

# Lab1 DSP Tools Review

---

张旭东 12011923

## 1. Introduction

Speech signal processing is a general term for various processing technologies, such as speech production process, statistical characteristics of speech signals, automatic recognition of speech, machine synthesis, and speech perception. Because modern speech processing technology is based on digital computing, and with the microprocessor, signal processor or general purpose computer to achieve, it is also called digital speech signal processing. This lab is actually a reviewing lab which reviews some DSP tools to be used later in speech signal processing, including filter design, varying sample rate, and so on.

The whole lab is separated into four sections and each section is corresponding to the questions in lab1 manual. The detail will be introduced in each sections.

## 2. Lab results and analysis

### 2.1 problem1

The exercise mainly review how to read and play a speech file and plot a certain number of sample value using MATLAB.

Firstly, the speech file named `mhint_01_01.wav` is asked to be loaded into the workspace and the sentence heard is:

他刚才说话时吞吞吐吐

Then, `N` sample values of the speech file are asked to be plotted, starting at sample `fstart` and ending with sample `fstart+N-1`, `M` samples of speech per line, with up to four lines of plotting per page. The idea is that total number of lines and that of pages are calculated firstly. Then, the started sample point and the end sample point of each line of each page is below:

```
lines=floor(N/M);% completed lines
```

```

disp(lines);
pages=floor(lines/4);%completed pages
disp(pages);
restoflines=lines-4*pages;
restofsamples=N-M*lines;
disp(restofsamples);
for i=1:pages
    figure
    for j=1:4
        startsampleofeachline=fstart+4*M*(i-1)+M*(j-1);
        endsampleofeachline=fstart+4*M*(i-1)+M*j-1;
        cut=x(startsampleofeachline:endsampleofeachline);
        subplot(4,1,j);

        plot(startsampleofeachline/fs:1/fs:endsampleofeachline/fs,cut);
        xlabel('t(s)');
        ylabel('value');
        axis([startsampleofeachline/fs,endsampleofeachline/fs
min(x) max(x)]);
    end
end
end

