{q^2}{12} = 0.00012815583 \tag{9}$$

O SNR será:

$$SNR = 20log\left(\frac{signal}{noise}\right) = 20log\left(\frac{4}{0.00012815583}\right) \approx 89.89\ dB \tag{10}$$

1.7 Exercício 2.14:

Generate a 10,000-point data set, where each value is the average of four random numbers produced by rand, the uniform random number generator. Plot the his- togram of this data set as a bar plot. Note how this distribution function closely resembles a Gaussian distribution despite the fact that the original noise source had uniform distribution.

O código MATLAB:

```
clc; clear all; close all;

for idx=1:1e4
    rand_4num = rand(4,1);              % gerando 4 nums
    rand_array(idx) = mean(rand_4num);  % media dos 4 nums
end

histogram(rand_array); % histograma
xlabel("Valor gerado por rand()");
ylabel("Ocorrência dos valores");
```

```
title("Histograma de 10000 valores de rand()");

saveas(gcf, 'semmlow_2_14.png'); % salvar imagem
```

Evidentemente, a distribuição da função **rand()** gera valores aleatórios que seguem uma distribuição uniforme (todos possuem a mesma probabilidade de ocorrerem). Como pode ser visto na Figura 5, a ocorrência em que os valores gerados aparecem segue uma Distribuição Gaussiana, lembrando também uma PDF (Probability Density Function) Normal. Isso acontece exatamente pelo fato de que a função função rand() gera valores que segue uma Distribuição Uniforme.

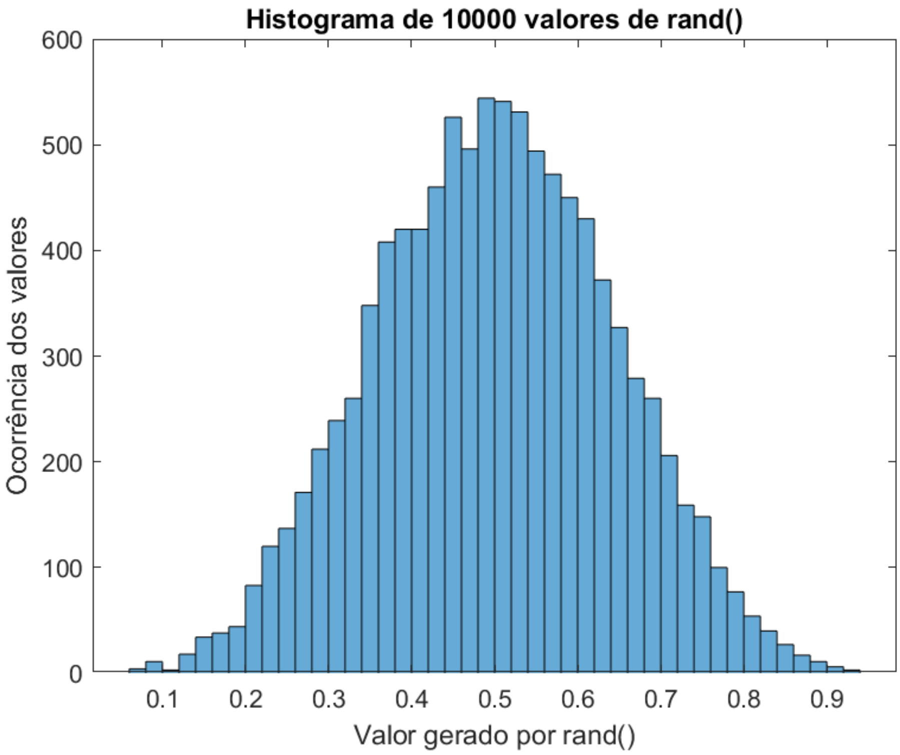

Figura 5: Histograma do Exercício 2.14

1.8 Exercício 2.20:

Modify Example 2.6 to test if a sine wave and cosine wave, both having a frequency, f = 1.5 Hz, are orthogonal. Make TT = 3 s and Ts = 0.01 s.

```
Ts = 0.01;            % Intervalo amostral
Tt = 3;               % Tempo total
t = 0:Ts:Tt;          % Vetor tempo
f = 1.5;              % Frequencia
x = sin(2*pi*f*t);    % Onda senoidal
y = cos(2*pi*f*t);    % Onda cosenoidal
Corr = sum(x.*y);     % Eq. 2.29
disp(Corr)
```

