Lab 5. Preparation for project 1

Cheng PENG
Department of Biomedical Engineering
pengc@sustech.edu.cn

Fall 2021

Lab Schedule

2021年	周 次	一	二	三	四	五	六	日	
	第10周 秋季学期	15 十一	16 十二	17 十三	18 十四	19 十五	20 十六	21 十七	← **lab 5**
	第11周 秋季学期	22 小雪	23 十九	24 二十	25 廿一	26 廿二	27 廿三	28 廿四	← Intro to project 1
12月	第12周 秋季学期	29 廿五	30 廿六	1 廿七	2 廿八	3 廿九	4 初一	5 初二	← Q&A
	第13周 秋季学期	6 初三	7 大雪	8 初五	9 初六	10 初七	11 初八	12 初九	← Presentation 1
	第14周 秋季学期	13 初十	14 十一	15 十二	16 十三	17 十四	18 十五	19 十六	← Intro to project 2
	第15周 秋季学期	20 十七	21 冬至	22 十九	23 二十	24 廿一	25 廿二	26 廿三	← Q&A
	第16周 秋季学期	27 廿四	28 廿五	29 廿六	30 廿七	31 廿八	1 元旦	2 三十	← Presentation 2

Overview

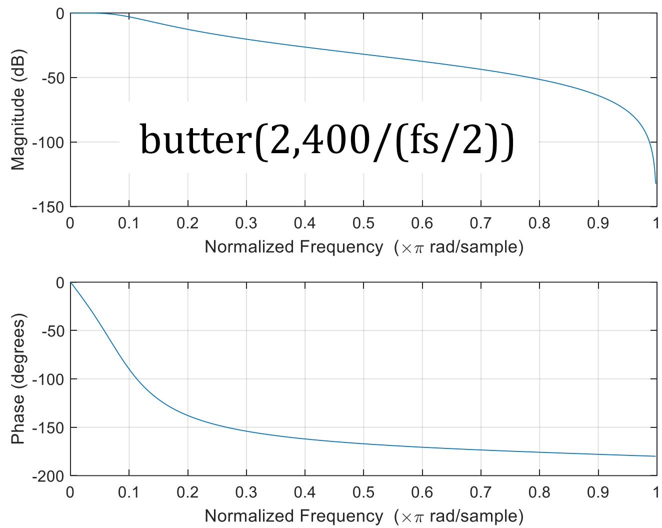
- In this tutorial, you will learn to:
  1. Design basic filters via Matlab
  2. Use filtering to
     ① generate speech-shaped noise
     ② extract signal envelope
  3. Adjust the signal intensity
     ① normalizing signal energy
     ② adjusting signal-to-noise ratio (SNR) level