```

If there are samples left, a new figure and a newline are created to plot these samples.

```

if restoflines~=0
    figure
    for i=1:restoflines
        startsample=fstart+4*M*pages+M*(i-1);
        endsample=startsample+M-1;
        cut=x(startsample:endsample);
        subplot(4,1,i);
        plot(startsample/fs:1/fs:endsample/fs,cut);
        xlabel('t(s)');
        ylabel('value');
        axis([startsample/fs endsample/fs min(x) max(x)]);
    end
end

if restofsamples~=0

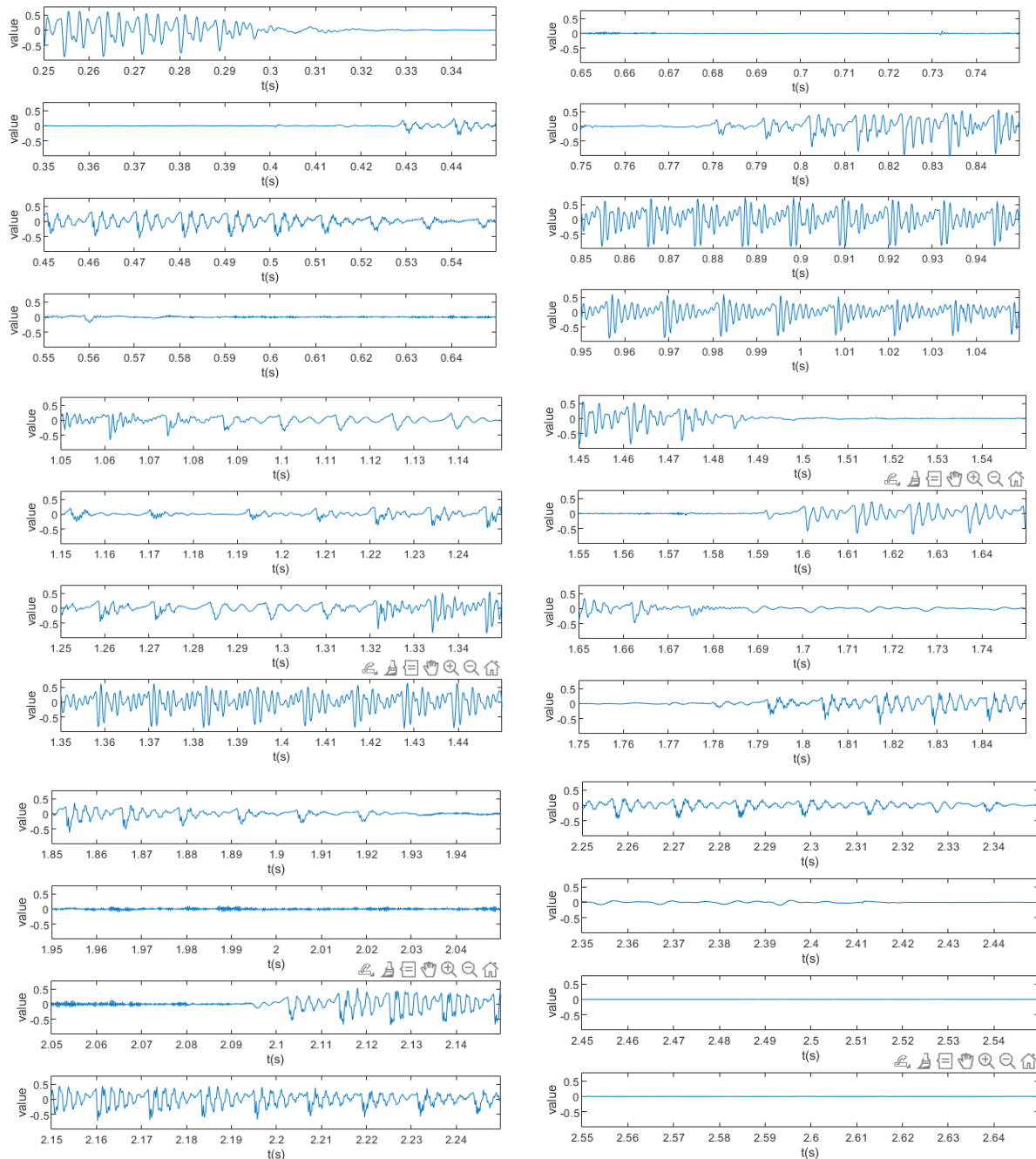
```

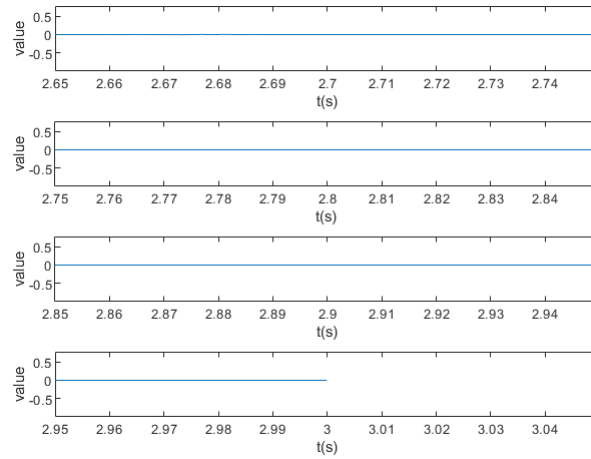
```

subplot(4,1,restoflines+1);
plot((fstart+N-restofsamples)/fs:1/fs:(fstart+N-
1)/fs,x(fstart+N-restofsamples:fstart+N-1));
xlabel('t(s)');
ylabel('value');
axis([(fstart+N-restofsamples)/fs (fstart+N-restofsamples+M-
1)/fs min(x) max(x)]);
end

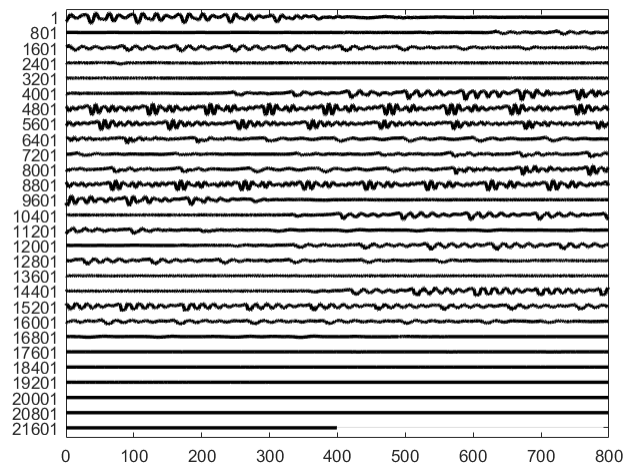
```

