

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



TEORIA DO TRATAMENTO DE SINAL

M.EMG

Trabalho Final - Versão Provisória

Estudantes:

João SÁ PEREIRA

Docentes:

Prof. Jorge CARVALHO

Prof. Teresa LAJINHA

13 de Abril de 2024

Conteúdo

1	Fourier Theory Made Easy (?)	1
1.1	Uma onda sinusoidal	1
1.2	Transformadas de Fourier Famosas	4
1.2.1	Função Seno → Função Delta	4
1.2.2	Função Gaussiana	5
1.2.3	Função Seno Cardinal → <i>Square Wave</i>	6
1.2.4	Função Exponencial → Lorentziana	8
1.3	FFT ou FID	8

Lista de Figuras

1.1	Onda sinusoidal.	1
1.2	Taxa de amostragem.	2
1.3	Sinal subamostrado.	3
1.4	Função Seno.	5
1.5	Função Delta.	5
1.6	Função Gaussiana.	6
1.7	Função Gaussiana - TF.	6
1.8	Função Sinc.	7
1.9	Espectro de Frequências.	7
1.10	Square Wave.	7
1.11	Função Exponencial.	8
1.12	Lorentziana.	8

Código

1.1	Slide 2.	1
1.2	Slide 3.	2
1.3	Slide 4.	3
1.4	Slide 12.	4
1.5	Slide 13.	5
1.6	Slides 14 e 15.	6
1.7	Slide 16.	8

1. Fourier Theory Made Easy (?)

1.1. Uma onda sinusoidal

Começaremos por reproduzir uma onda sinusoidal. A sua forma mais básica, como função do tempo, é dada pela seguinte expressão:

$$y(t) = A \cdot \sin(2\pi ft + \phi) \quad (1.1)$$

Para o primeiro exemplo, iremos representar em MATLAB uma onda sinusoidal com amplitude 5 e frequência de 4 Hz. Para isso, é necessário definir o eixo do tempo. Neste caso, o eixo do tempo começa em zero e acaba em 1 (segundos), com incrementos de 0.01. O código que fornece a imagem é o seguinte:

Código 1.1: Slide 2.

```
1 dt = 0.01;  
2 t = 0:dt:1;  
3 F = 4;  
4 A = 5;  
5 y = A * sin(2 * pi * F * t);  
6 plot(t,y);  
7 xlabel('Segundos');
```

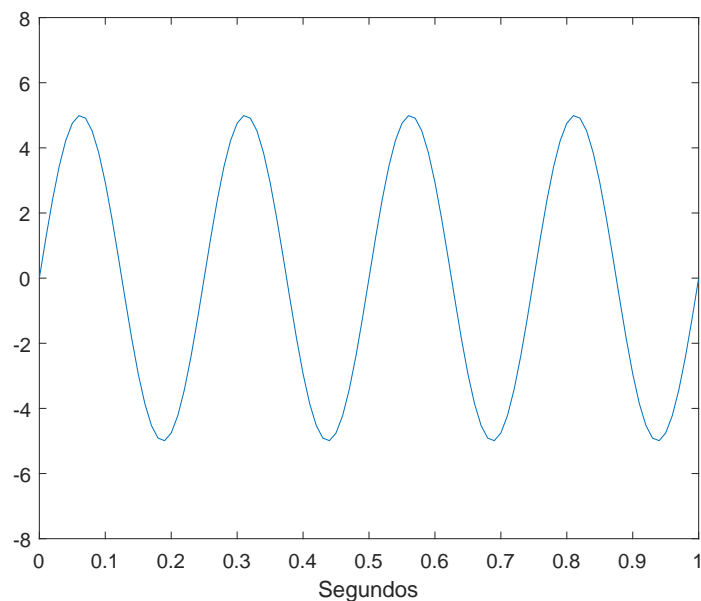


Figura 1.1: Onda sinusoidal.

Se definirmos a taxa de amostragem (*sampling rate*) como sendo igual a 256 e a duração da amostragem (*sampling duration*) como sendo igual a 1, podemos definir a variável *dt* como sendo o quociente entre a *sampling duration* e o *sampling rate*. E se representarmos a figura com pontos, podemos verificar que por cada segundo, temos 256 pontos.

