Processamento Digital de Sinais e Aplicações em Acústica
Tutorial 05 - Série de Fourier

https://github.com/fchirono/Aulas_PDS_Acustica

Objetivos do Tutorial

Ao final desta sessão, você será capaz de:

- Sintetizar sinais a partir dos coeficientes de Fourier, e entender os efeitos de adicionar harmônicos a um sinal do ponto de vista da qualidade sonora;
- Determinar os coeficientes de Fourier de uma função complexa periódica.

Sumário

1 Síntese de Fourier

1.1 Revisão

Sob certas condições, um sinal $y(t)$ pode ser decompostos em uma série de Fourier. Para este exercício, o sinal $y(t)$ será expresso como uma combinação linear de cossenos:

$$y(t) = \sum_{k=1}^{K} a_k \cos(2\pi f_0 k t), \tag{1}$$

onde f_0 é a frequência fundamental em Hertz, t é o tempo em segundos, $k = 1, \ldots, K$ é o índice do k-ésimo harmônico (em relação à frequência fundamental f_0), e a_k é um coeficiente de valor real.

1.2 Tarefas

1. Crie uma função Python que implemente a Equação (1)

```
y = sintetizar(f0, fs, T_max, A_k)
```

   onde `f0` é a frequência fundamental, `fs` é a frequência de amostragem em Hertz, `T_max` é a duração do sinal em segundos e `A_k` é o vetor $[a_1, \cdots, a_K]$ que contém os coeficientes da série de Fourier (o primeiro elemento refere-se à amplitude da frequência fundamental). Por exemplo, para gerar um sinal amostrado a 44100 Hz, frequência fundamental A4 (440 Hz), duração 1 s, com três harmônicos de amplitude $[1, 0.4, 0.2]$, a chamada da função será:

```python
import numpy as np
y = sintetizar(440, 44100, 1, np.array([1, 0.4, 0.2]))
```

Ouça o sinal de saída usando `sounddevice` ou salvando os dados em um arquivo `.wav`.

2. A tabela abaixo fornece as frequências fundamentais de várias notas musicais:

Nota	Freq.	Nota	Freq.	Nota	Freq.	Nota	Freq.	Nota	Freq.	Nota	Freq.
C_1	32.70	C_2	65.41	C_3	130.81	C_4	261.63	C_5	523.25	C_6	1046.50
$C_1^\#$	34.65	$C_2^\#$	69.30	$C_3^\#$	138.59	$C_4^\#$	277.18	$C_5^\#$	554.37	$C_6^\#$	1108.73
D_1	36.71	D_2	73.42	D_3	146.83	D_4	293.66	D_5	587.33	D_6	1174.66
$D_1^\#$	38.89	$D_2^\#$	77.78	$D_3^\#$	155.56	$D_4^\#$	311.13	$D_5^\#$	622.25	$D_6^\#$	1244.51
E_1	41.20	E_2	82.41	E_3	164.81	E_4	329.63	E_5	659.25	E_6	1318.51
F_1	43.65	F_2	87.31	F_3	174.61	F_4	349.23	F_5	698.46	F_6	1396.91
$F_1^\#$	46.25	$F_2^\#$	92.50	$F_3^\#$	185.00	$F_4^\#$	369.99	$F_5^\#$	739.99	$F_6^\#$	1479.98
G_1	49.00	G_2	98.00	G_3	196.00	G_4	392.00	G_5	783.99	G_6	1567.98
$G_1^\#$	51.91	$G_2^\#$	103.83	$G_3^\#$	207.65	$G_4^\#$	415.30	$G_5^\#$	830.61	$G_6^\#$	1661.22
A_1	55.00	A_2	110.00	A_3	220.00	A_4	440.00	A_5	880.00	A_6	1760.00
$A_1^\#$	58.27	$A_2^\#$	116.54	$A_3^\#$	233.08	$A_4^\#$	466.16	$A_5^\#$	923.33	$A_6^\#$	1864.66
B_1	61.74	B_2	123.47	B_3	246.94	B_4	493.88	B_5	987.77	B_6	1975.53

Gere os tons dados na tabela a seguir, concatene-os juntos e ouça a melodia resultante. Sugerimos usar a função `numpy.concatenate`; leia a documentação da função para se familiarizar com sua operação. **Cuidado com o volume de reprodução!**

No.	1	2	3	4	5	6	7	8
Nota	E_4	$D_4^\#$	E_4	F_4	E_4	-	A_4	-
Duração em s	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15
No.	9	10	11	12	13	14	15	16
Nota	E_4	$D_4^\#$	E_4	F_4	E_4	-	$G_4^\#$	-
Duração em s	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15

3. Ouça a melodia geradas na tarefa anterior usando diferentes coeficientes (por exemplo, $\mathbf{a} = [1, 1, 1, 1]$ ou $\mathbf{a} = [0.8, 0.1, 0.2, 0.6]$). Qual é o efeito da variação dos coeficientes de Fourier?