$N = 22000$ ,  $fstart = 2000$ ,  $M = 100$  are chosen and the result for speech file `s5.wav` are follows:





The problem 2.33 ask to plot the speech file with the same parameters but plotting all the samples in a contiguous format on a single page using function `strips_modified.m`. The result are as following:



## 2.2 problem2

The purpose of problem2 is to review the design of filter. The requirement is to design an order  $n$  low-pass digital **Butterworth** filter with normalized cutoff frequency  $W_n = \frac{f_{cut}}{0.5f_s}$ , where  $f_{cut}$  is a low-pass cut frequency and  $f_s$  is sampling rate for the digital signal.

In MATLAB, there is a function `butter` to achieve the **Butterworth** filter.

```
[b,a]=butter(n,wn)
```

**b,a**: Transfer function coefficients, specified as vectors.

**n**: the order of low-pass digital **Butterworth** filter.

**wn**: normalized cutoff frequency.

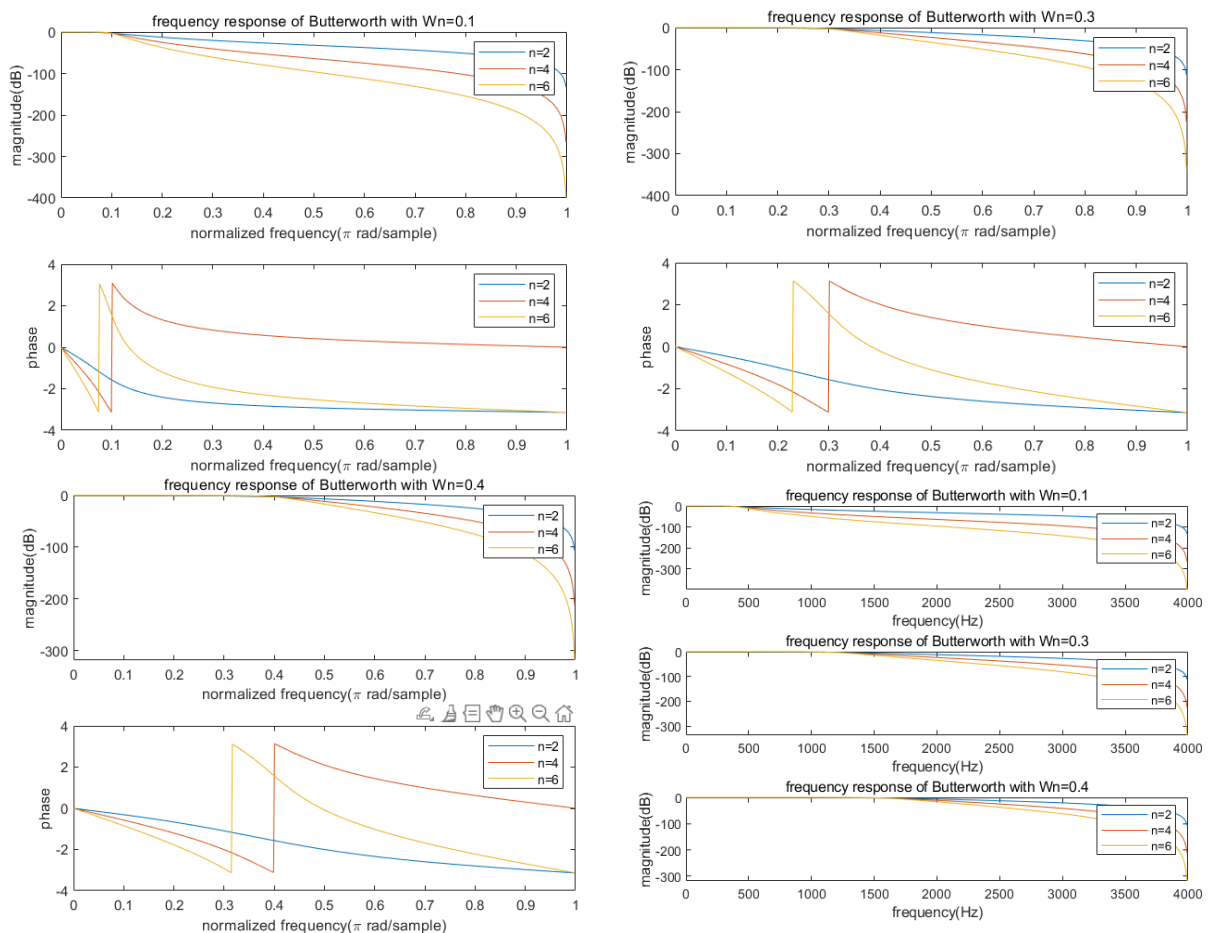
In MATLAB, there is a function **freqz** to calculate the frequency response of digital filter.

```
[h,w]=freqz(b,a)
```

