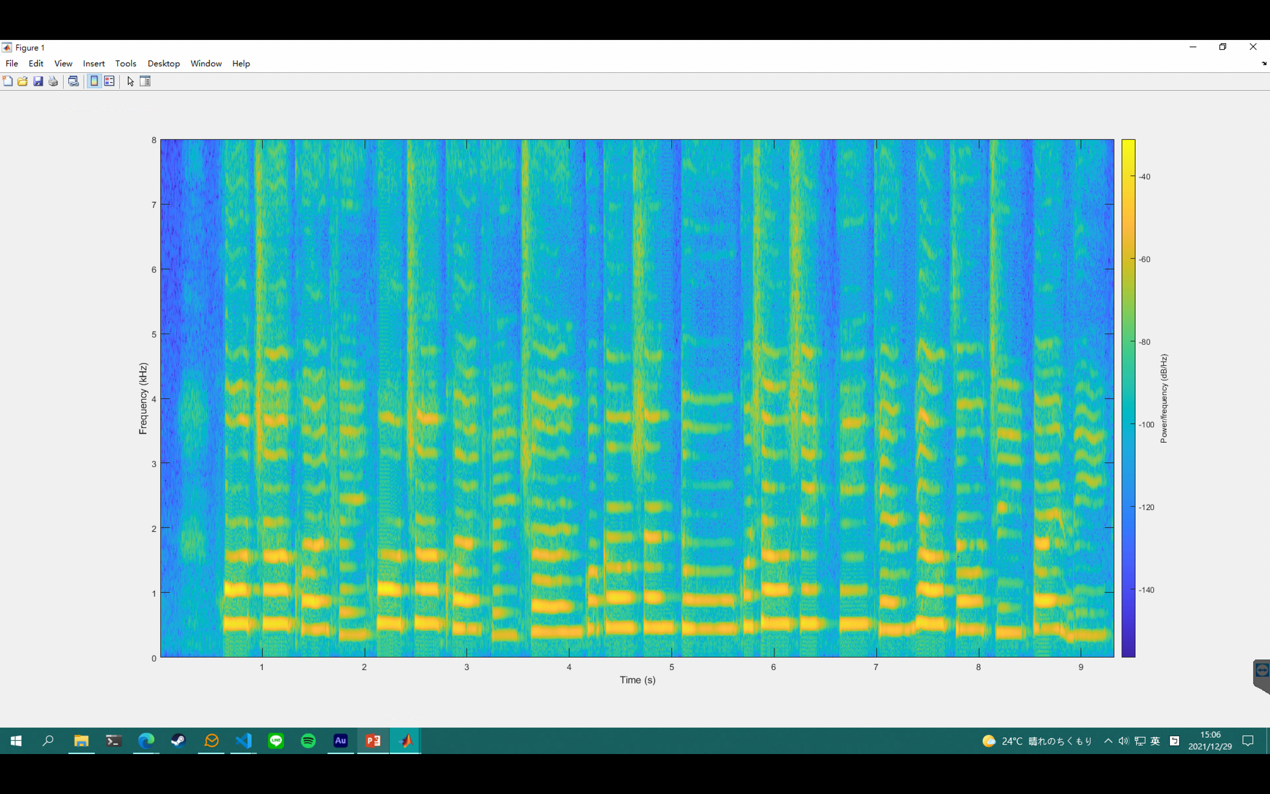
DSP Projects of Chapter 7

NE6101034 AI碩一 柳譯筑

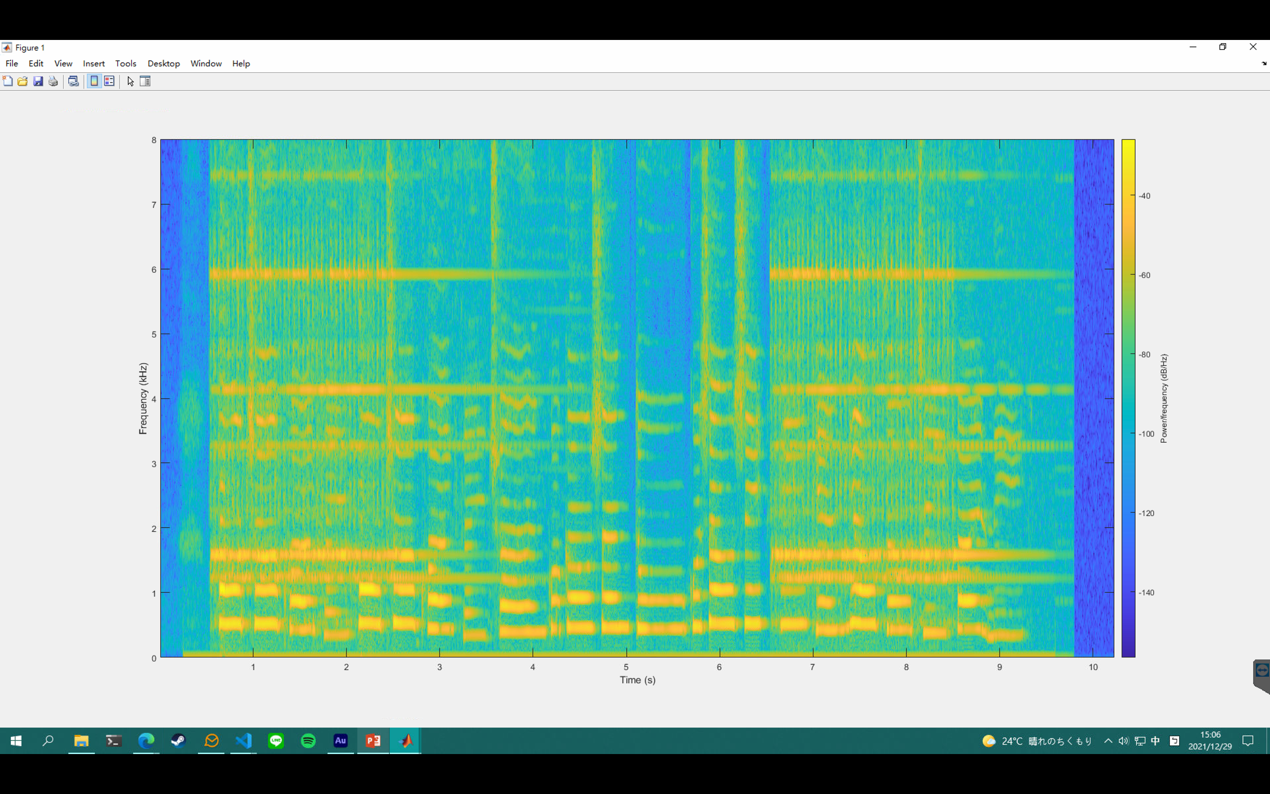
1. Download the two audio signal files with a sampling rate of 16 KHz (one is clean and the other is noisy) from the course Web site and process the signal as follows.