Código 1.2: Slide 3.

```
1 sampling_rate = 256;  
2 sampling_duration = 1;  
3 dt = sampling_duration/sampling_rate;  
4 t = 0:dt:1;  
5 F = 4;  
6 A = 5;  
7 y = A * sin(2 * pi * F * t);  
8 plot(t, y, 'o', 'MarkerSize',4);  
9 xlabel('Segundos')
```

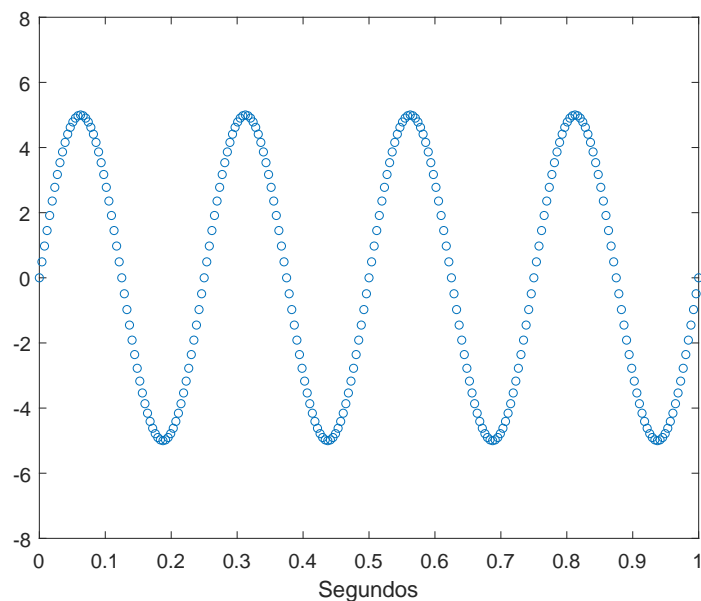
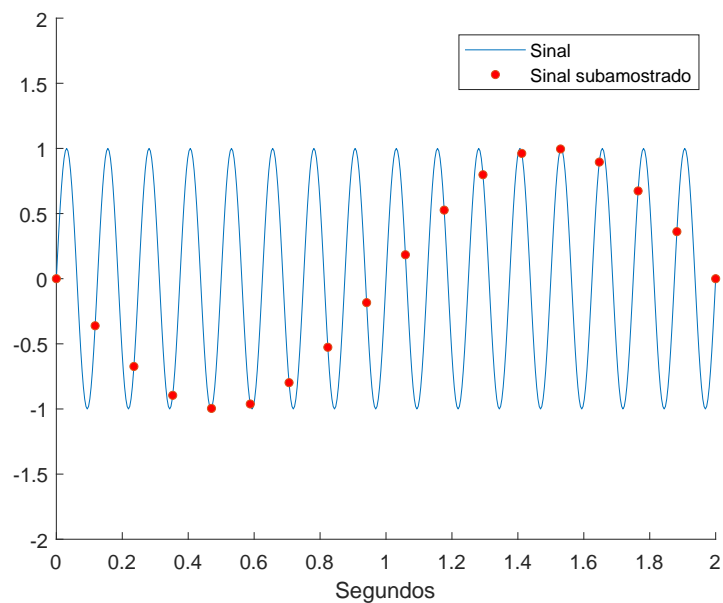


Figura 1.2: Taxa de amostragem.

Agora uma função sinusoidal com frequência de 8 Hz e com uma determinada taxa de amostragem. Quando se altera o valor do passo de amostragem, para 8.5 Hz, obtêm-se um sinal com frequência de 0.5 Hz. Este exercício, recriado em MATLAB no código da página seguinte, mostra a importância da escolha do passo de amostragem.