**h**: frequency response, returned as a vector.

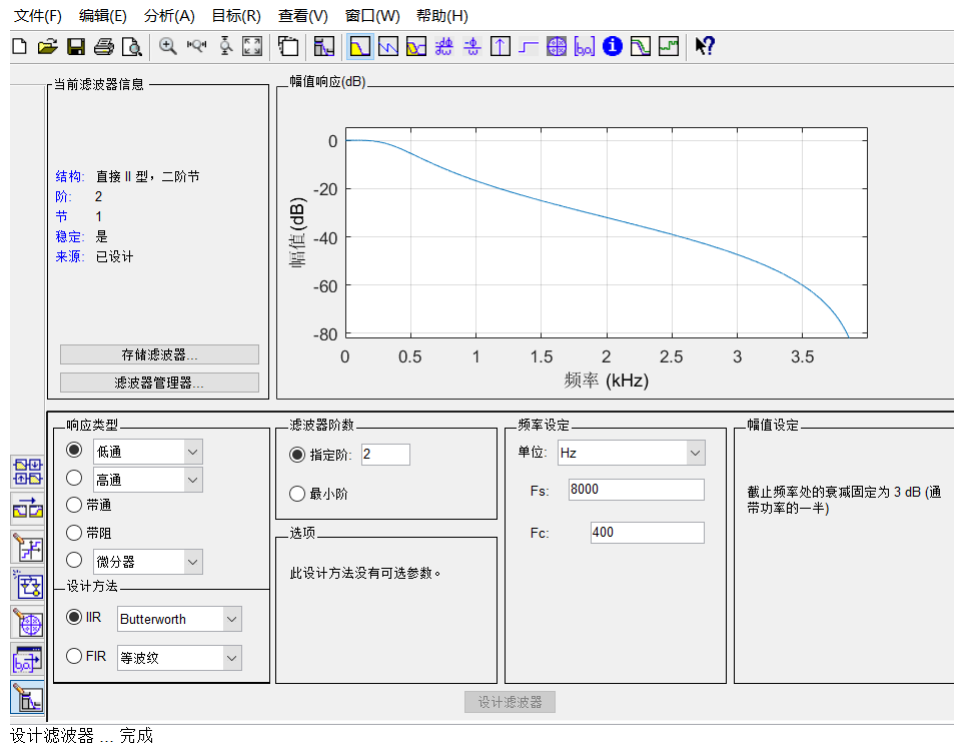
**w**: angular frequencies, returned as a vector.