O valor mostrado foi 1.7994e-14, um valor muito próximo de zero, o que indica ortogonalidade, como esperado da relação de seno e cosseno de mesma frequência e fase.

1.9 Exercício 2.24:

Repeat Problem 2.22 but solve for the Pearson correlation coefficient to determine if the two variables are orthogonal.

```
clc; clear all; close all;

load ("correl1.mat"); %carregar arquivo "correl1.mat"

N=length(x); %comprimento de um dos vetores

rxy_pearson = 1/((N-1)*sqrt(var(x)*var(y)))*sum((x-mean(x)).*(y-mean(y)));
%fórmula do coeficiente de pearson, conforme eq. 2.43


plot(x,"DisplayName","Signal x"); %plotagem sinal x

hold on; %mantém mesmo gráfico

plot(y,"DisplayName","Signal y"); %plotagem sinal y

legend('Orientation','horizontal','Box','on','Location','southoutside'); %legenda
title({"Pearson correlation:";rxy_pearson}); % rxy_pearson exibido no titulo
```

```
xlabel("Position in the array"); %label do eixo x
ylabel("Amplitude"); %label do eixo y

saveas( gcf , "semmlow_2_24.png"); %salvar imagem
```

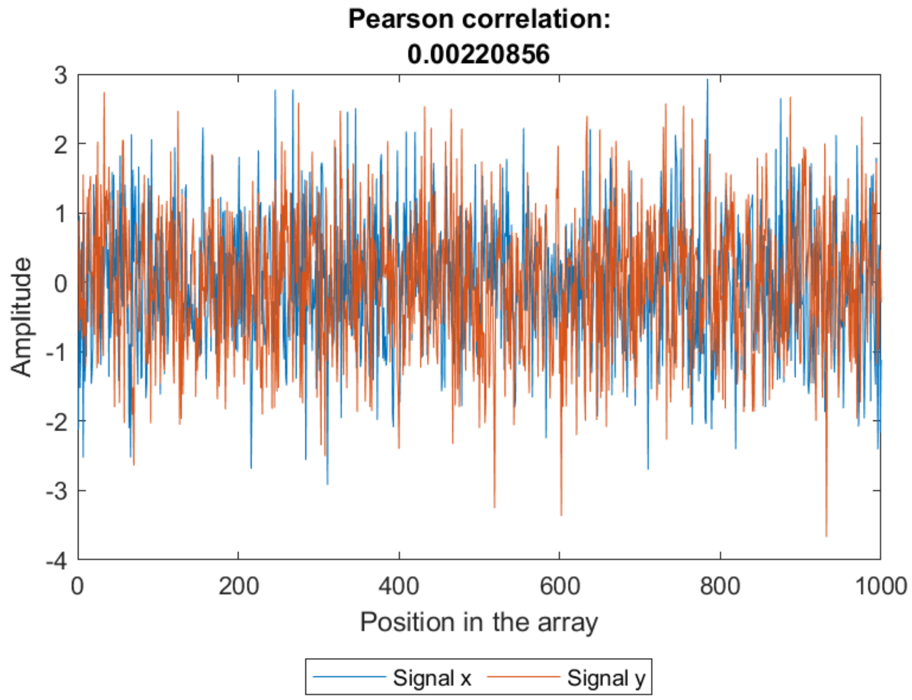

Figura 6: Gráfico do Exercício 2.24 e o valor $r_{xy} = 0.0022$

O valor do coeficiente resultou em um número próximo de zero. Essa métrica é normalizada, sendo sempre um valor entre -1 e +1, sendo -1 (correlação máxima negativa), 0 (nenhuma correlação) e 1 (correlação máxima positiva). O que é um forte indício de que os sinais são ortogonais.

