MATLAB Projects

Problem 8.52

Digital speech and audio equalizer Design a seven-band audio equalizer using fourth-order bandpass filters with a sampling rate of 44.1 kHz. The center frequencies are listed in **Table 8.14**. In this project, use the designed equalizer to process a stereo audio ("No9seg.wav").

- Plot the magnitude response for each filter bank.
- Listen and evaluate the processed audio with the following gain settings:

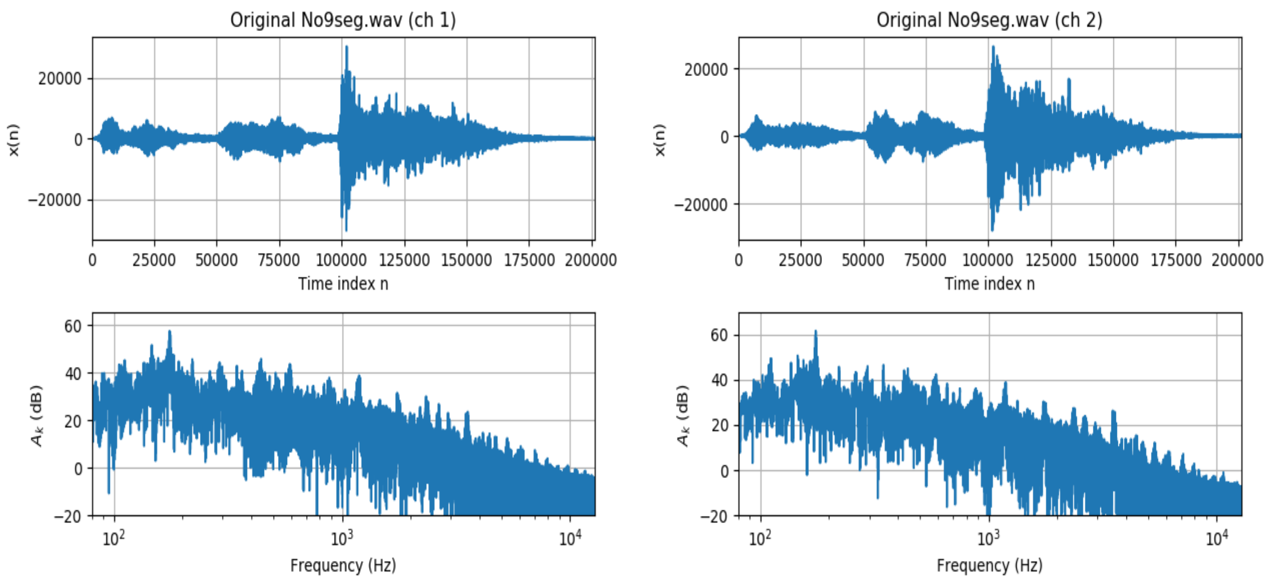
1. each filter bank gain=0 (no equalization)
2. low-pass filtered
3. band-pass filtered
4. high-pass filtered

Table 8.14 Specification for Center Frequencies and Bandwidths

Center Frequency (Hz)	**160**	**320**	**640**	**1280**	**2560**	**5120**	**10,240**
Bandwidth (Hz)	80	160	320	640	1280	2560	5120