Firstly, the frequency responses of the **Butterworth** filter with order **n** from 2, 4, 6 and **wn** from 0.1, 0.3 and 0.4 are as follows:



From the above four pictures, the normalized cutoff frequency **wn** determines the position where the magnitude start attenuating. At the same **wn**, with the order **n** increasing, the rate of attenuation becomes faster and faster when the normalized frequency increases.

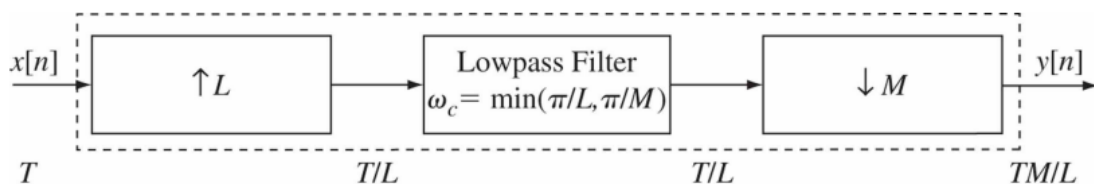
Next, the function **fdatool.m** is used to design the filters described above.



Also, click the button **file** and the option of generating MATLAB code is helpful and convenient to design the digital filter.

## 2.3 problem3

The purpose of problem3 is to be familiar with change the sample rate of original signal and observe the influence of sample rate to a signal. The theoretical principle of changing sample rate is as follows:



Firstly do the interpolation to increase the sampling rate by factor **L** and then pass a low pass filter with cutoff frequency  $\min(\frac{\pi}{L}, \frac{\pi}{M})$ , which is used to prevent aliasing in frequency domain during decimation. After that, do decimation to decrease the sample rate by factor **M**.

The sample rate of original signal is 16000 **Hz**. In MATLAB, there is a function **resample** to achieve the change of sample rate of original signal.

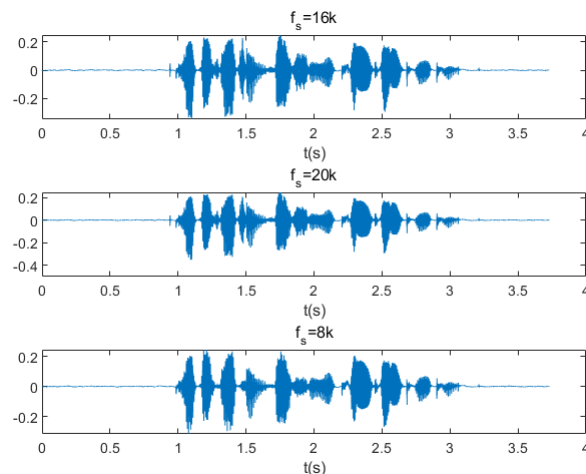
```
y=resample(x,p,q)
```

**x**: the original input sequence or signal.

**p/q**: the ratio of the target sample rate to the original sample rate.

**y**: the resampled sequence or signal.

The results of changing the sample rate of original signal to 20 kHz and 8 kHz are as follows:



In terms of hearing, the signal whose sample rate is 16 kHz is almost with that whose sample rate is 20 kHz. However, the signal whose sample rate is 8 kHz is different to the above two. The above two sounds clearer and has a higher pitch than the signal whose sample rate is 8 kHz. From the above picture, it is clear that some samples occurred in the original signal don't appear in the signal whose sample rate is 8 kHz.

What's more, there are also another function `interp1.m` to achieve the change of sample rate of original signal.

```
vq=interp1(x,v,xq)
```

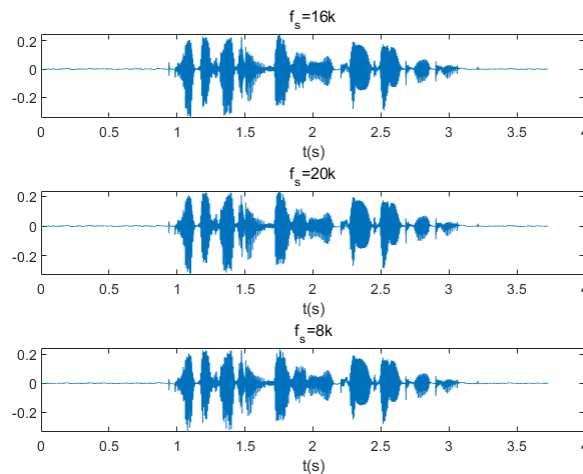
**vq**: resampled sequence or signal.

