

# 信号处理原理 实验 1

## 实验报告

李祥泽 2018011331

## 实验原理

对于周期为  $T_1$  的函数  $f(t)$ , 其傅里叶级数展开可以写为以下的三角函数形式:

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega_1 t + b_n \sin n\omega_1 t)$$

其中  $\omega_1 = \frac{2\pi}{T_1}$  是原函数的角频率.

## 可视化方波信号

对于周期为  $2\pi$  的方波信号  $f(t) = 0.5 \cdot \text{sgn}(\sin t) + 0.5$ , 或写成分段函数的形式:

$$f(t) = \begin{cases} 1 & \sin t \geq 0 \\ 0 & \sin t < 0 \end{cases}$$

其傅里叶级数的各项系数为:

$$\begin{aligned} a_0 &= \frac{1}{T_1} \int_0^{T_1} f(t) dt &&= \frac{1}{2} \\ a_n &= \frac{1}{T_1} \int_0^{T_1} f(t) \cos(nt) dt &&= 0 \\ b_n &= \frac{1}{T_1} \int_0^{T_1} f(t) \sin(nt) dt &&= \begin{cases} 0 & n = 2, 4, 6, \dots \\ \frac{2}{n\pi} & n = 1, 3, 5, \dots \end{cases} \end{aligned}$$

因此, 相关代码为 (注意到 `fourier_coefficient` 函数传入的 `n` 与上面定义不同):

```
1 def square_wave(t):
2     if math.sin(t) >= 0:
3         return 1
4     else:
5         return 0
6
7 def fourier_coefficient(n):
8     if signal_name == "square":
9         if n == 0:
10            return 1/2
11         if n % 2 == 0:
12            return 0
13         # else : n % 2 == 1
14         m = (n+1)/2
15         if m % 2 == 0:
16            return 0
17         else:
18            return 2/m/math.pi
```

## 实验总结

可以发现, 随着项数的增加, 得到的傅里叶级数展开与原函数的误差逐渐减小. 理论上说, 如果项数趋向于无穷, 该误差将趋向于 0, 从而傅里叶级数展开能精确地表示原函数.

**另:** 建议将实验依赖的 python 包写成 `requirements.txt`, 以便一次性管理.