	Khoa Điện tử Viễn thông- Trường Đại học Khoa học tự nhiên Bài 4: THIẾT KẾ LỌC FIR
	Nhóm: Ca02 Họ và tên: Nguyễn Anh Tuấn MSSV:20200399
	 Mục đích: Xác định được các thông số bộ lọc cần thiết kế Thiết kế được bộ lọc FIR Phân tích tín hiệu trước và sau lọc
	Tốm tắt lý thuyết: Để thiết kế lọc FIR với phương pháp cửa sổ ta sử dụng hàm scipy.signal.firwin(numtaps, cutoff, width=None, window='hamming', pass_zero=True) Tham số đầu vào: • numtaps: Bậc lọc
	 cutoff: Tần số cắt chuẩn hóa fc = f/(fs/2) width: độ rộng chuyển tiếp của lọc window: dạng cửa sổ của bộ lọc. Một số loại thông dụng: "triang", "blackman", "hamming", "hann", "kaiser", pass_zero: {True, False, 'bandpass', 'lowpass', 'bandstop'} True: Độ lợi tại zero = 1
	 False: Độ lợi tại tần số zero = 1 Đối với phiên bản mới hơn 1.3.0, có thể sử dụng chuỗi để xác định loại lọc Hàm trả về các hệ số đáp ứng xung của lọc FIR. Lưu ý: Chia làm 2 loại: Loại 1 đối với bậc lọc lẻ và loại 2 đối với bậc lọc chẵn.
In [2]:	Lọc loại 2 có đáp ứng zero tại tần số Nyquist. Nếu chúng ta sử dụng lọc bậc chẵn mà có dải qua kết thúc bên phải của tần số Nyquist sẽ phát sinh lỗi Thêm thư viện vào project
In [3]:	import numpy as np Bài 1a: Chương trình sau đây thực hiện lọc FIR thấp qua 8 bậc với cửa sổ tam giác (triang), tần số cắt là 10 Hz
	nSamples = 400 #Số mẫu t = np.arange(nSamples)/sampleRate #Khoảng thời gian x = np.cos(2*np.pi*0.5*t) + 0.2*np.sin(2*np.pi*2.5*t+0.1) + \
	<pre>wC = cutOffFreq/nyquistRate # Tần số cắt chuẩn hóa b = signal.firwin(N, wC, window="triang") y = signal.lfilter(b, 1, x) w, h = signal.freqz(b,1, worN=1024) plt.subplot(2,1,1) plt.plot((w/np.pi)*nyquistRate, abs(h), linewidth=2) plt.title('Đáp ứng tần số') plt.xlabel('Tần số (Hz)') plt.ylabel('Biên độ')</pre>
	plt.ylabel('Blen do') plt.subplot(2,1,2) plt.plot(t, x, linewidth=2) plt.title('Tín hiệu vào và ra') plt.xlabel('Thời gian') plt.ylabel('Biên độ') plt.ylabel('Biên độ') plt.plot(t, y, 'r-', linewidth=2) plt.subplots_adjust(top=1.5, hspace=0.5)
	Đáp ứng tắn số 1.0
	0.2 - 0.0 - 0 10 20 30 40 50 Tán số (Hz) Tín hiệu vào và ra
	15 10 10 10 10 10 10 10 10 10 10 10 10 10
	-1.5
In [4]:	#tín hiệu dạng sóng ra ít dao động hơn tín hiệu vào vì các thành phần tần số cao bị loại bỏ Bài 1c: Thay đổi bậc lọc từ 8 sang 16, 32, 64.