**x**: contains the sample points

**v**: contains the corresponding values,  $v(x)$ .

**xq**: contains the coordinates of the query points.

The results of changing the sample rate of original signal to 20kHz and 8kHz using `interp1.m` are as follows:



The result using `interp1.m` is a little different to the result using `resample.m`.

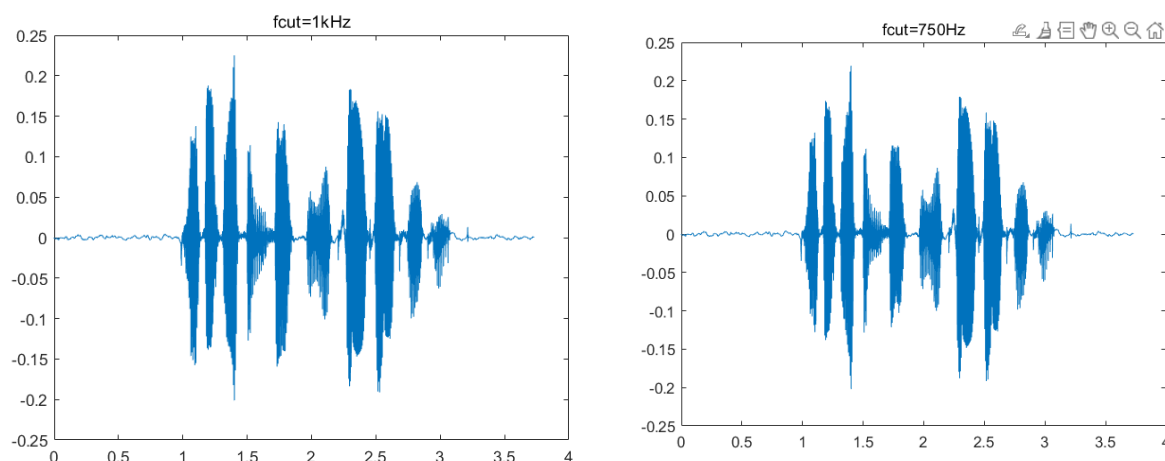
That is because the former uses linear interpolation to return the insertion value of one dimensional function at a specific query point while the latter applies an FIR Antialiasing lowpass filter to the original signal and compensates for the delay introduced by the filter.

