ENG1400 - PUC-Rio, 2020.1

Trabalho 2: ENG1400

2/06/2020

Lecturer: Marco Antônio Grivet *Scribe: Rafael Vilela*

1 Introdução

De acordo com a função no tempo fornecida pelo enunciado:

$$f(t) = cos(20 * \pi * t) + 2 * cos(30 * \pi * t) - 3 * cos(40 * \pi * t) + cos(50 * \pi * t)$$

calculou-se a energia no domínio do tempo e frequência além dos gráficos do sinal em ambos domínios. Pela propriedade da linearidade, considerando a forma exponencial complexa do cosseno:

$$cos(w_0 * t) = (1/2) * [e^{i*w_0*t} + e^{-i*w_0*t}]$$

Transformada de Fourier:

$$\int_{-\infty}^{\infty} f(t) * e^{-i*w*t} dt$$

Aplicando $f(t)$ na transformada:

$$\pi * \delta(w - w_0) + \pi * \delta(w + w_0)$$

Logo (em função de ω):

$$X(w) = \pi * (\delta(w - 20 * \pi) + \delta(w + 20 * \pi)) + 2 * \pi * (\delta(w - 30 * \pi) + \delta(w + 30 * \pi))$$

$$-3 * \pi * (\delta(w - 40 * \pi) + \delta(w + 40 * \pi)) + \pi * (\delta(w - 50 * \pi) + \delta(w + 50 * \pi))$$

δ é a função impulso.

O software utilizado foi o MATLAB versão R2019b e o código está na seção seguinte. A forma que os gráficos foram plotados não funciona em algumas versões mais antigas do MATLAB.

O gráfico em função de frequência é o gráfico da Transformada de Fourier do sinal, no caso ela é calculada através do algoritmo FFT (Fast Fourier Transform), que discretiza a função, para que seja possível calcular através de operações aritméticas. A sua complexidade computacional é de $O(n * log(n))$, mais eficiente que n^2 da DFT (algoritmo).

A ideia principal do algoritmo é dividir as n amostras em n/2 amostras, calculando suas transformadas e recombinando. Essa lógica é aplicada recursivamente até que seja indivisível (resultado não inteiro), depois somando tudo e achando um valor global da Transformada de Fourier Discreta.

2 Código

A seguir será apresentado o código comentado (dividido em 2 imagens) e os respectivos gráficos do sinal no tempo e na frequência e a fase.

Em relação ao código, T representa o domínio, F a frequência de amostragem (que indicará o dt, os intervalos de tempo) e t o vetor tempo. No código n é o tamanho do vetor "sinal", como uma potência de 2 (no caso n=128 ou 2^7); produto do domínio pela frequência de amostragem.

Há várias forma de representar vetores no MATLAB, porém foi preferível a "colon" porque é a mais utilizada com mais exemplos de resoluções de exercícios. O tamanho do vetor não é fixado como é usando "linspace".

A função "vpa(x)" (variable-precision floating-point arithmetic) calcula cada elemento de x até pelo menos "d" algarismos significativos, o padrão é 32. Foi utlizada para limitar os algarismos significativos do resultado de cada integração.

Valor da energia no domínio da frequência (aprox): 3.7416J

Valor da energia no domínio do tempo: 3.75J

Um erro absoluto em relação à energia no domînio no tempo menor que 10^-2, erro relativo por volta de $0,026\%$.

A função rectangularPulse(0,0.5,t) não altera o resultado do cálculo em "etot" e "etotf", é um artifício para calcular as integrais. Como está definida no mesmo domínio de f(t), ela multiplica por 1 no domínio e zera para todo valor fora de [0,0.5]. Calculando a integral de 0 a 5 é possível achar 3.75J. Já para a frequência há um efeito de espalhamento (spectral leakage), sendo necessário calcular a parte negativa também do domínio (similar ao gráfico de fft). A causa desse espalhamento é a convolução circular entre o sinal real, que será x[n], e o pulso retangular.

```
F=256;   %freq amostragem
dt=1/F;  %delta t = 5*10^-3
T=0.5;    %tempo / dominio sinal

t = 0:dt:T-dt;

sinal=cos(20*pi*t)+ 2*cos(30*pi*t)- 3*cos(40*pi*t)+ cos(50*pi*t);

n=length(sinal);
y=fft(sinal)/n;
%N=TF= 128

fshift=(-n/2:n/2-1)*(F/n);

yshift= fftshift(abs(y));

yshiftFase= fftshift(angle(y));

%plots
tiledlayout(3,1)

nexttile
plot(t,sinal)
title('Sinal no tempo')
xlabel('s')