solution

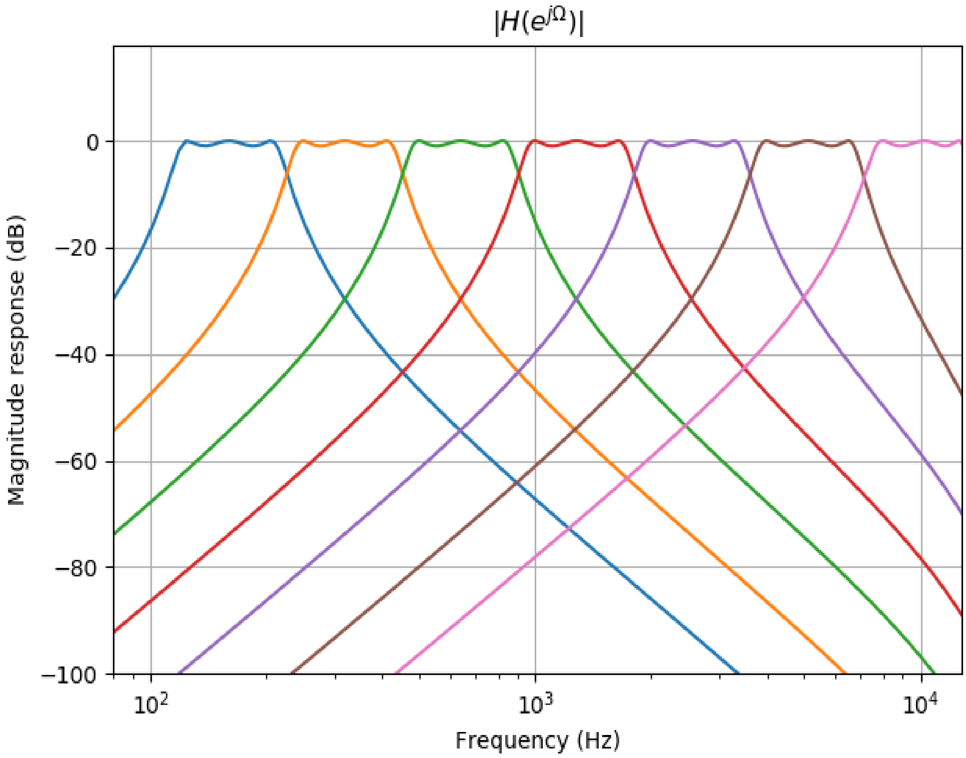
Original signal



Band-pass filter

Here we design the filter:

- Chebyshev I filter: order $n = 3$
- Pass band $A_p = 1$ dB
- Filter type: band-pass
- center frequency $f_c = [160, 320, 640, 1280, 2560, 5120, 10240]$
- Band-width $BW = [80, 160, 320, 640, 1280, 2560, 5120]$
- Sampling frequency $f_s = 44.1$ kHz



Design Chebyshev filter, $A_p = 1dB, \varepsilon = \sqrt{10^{A_p/10} - 1} = 0.5088$

Now with order $n = 3, \varepsilon = 0.5088$, design the Chebyshev filter, n is odd

$$H(s') = \frac{1}{\varepsilon 2^{n-1}} \frac{1}{(s + \text{sh}) \prod_{m=0}^{(\frac{n-1}{2})-1} \left(s^2 + [2 \times \text{sh} \times s(m)]s + [\text{sh}^2 + 1 - s^2(m)]\right)}$$
$$= 0.4913 \times \frac{1}{s + 0.4942} \times \frac{1}{s^2 + 0.4942s + 0.9942}$$

Here, $\text{sh} \equiv \sinh\left(\frac{1}{n}\text{arsinh}\left(\frac{1}{\varepsilon}\right)\right), s(m) \equiv \sin\left(\frac{\pi}{2}\left(\frac{1}{n}\right) + \pi\left(\frac{m}{n}\right)\right)$

Then substitute $s' = \frac{s(\omega_H - \omega_L)}{s^2 + \omega_H \omega_L}$, band-stop filter

Here with $\omega = (2f_s) \times \tan(\pi \frac{f}{f_s}) = (2f_s) \times \tan(\frac{2\pi f}{2f_s})$,

we have $\omega_L = [754.0, 1508.11, 3017.1, 6041.28, 12139.51, 24747.84, 53725.46]$

and $\omega_H = [1340.5, 2681.49, 5367.0, 10766.21, 21794.31, 45827.05, 115719.83]$

$$H(s)$$

$$= 0.4913 \times \frac{586.4957s}{1.0s^2 + 289.8289s + 1010735.0299}$$

$$\times \frac{343977.2177s^2}{s^4 + 289.8289s^3 + 2363453.7874s^2 + 292940262.0727s + 1021585300762.115} \quad [1st]$$

$$= 0.4913 \times \frac{1173.382s}{1.0s^2 + 579.8509s + 4043990.9005}$$

$$\times \frac{1376825.3819s^2}{s^4 + 579.8509s^3 + 9456827.9109s^2 + 2344911788.4637s + 16353862403450.79} \quad [2nd]$$

