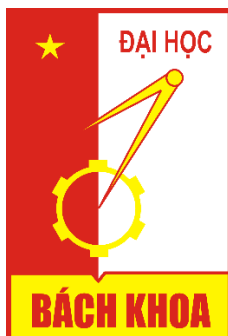


ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CNTT VÀ TT

----- ∞  ∞ -----



Xử lý tín hiệu

Giảng viên: TS. Nguyễn Hồng Quang

Sinh viên thực hiện:

Mai Quốc Đạt 2022277

Hà Nội, năm 2024

Yêu cầu:

Bài tập lập trình (nộp code và báo cáo):

- Tổng hợp tín hiệu xung vuông liên tục tuần hoàn $x(t)$ từ N thành phần tần số với N là tham số cho chương trình.
- Vẽ tín hiệu tổng hợp được với $N = 1, 5$ và 50 . Nhận xét hình dạng $x(t)$ thu được.

Lý thuyết :

Tín hiệu xung vuông liên tục tuần hoàn có thể biểu diễn dưới dạng tổng các sóng sin và cosin theo chuỗi Fourier. Một tín hiệu xung vuông với chu kỳ T_0 có thể được mô tả bằng chuỗi Fourier như sau:

$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t},$$

với $\omega_0 = \frac{2\pi}{T_0}$

Tuy nhiên, trong trường hợp tín hiệu xung vuông, các hệ số Fourier chỉ có giá trị tại các bội số lẻ của c_n . Các hệ số Fourier c_n cho tín hiệu xung vuông có dạng:

$$c_n = \begin{cases} \frac{4}{n\pi}, & n \text{ là lẻ} \\ 0, & n \text{ là chẵn} \end{cases}$$

Do đó, tín hiệu $x(t)$ có thể được biểu diễn như sau:

$$x(t) = \sum_{n=1,3,5,\dots,N} \frac{4}{n\pi} \sin(n\omega_0 t),$$

Thực hiện:

Code đi kèm :

```
import numpy as np
import matplotlib.pyplot as plt

# Định nghĩa các tham số
```

```

T0 = 2 * np.pi # Chu kỳ của tín hiệu
omega_0 = 2 * np.pi / T0 # Tần số cơ bản
t = np.linspace(0, T0, 1000) # Tạo dãy thời gian

# Hàm tổng hợp tín hiệu xung vuông từ N thành phần Fourier
def square_wave(N):
    x_t = np.zeros_like(t)
    for n in range(1, N+1, 2): # Lấy các thành phần tần số lẻ
        x_t += (4 / (n * np.pi)) * np.sin(n * omega_0 * t)
    return x_t

# Vẽ tín hiệu cho N = 1, 5, 50
plt.figure(figsize=(10, 6))

for N in [1, 5, 50]:
    x_t = square_wave(N)
    plt.plot(t, x_t, label=f'N = {N}')

plt.title("Tín hiệu xung vuông liên tục tuần hoàn tổng hợp từ N thành phần")
plt.xlabel("Thời gian (t)")
plt.ylabel("Giá trị x(t)")
plt.legend()
plt.grid(True)
plt.show()

```

Hình ảnh kết quả :