1. Show the spectrogram of the two audio signals

下圖為乾淨的音樂spectrum

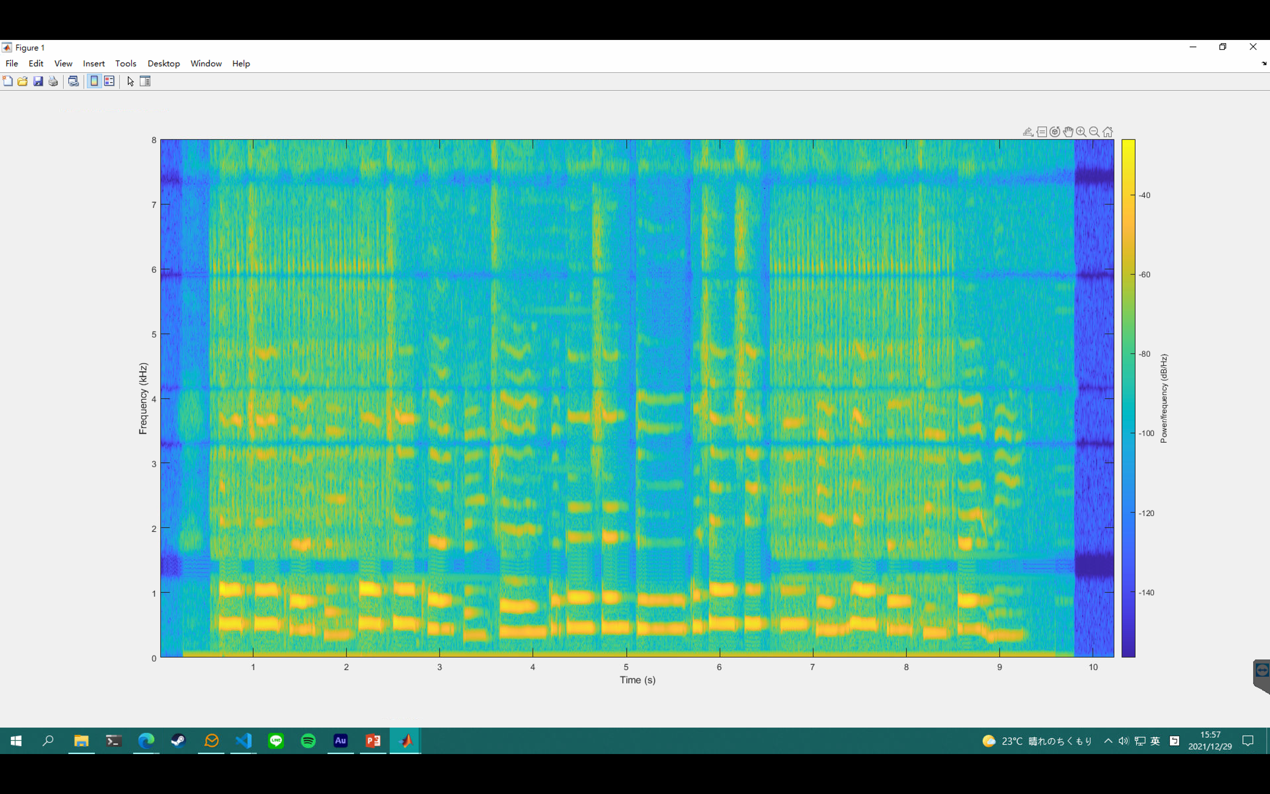


下圖為有雜音的spectrum



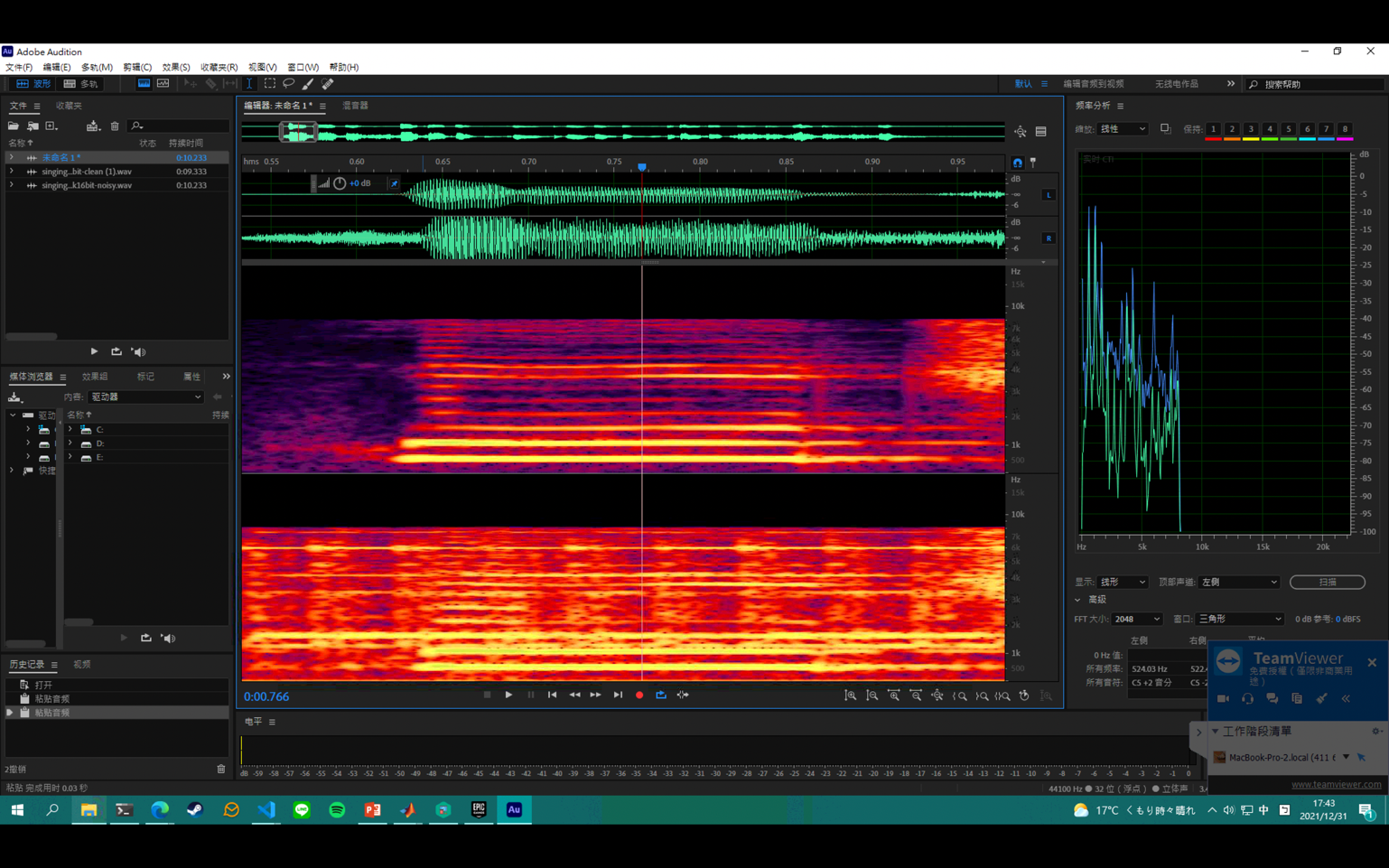
b. Remove the noise. You need to analyze the spectrum of the noise from the two audio files and design a filter to remove the noise.

