Southern University of Science and Technology

Speech Signal Processing

Lab 5 Report

11510478 郭锦岳

Question 1

I write the function myCepstrum to let it able to choose generating complex or real cepstrum. When generating complex ones, I use unwrap() to get the unwrapped angle of the result of fft.

Code:

```
function [ output ] = myCepstrum( input, cr )
%MYCEPSTRUM computes the complex or real cepstrum of the input
signal.
% cr: complex or real. 'C' for complex, 'R for real.

fftw('dwisdom',[]);% clear the current fft setting
fftw('planner','measure');% set fft method to measure
H = fft(input);
if cr == 'C'
    H_log = log(abs(H)) + 1i*unwrap(angle(H));
elseif cr =='R'
    H_log = log(abs(H));
end
output = real(ifft(H_log));

end
```

Then manually generate the signal $x_1[n] = a^n u[n], a = 0.5$

```
x1 = 0.5.^(1:128);
x1_c = myCepstrum(x1,'C');
```

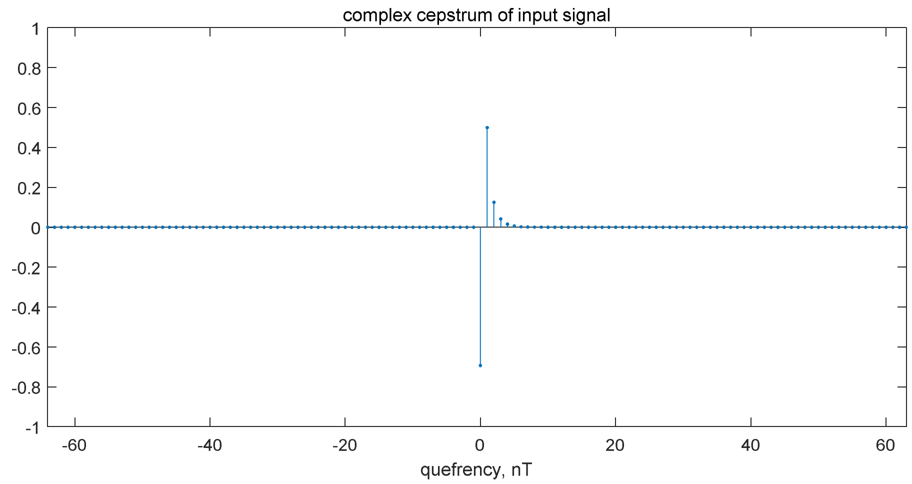

Figure 1, fft number = 128

Since the signal is aperiodic, its cepstrum does not show any periodic component. As told in class, the cepstrum decays along with quefrency.

Question 2

For question 2, I use the same function as in question 1, and subplot the results in a single figure.

Code:
```
x2 = [1, zeros(1,99), 0.85];
subplot(3,1,1);stem(x2);title('original
signal');xlabel('Time');ylim([-1,1]);
subplot(3,1,2);stem(myCepstrum(x2,'C'));title('complex
cepstrum');xlabel('Quefrency');axis([0,2048,-1,1]);
subplot(3,1,3);stem(myCepstrum(x2,'R'));title('real
cepstrum');xlabel('Quefrency');axis([0,2048,-1,1]);
```