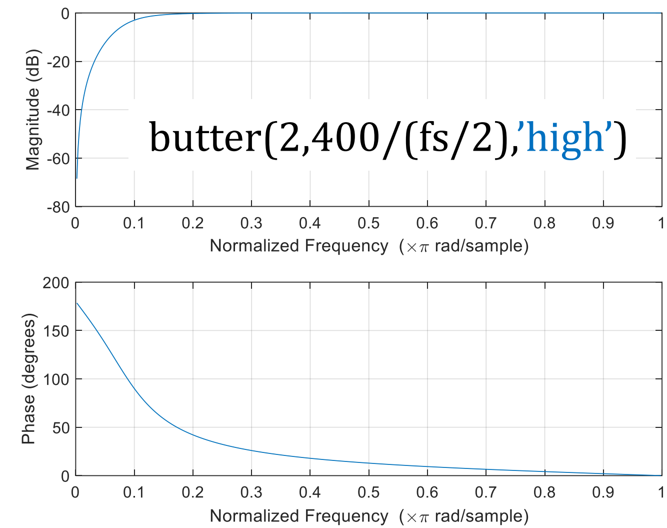
1. Butterworth filter design

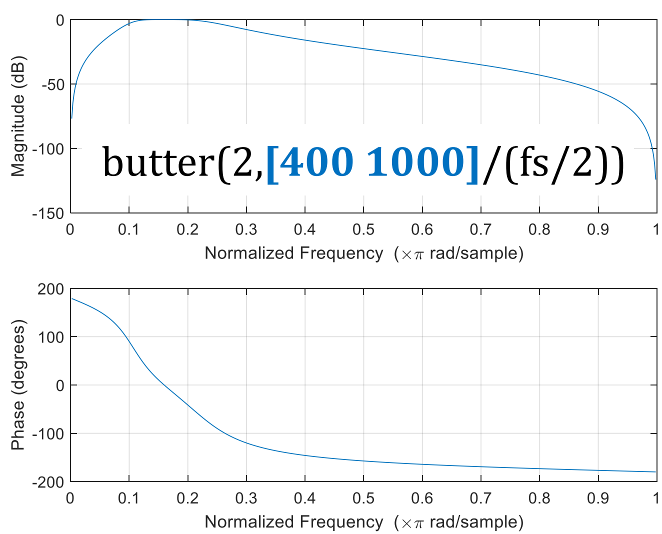
- [b,a]=butter(*n*,*Wn*)
  - Design an order *n* low-pass digital Butterworth filter with normalized cutoff frequency *Wn*. *Wn* must be 0.0 < *Wn* < 1.0, with 1.0 corresponding to half the sample rate (fs/2 in Hz, 2*pi*fs/2 in radian/s).
  - Return the filter coefficients in length n+1 vectors b and a.
- [b,a]=butter(n,Wn,'ftype')
  - Design a highpass, lowpass, or bandstop filter, where the string ‘ftype’ is ‘high’, ‘low’, or ‘stop’,
  - e.g.,

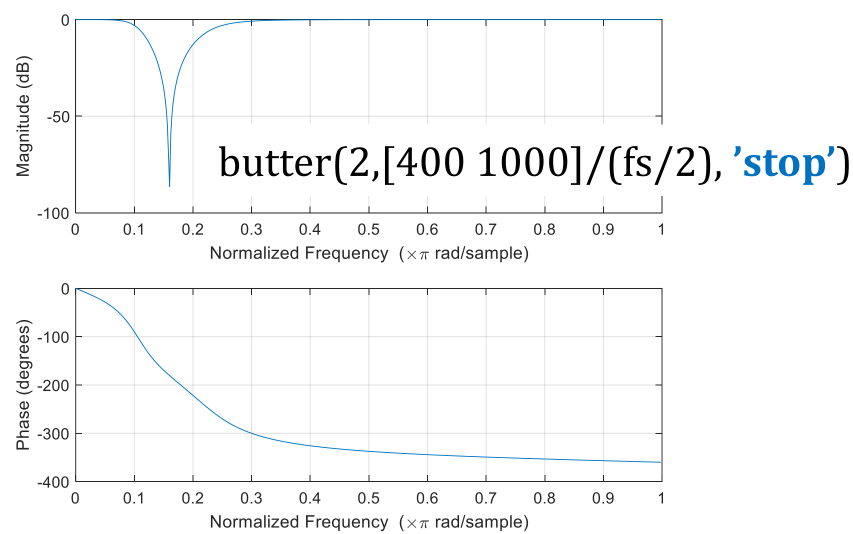
截止频率

```
fs=8000;
[b, a]=butter(2,400/(fs/2)); % low-pass
[b, a]=butter(2,400/(fs/2), 'high'); % high-pass
[b, a]=butter(2,[400 1000]/(fs/2)); % band-pass
[b, a]=butter(2,[400 1000]/(fs/2), ’stop’); % band-stop
```


butter(2,400/(fs/2))
Magnitude (dB)
Normalized Frequency (×π rad/sample)
Phase (degrees)
Normalized Frequency (×π rad/sample)