$$= 0.4913 \times \frac{2349.8925s}{1.0s^2 + 1161.2478s + 16192794.6791}$$

$$\times \frac{5521994.934s^2}{s^4 + 1161.2478s^3 + 37875582.0497s^2 + 18803847462.5707s + 2.6221 \times 10^{14}} \quad [3rd]$$

$$= 0.4913 \times \frac{4724.9293s}{s^2 + 2334.9212s + 65041670.5048}$$

$$\times \frac{22324957.049s^2}{s^4 + 2334.9212s^3 + 152278915.7077s^2 + 151867173966.5871s + 4.2304 \times 10^{15}} \quad [4th]$$

$$= 0.4913 \times \frac{9654.8014s}{s^2 + 4771.119s + 264572327.7615}$$

$$\times \frac{93215190.0733s^2}{s^4 + 4771.119s^3 + 621819625.0523s^2 + 1262306072671.3513s + 6.9999 \times 10^{16}} \quad [5th]$$

$$= 0.4913 \times \frac{21079.2088s}{s^2 + 10416.7253s + 1134120477.0311}$$

$$\times \frac{444333041.5525s^2}{1.0s^4 + 10416.7253s^3 + 2709998902.0356s^2 + 11813821511707.322s + 1.2862 \times 10^{18}} \quad [6th]$$

$$= 0.4913 \times \frac{61994.3763s}{s^2 + 30635.7984s + 6217101105.0179}$$

$$\times \frac{3843302690.7742s^2}{1.0s^4 + 30635.7984s^3 + 16255231373.6225s^2 + 190465856278593.03s + 3.8652 \times 10^{19}} \quad [7th]$$

Thus, H(z)