Código 1.3: Slide 4.

```
1 sampling_rate = 8.5;
2 sampling_duration = 1;
3 dt1 = 1/256;
4 dt2 = sampling_duration/sampling_rate;
5 % Vetor tempo e função seno 1
6 t1 = 0:dt1:2;
7 y1 = sin(2 * pi * 8 * t1);
8 % Vetor tempo e função seno 2
9 t2 = 0:dt2:2;
10 y2 = sin(2 * pi * 8 * t2);
11 % Plots
12 hold on
13 plot(t1,y1)
14 plot(t2,y2, 'o', 'MarkerFaceColor', 'red', 'MarkerSize', 4)
15 legend('Sinal', 'Sinal subamostrado')
16 xlabel('Segundos')
17 hold off
18 axis([0 2 -2 2])
```

**Figura 1.3:** Sinal subamostrado.

1.2. Transformadas de Fourier Famosas

Uma transformada pega numa função (ou sinal) e transforma-a numa outra função (ou sinal). A transformada discreta de Fourier é dada pela seguinte expressão:

$$H(f) = \int_{-\infty}^{+\infty} h(t) \cdot e^{2\pi i f t} \cdot dt \quad (1.2)$$

A transformada rápida de Fourier (do inglês: *Fast Fourier Transform*, abreviado FFT), é um algoritmo que calcula a transformada discreta de Fourier. A análise de Fourier converte um sinal do domínio original (tempo) para uma representação no domínio das frequências e vice-versa.

1.2.1. Função Seno → Função Delta

A transformada de Fourier de uma função sinusoidal é uma função delta de Dirac, pelo menos em parte. Quando aplicamos a transformada de Fourier no MATLAB (função *fft*) a uma função sinusoidal, o resultado é uma combinação de deltas de Dirac localizados nas frequências da função sinusoidal. A transformada de Fourier da função é zero em todas as frequências exceto na frequência da função seno, onde há um Dirac localizado.

Código 1.4: Slide 12.

```
1 % Criar os incrementos dt
2 sampling_duration = 1;
3 sampling_rate = 250;
4 dt = sampling_duration/sampling_rate;
5
6 % Vetor tempo e função Seno
7 t = 0:dt:2;
8 A = 1;
9 F = 8;
10 y = sin(2 * pi * F * t);
11 figure();
12 plot(t,y);
13 title('Função Seno');
14 axis([0 2 -2 2]);
15
16 % Função Delta
17 ft_y = fft(y);
18 df = (1/dt)/length(ft_y);
19 freq = (0:length(ft_y)-1)*df;
20 figure();
21 plot(freq, abs(ft_y));
22 title('Função Delta');
23 axis([0 125 0 300]);
```

É possível verificar pelos gráficos produzidos pelo código anterior, que o Dirac ocorre na frequência $7.98403 \approx 8$, que é a frequência da função sinusoidal.

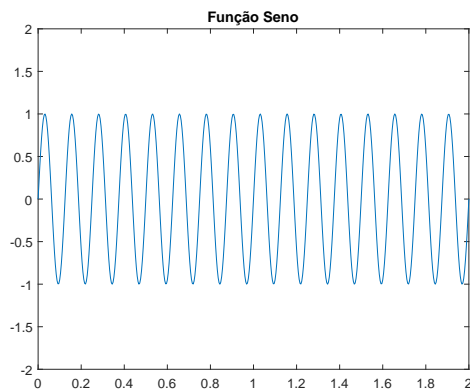


Figura 1.4: Função Seno.

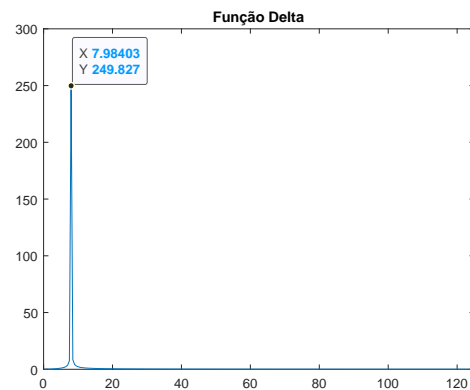


Figura 1.5: Função Delta.

1.2.2. Função Gaussiana

A transformada de Fourier de uma função gaussiana é também uma função gaussiana mas com parâmetros diferentes.