```

(1)

```
nexttile
plot(fshift,10*log10(yshift))
title('Sinal na freq')
xlabel('Hz')
ylabel('dB')

nexttile
plot(fshift,yshiftFase)
title('Fase')
xlabel('Hz')
ylabel('Rad')

syms t w

f=rectangularPulse(0,0.5,t)*(cos(20*pi*t)+ 2*cos(30*pi*t)- 3*cos(40*pi*t)+ cos(50*

energia= int(abs(f).^2,t,[0 0.5]);
etot=vpa(energia);

W=fourier(f,t,w);
energiaf=int(abs(W).^2,w,[-1000 1000]);
etotf= (1/(2*pi))*vpa(energiaf);

%%%%%%%%%%%%%%%
```

(2)

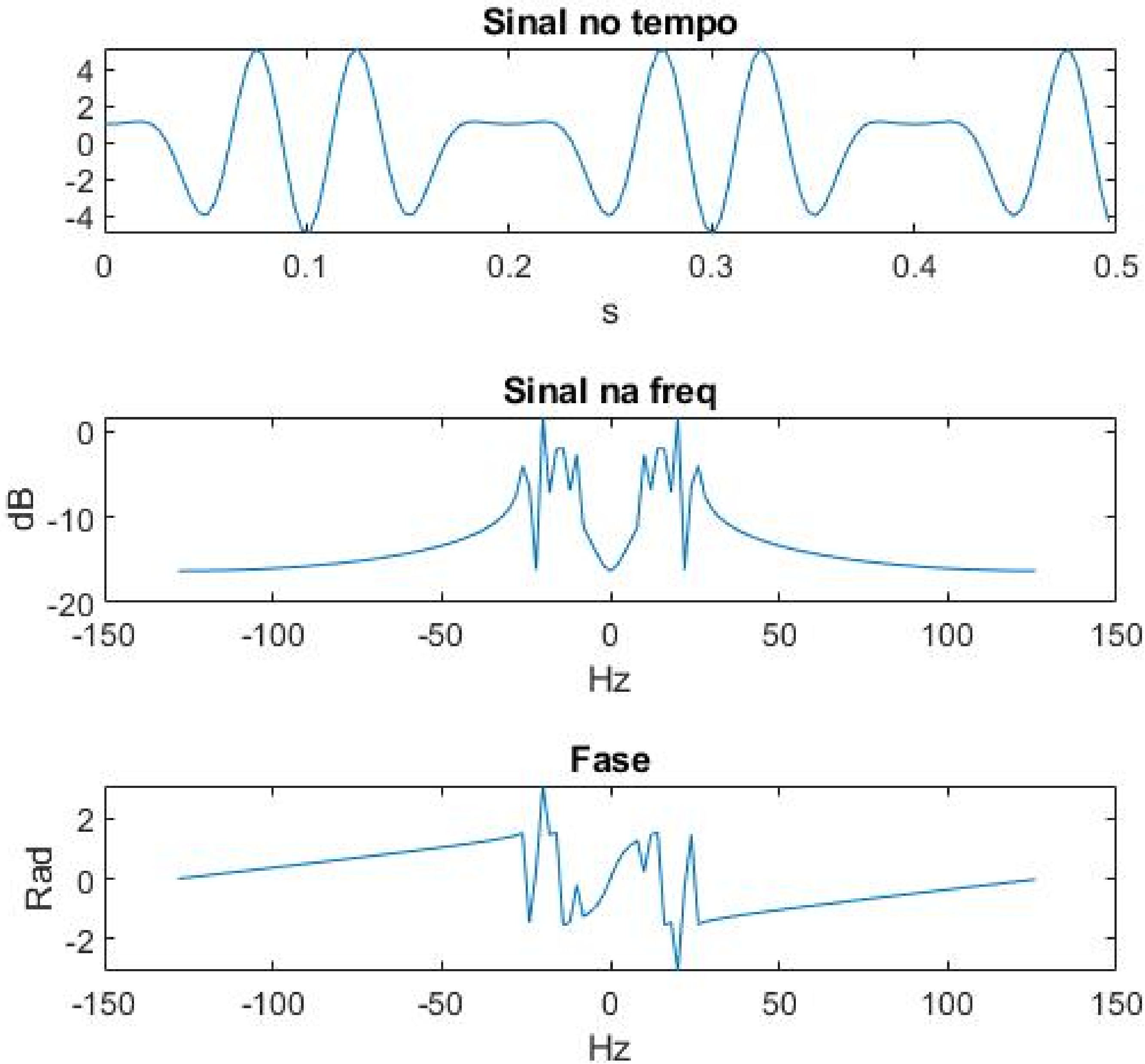
2.1 Gráficos

O primeiro gráfico do sinal no tempo está representado no domínio proposto (em segundos), mostrando uma função periódica. A Transformada de Fourier serve também para funções não periódicas. Diferentemente dos 2 últimos gráficos, não está espelhado no eixo "y" devido às funções "com shift" (como fftshift) no MATLAB, usualmente utilizadas em gráficos de sinais e sistemas para uma melhor visualização do sinal.

O gráfico de fase do sinal representa o ângulo em radianos com valores entre $[-\pi : \pi]$, no mesmo domínio (frequência) que a Transformada de Fourier, 128 para esse exemplo, entretanto o plot do MATLAB está representando entre -150 e 150 (Hz).

Era esperado que o gráfico da Transformada de Fourier possuísse 4 picos porque $f(x)$ é a soma de 4 funções cossenos, no caso o maior em amplitude na frequência de 20Hz e os outros 3 picos entre 0 e 40Hz, conforme a frequência de amostragem aumenta, mais fiel o gráfico fica à representação do sinal na frequência. O eixo vertical foi representado em dB.

O código foi testado com outra função (dentre as 19 do Trabalho II) também e a representação não ficou tão fiel porque a frequência de amostragem era pequena. Portanto, dependendo da função, é necessário ampliar a frequência de amostragem.



(3)

3 Resultados

Pela teoria a energia no tempo é igual a energia na frequência normalizada por $2 * \Pi$. Isso se deve às Relações (Teoremas) de Parseval, essa equação também é conhecida como Teorema de Energia de Rayleigh:

$$\int_{-\infty}^{\infty} \mid x(t) \mid^2 dt = \frac{1}{2 * \pi} * \int_{-\infty}^{\infty} \mid X(j * w) \mid^2 dw$$

