HW4

Q36111281 張偉治

1. 音訊分析

為了設計能在信噪比(Signal-to-noise ratio)和失真率(information rate-distortion theory)之間有良好平衡的濾波器，需要透過比對乾淨的音檔與含有噪音的音檔之間的時頻譜(spectrogram)。

透過以下程式碼載入音檔，並利用 spectrogram 函式對音檔做 SFFT，再將得到的數據進行作圖得到 figure 1。

```
% load data
[y1,fs1] = audioread('singing16k16bit-clean.wav');
[y1s, ~]= size(y1);
s1 = spectrogram(y1, 256);
[y2,fs2] = audioread('singingWithPhoneRing16k16bit-noisy.wav');
[y2s, ~]= size(y2);
s2 = spectrogram(y2, 256);

% spectrogram compare
if plot_flag1==1
    tiledlayout(2,1);

    ax1 = nexttile;
    t1 = linspace(0, y1s/fs1, size(s1,1));
    f1 = linspace(0, fs1/2, size(s1,2));
    imagesc(t1, f1, 20*log10((abs(s1))));xlabel('Samples');
ylabel('Freqency');
    title('clean sound')
    colorbar;

    ax2 = nexttile;
    t2 = linspace(0, y2s/fs2, size(s2,1));
    f2 = linspace(0, fs2/2, size(s2,2));
    imagesc(t2, f2, 20*log10((abs(s2))));xlabel('Samples');
ylabel('Freqency');
    title('noise sound')
    colorbar;
end
```

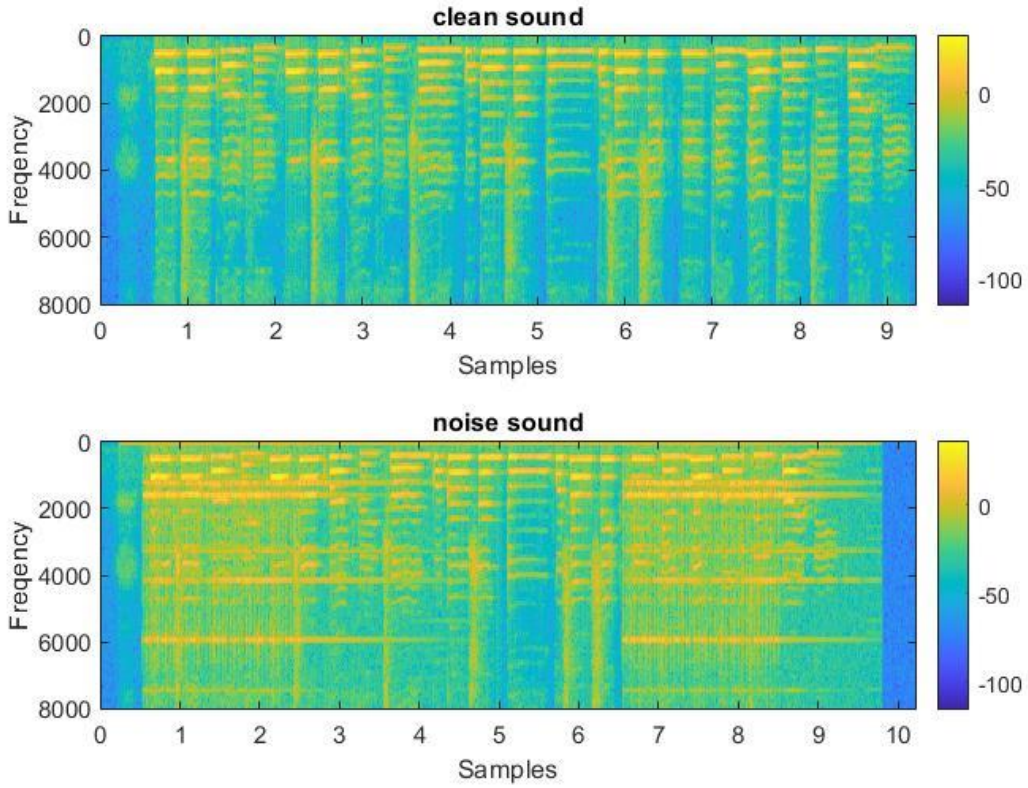

figure 1

從上圖可以看出，在 6000Hz、4000Hz、1800 Hz 與 1600Hz 處，含有噪音的音檔的時頻譜明顯和乾淨音檔的時頻譜明顯不同。高於 2000Hz 的部分可以使用帶通率波將其濾除，因為乾淨音檔的主頻基本落在 200Hz 以下，不容易破壞到人聲泛音的部分。