## 2.4 problem4

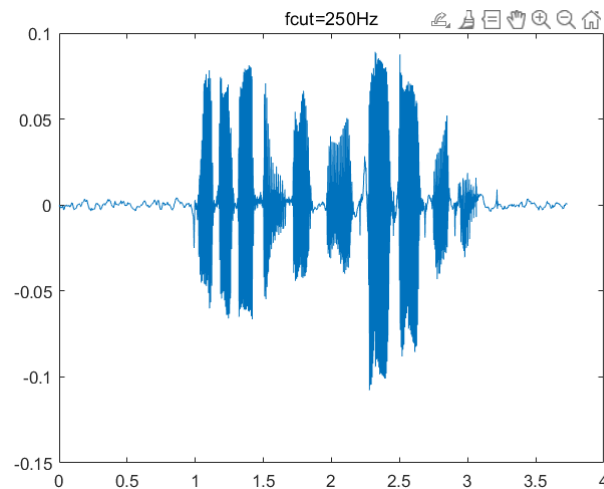
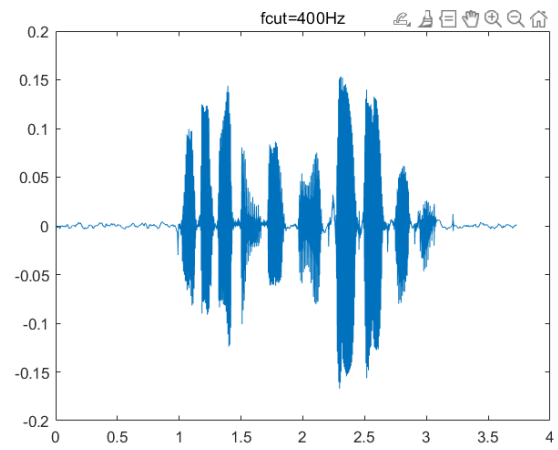
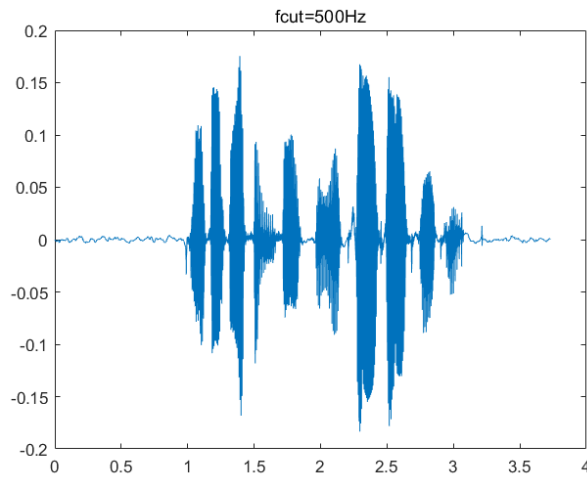
The work of problem4 is to process the signal with low-pass filtering and high-pass filtering and describe their affect on speech understanding.

### 2.4.1 low-pass filtering

In this part, `Butterworth` filter is chosen and its order is 6. The result is shown as below:



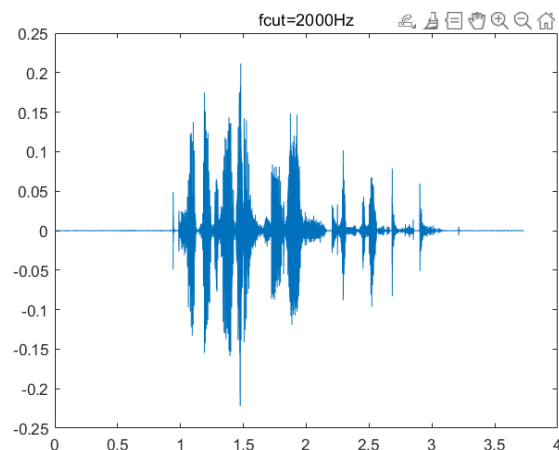
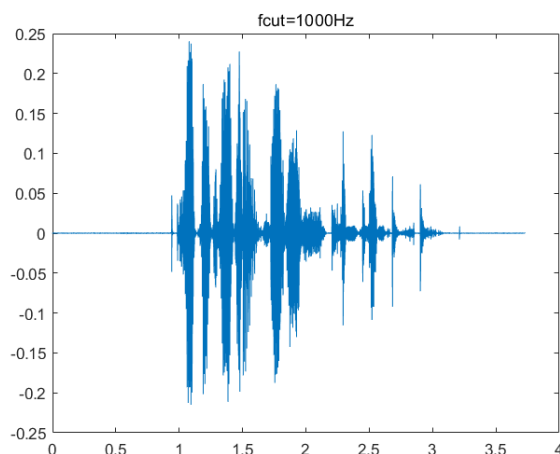