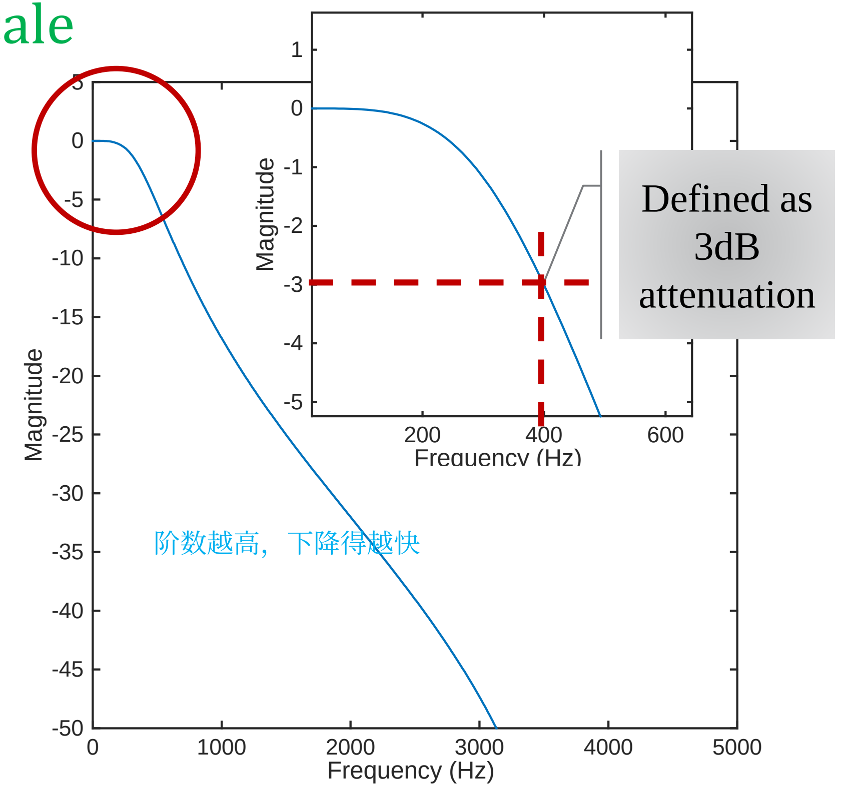

butter(2,400/(fs/2),'high')
Magnitude (dB)
Normalized Frequency (×π rad/sample)
Phase (degrees)
Normalized Frequency (×π rad/sample)


butter(2,[400 1000]/(fs/2))
Magnitude (dB)
Normalized Frequency (×π rad/sample)
Phase (degrees)
Normalized Frequency (×π rad/sample)


butter(2,[400 1000]/(fs/2), 'stop')
Magnitude (dB)
Normalized Frequency (×π rad/sample)
Phase (degrees)
Normalized Frequency (×π rad/sample)

- fs=8000; % sampling rate for discrete signal
- [b, a]=butter(2,400/(fs/2)); % low pass filter
- [h,f]=freqz(b,a,512,fs); % Digital filter frequency response
- plot(f,20*log10(abs(h))); % in dB scale
- axis([0 5000 -50 5]);
- xlabel('Frequency (Hz) ');
- ylabel('Magnitude (dB)');

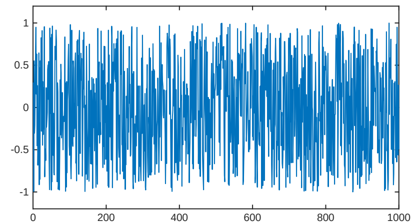
How about 'high-pass' and 'band-pass'?
How about high-order n?

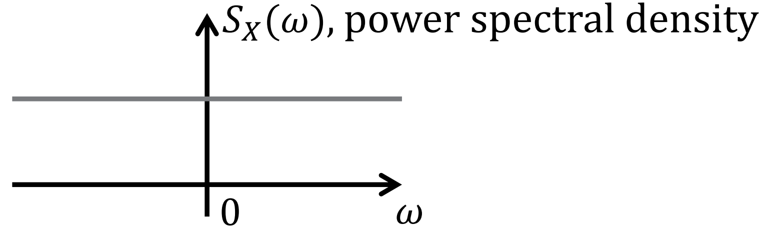


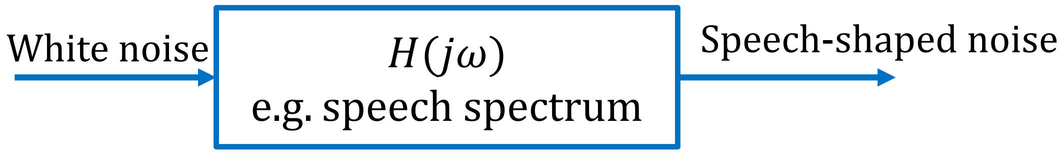
2. Use filtering to ...