Código 1.5: Slide 13.

```

1 % Parâmetros e função Gaussiana
2 media = 25;
3 std = 1;
4 A = 0.4;
5 dt=0.1;
6 t=0:dt:50;
7 s = A * exp(-0.5 * ((t - media)/std).^2);
8 figure();
9 plot(t,s);
10 title('Função Gaussiana');
11 axis([0 50 0 0.5]);
12
13 % Transformada de Fourier
14 S = fftshift(abs(0.5*fft(s))); % Multiplicação por 0.5 para igualar ao
    apresentado no ppt
15 f = 0:0.5:250;
16 figure();
17 plot(f,S, Color='#A2142F');
18 title('Função Gaussiana - TF')

```

Neste caso, a amplitude da transformada de Fourier é 5 e a média é 125. As figuras da página seguinte mostram a resposta do código apresentado.

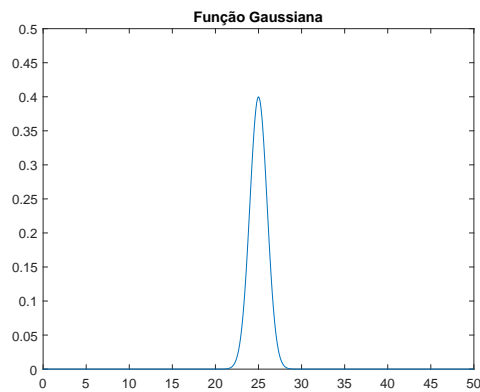


Figura 1.6: Função Gaussiana.

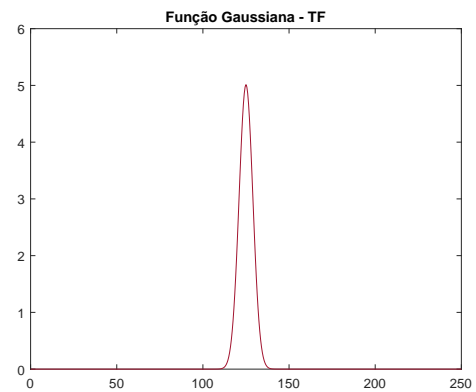


Figura 1.7: Função Gaussiana - TF.

1.2.3. Função Seno Cardinal → Square Wave

Código 1.6: Slides 14 e 15.

```

1 % Parâmetros do sinal Sinc
2 F = 1000;
3 T = 1;
4 f0 = 50;
5 t = -T:1/F:T; % Vetor de tempo
6 s = sinc(f0 * t); % Sinal Sinc
7
8 S = abs(0.25*fftshift(fft(s))); % Calcula a TF do sinal
9 N = length(s);
10 f = (-N/2 : N/2-1) * (F/N); % Vetor de frequências correspondentes
11
12 w = hann(length(s))'; % Função de Hann
13 sw = s .* w; % Sinal Sinc 'filtro'
14 Sw = 0.25*fftshift(fft(sw));
15 f = (-N/2 : N/2-1) * (F/N); % Frequências correspondentes
16
17 figure();
18 plot(t, s);
19 title('Sinal Sinc filtrado');
20 ylim([-0.5 1.5])
21 figure();
22 plot(f, abs(Sw));
23 title('Espectro de Frequência (Sinal filtrado)');
24 xlim([-125 125]);
25 figure();
26 plot(f, abs(S));
27 title('Square wave');
28 xlim([-125 125]);

```

A transformada de Fourier da função seno cardinal (sinc) resulta numa *square wave* (pedestal).

De forma intuitiva, a função seno cardinal tem oscilações até ao infinito e quando a sua transformada de Fourier é calculada, envolve a soma de componentes sinusoidais infinitas, o que resulta numa onda quadrada.

Essencialmente, a transformada de Fourier da função seno cardinal capta a propriedade fundamental desta função, nomeadamente o seu comportamento oscilatório, que no eixo das frequências, resulta numa *square wave*.