因為噪音的主頻也在 2000Hz 以下，為了更好的觀察噪音對人聲的影響，引此對全部音檔進行 FFT。程式碼如下所示。

```
% fft compare
if plot_flag2 == 1
    tiledlayout(2,1);

    m1 = length(y1);
    n1 = pow2(nextpow2(m1));
    y1_f = fft(y1,n1);
    f1 = (0:n1-1)*(fs1/n1)/10;
    power1 = abs(y1_f).^2/n1;

    m2 = length(y2);
    n2 = pow2(nextpow2(m2));
    y2_f = fft(y2,n2);
```

```
    f2 = (0:n2-1)*(fs2/n2)/10;
    power2 = abs(y2_f).^2/n2;

    ax1 = nexttile;
    plot(f1(1:floor(n1/2)),power1(1:floor(n1/2)))
    title('clean sound')
    xlabel('Frequency')
    ylabel('Power')

    ax2 = nexttile;
    plot(f2(1:floor(n2/2)),power2(1:floor(n1/2)))
    title('noise sound')
    xlabel('Frequency')
    ylabel('Power')
end
```

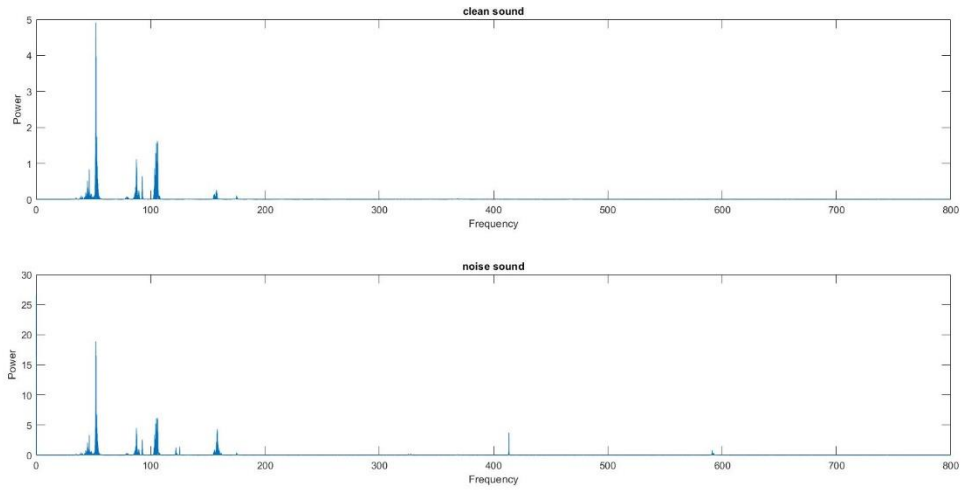

figure 2

從 figure 2 可以發現在頻率低於 2000Hz 時，噪音的聲音比人聲高出數倍，例如在 500Hz 處，可以發現有噪音的音檔在該處的訊號強度比乾淨音檔的訊號強度高出 4 倍，因此除了帶通濾波器外，我也同採用動態範圍壓縮(dynamic range compression)來降噪。