In [5]:	Nhận xét đáp ứng tần số của bộ lọc và tín hiệu ngõ ra? Tín hiệu ngõ ra có được lọc hết được các tần số lớn hơn 2.5? sampleRate = 100 #Tần số lấy mẫu nSamples = 400 #Số mẫu t = np.arange(nSamples)/sampleRate #Khoảng thời gian x = np.cos(2*np.pi*0.5*t) + 0.2*np.sin(2*np.pi*2.5*t+0.1) + \
	0.2*np.sin(2*np.pi*15.3*t) + 0.1*np.sin(2*np.pi*16.7*t + 0.1) + \
	<pre>b = signal.firwin(N, wC, window="triang") y = signal.lfilter(b, 1, x) w, H = signal.freqz(b,1, worN=1024) plt.subplot(3,2,1) plt.plot((w/np.pi)*nyquistRate, abs(H), linewidth=2) plt.title('Đáp ứng tần số') plt.xlabel('Tần số (Hz)') plt.ylabel('Biên độ')</pre>
	<pre>plt.subplot(3,2,2) plt.plot(t, x, linewidth=2) plt.title('Tín hiệu vào và ra') plt.xlabel('Thời gian') plt.ylabel('Biên độ') plt.plot(t, y, 'r-', linewidth=2) plt.subplots_adjust(top=1.5, hspace=0.5)</pre> #N=32
	N = 32 wC = cutOffFreq/nyquistRate # Tần số cắt chuẩn hóa b = signal.firwin(N, wC, window="triang") y = signal.lfilter(b, 1, x) w, H = signal.freqz(b,1, worN=1024) plt.subplot(3,2,3) plt.plot((w/np.pi)*nyquistRate, abs(H), linewidth=2) plt.title('Đáp ứng tần số')
	<pre>plt.xlabel('Tẩn số (Hz)') plt.ylabel('Biên độ') plt.subplot(3,2,4) plt.plot(t, x, linewidth=2) plt.title('Tín hiệu vào và ra') plt.xlabel('Thời gian') plt.ylabel('Biên độ') plt.plot(t, y, 'r-', linewidth=2) plt.subplots_adjust(top=1.5, hspace=0.5)</pre>
	<pre>#N=64 N = 64 wC = cutOffFreq/nyquistRate # Tẩn số cắt chuẩn hóa b = signal.firwin(N, wC, window="triang") y = signal.lfilter(b, 1, x) w, H = signal.freqz(b,1, worN=1024) plt.subplot(3,2,5)</pre>
	plt.plot((w/np.pi)*nyquistRate, abs(H), linewidth=2) plt.title('Đáp ứng tần số') plt.xlabel('Tẩn số (Hz)') plt.ylabel('Biên độ') plt.subplot(3,2,6) plt.plot(t, x, linewidth=2) plt.title('Tín hiệu vào và ra') plt.xlabel('Thời gian') plt.ylabel('Biên độ')
	plt.plot(t, y, 'r-', linewidth=2) plt.subplots_adjust(top=1.5, hspace=0.5) Dáp ứng tần số Tín hiệu vào và ra
	0.0 -
	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
	0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
In [6]:	Bài 1d: Thực hiện lại bộ lọc với cửa sổ blackman, hamming, hann sampleRate = 100 #Tẩn số lấy mẫu nSamples = 400 #Số mẫu t = np.arange(nSamples)/sampleRate #Khoảng thời gian x = np.cos(2*np.pi*0.5*t) + 0.2*np.sin(2*np.pi*2.5*t+0.1) + \
	0.2*np.sin(2*np.pi*15.3*t) + 0.1*np.sin(2*np.pi*16.7*t + 0.1) + \
	<pre>wC = cutOffFreq/nyquistRate # Tẩn số cắt chuẩn hóa b = signal.firwin(N, wC, window="blackman") y = signal.lfilter(b, 1, x) w, H = signal.freqz(b,1, worN=1024) plt.subplot(3,2,1) plt.plot((w/np.pi)*nyquistRate, abs(H), linewidth=2) plt.title('Đáp ứng tẩn số') plt.xlabel('Tẩn số (Hz)') plt.ylabel('Biên độ')</pre>
	<pre>plt.subplot(3,2,2) plt.plot(t, x, linewidth=2) plt.title('Tín hiệu vào và ra') plt.xlabel('Thời gian') plt.ylabel('Biên độ') plt.plot(t, y, 'r-', linewidth=2) plt.subplots_adjust(top=1.5, hspace=0.5)</pre> # windown = hamming
