Universidade Federal de Viçosa
ELT 410 - Processamento digital de sinais
Aula prática 7: Série de Fourier - Gibbs - Prof. Leonardo Bonato Felix

1 Introdução

Esta prática refere-se a representação de sinais periódicos através da série de Fourier. Em particular, será estudado a reconstrução de sinais periódicos usando a série de Fourier truncada.

2 Comandos úteis

- `abs`, calcula o módulo.
- `angle`, calcula a fase.
- `sum`, realiza o somatório de uma matriz.
- `help`, ajuda para todos os comandos.

3 Roteiro

Fenômeno Gibbs - Exercício 1

Nesta seção, o sinal $x(t)$ (Fig.1) será reconstruído a partir de sua série de Fourier e de um número de coeficientes variável. Além disso, será observado o fenômeno de Gibbs.

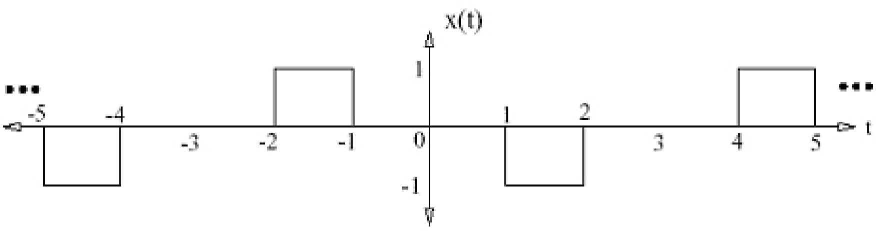

Figura 1: Sinal periódico.

1. Crie um arquivo chamado gibbs.m para este exercício.

2. Acesse a página da disciplina no PVAnet e baixe a função Ck.m. Verifique o conteúdo da função. Esta função usa um argumento k para criar o k-ésimo coeficiente da onda quadrada acima.

3. Plote (usando `subplot` e `stem`) o módulo e a fase de Ck para $k \epsilon [-k_{max}, k_{max}]$. $k_{max} = 10$

4. Escreva um código[1] que implemente a síntese de $x(t)$ com uma série truncada para um dado k_{max}.

$$x(t) = \sum_{k=-k_{max}}^{k_{max}} (C_k e^{jk\omega_0 t})$$

$$x(t) = \sum_{k=0}^{k_{max}} (2|C_k| cos(k\omega_0 t + \angle Ck))$$

5. Plote $x(t)$ para $t \epsilon [-5, 5]$ para cada um dos casos: $k_{max} = 5$, 15 e 30. Use o vetor de tempo `t=-5:.01:5`.

   Conforme se adiciona cossenos, nota-se que a síntese de Fourier aproxima-se cada vez mais da onda quadrada. Todavia, começam a aparecer algumas discrepâncias próximo das descontinuidades do sinal. Esta distorção é chamada de fenômeno Gibbs. Finalize este exercício descrevendo o fenômeno de Gibbs e sua implicações na análise de sinais contínuos no tempo.

Minimização do efeito Gibbs - Exercício 2

Como verificado anteriormente, o efeito Gibbs compromete a síntese de um sinal a partir de sua série de Fourier, através da distorção da forma de onda sintetizada em pontos de descontinuidade. Todavia, é possível minimizar esta distorção através da aplicação de janelamentos especiais.

1. Calcule a série de Fourier para o sinal da Fig. 1 e implemente sua síntese para 2, 5, 10, 50 e 100 termos.

2. Utilize o janelamento Fejer para diminuição do efeito Gibbs. Neste método, se N harmônicos são incluídos na série reconstruída, então a amplitude do k-ésimo harmônico é multiplicada por $(N - k)/N$. Em seguida, utilize o janelamento Hamming, que consiste na multiplicação do k-ésimo harmônico por $0,54 + 0,46cos(k\pi/N)$. Compare os dois métodos.

[1] Pode-se evitar problemas numéricos usando a forma cossenoidal. Neste exemplo, $\omega_0 = 1$

Sintetizador de som - Exercício 3

A série de Fourier pose ser usado para sintetizar outros tipos de sinais, por exemplo, instrumentos musicais.

1. Crie um novo arquivo de código.

2. Acesse o PVAnet e baixe o arquivo *trumpet.mat*. A freqüência de amostragem é 11.025 Hz. Reproduza este arquivo com o comando `sound`.

3. Plote três diferentes trechos do sinal em janelas separadas. Verifique se ele parece o mesmo em toda a extensão.

4. Olhe o espectro do sinal

   ```
   Fs = 11025; Y = fft(trumpet, 512); Ymag = abs(Y); f = Fs * (0:256)/512;
   plot(f, Ymag(1:257)); xlabel('Frequency (Hz)'); ylabel('Magnitude');
   ```

   O resultado esperado é um conjunto de picos (estes são os harmônicos do instrumento).

5. O instrumento pode ser sintetizado usando apenas a informação de picos. Usando o *data cursor* na figura do Matlab, anote os cinco maiores valores dos picos e suas respectivas abscissas.

6. Crie uma função que receba três vetores: vetor de tempo t, vetor de freqüências $freq$ e vetor de módulo mag. Esta função deve somar cossenos para cada par freqüência/modulo dos vetores $freq$ e mag e entregar esta soma na saída. Lembre-se de normalizar a saída entre -1 e 1.

7. Algumas dicas: Use um `for` para somar os cossenos. A função cosseno deve se parecer com isto: `mag(i)*cos(2*pi*freq(i)*t);`. Lembre-se que o vetor de tempo deve ter a forma $0 : 1/Fs :< tempo_e m_s egundos >$.

8. O comando `soundsc` normaliza o som antes de reproduzi-lo.

9. Por exemplo, se dois harmônicos forem considerados, por exemplo em 100 Hz com magnitude 1 e outro em 150 Hz com magnitude 2, então os vetores de entrada serão

   ```
   t = 0:1/Fs:1;

   freq = [100 150];

   mag = [1 2];
   ```

10. Toque o trompete e o som sintetizado. Se assemelham de alguma forma? Use `subplot` para plotar algumas porções de ambos sinais. Se assemelham de alguma forma?

11. Experimente sintetizar com mais e menos harmônicos. Como isto afeta o som do nosso trompete?

12. Use a informação de fase na sintetização. Para isso, plote a fase da série (usando o comando `angle`), verifique o valor da fase em cada harmônico que deseja usar e adicione a fase no cálculo da síntese, i.e. some este valor de fase no argumento do cosseno. Qual(is) a(s) diferença(s), em termos perceptíveis visual e auditivamente, entre a sintetização com e sem a fase?