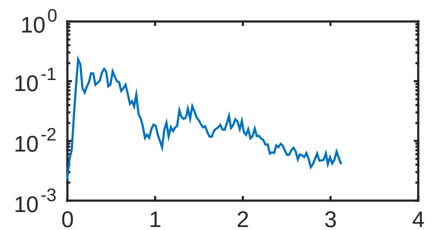
① Generate speech-shaped noise

- White noise: `N=1000; noise = 1-2*rand(1,N);`









Steps:

1) generate long-term spectrum of speech signal 不会拼接成矩阵
   - concatenate 10 speech waveforms into one signal
     - sig =[x,x,x,x,x,x,x,x,x,x]; Or, repmat(x,1,10); % x is a row vector
     - sig =[x; x; x; x; x; x; x; x; x; x]; Or, repmat(x,10,1); % x is a column vector
   - estimate the power spectral density of the speech signal
     - [Pxx,w] =psd(sig,[],512,fs);
     - Or [Pxx,w] = periodogram(sig,[],512,fs)
   - plot(w,Pxx)
     - Or plot(w,10*log10(Pxx))

> ❖ psd for lower version of Matlab
> ❖ periodogram for higher version

- 有同学可能从其他途径了解到 pwelch
  - pwelch 是对 periodogram(周期图法)的一种改进 (已超纲)
  - 两个方法给出的结果看上去并不一样 (因为改进了嘛)
  - periodogram 和 pwelch 可以任选一个使用，但全程要**保持不变**
    - 选了periodogram就一直用periodogram
    - 选了pwelch就一直用pwelch
- 不知道pwelch的同学不用管(因为已超纲)
  - 并没有什么损失
- 其他做功率谱估计(power spectral estimation)的方法不需要去了解 (因为已超纲)
  - 把时间用在关键问题上 (完成lab task)

Steps:

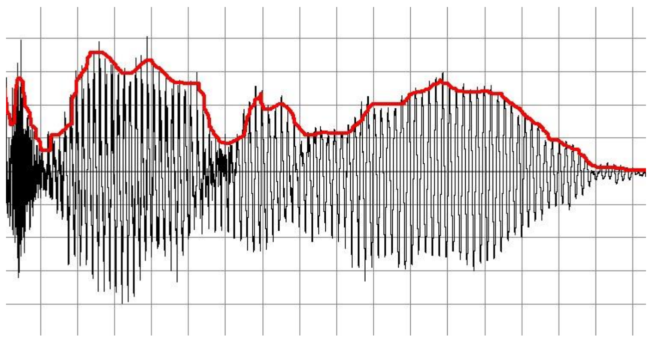
2) generate filter coefficients
   - b = fir2(3000,w/(fs/2),sqrt(Pxx/max(Pxx))); % fir2: from frequency response to coefficients, a=1
   - [h,wh] = freqz(b,1,128); %check frequency response here
3) perform filtering on white noise signal
   - y= filter(b,1,noise);

$$H(j\omega) = \frac{\sum_{k=0}^{N} b_k (j\omega)^k}{\sum_{k=0}^{M} a_k (j\omega)^k}$$

- Then you'll get the speech-shaped noise

Use filtering to ...

② Extract speech envelope (包络) waveform



Original waveform

Envelope

https://commons.wikimedia.org/wiki/File%3AC_Envelope_follower.jpg

Signal → abs(·) → Low-pass filtering → envelope

rectification

1) full wave: abs(·)

2) half wave: if s<0, s=0

2. Use filtering to ... -- Summary

① Generate speech-shaped noise

1) generate long-term spectrum of speech signal
2) generate filter coefficients based on speech signal's power spectrum density (psd) **fir2**
3) perform filtering on white noise signal

② Extract speech envelope (包络) waveform

1) Pass the 'abs(sig)' through a lowpass filter **butter**

3. Adjust Signal Intensity

- Signal Energy

$$E = \sum_{n=n_1}^{n_2} |x[n]|^2$$

  - $E \rightarrow a \times E, |x[n]| \rightarrow a^2 \times |x[n]|$
  - In Matlab,
    - $\text{norm(sig)} = \sqrt{\sum_{n=n_1}^{n_2} |x[n]|^2} = \sqrt{E_{sig}},\ E_{sig} = \text{norm(sig)\^{}2}$
    - n = norm(v) returns the Euclidean norm of vector v. This norm is also called the 2-norm, vector magnitude, or Euclidean length.