	<pre>N = 16 wC = cutOffFreq/nyquistRate # Tẩn số cắt chuẩn hóa b = signal.firwin(N, wC, window="hamming") y = signal.lfilter(b, 1, x) w, H = signal.freqz(b,1, worN=1024) plt.subplot(3,2,3) plt.plot((w/np.pi)*nyquistRate, abs(H), linewidth=2) plt.title('Đáp ứng tần số')</pre>
	plt.title('Dáp ứng tẩn số') plt.xlabel('Tấn số (Hz)') plt.ylabel('Biên độ') plt.subplot(3,2,4) plt.plot(t, x, linewidth=2) plt.title('Tín hiệu vào và ra') plt.xlabel('Thời gian') plt.xlabel('Biên độ') plt.ylabel('Biên độ') plt.plot(t, y, 'r-', linewidth=2) plt.subplots_adjust(top=1.5, hspace=0.5)
	<pre>plt.subplots_adjust(top=1.5, hspace=0.5) # windown = hann N = 16 wC = cutOffFreq/nyquistRate # Tần số cắt chuẩn hóa b = signal.firwin(N, wC, window="hann") y = signal.lfilter(b, 1, x) w, H = signal.freqz(b,1, worN=1024) plt.subplot(3,2,5)</pre>
	plt.subplot(3,2,5) plt.plot((w/np.pi)*nyquistRate, abs(H), linewidth=2) plt.title('Đáp ứng tần số') plt.xlabel('Tần số (Hz)') plt.ylabel('Biên độ') plt.subplot(3,2,6) plt.plot(t, x, linewidth=2) plt.title('Tín hiệu vào và ra') plt.xlabel('Thời gian')
	plt.ylabel('Biên độ') plt.plot(t, y, 'r-', linewidth=2) plt.subplots_adjust(top=1.5, hspace=0.5) Dáp ứng tắn số Tín hiệu vào và ra
	S o o o o o o o o o o o o o o o o o o o
	0.0
	9 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
In [7]:	nSamples = $400 #S ilde{n} ilde{u}$ t = np.arange(nSamples)/sampleRate #KHoảng thời gian x = np.cos(2*np.pi*0.5*t) + 0.2*np.sin(2*np.pi*2.5*t+0.1) + \
	0.2*np.sin(2*np.pi*15.3*t) + 0.1*np.sin(2*np.pi*16.7*t + 0.1) + \ 0.1*np.sin(2*np.pi*23.45*t+.8) nyquistRate = sampleRate/2 #Tần số Nyquist cutOffFreq = 10 #Hz #Độ rộng chuyển tiếp từ dải thông (passband) sang dải chặn (stopband) là 5Hz width = 5.0/nyquistRate rippleInDB = 60 # Độ dợn sóng dải chặn N, beta = signal.kaiserord(rippleInDB, width) #
	# wC = cutOffFreq/nyquistRate # Tẩn số cắt b = signal.firwin(N, wC, window=('kaiser', beta)) y = signal.lfilter(b, 1, x) w, h = signal.freqz(b,1, worN=1024) #Độ dịch pha delay = 0.5 * (N-1) / sampleRate #Vẽ tín hiệu và đáp ứng
	#Vē tín hiệu và đáp ứng plt.subplot(2,1,1) plt.plot((w/np.pi)*nyquistRate, abs(h), linewidth=2) plt.title('Đáp ứng tần số') plt.xlabel('Tẩn số (Hz)') plt.ylabel('Biên độ') plt.subplot(2,1,2) plt.plot(t, x, linewidth=2) plt.plot(t -delay, y, 'r-', linewidth=2)
	plt.plot(t -delay, y, 'r-', linewidth=2) plt.title('Tín hiệu vào') plt.xlabel('Thời gian') plt.ylabel('Biên độ') plt.subplots_adjust(top=1.5, hspace=0.5) Dáp ứng tấn số
	受 0.6 -
	0 10 20 30 40 50 Tấn số (Hz) Tín hiệu vào 1.5 1.0 0.5
	0.5 -0.5 -1.0 -1.5 0 1 2 3 4
	Bài 2: Áp dụng cho tín hiệu và bộ lọc ở câu 1 e: Vẽ đáp ứng bộ lọc tại subplot(2,1,1). Vẽ phổ tín hiệu vào và ra tại subplot(2,1,2)
In [8]:	nSamples = 400 #Số mẫu t = np.arange(nSamples)/sampleRate #KHoảng thời gian x = np.cos(2*np.pi*0.5*t) + 0.2*np.sin(2*np.pi*2.5*t+0.1) + \ 0.2*np.sin(2*np.pi*15.3*t) + 0.1*np.sin(2*np.pi*16.7*t + 0.1) + \