2 Análise de Fourier

2.1 Revisão

Os coeficientes de Fourier c_k de um sinal periódico, de valor complexo e tempo contínuo $x(t)$ são dados através da integral

$$c_k = \frac{1}{T} \int_{t_0}^{t_0+T} x(t)\ e^{-jkt\frac{2\pi}{T}} dt$$

onde T é o período.
Para um sinal de tempo discreto, os coeficientes de Fourier são definidos da seguinte forma:

$$c_k = \frac{1}{N} \sum_{n=n_0}^{n_0+N-1} x[n]\ e^{-jk\frac{2\pi}{N}n},$$

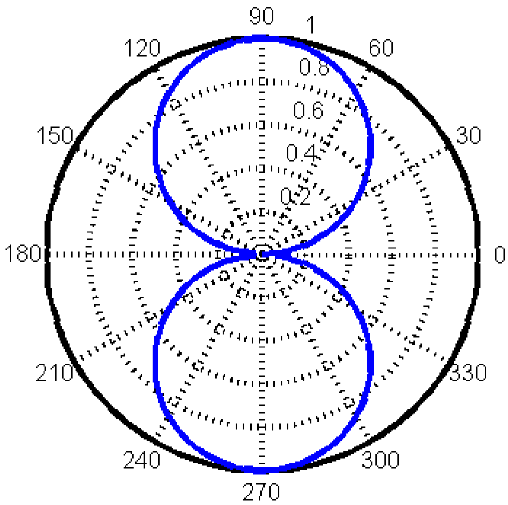
onde n representa o índice da amostra.

2.2 Tarefas: Analisando Padrões de Radiação

O padrão de radiação de uma fonte sonora descreve o campo acústico gerado por esta fonte em função da direção do observador, e geralmente possui magnitude e fase (isto é, assume valores complexos). Para um alto-falante tipo dipolo ideal, por exemplo, este padrão pode ser considerado uma função do ângulo polar ϕ

$$R(\phi) = |\sin(\phi)|.$$

Como tal, também pode ser considerado periódico em 2π e a visualização pode ser vista na figura a seguir.



Tarefas

1. Gere um vetor `phi` contendo ângulos ϕ no intervalo $[0, 2\pi)$. Note que para evitar erros numéricos, é importante usar uma distribuição uniforme de pontos sobre todo o espaço angular.

2. As funções `PadraoRadiacao1` e `PadraoRadiacao2` implementam cada uma um padrão de radiação complexo. A sintaxe para obtê-los é

```
import tutorial2_funcoes as Tutorial2

rad1 = Tutorial2.PadraoRadiacao1(phi)
rad2 = Tutorial2.PadraoRadiacao1(phi)
```

3. Plote ambos os padrões de radiação usando o seguinte código:

```
# Cria um objeto figura com tamanho especificado
FigPolar1 = plt.figure(figsize = (7, 7))

# Cria um objeto "subplot"/eixos com uma grade polar
EixoPolar1 = FigPolar1.add_subplot(111, polar=True)

# Especifica a direcao 'Norte' como a direcao '0'
EixoPolar1.set_theta_zero_location("N")
```

```python
# Cria um grafico polar nos eixos especificados
EixoPolar1.plot(phi, rad1)

# Adiciona um titulo ao eixo
EixoPolar1.set_title("Padrao de Radiacao", size=18)
```

NOTA: Devido aos valores complexos retornados pelas funções, você terá que aplicar a função `np.abs()` aos dados antes de plotá-los.

4. Escreva sua própria função para determinar os coeficientes da série de Fourier para ambos os padrões de radiação. Lembre-se de que os coeficientes de Fourier podem ser complexos!
   Plote a amplitude e fase dos coeficientes de Fourier obtidos. Considere usar `plt.stem` para plotar valores discretos. Qual é a consequência da simetria em relação a $0°$ do primeiro padrão de radiação?

5. Re-sintetize os dois padrões de radiação a partir dos coeficientes de Fourier obtidos e compare os resultados com os originais.