2. 程式碼說明

除了載入音檔和 spectrogram 及 FFT 的作圖外，程式碼還分為動態範圍壓縮設計、帶通濾波器設計、聲音重建與儲存。

2.1 濾波器設計

根據時頻譜的結果設計帶通路波器如下圖 figure 3 所示，是五組帶通濾波器的組合，當然也可以將藍色與綠色的帶通濾波器分別用低通濾波器和高通濾波器分別取代。

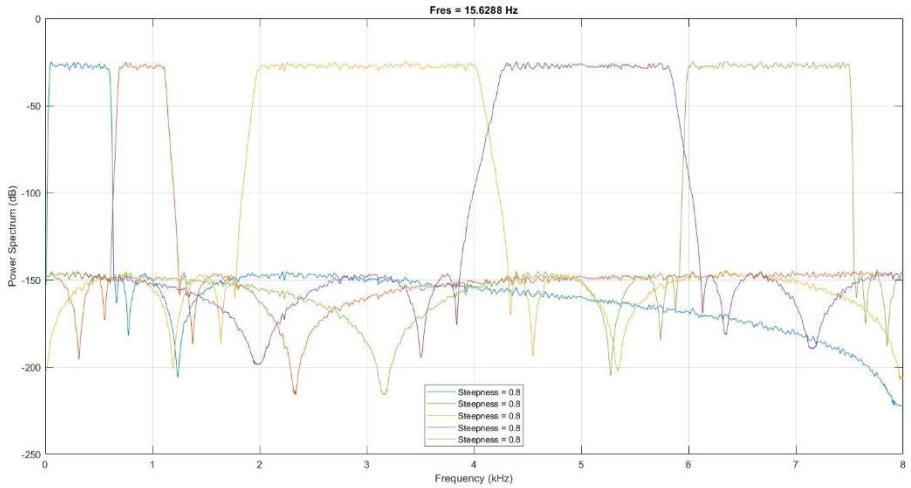

figure 3

```
% filter design,and apply Compressor and filter at the same time
% note：when apply filter, must be sure filter_control=1
if filter_control==0
    x = randn(20000,1);
else
    x = y2;
end
[y10,d10] = bandpass(x,[ 50
600],fs1,ImpulseResponse="iir",Steepness=0.8);
y10 = dRG(y10);
[y11,d11] = bandpass(x,[ 700
1100],fs1,ImpulseResponse="iir",Steepness=0.8);
[y12,d12] = bandpass(x,[2000
4000],fs1,ImpulseResponse="iir",Steepness=0.8);
y12 = dRCompressor(y12);
[y13,d13] = bandpass(x,[4300
5800],fs1,ImpulseResponse="iir",Steepness=0.8);
y13 = dRCompressor(y13);
[y14,d14] = bandpass(x,[6000
7500],fs1,ImpulseResponse="iir",Steepness=0.8);
y14 = dRCompressor(y14);
```

```
pspectrum([y10 y11 y12 y13 y14],fs1);
legend("Steepness = " + [0.8 0.8 0.8 0.8 0.8],Location="south");
```

