Southern University of Science and Technology

Speech Signal Processing

Lab 1 Report

11510478 郭锦岳

1.1 Recording and analysis using matlab

First, we use matlab to record the vowels, /a:/, /i/, and /u/ respectively

```
% Record your voice for 2 seconds.

recObj = audiorecorder;
disp('Start speaking.')
recordblocking(recObj, 2);
disp('End of Recording.');

aa = getaudiodata(recObj);%save audio file(take 3 times
for /a:/, /i/, /u/)
audiowrite('aa.wav', aa, 8000);%write audio file into
system(take 3 times)
```

Then use the function DTFT to generate the spectrum of these three vowels:

```
aa = audioread('aa.wav');
[aa_spec, w] = DTFT(aa, length(aa));
plot(w*8000/(2*pi), 10*log(abs(aa_spec)));title('spectrum
of vowel /a:/ (dB)');axis([0 4000 -10 80]);
```

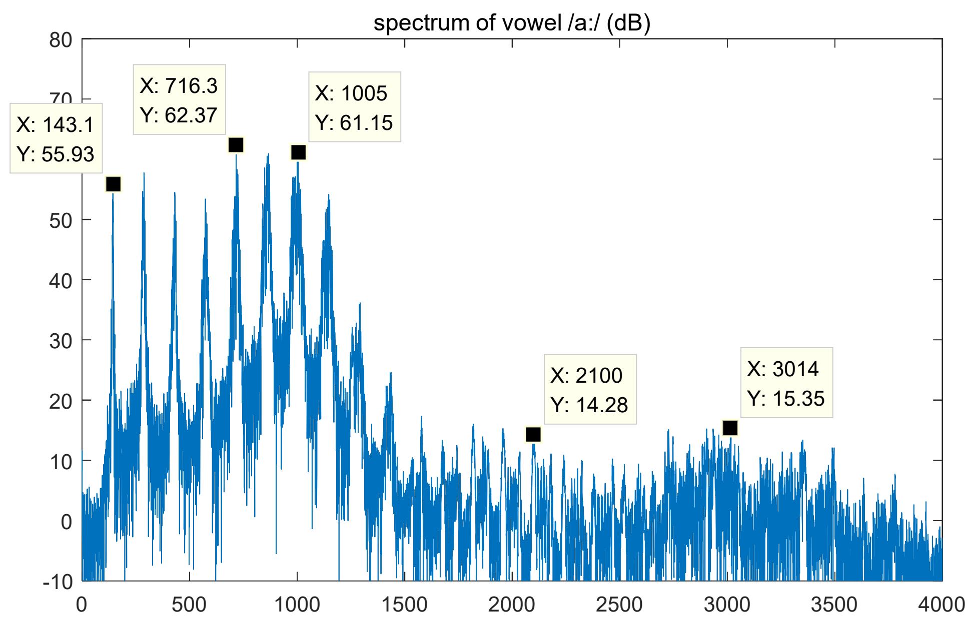

Figure 1. 1

In figure 1.1, we can see that the fundamental frequency f0 is 143.1 Hz. The formants of this specific vowel are f1 = 716.3Hz, f2 = 1005Hz, and f3 = 2100Hz, which is similar to the result on the text book.

In the same way, we record and generate the spectrum of /i/ and /u/, as shown in Figure 1.2 and 1.3 below:

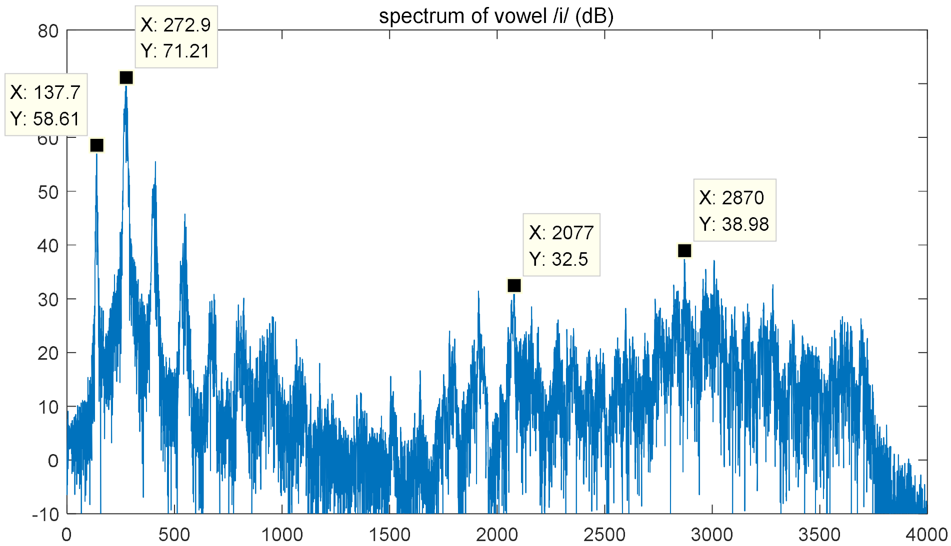

Figure 1. 2

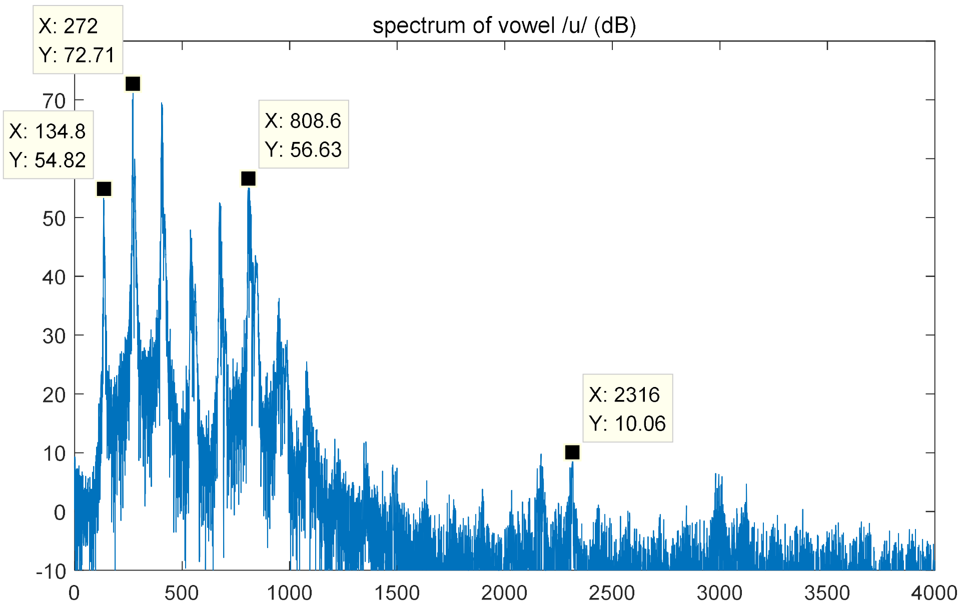

Figure 1. 3

We can see that f0 is about the same for all three spectrums, since they are all record of my own voice. However, the formants changes, which can show the difference of phonemes.

1.2 Analysis using praat

In praat we can easily generate the spectrum by a simple click:

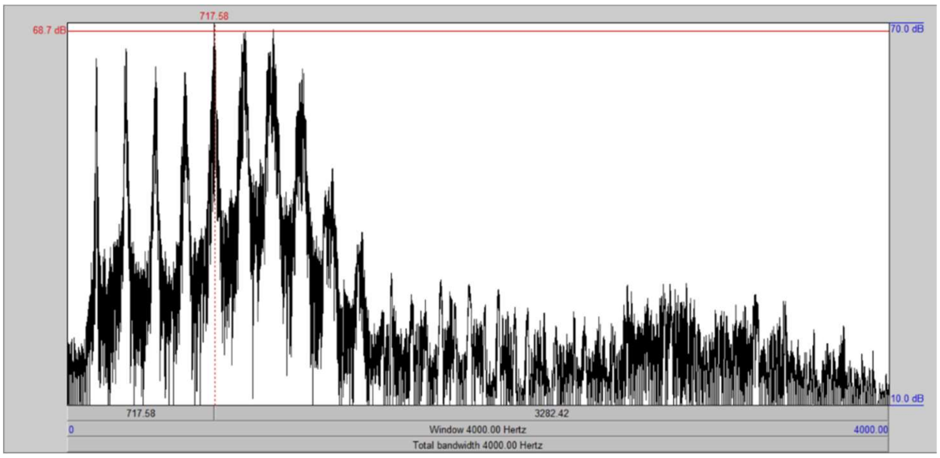

Figure 1. 4 spectrum of /a:/

In figure 1.4, the cursor is on the first formant, showing its frequency of 717.58Hz, which is the same as the result in figure 1.1.

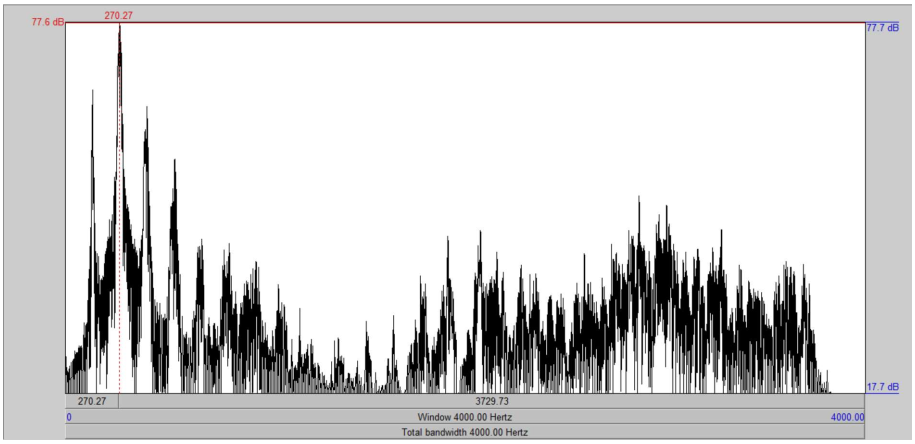

Figure 1. 5 spectrum of /i/

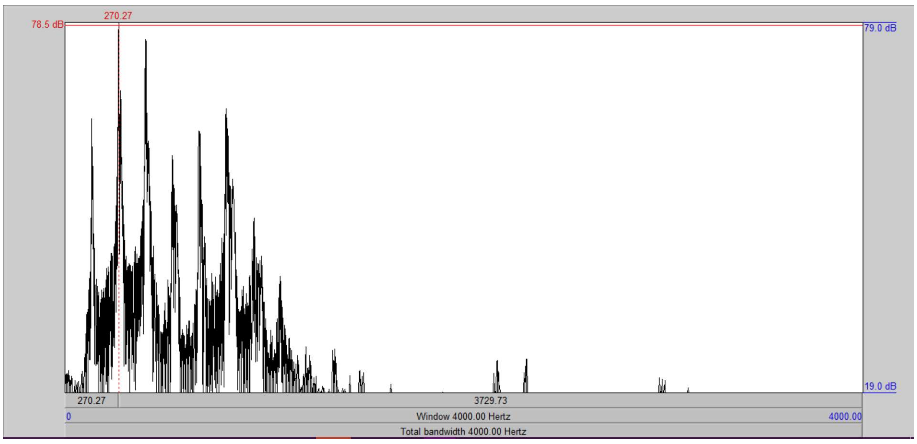

Figure 1. 6 spectrum of /u/

2.1 Speech production using Matlab

Following the instruction of lab1, I wrote the script to synthesis the sound.