① Normalizing signal energy

- Energy normalization
  - y=y/norm(y); % then E(y) is normalized to 1
  - y=y/norm(y)*3; % then E(y) is normalized to 9 (= 3^2)
- Energy adjustment
  - y=y/norm(y)*norm(x); % then E(y) = E(x)
  - y=y/norm(y)*2*norm(x); % then, norm(y)=2norm(x), E(y)=4*E(x)

```
x = randn(1000,1)*rand(1,1); % generate 2 random signals with random norms
y = randn(1000,1)*rand(1,1);
y=y/norm(y)*2*norm(x);  % norm(y)=2norm(x), E(y)=4*E(x)
norm(y)/norm(x)          % norm(y)/norm(x) = 2
norm(y)^2/ norm(x)^2  % E(y)/E(x) = 4
```

② Adjusting signal-to-noise ratio (SNR) level

- Signal-to-noise ratio (SNR)

$$SNR = 10log_{10}\left(\frac{E_{signal}}{E_{noise}}\right) = 20log_{10}\left(\frac{\sqrt{E_{signal}}}{\sqrt{E_{noise}}}\right)$$

  - E is energy, and SNR in dB scale
  - In Matlab:
    - SNR=20*log10(norm(sig)/norm(noise)); % sig and noise must have the same length

$\sqrt{E}$

- How to adjust SNR when adding noise to a noise-free signal?
  1. Adjust the energy of noise to target value
     - $E_{signal}/E_{noise} = 10^{\frac{SNR}{10}} \Rightarrow E_{noise} = f(SNR, E_{signal})$, target
     - noise = noise/norm(noise) * (?)
  2. Verify the resultant SNR after adjusting the noise energy
     - norm(sig)^2/norm(noise)^2 = Specified SNR ?
  3. sigwithnoise = sig+noise; % sig and noise must have the same length
     - sigwithnoise has the specified SNR

More on audio file ...

- Read audio file ('*.wav' in this lab)
  - [y, fs] = audioread(filename)
    - y: signal, fs: sampling rate in Hertz
- Play sound
  - sound(y, fs)
  - Or,
    - player = audioplayer(y, fs)
    - play(player)
- Save audio file
  - audiowrite(filename,y,fs)

- Record sound using PC-based audio input device
  - recObj = audiorecorder(fs, nbits, NumChannels)
  - recordblocking(recObj, length)
    - nbits: the number of bits per sample used to encode the data (e.g., 8, 16, 32)
    - NumChannels: number of channels
    - length: recording length in seconds

- An example

```
fs = 11025; nbits = 8; length = 5;
% Record your voice for 5 seconds.
recObj = audiorecorder(fs,nbits,1);
disp('Start speaking.')
recordblocking(recObj, length);
disp('End of Recording.');
% Play back the recording.
play(recObj);
% Store data in double-precision array
myRecording = getaudiodata(recObj);
subplot(2,1,1); plot(myRecording); % Plot the waveform.
audiowrite('Myvoice.wav', myRecording, fs)
[sigread,fsread] = audioread('Myvoice.wav');
subplot(2,1,2); plot(sigread); % Plot the waveform.
```

See Matlab documentation for more details

Lab 5 Assignment

1. Generate a speech-shaped noise (SSN), and plot the spectra of the speech signal and SSN (e.g., use Matlab function ‘psd’ or ‘periodogram’).
    a. generate long-term spectrum of speech signal
    b. generate filter coefficients based on the psd of above speech signal (fir2)
    c. generate white noise (how many points?)
    d. perform filtering on white noise signal

Lab 5 Assignment

2. Read a speech signal x(t), adjust the SNR (x(t) to the above SSN in last part) to -5 dB, let $y = x + SSN$, and normalize the energy of y in relative to x(t), i.e., modify the energy of y so that it equals to the energy of x.

   a. Adjust the intensity of SSN so that

   Be sure to verify SNR is -5 dB

$$SNR = 20log_{10}(norm(x)/norm(SSN)) = -5$$

   b. Adjust the intensity of y so that

$$norm(y) = norm(x) \Leftrightarrow Energy(y) = Energy(x)$$

Lab 5 Assignment (cont.)

3. Extract speech envelope
   a. Full wave rectification : y =abs(y)
   b. Low-pass filtering (butter)
      ① With 2^{nd}-order low-pass filter and cutoff frequency f_{cut}= 100, 200, and 300 Hz. Plot these three envelope waveforms in one plot, and describe the difference among them.
      ② With 2^{nd} and 6^{th}-order low-pass filter and cutoff frequency f_{cut}= 200 Hz. Plot these two envelope waveforms in one plot, and describe the difference between them.

Assignment for lab 5 (cont.)

- Two *.wav files for lab 5 and the coming Project 1
  - ‘C_01_01.wav’ & ‘C_01_02.wav’
- For lab 5, use the files above or record speech signal by yourself via Matlab
- Submit your lab 5 report + codes onto Blackboard before 10:00 am Nov. 25th

- Any questions?