2.2 動態壓縮設計

設計兩組動態範圍壓縮，分別為 figure 4 和 figure 5。

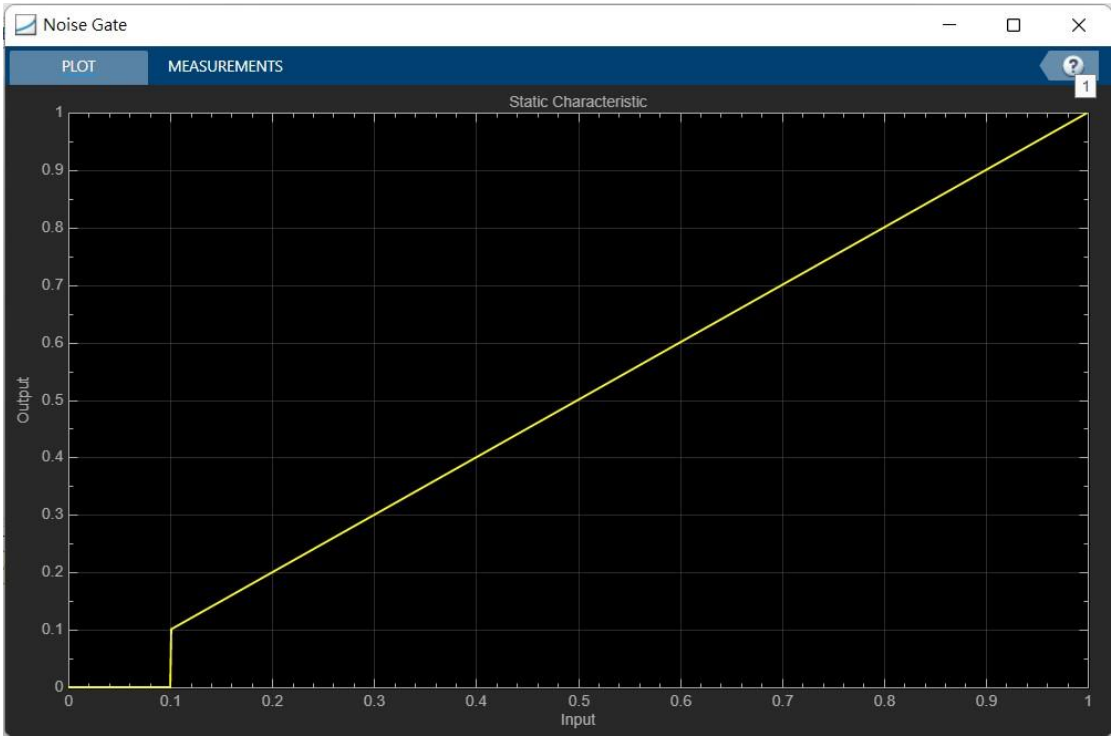

figure 4

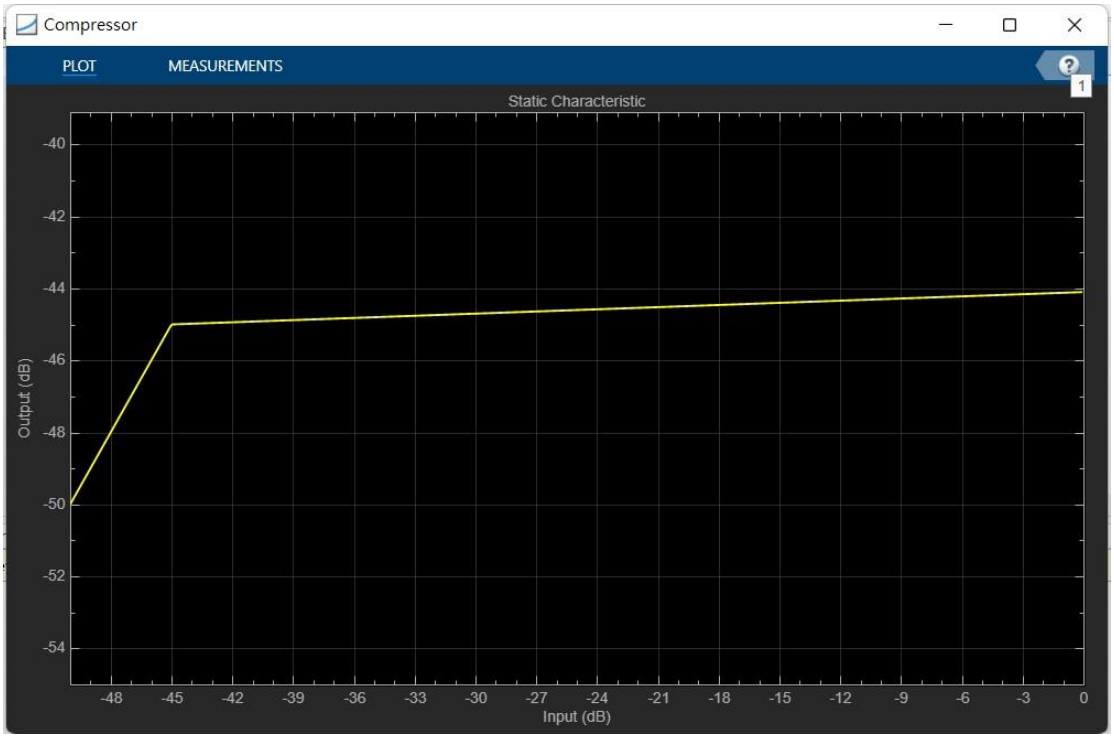

figure 5

figure 4 是應用於高頻率部分的 noise gate，功能是把某頻率段的訊號，將為低於閥值的訊號切掉，高於閥值的訊號保留。因為雖然很微弱，但高頻部分仍有噪音的泛音，但能量遠低於人聲部分。

figure 5 是應用於低頻部分的 compressor，功能是把某頻率段的訊號，將為低於閥值的訊號保留，高於閥值的訊號切掉。因為在低頻時，噪音的訊號能量遠高於人聲。

2.3 整體濾波器設計

完整濾波器設計參考 figure 6，該圖是透過將噪音輸入濾波器得到的頻率響應。因為 compressor 是動態壓縮，因此不同音檔輸入我所設計的濾波器都會得到不同的頻率響應，實際含噪音音檔的頻率響應為 figure 7。