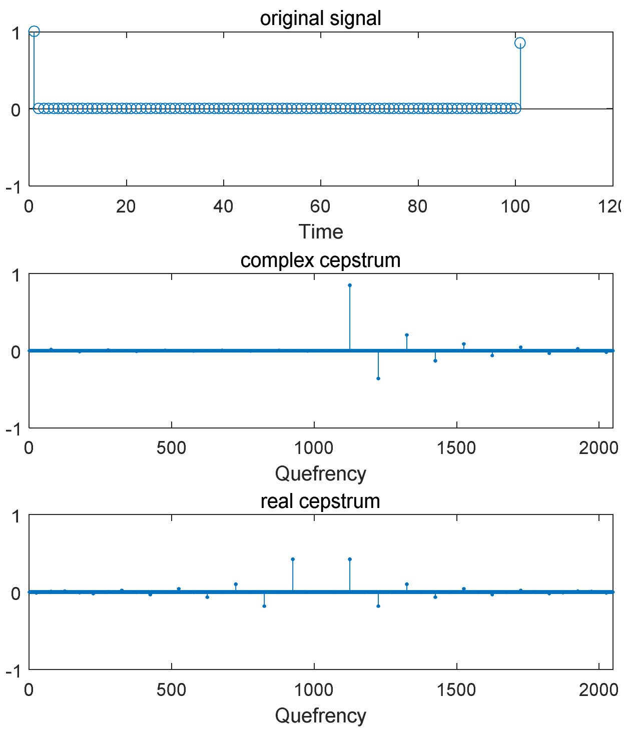

Figure 2, fft number = 2048

For the topic of the number of samples of fft, even though my signal length in question 1 and 2 is at the same level, the cepstrum of x1 only need 128 samples to get rid of aliasing. However, x2 needs at least 2048-point fft to clear aliasing. It seems that the frequency components of the signal can also affect the cepstrum result.

Actually, since the process of computing cepstrum needs both fft and ifft operation, if the temporal signal has more high-frequency component, the relative spectrum could be also with high-frequency component. Therefore, when doing ifft, you need higher sampling frequency to avoid aliasing in frequency domain.

Question 3

The program for question 3 is a little bit complicating.

When doing quefrency liftering, the voiced signal has an impulse at $n = 150$, so I set the cut-off quefrency at $n = 140$.

Code:

```
function [  ] = speAnal( file_name, start, nsamples )
%SPEANAL answer of lab5q3
% file_name: string of the file name
% start: the staring sample of the signal in file
% length: length of signal in sample
signal = audioread(file_name);
signal = signal(start:start+nsamples-1);
subplot(2,2,1);plot(signal);title('windowed signal');ylim([-
1,1]);
signal = signal.*hamming(nsamples);% use hamming window

spectrum = 20*log(abs(fft(signal)));
cepstrum = myCepstrum(signal,'R');
% do liftering by a 25% low-pass lifter, then turn back to log
spectrum,
% then times 10 to dB scale
spectrum_liftered =
real(20*(fft(cepstrum(1024:1024+100),2048)));

[spectrum,x1] = fftshift(spectrum);
subplot(2,2,2);stem(x1,
spectrum,'Marker','none');title('spectrum (dB)');xlim([-
pi/2,pi/2]);
subplot(2,2,3);stem(cepstrum,'Marker','none');title('real
cepstrum');axis([0,2048,-1,1]);
[spectrum_liftered,x2] = fftshift(spectrum_liftered);
subplot(2,2,4);stem(x2,spectrum_liftered,'Marker','none');titl
e('low pass liftered spectrum (dB)');xlim([-pi/2,pi/2]);

end

function [outSpec, outx] = fftshift(inSpec)
% shift the fft result and generate the modified x value
inSpec = inSpec';
L = length(inSpec);
outSpec = [inSpec(round(L/2):L), inSpec(1:round(L/2))];
outx = (-L/2:L/2)*pi/L;

end
```