我使用matlab butterworth bandpass filter嘗試把電話聲的基頻( fundamental frequency f0) 跟 formants(f1 to f4) 濾掉，頻率的數字我是直接放大圖表用肉眼去判斷，把3200~3400, 4080~4230, 5800~6000, 7230~7600 bandpass filter掉。

濾完的 spectrum長這樣：

不過還是會有比較小聲的低音雜音。

原因是因為在低音的部分沒辦法完整濾除掉，如果濾掉這個頻段的聲音會連人聲一起濾掉。



分析這張頻譜可以看到最下面三條能量最強的地方（電話聲），有其中一部分能量跟人聲重疊。

附上程式碼：

[audioIn,fs] = audioread('C:\Users\User\Desktop\Kay\DSP\HW4\singingWithPhoneRing16k16bit-noisy.wav');

[audioIn\_clean,fs] = audioread('C:\Users\User\Desktop\Kay\DSP\HW4\singing16k16bit-clean (1).wav');

%%

windowSize = 256;

windowOverlap = [];

freqRange = 0:fs;

spectrogram(audioIn(:,1), windowSize, windowOverlap, freqRange/2, fs, 'yaxis');

%%

spectrogram(audioIn2(:,1), windowSize, windowOverlap, freqRange/2, fs, 'yaxis');

%%

Wp = 1200/fs; % Passband Frequency

Ws = 1710/fs; % Stopband Frequency

n = 7;

beginFreq = 1150/(fs/2);

endFreq = 1710/(fs/2);

[b,a] = butter(n, [beginFreq, endFreq], 'stop');

audioOut1 = filter(b, a, audioIn);

beginFreq1 = 4080/(fs/2);

endFreq1 = 4230/(fs/2);

[b1,a1] = butter(n, [beginFreq1, endFreq1], 'stop');

audioOut2 = filter(b1, a1, audioOut1);

beginFreq2 = 5800/(fs/2);

endFreq2 = 6000/(fs/2);

[b2,a2] = butter(n, [beginFreq2, endFreq2], 'stop');

audioOut3 = filter(b2, a2, audioOut2);

beginFreq3 = 3200/(fs/2);

endFreq3 = 3400/(fs/2);

[b3,a3] = butter(n, [beginFreq3, endFreq3], 'stop');

audioOut4 = filter(b3, a3, audioOut3);

beginFreq4 = 7230/(fs/2);

endFreq4 = 7600/(fs/2);

[b4,a4] = butter(n, [beginFreq4, endFreq4], 'stop');

audioOut = filter(b4, a4, audioOut4);

spectrogram(audioOut(:,1), windowSize, windowOverlap, freqRange/2, fs, 'yaxis');

p = audioplayer(audioOut, fs);

p.play;