In this code, we use the record of /u/ as example.

```
% take the envelope of the spectrum, only YUPPER is used
[YUPPER,YLOWER] = envelope(abs(u_spec),50,'peak');
w = w*8000/(2*pi);
plot(w, 10*log(abs(YUPPER)));title('envelope of spectrum
of vowel /u/ (dB)');axis([0 4000 -10 80]);


frequency = 50:50:4000; % take frequency samples
t = 1:16000; % generate time series

% calculate the index of the referencing frequency
frequency_YUPPER =
round(length(w)/8000*frequency+(length(w)/2));
YUPPER = abs(YUPPER);

% synthesis the waveform by multiplexing pure tones with
respecting amplitude
u_synth = zeros(1,16000);
for i = 1:length(frequency)
    u_synth = u_synth + cos(2*pi*frequency(i)/8000*t) *
YUPPER(frequency_YUPPER(i));
end

% normalize the waveform amplitude
m = max(u_synth);
u_synth = u_synth/m;

[u_syn_spe, w] = DTFT(u_synth, length(u_synth)); % show
spectrum

soundsc(u_synth); % hear the sound
```

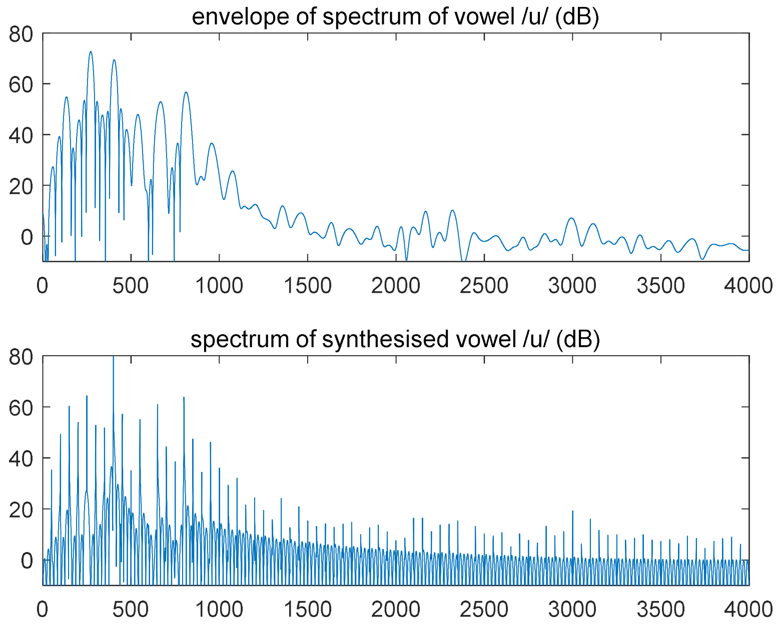

Figure 2. 1

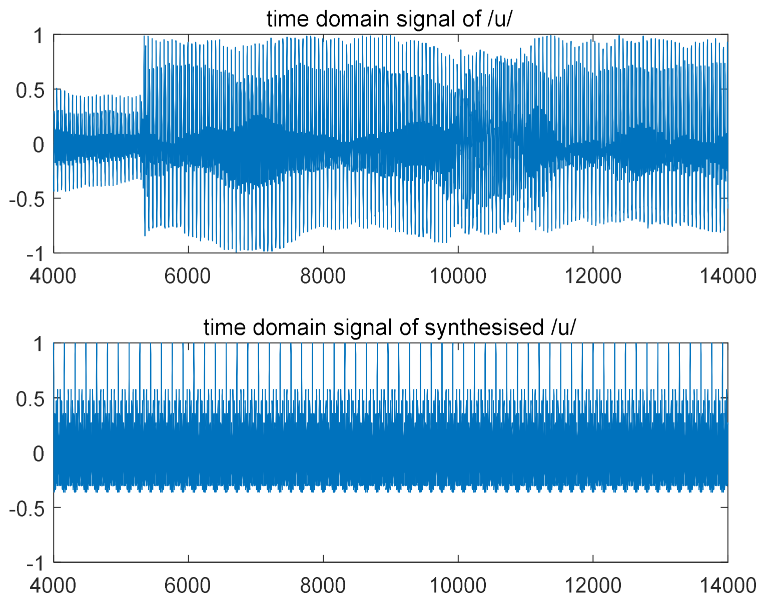

Figure 2. 2

The spectrum of synthesized signal is ideal, having the shape of a sampled spectrum. But the time domain signal, it is hugely different.

When listening at the signal, it can be recognized as the vowel u, but the sound is very unnatural. I think this is due to the absence of adjustments, e.g., critical band, equal LL curves.

Further study is needed to improve my result.