	0.1*np.sin(2*np.pi*23.45*t+.8) nyquistRate = sampleRate/2 #Tần số Nyquist cutOffFreq = 10 #Hz tần số cắt #Độ rộng chuyển tiếp từ dải thông (passband) sang dải chặn (stopband) là 5Hz width = 5.0/nyquistRate rippleInDB = 60 # Độ dợn sóng dải chặn N, beta = signal.kaiserord(rippleInDB, width) #
	<pre>wC = cutOffFreq/nyquistRate # Tần số cắt b = signal.firwin(N, wC, window=('kaiser', beta)) y = signal.lfilter(b, 1, x) w, h = signal.freqz(b,1, worN=1024) #Độ dịch pha delay = 0.5 * (N-1) / sampleRate #Vẽ tín hiệu và đáp ứng plt.subplot(2,1,1)</pre>
	plt.plot((w/np.pi)*nyquistRate, abs(h), linewidth=2) plt.title('Đáp ứng tần số') plt.xlabel('Tần số (Hz)') plt.ylabel('Biên độ') #phân tích phổ tín hiệu vào w, X = signal.freqz(x, 1, worN=1024) w, Y = signal.freqz(y, 1, worN=1024)
	<pre>plt.subplot(2,1,2) plt.plot((w/np.pi)*nyquistRate, abs(X), 'b') plt.plot((w/np.pi)*nyquistRate, abs(Y), 'r') plt.title('phổ biên bộ') plt.xlabel('tấn số') plt.ylabel('Biên độ') plt.subplots_adjust(top=1.5, hspace=0.5)</pre>
	Đáp ứng tấn số 1.0 -
	0.2 - 0.0 - 0 10 20 30 40 50 Tán số (Hz) phổ biên bộ
	200 - 150 - 100 - 50 -
	Bài 3a: [Mẫu] Đoạn chương trình sau đây thực hiện các bộ lọc với dải thông khác nhau với bậc lọc N = 15, tần số cắt chuẩn hóa ω
In [9]:	1. Lọc cao qua ("stop band" từ 0 đến ω): from scipy import signal sampleRate = 100 #Tần số lấy mẫu nyquistRate = sampleRate/2 #Tần số Nyquist N = 15
	<pre>wc = 0.1 # w = fc/nyquistRate b = signal.firwin(N, wc, pass_zero=False) #b là các hệ số lọc print(b) w, H = signal.freqz(b,1, worN=1024) plt.plot((w/np.pi)*nyquistRate, abs(H), linewidth=2) plt.title('Đáp ứng tần số của bộ lọc') plt.xlabel('Tần số (Hz)') plt.ylabel('Biên độ')</pre>
Out[9]:	[-0.00293555 -0.00631867 -0.0160777 -0.03303718 -0.05499888 -0.07714916 -0.09364246 0.89770084 -0.09364246 -0.07714916 -0.05499888 -0.03303718 -0.0160777 -0.00631867 -0.00293555] Text(0, 0.5, 'Biên độ') Dáp ứng tắn số của bộ lọc
	0.8 - 18 0.7 - 19 0.6 - 0.5 -
	1. Lọc dải thông (band pass):
In [10]:	<pre>sampleRate = 100 #Tẩn số lấy mẫu nyquistRate = sampleRate/2 #Tẩn số Nyquist N = 15 w1, w2 = 0.2, 0.5 b = signal.firwin(N, [w1, w2], window = 'blackmanharris', pass_zero=False) #Thêm cửa số print(b) w, H = signal.freqz(b,1, worN=1024) plt.plot((w/np.pi)*nyquistRate, abs(H), linewidth=2)</pre>
	plt.title('Đáp ứng tần số của bộ lọc') plt.xlabel('Tần số (Hz)') plt.ylabel('Biên độ') [-2.02441315e-07 2.03442146e-04 3.22270010e-03 -9.22464023e-03 -1.04454389e-01 -1.42554766e-01 1.76911861e-01 4.54801815e-01 1.76911861e-01 -1.42554766e-01 -1.04454389e-01 -9.22464023e-03 3.22270010e-03 2.03442146e-04 -2.02441315e-07]
Out[10]:	Đáp ứng tấn số của bộ lọc 1.0
	9 0.6 - 0.4 - 0.2 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0
In [11]:	0 10 20 30 40 50 Tấn số (Hz) 1. Lọc dải chặn (band stop): sampleRate = 100 #Tấn số lấy mẫu nyquistRate = sampleRate/2 #Tần số Nyquist
	<pre>N = 15 w1, w2 = 0.2, 0.5 b = signal.firwin(N, [w1, w2]) print(b) w, H = signal.freqz(b,1, worN=1024) plt.plot((w/np.pi)*nyquistRate, abs(H), linewidth=2) plt.title('Đáp ứng tần số của bộ lọc') plt.xlabel('Tần số (Hz)')</pre>
