**A blue and grey sign with white text

Description automatically generated*Trường Đại học Khoa học Tự nhiên TP.HCM***

***Khoa Điện tử - Viễn thông***

**NHẬP MÔN XỬ LÍ ẢNH VÀ VIDEO**

**GVPT:** Huỳnh Hữu Thuận

|  |  |  |
| --- | --- | --- |
| Họ và tên | MSSV | Lớp |
| Võ Thành Nhân | 22200114 | 22Nhung |

**BÀI TẬP VỀ NHÀ LẦN 2**

**1. Tạo tín hiệu sin x1 có tần số 2khz và x2 có tần số 10khz, tần số lấy mẫu là 40khz Thời gian lấy mẫu 10s**

Tạo tín hiệu sin x1 và x2 (Câu 1):

* Sử dụng công thức tín hiệu sin: x(t)=sin(2πft) với f1= 2 kHz (tần số thấp), f2=10 kHz (tần số cao).
* Lấy mẫu: Áp dụng định lý lấy mẫu Nyquist (fs = 40 kHz > 2f\_max = 20 kHz). Vector thời gian rời rạc: tn = n/fs; n=0,1,…,N−1 (N = fs × t = 400,000 mẫu cho t = 10 s).
* Kết quả: x1 là sóng chậm (chu kỳ dài), x2 là sóng nhanh (chu kỳ ngắn).

Code:

fs = 40000 # sample rate 40 kHz

t = 10 # time 10 seconds

N = int(fs \* t) # number of samples

n = np.linspace(0, t, N, endpoint=False) # time vector

# 1) Create signals x1 (2 kHz), x2 (10 kHz)

f1 = 2000

f2 = 10000

x1 = np.sin(2 \* np.pi \* f1 \* n)

x2 = np.sin(2 \* np.pi \* f2 \* n)

Đồ thị sóng sin có tần số 2khz và 10khz

A blue and green lines

AI-generated content may be incorrect.

**2. Tạo tín hiệu tổng x = x1 + x2**

**Tạo tín hiệu tổng x = x1 + x2**: Trong miền thời gian, tổng hai tín hiệu sin là tín hiệu tổng hợp, không thay đổi biên độ/tần số gốc. Phổ tần số chứa hai đỉnh tại 2 kHz và 10 kHz.

Code:

# 2) Create signal x = x1 + x2

x = x1 + x2

Đồ thị sóng tổng x

A purple lines on a white background

AI-generated content may be incorrect.

**3. Sử dụng lọc FIR để loại x1 ra khỏi x. Cho biết các hệ số lọc và tần số cắt.**

Để loại bỏ thành phần tần số thấp (x1= 2 kHz), giữ thành phần cao (x2 = 10 kHz) => chọn tần số cắt = 5kHz nằm giữa 2kHz và 10kHz và bộ lọc cao qua với hệ số lọc 101 (chọn số lẻ để đối xứng)

Lọc với đáp ứng xung đầu vào h(n) và tín hiệu x

Code:

# 3) FIR filter to remove x1 from x (using high-pass filter - keep x2)

# Choose cutoff frequency = 5000 Hz (between f1 and f2)

wC = 5000  # Cutoff frequency (Hz)

Nf = 101  # Filter order

h\_hp = signal.firwin(Nf, wC, pass\_zero=False, window='hamming', fs=fs)

# Apply the high-pass filter to x to get y\_hp

y\_hp = signal.lfilter(h\_hp, 1.0, x)

Đồ thị sóng sau lọc FIR highpass

A red and purple lines

AI-generated content may be incorrect.

**4. Sử dụng lọc FIR để loại x2 ra khỏi x. Cho biết các hệ số lọc và tần số cắt**

Tương tự câu 3) để loại bỏ thành phần tần số cao (x2 = 10 kHz), giữ thành phần thấp (x1= 2 kHz) => chọn tần số cắt = 5kHz nằm giữa 2kHz và 10kHz và bộ lọc thấp qua với hệ số lọc 101 (chọn số lẻ để đối xứng)

Lọc với đáp ứng xung đầu vào h(n) và tín hiệu x

Code:

# 4) FIR filter to remove x2 from x (using low-pass filter - keep x1)

# Choose cutoff frequency = 5000 Hz (between f1 and f2)

h\_lp = signal.firwin(Nf, wC, pass\_zero=True, window='hamming', fs=fs)

# Apply the low-pass filter to x to get y\_lp

y\_lp = signal.lfilter(h\_lp, 1.0, x)

Đồ thị sóng sau lọc FIR lowpass

A purple and blue line

AI-generated content may be incorrect.

--------------------------------------------------------------------------------------

**Script Python:** python exercise\_2.py

Sau khi chạy soucre code thì sinh ra 4 file png gồm: signal, signal\_sum, signal\_lowpass, signal\_highpass. Các file này sẽ chứa các dạng sóng tương ứng.