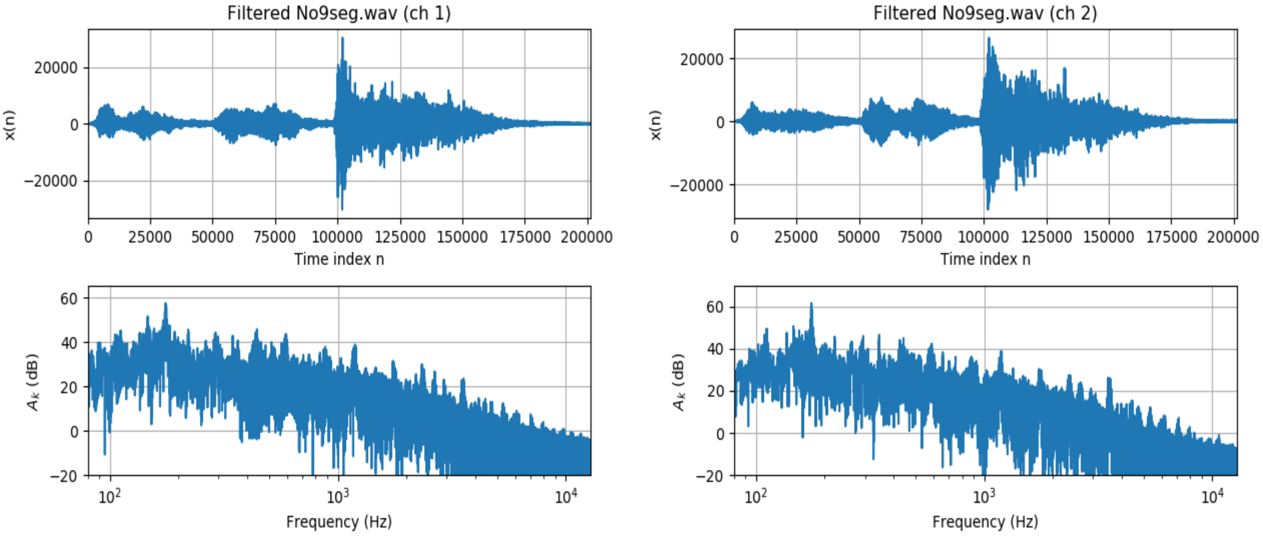
$$
\begin{aligned}
&H(z) \\
&= [(1.4345 \times 10^{-7} - 4.3035 \times 10^{-7} z^{-2} - 0.0 z^{-3} + 4.3035 \times 10^{-7} z^{-4} - 1.4345 \times 10^{-7} z^{-6}) \\
&/(1.0 - 5.9852 z^{-1} + 14.9276 z^{-2} - 19.8588 z^{-3} + 14.8624 z^{-4} - 5.933 z^{-5} + 0.9869 z^{-6})] && [1st] \\
&= [(1.1398 \times 10^{-6} - 3.4194 \times 10^{-6} z^{-2} + 0.0 z^{-3} + 3.4194 \times 10^{-6} z^{-4} - 1.1398 \times 10^{-6} z^{-6}) \\
&/(1.0 - 5.967 z^{-1} + 14.8421 z^{-2} - 19.6985 z^{-3} + 14.7127 z^{-4} - 5.8633 z^{-5} + 0.9741 z^{-6})] && [2nd] \\
&= [(8.9896 \times 10^{-6} - 2.6969 \times 10^{-5} z^{-2} - 0.0 z^{-3} + 2.6969 \times 10^{-5} z^{-4} - 8.9896 \times 10^{-6} z^{-6}) \\
&/(1.0 - 5.9207 z^{-1} + 14.6327 z^{-2} - 19.3225 z^{-3} + 14.3786 z^{-4} - 5.7169 z^{-5} + 0.9488 z^{-6})] && [3rd] \\
&= [(6.9757 \times 10^{-5} - 0.0002093 z^{-2} + -0.0 z^{-3} + 0.0002093 z^{-4} - 6.9757 \times 10^{-5} z^{-6}) \\
&/(1.0 - 5.7893 z^{-1} + 14.0683 z^{-2} - 18.3659 z^{-3} + 13.5846 z^{-4} - 5.3981 z^{-5} + 0.9004 z^{-6})] && [4th] \\
&= [(0.0005209 - 0.001563 z^{-2} + 0.0 z^{-3} + 0.001563 z^{-4} - 0.000521 z^{-6}) \\
&/(1.0 - 5.3813 z^{-1} + 12.4453 z^{-2} - 15.7999 z^{-3} + 11.6086 z^{-4} - 4.6826 z^{-5} + 0.812 z^{-6})] && [5th] \\
&= [(0.0035 - 0.0 z^{-1} - 0.0106 z^{-2} + 0.0106 z^{-4} + 0.0 z^{-5} - 0.0035 z^{-6}) \\
&/(1.0 - 4.0872 z^{-1} + 8.1194 z^{-2} - 9.5695 z^{-3} + 7.0896 z^{-4} - 3.1149 z^{-5} + 0.6669 z^{-6})] && [6th] \\
&= [(0.0183 - 0.0548 z^{-2} + 0.0548 z^{-4} - 0.0183 z^{-6}) \\
&/(1.0 - 0.5511 z^{-1} + 2.0431 z^{-2} - 0.7891 z^{-3} + 1.6473 z^{-4} - 0.3362 z^{-5} + 0.4822 z^{-6})] && [7th]
\end{aligned}
$$