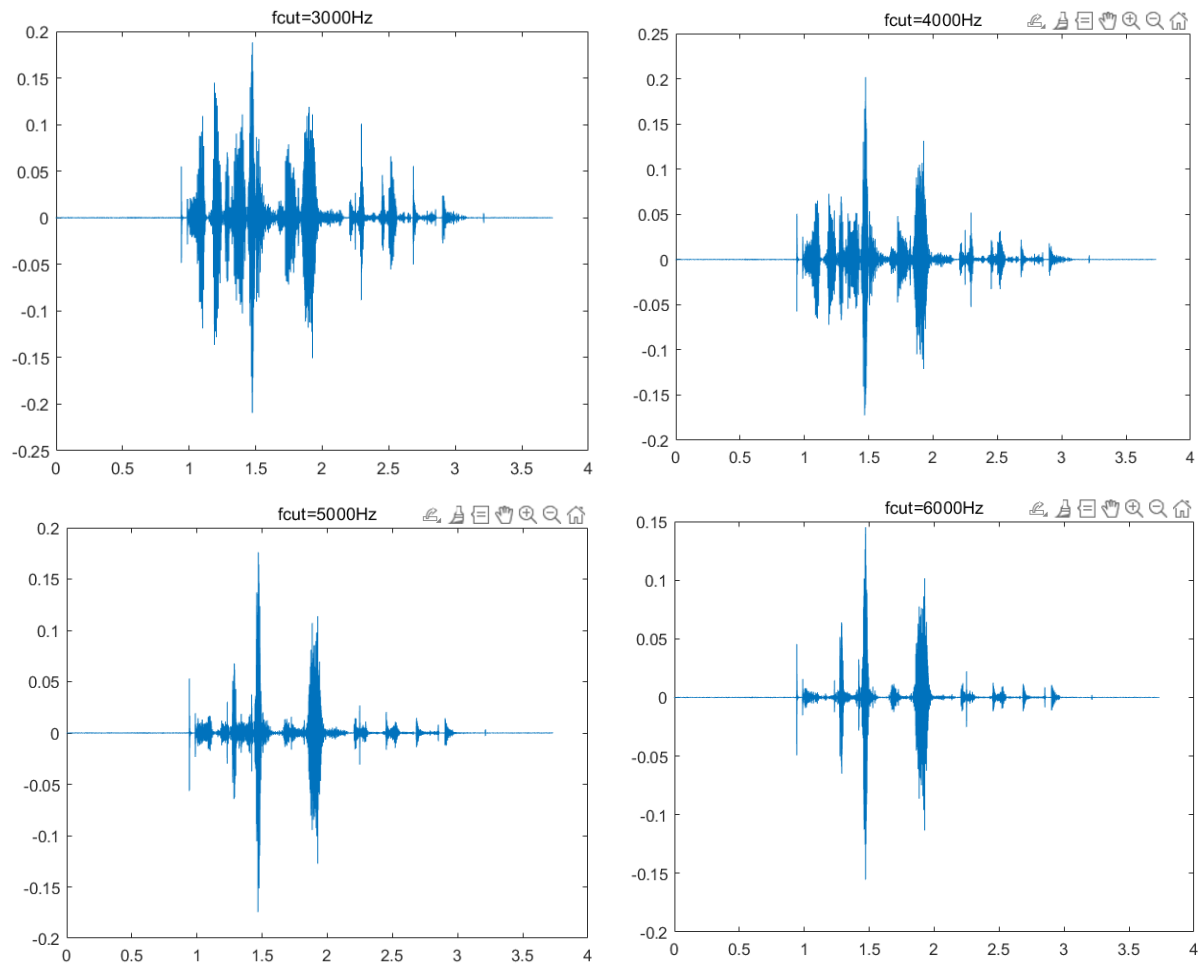


From all the above five figures, it is clear that the average magnitude is getting lower and lower and the energy of wave is less and less with the cutoff frequency decreasing. In terms of hearing, the lower cutoff frequency, the harder to distinguish the speech voice and the lower the voice is.

## 2.4.2 high-pass filtering

In this part, **Butterworth** filter is chosen and its order is 6. The result is shown as below:





The results are quite similar to the ones in the previous part. We get less energy and less magnitude for the overall wave. For high pass filtering, the change is little with the cutoff frequency increasing because these part contains more information as we expected.

Generally from the 11 test, we can guess the most important frequency is around 1 kHz to 3 kHz.

### 3. Conclusion

The main purpose of this lab is to review some basic DSP tools in speech signal processing, including filter design, varying sampling rate. The main harvest is to use `fdatool.m` to design the filter conveniently.