Conforme dito anteriormente, o domínio variando entre [-1000:1000] não resulta em exatamente o valor da energia no tempo, mas é uma boa aproximação, com erro pequeno. O domínio da integral deveria conter todas as frequências do sinal original para que ambas fossem iguais, satisfazendo o teorema acima.

A freqûencia de amostragem do sinal é muito importante por causa do funcionamento discreto dos computadores (por consequência dos softwares). Caso o valor fosse muito pequeno o sinal ficaria deformado. Com esse sinal de soma de cossenos essa frequência funcionou, mas poderia não funcionar corretamente em outros sinais. Não menos importante, "n" ser uma potência de 2 porque o algoritmo foi desenvolvido para trabalhar dessa forma.

O MATLAB é uma ferramenta muito útil para códigos a fim de trabalhar com as propriedades de sinais e sistemas, inclusives com algumas funções nativas como a fourier, também utilizada no código para calcular a energia no domínio da frequência. Mesmo a FFT sendo um algoritmo eficiente, há uma demora em processar em computadores convencionais. Há outros algoritmos mais eficientes que a FFT como a sFFT (Sparce Fast Fourier Transform) utilizado principalmente em Big Data, todos melhorias advindas do algoritmo original.

Outros softwares como o Maple também possuem ferramentas para análise de sinais, incluindo análise de Fourier, entretanto com menos conteúdo e exemplos disponíveis online para consulta e aprendizagem.

References

[1] M. G. Côrtes. *Análise comparativa dos parâmetros espectrográficos da voz antes e depois da fonoterapia*. 2007.

[2] B. Haykin, Simon; Van Veen. *Sinais e sistemas*. Bookman, 2001.

[3] M. B. Joaquim. *Processamento Digital de Sinais*. 2010.

[4] V. K. Madisetti. *The Digital Signal Processing Handbook*. CRC Press, 2010.

[5] A. OPPENHEIM, Alan; WILLSKY. *Sinais e Sistemas*. Pearson, 2010.

[6] R. Rabiner, Lawrence; Schafer. *Digital Processing of Speech Signals*. Prentice Hall, 1978.

[6] [4] [5] [2] [1] [3]