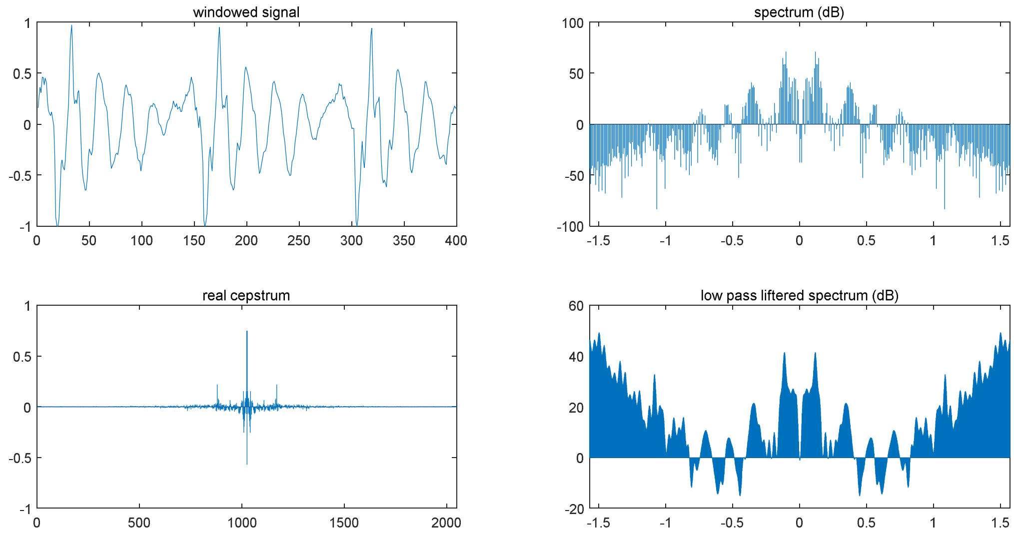

Figure 3, voiced speech

For the voiced speech, we can see it contains more low frequency components, and can clearly see the f1, f2 and f3. Also, the f0 can be seen as the 'sampling' of the spectrum.

In the real cepstrum, the impulse at $n = 150$ is the component of the excitation signal. We cut the cepstrum and transfer it back into frequency domain. The liftered spectrum becomes smoother, since the excitation has been removed. Now the liftered spectrum is the convolution of glottal pulse, vocal tract and radiation load.

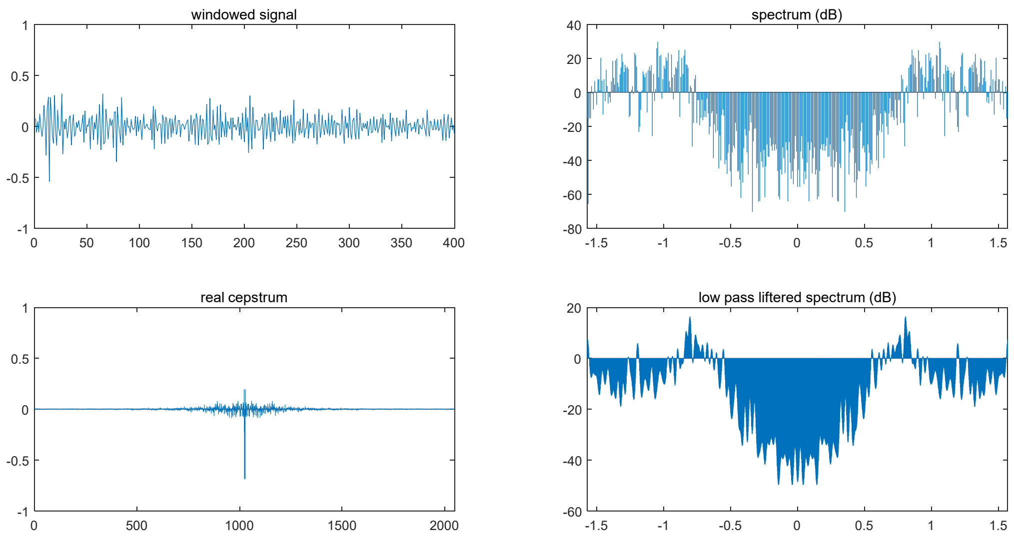

Figure 4, unvoiced speech

The unvoiced speech contains more high-frequency components. After liftering, the spectrum also becomes smoother. Another significant effect is that the high frequency components are reversed, which is the same with the voiced speech. Further study is needed to explore this phenomenon and explain the reason.