1.10 Exercício 2.25:

Use Equation 2.30 to find the normalized correlation between a cosine and a square wave as shown below. This is the same as Example 2.6 except that the sine has been replaced by a cosine and the mean is taken.

O código MATLAB:

```
clc; clear all; close all;

% Cos e Onda quadrada com a mesma f

N = 500;
Ts = 2e-3;
A = 1;
t = (0:N-1)*Ts;
f1 = 1;
s1 = A*cos(2*pi*f1*t);      % cossenoide
s2 = A*square(2*pi*f1*t); % onda quadrada

rxy = mean(s1.*s2); % correlacao normalizada (= sum(s1.*s2)/N )

% plotando os sinais s1 e s2
plot(t, s1, "DisplayName", "Cosseno");
hold on;
plot(t, s2, "DisplayName", "Onda Quadrada");
ylim([-1.1*A, 1.1*A]);
xlabel("Time (s)");
ylabel("Amplitude (V)");
legend('Orientation','horizontal','Box','on','Location','southoutside');
title({"Normalized correlation (rxy):"; rxy}); % rxy plotado no titulo

saveas(gcf,'semmlow_2_25.png'); % salvar imagem
```

Como pode ser visto pela Figura 7, o valor de correlação normalizado é $r_{xy} = 0.004$. Sendo $r_{xy} \approx 0$, é possível dizer que os sinais $s1$ e $s2$ são ortogonais, ou seja, não possuem correlação entre si. Visualmente, é possível perceber que o pico da onda quadrada coincide com o vale da cossenoide, justificando determinada conclusão de ortogonalidade.

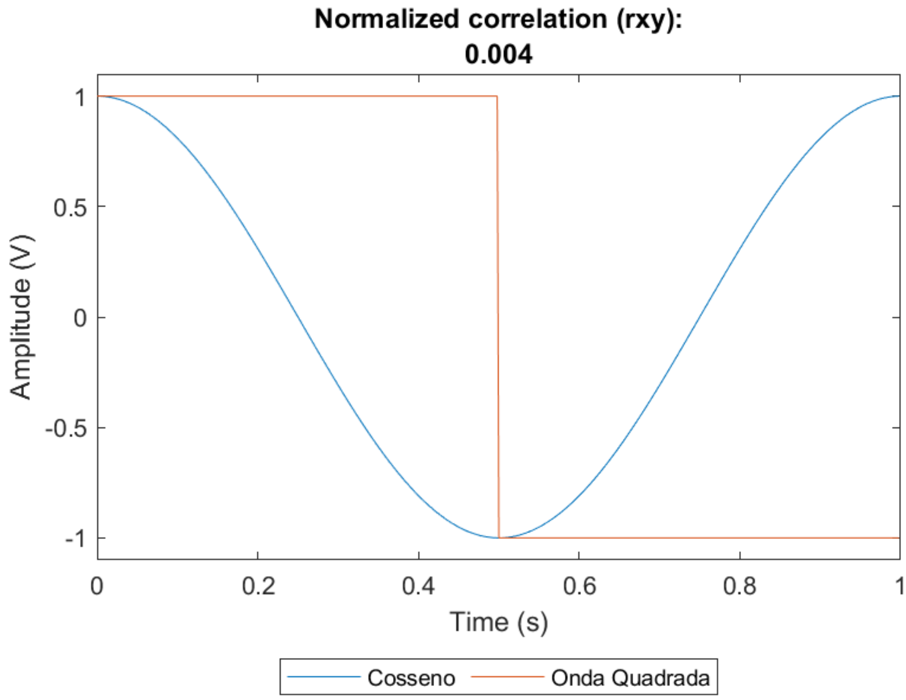

Figura 7: Gráfico do Exercício 2.25 e o valor $r_{xy} = 0.004$