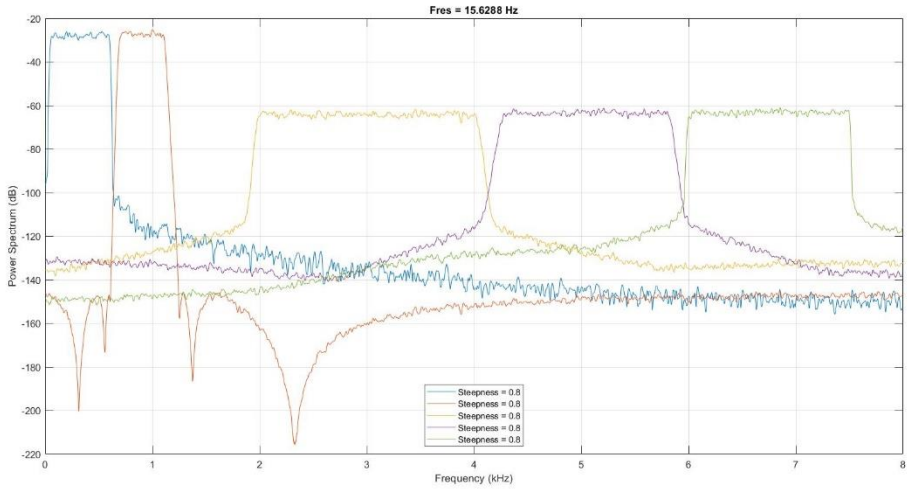

figure 6

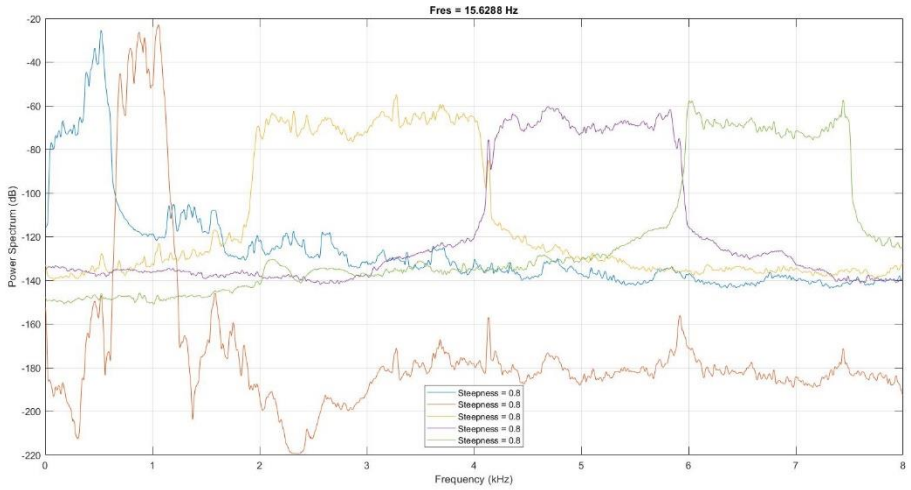

figure 7

3. 輸出結果

Figure 8 是經過去濾波器與動態範圍壓縮去噪後的音檔的時頻譜，從圖中可以看出噪音已被有效清除，不過因為人耳是非常敏銳的感測器，因此實際聽音檔時，還是能聽到在低頻的部分有鈴聲的噪音，不過也可以發現聲音保留的較為完整。綜合時頻譜與實際聆聽，我所設計的濾波器能有效消除高頻部分的鈴聲噪音並且有著極低的失真率。

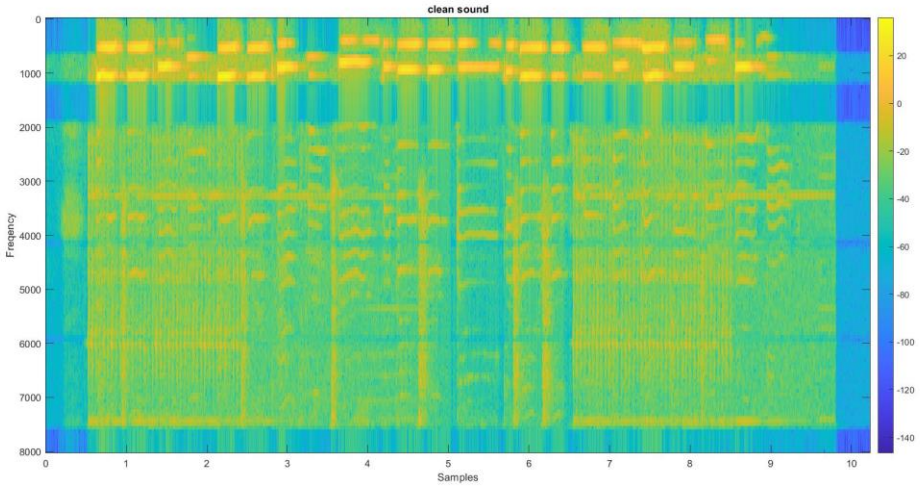

figure 8

4. FL studio 輸出

最後使用專業的音樂編輯軟體進行去噪的實驗(以 FL studio 示範之)，其去噪的策略是透過擷取雜訊的音檔，將其轉換成時頻譜特徵後，並自動化利用該特徵設計帶通與動態壓縮二合一的濾波器，因此能做到有著極高去噪比的同時仍有極低的失真率，此外亦可透過 dry and wet 的旋鈕調整該自動化濾波器的應用比率。

figure 9