$$Y(z) \equiv X(z) + \sum_{k=1}^{7} \mathrm{Gain}_k \times H(z)_k X(z)$$

$$y(n) = x(n) + \sum_{k=1}^{7} \mathrm{Gain}_k \times [h_k(n) * x(n)]$$

no equalization

$\mathrm{Gain} = [0, 0, 0, 0, 0, 0, 0]$



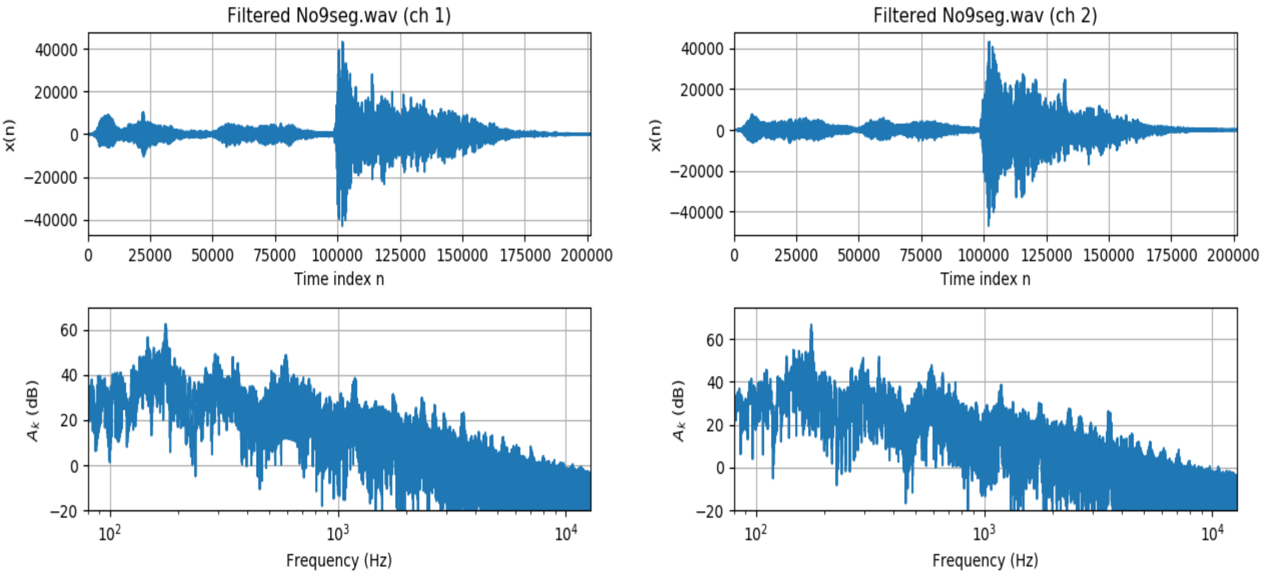
low-pass filtered

$\mathrm{Gain} = [1, 1, 1, 0, 0, 0, 0]$

We can see there are 3 peaks [160, 320, 640] in the filtered spectrum,

components in these band [80, 160, 320] are strengthened by the low-pass filter.

We can hear low frequency components more clearly.



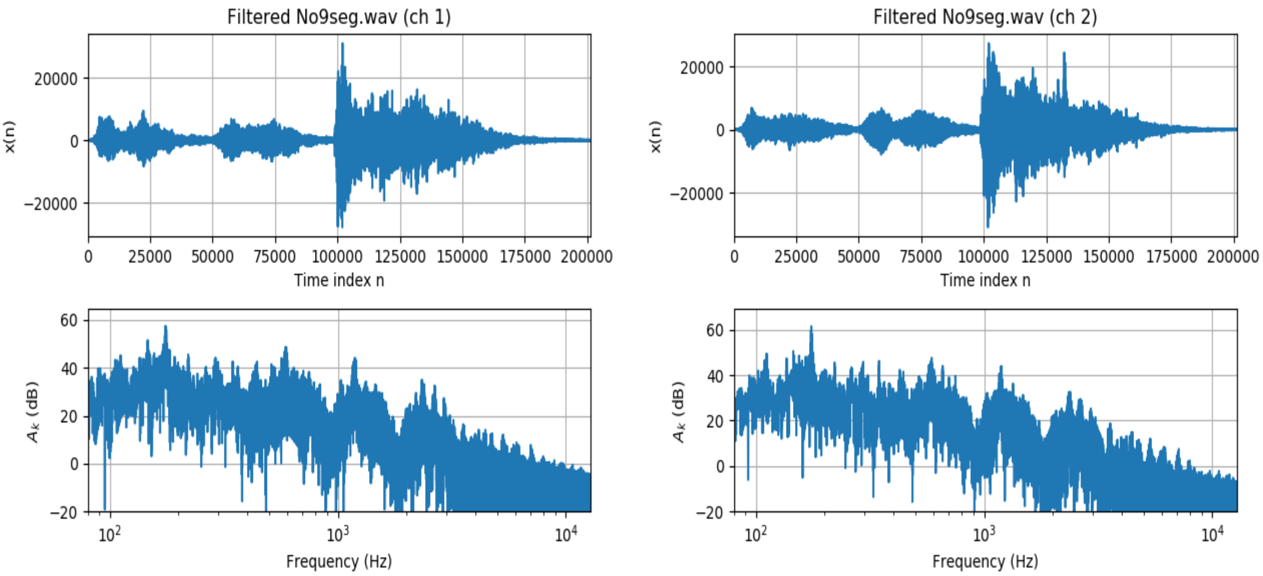
band-pass filtered

$$\mathrm{Gain} = [0, 0, 1, 1, 1, 0, 0]$$

We can see there are 3 peaks[640, 1280, 2560] in the filtered spectrum,

components in these band[320, 640, 1280] are strengthened by the band-pass filter.