Out[11]:	plt.ylabel('Biên độ') [1.84121573e-04 -4.04871261e-03 -1.66687274e-02
	Đáp ứng tần số của bộ lọc 10 -
	0.4 - 0.2 - 0.10 20 30 40 50 Tấn số (Hz)
In [12]:	1. Lọc multi band, dải thông là [0, w1], [w2, w3] và [w4, 1] sampleRate = 100 #Tần số lấy mẫu nyquistRate = sampleRate/2 #Tần số Nyquist N = 15 w1, w2, w3, w4 = 0.2, 0.5, 0.6, 0.8 b = signal.firwin(N, [w1, w2, w3, w4]) print(b)
	print(b) w, H = signal.freqz(b,1, worN=1024) plt.plot((w/np.pi)*nyquistRate, abs(H), linewidth=2) plt.title('Đáp ứng tần số của bộ lọc') plt.xlabel('Tẩn số (Hz)') plt.ylabel('Biên độ') [0.00594575 -0.01458124 -0.01659233 0.07623798 0.02892033 0.17803264 -0.01530625 0.51468624 -0.01530625 0.17803264 0.02892033 0.07623798
Out[12]:	-0.01659233 -0.01458124 0.00594575]
	18 0.6 - 0.4 - 0.2 -
In [13]:	1. Lọc multi-band với dải thông [w1, w2] và [w3, w4] sampleRate = 100 #Tần số lấy mẫu
[61]	<pre>nyquistRate = sampleRate/2 #Tẩn số Nyquist N = 15 w1, w2, w3, w4 = 0.2, 0.5, 0.6, 0.8 b = signal.firwin(N, [w1, w2, w3, w4], pass_zero=False) print(b) w, H = signal.freqz(b,1, worN=1024) plt.plot((w/np.pi)*nyquistRate, abs(H), linewidth=2) plt.title('Đáp ứng tẩn số của bộ lọc') plt.xlabel('Tẩn số (Hz)')</pre>
Out[13]:	plt.xlabel('Tẩn số (Hz)') plt.ylabel('Biên độ') [-0.00666967
	0.2 - 0.0 - 10 - 20 - 30 - 40 - 50 Tấn số (Hz) Bài 3b:
In [14]:	Thực hiện lại bài 3a.2 với bậc lọc bằng 3. Nhận xét và giải thích kết quả sampleRate = 100 nyquisRate = sampleRate/2 N=3 w1, w2 = 0.2, 0.5 b = signal.firwin(N, [w1, w2], window = 'blackmanharris', pass_zero=False) #Thêm của số
	print(b) w, H = signal.freqz(b,1, worN=1024) plt.plot((w/np.pi)*nyquistRate, abs(H), linewidth=2) plt.title('Đáp ứng tần số của bộ lọc') plt.xlabel('Tần số (Hz)') plt.ylabel('Biên độ') #nhận xét: #đáp ứng của bộ lọc không thể hiện rõ lọc dải qua bậc lọc thấp
Out[14]:	[2.62417806e-05 9.99976173e-01 2.62417806e-05] Text(0, 0.5, 'Biên độ') +1
	-0.00004 - -0.00006 - -0.00008
	Bài 4: Cho tín hiệu x a. Thực hiện mạch lọc loại bỏ tín hiệu 0.5Hz và 2.5Hz
In [15]:	b. Thực hiện mạch lọc chỉ giữ lại tần số 15.3Hz sampleRate = 100 #Tần số lấy mẫu nSamples = 400 #Số mẫu t = np.arange(nSamples)/sampleRate #Khoảng thời gian x = np.cos(2*np.pi*0.5*t) + 0.2*np.sin(2*np.pi*2.5*t+0.1) + \ np.sin(2*np.pi*15.3*t) + 0.1*np.sin(2*np.pi*18.7*t + 0.1) + \
	<pre>0.1*np.sin(2*np.pi*23.45*t+.8) #Viết tiếp chương trình ở đây w, X = signal.freqz(x, 1, worN=1024) fc=10 N=39 wC=fc/50 # w = fc/nyquistRate nyquistRate = sampleRate/2 b = signal.firwin(N, wc, pass_zero=False)</pre>
	<pre>w, H = signal.freqz(b, 1, worN=1024) y = signal.lfilter(b, 1, x) w, Y = signal.freqz(y, 1, worN=1024) plt.subplot(2,1,1) plt.plot(w/np.pi*nyquistRate, abs(H), linewidth=2) plt.title('Dáp ứng tần số') plt.xlabel('Tẩn số (Hz)')</pre>
	plt.ylabel('Biên độ') plt.subplot(2,1,2) plt.plot(w/np.pi*nyquistRate, abs(X),'b') plt.plot(w/np.pi*nyquistRate, abs(Y),'r') plt.title('phổ biên độ') plt.xlabel('tần số') plt.ylabel('Biên độ') plt.subplots_adjust(top=1.5, hspace=0.5)
	## Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dáp ứng tần số Dá
	0.2 - 0.0 - 0 10 20 30 40 50 Tán số (Hz) phổ biên độ
	200 - 150 - 150 - 100 - 150 - 50 -
In [18]:	sampleRate = 100 #Tần số lấy mẫu nSamples = 400 #Số mẫu
	nSamples = 400 #Số mẫu t = np.arange(nSamples)/sampleRate #Khoảng thời gian x = np.cos(2*np.pi*0.5*t) + 0.2*np.sin(2*np.pi*2.5*t+0.1) + \
	<pre>nyquistRate = sampleRate/2 w1, w2 = 0.24, 0.33 b = signal.firwin(N, [w1, w2], window='blackmanharris', pass_zero=False) w, H = signal.freqz(b, 1, worN=1024) y = signal.lfilter(b, 1, x) w, Y = signal.freqz(y, 1, worN=1024)</pre>
	<pre>plt.subplot(2,1,1) plt.plot((w/np.pi)*nyquistRate, abs(H),linewidth=2) plt.title('Dáp ứng tấn số') plt.xlabel('Tẩn số (Hz)') plt.ylabel('Biên độ') plt.subplot(2,1,2) plt.plot(w/np.pi*nyquistRate, abs(X),'b') plt.plot(w/np.pi*nyquistRate, abs(Y),'r') plt.plot(w/np.pi*nyquistRate, abs(Y),'r')</pre>
	plt.plot(w/np.pi*nyquistRate, abs(Y),'r') plt.title('phổ biện độ') plt.xlabel('tẩn số') plt.ylabel('Biên độ') plt.subplots_adjust(top=1.5, hspace=0.5) Dáp ứng tấn số 10 08
	98' 0.6 - 0.2 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.0 - 0.
	phổ biên độ 200 - 150 -
	200 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 - 150 -
In [32]:	Bài 5: Cho file âm thanh gốc là "violin_origional.wav" và file âm thanh gốm nhiều "violin_noise.wav" a. Phân tích phổ của 2 file trên b. Xác định tấn số của nhiều c. Thiết kế mạch lọc để loại nhiều trong file "violin_noise.wav"
In [32]:	Bài 5: Cho file âm thanh gốc là "violin_origional.wav" và file âm thanh gốm nhiều "violin_noise.wav" a. Phân tích phổ của 2 file trên b. Xác định tần số của nhiều c. Thiết kế mạch lọc để loại nhiều trong file "violin_noise.wav" #Thi/ viên from scipy .io import signal import I Pṛtpack as fft from scipy inport signal import I Pṛtpack as ind #Thêm thư viên để play Audio # Doc file Tile_name= "violin_noise.wav" fs, s = wavfile_read(file_name) #Sinh viên thêm đoạn code vào đây
In [32]:	Bài 5: Cho file âm thanh góc là "violin_origional.wav" và file âm thanh góm nhiều "violin_noise wav" a. Phân tích phổ của 2 file trên b. Xào định tần số của ch nhiều c. Thiết kế mạch kọc để loại nhiều trong file "violin_noise.wav" #Thư viện from scipy, io import signal import IPython display as tipd #Thêm thư viên để play Audio # Độc file_name= "violin_noise.wav" [8] 5. s = wavfile read (file_name)
<pre>In [32]: In []: In []:</pre>	BAN 5: Cho lite dam thanh góc là "woll _ongional.wav" và lite âm Pranh góm nhiều "violin_noise wav" a _Phân lich ghổ của 2 liệu với b. Xão dâm tần số của nhiều c. Thiết kế mạch lọc để loại nhiều trong file "violin_noise wav" 277// Vita From scilpy lamport stylank as fit From scilpy lamport stylank as fit From scilpy lamport signal lamport litylank display as job = lhôn thự viện để play Audio file_name "violin_noise.wav" file_name "violin_noise.wav" ###################################