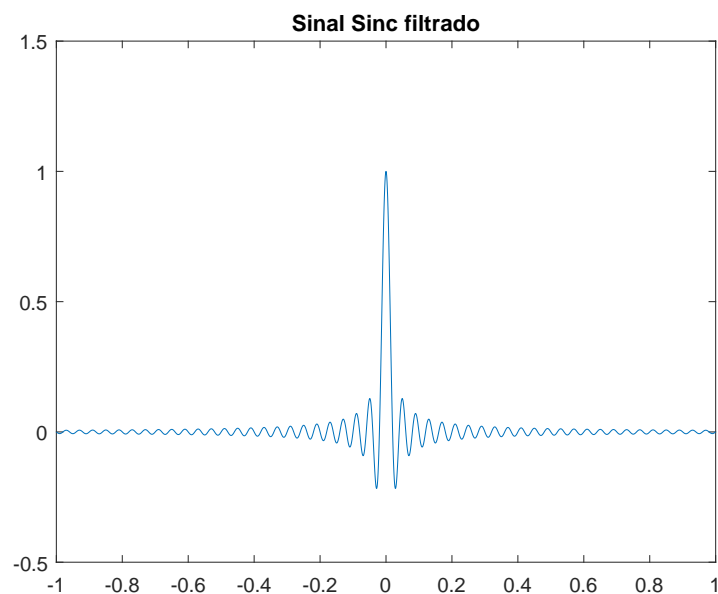


Figura 1.8: Função Sinc.

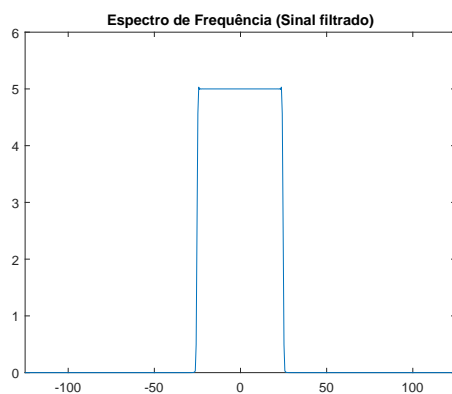


Figura 1.9: Espectro de Frequências.

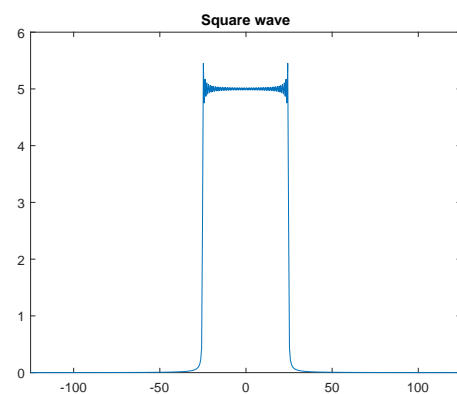


Figura 1.10: Square Wave.

1.2.4. Função Exponencial → Lorentziana

Código 1.7: Slide 16.

```
1 dt = 0.1;
2 t = 0:dt:10;
3 % Função Exponencial
4 s = exp(-t);
5 S = fft(s);
6 Sr = real(S);
7 Si = imag(S);
8 Sabs = abs(S);
9
10 figure();
11 plot(t,s); title('Sinal Exponencial');
12 figure();
13 plot(fftshift(Sabs)); title('Transformada de Fourier Lorentzian')
14 xlim([0, length(Sabs)]);
```

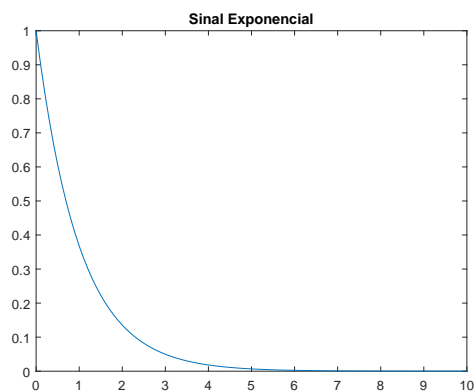


Figura 1.11: Função Exponencial.

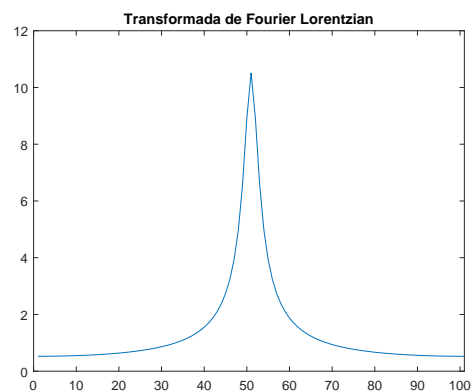


Figura 1.12: Lorentziana.

1.3. FFT ou FID