We can hear the frequency components in specific frequency band more clearly.



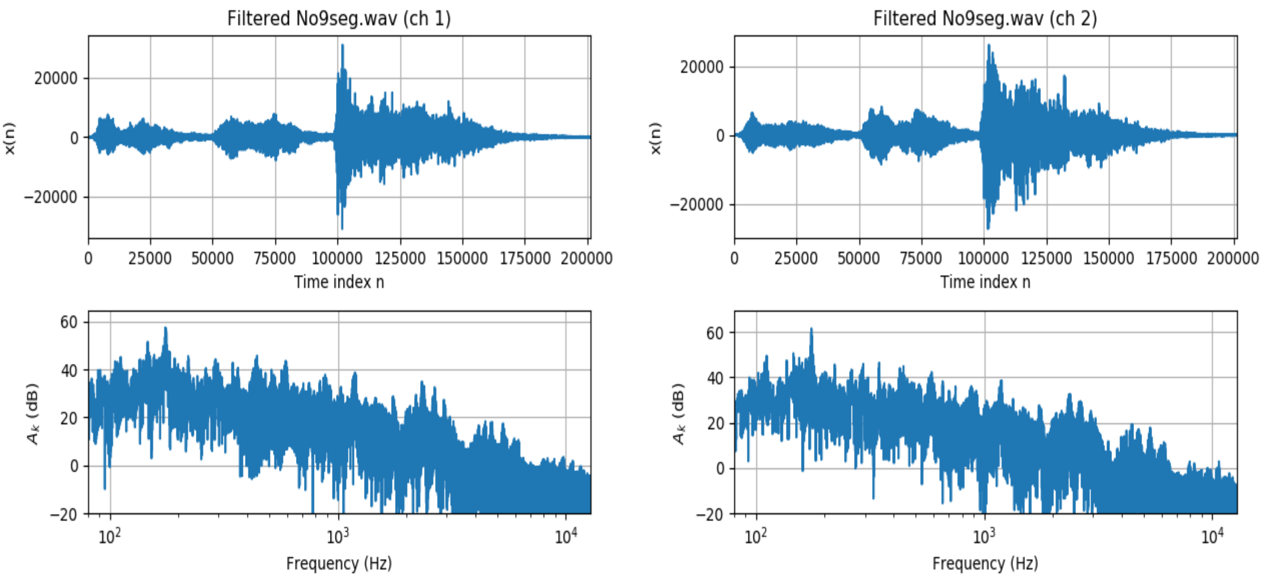
high-pass filtered

$$\mathrm{Gain} = [0, 0, 0, 0, 1, 1, 1]$$

We can see there are 3 peaks[2560, 5120, 10240] in the filtered spectrum,

components in these band[1280, 2560, 5120] are strengthened by the high-pass filter.

We can hear low frequency components more clearly.



Here is the main **Python** script with my IIR implementation library.

The **IIR library** implements:

- Calculation, substitution of Polynomial, Fraction of Polynomial $H(s), H(z)$
- BLT, unit low-pass filter $H(s')$ to low-pass, high-pass, band-pass, band-stop
- Magnitude$|H|$, Phase $\angle H$ of $H(s), H(z)$
- FFT to calculate A_k of $X(k), Y(k)$
- Butterworth, Chebyshev I filter: $H(s)$
- IIR filter: $y(n) = \sum_{k=0}^{M} b_k x(n-k) - \sum_{k=1}^{N} a_k y(n-k)$
- Pole-zero Placement parameters
- Plot of Impulse invariance

```
from scipy.io.wavfile import write, read # save sounds
from iir_filter.fft1d import plot_spectrum_dB
from iir_filter.frac import Frac, convert_s2z
from iir_filter.poly import Poly, Polyz
from iir_filter.util import convert_omega_z2s, filter_subs, calc_omega_pass
from iir_filter.calc_mag_angle import calc_mag_angle, plot_mag_freq_multiple
from iir_filter.protype import chebyshev_protype, calc_cheby_eps2
from iir_filter.iir_filter import iir_filter
from math import pi, sqrt, ceil
from functools import reduce

f_sample, list_input = read("./No9seg.wav") # sample rate,  input
list_input_ch1, list_input_ch2 = list_input.T[0], list_input.T[1]
plot_spectrum_dB(list_input_ch1, f_sample, path_fig="../p8_52_input_ch1.png",
str_title="Original No9seg.wav (ch 1)")
plot_spectrum_dB(list_input_ch2, f_sample, path_fig="../p8_52_input_ch2.png",
str_title="Original No9seg.wav (ch 2)")

list_f_center = [160, 320, 640, 1280, 2560, 5120, 10240]
list_BW = [80, 160, 320, 640, 1280, 2560, 5120]
list_omega_pass_low = [2*pi*(f_center - 0.5 * BW) for f_center, BW in
list(zip(list_f_center, list_BW))]
list_omega_pass_high = [2*pi*(f_center + 0.5 * BW) for f_center, BW in
list(zip(list_f_center, list_BW))]
list2D_omega_pass_z = list(zip(list_omega_pass_low, list_omega_pass_high))
num_filter = len(list2D_omega_pass_z)
order = 3
A_p = 1
epsilon = sqrt( calc_cheby_eps2(A_p) )
print("epsilon = " + str(epsilon))
list_H_s = chebyshev_protype(order, epsilon)
print(list_H_s)
list_H_z = []
list2D_mag, list2D_omega = [], []
for ind in range(num_filter):
    list_omega_pass_z = list2D_omega_pass_z[ind]
    list_omega_pass_s = calc_omega_pass(list_omega_pass_z, f_sample,
str_filter_type="band_pass")
    print(list_omega_pass_s)
```

```
    list_H_subs = [filter_subs(H_s, list_omega_pass_s, str_filter_type="band_pass") for
H_s in list_H_s]
    print(list_H_subs)
    H_subs = reduce(lambda x,y: x * y, list_H_subs)
    print(H_subs)
    H_z = convert_s2z(H_subs, f_sample)
    print(H_z)
    list_H_z.append(H_z)
    list_mag, list_angle, list_omega = calc_mag_angle(H_z, num_point=4096)
    list2D_mag.append(list_mag)
    list2D_omega.append(list_omega)
plot_mag_freq_multiple(list2D_mag, list2D_omega, f_sample, path_fig="../p8_52_H_z.png")
# band_gain = [1] + [0, 0, 0, 0, 0, 0, 0] # the first 1 represent original input gain: no
equalization
# band_gain = [1] + [1, 1, 1, 0, 0, 0, 0] # low pass
# band_gain = [1] + [0, 0, 1, 1, 1, 0, 0] # band pass
band_gain = [1] + [0, 0, 0, 0, 1, 1, 1] # high pass
list2D_output_ch1 = [list_input_ch1]
list2D_output_ch2 = [list_input_ch2]
for H_z in list_H_z:
    list2D_output_ch1.append( iir_filter(list_input_ch1, H_z) )
    list2D_output_ch2.append( iir_filter(list_input_ch2, H_z) )
list2D_output_ch1 = list(map(list, zip(*list2D_output_ch1))) # transpose
list2D_output_ch2 = list(map(list, zip(*list2D_output_ch2)))
list_output_ch1, list_output_ch2 = [], []
for out_ch1, out_ch2 in list(zip(list2D_output_ch1, list2D_output_ch2)):
    list_output_ch1.append( int(sum([elem * gain for elem, gain in list(zip(out_ch1,
band_gain))])) )
    list_output_ch2.append( int(sum([elem * gain for elem, gain in list(zip(out_ch2,
band_gain))])) )
plot_spectrum_dB(list_output_ch1, f_sample, path_fig="../p8_52_output_high pass_ch1.png",
str_title="Filtered No9seg.wav (ch 1)")
plot_spectrum_dB(list_output_ch2, f_sample, path_fig="../p8_52_output_high pass_ch2.png",
str_title="Filtered No9seg.wav (ch 2)")
import numpy as np
list_output = np.asarray([list_output_ch1, list_output_ch2]).T
max_output = max(np.max(list_output), -np.min(list_output))
factor = (2**(16-1)/max_output)
list_output_scaled = np.floor(list_output * factor).astype(np.int16)
write("../No9seg_high pass.wav", f_sample, list_output_scaled)
```