

多媒體技術概論 HW3 - 101062124 戴宏穎

檔案結構：

- ├─ AnJing_4bit.wav (HW3 - part2 Origin)
- ├─ AnJing_8bit.wav (HW3 - part2 with Hanning)
- ├─ Blackman (HW3 - part1 with Blackman)
 - ├─ band-pass.wav
 - ├─ high-pass.wav
 - └─ low-pass.wav
- ├─ Hamming (HW3 - part1 with Hamming)
 - ├─ band-pass.wav
 - ├─ high-pass.wav
 - └─ low-pass.wav
- ├─ Hanning (HW3 - part1 with Hanning)
 - ├─ band-pass.wav
 - ├─ high-pass.wav
 - └─ low-pass.wav
- ├─ hw3.m (HW3 matlab code)
- ├─ hw3_mix.wav (HW3 - part1 Origin)
- └─ myFilter.m (HW3 filter with windowing function part)

Part A.

給一個混合了三個頻率有分層的音檔，希望我們用 FIR filter 把他們分離還原。

撰寫在 myFilter.m 裡面：

Step 1 Normalization, 我取 $f_c = \text{para}/f_s$, 中間值 $\text{mid} = \text{floor}(N/2)$

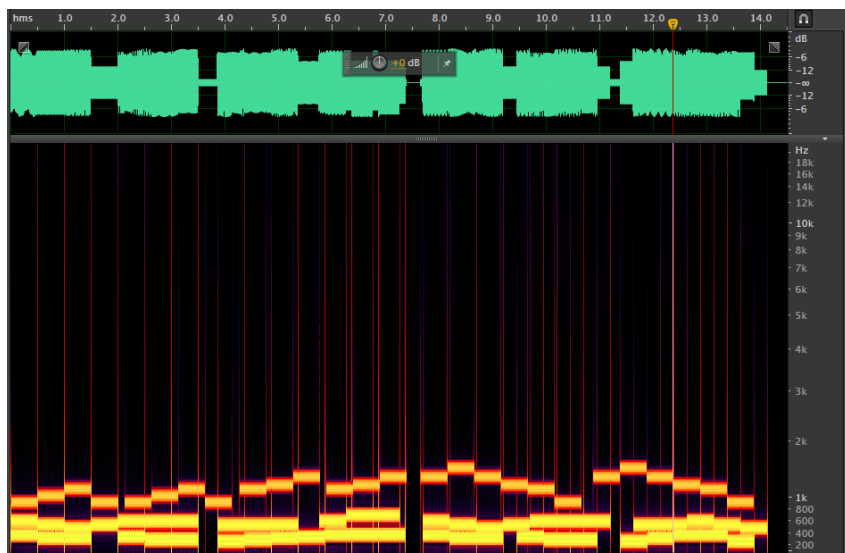
Step 2 看輸入是哪一種模型，直接套用講義上的公式，這邊其實一直遇到問題就是 band-pass 的時候，原本講義上有需要乘以係數，我乘了以後怪怪的，還好助教後來有公布說要拿掉係數就正常了。

Step 3, 4 我一起做掉，因為不是在 zero-phase 所以我是採用 wiki 上的公式，這樣聽起來也比較好聽。

Step 5 做 convolution 把 filter 出來的值與 input 的資料做結合 ($\text{filter}(k) * \text{in}(n-k)$)

在 hw3.m Q1 part 中：

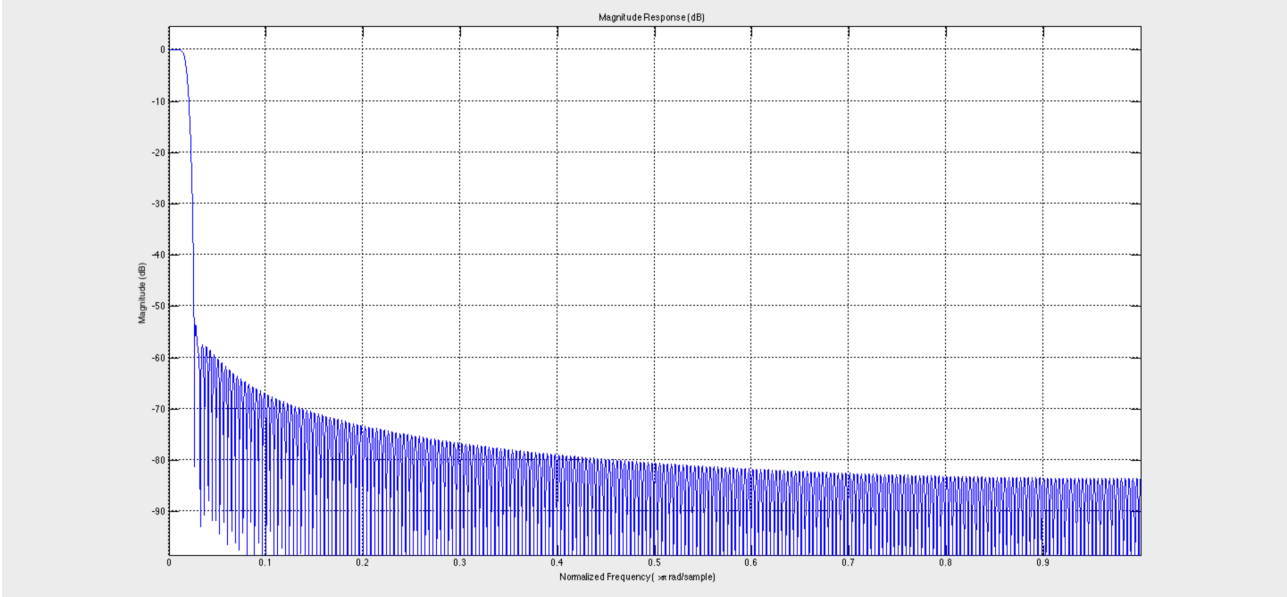
一開始不知道怎麼選出 N 值，後來助教有說明說在 500~1500 內都可以，因此我就選了 501 意思一下。



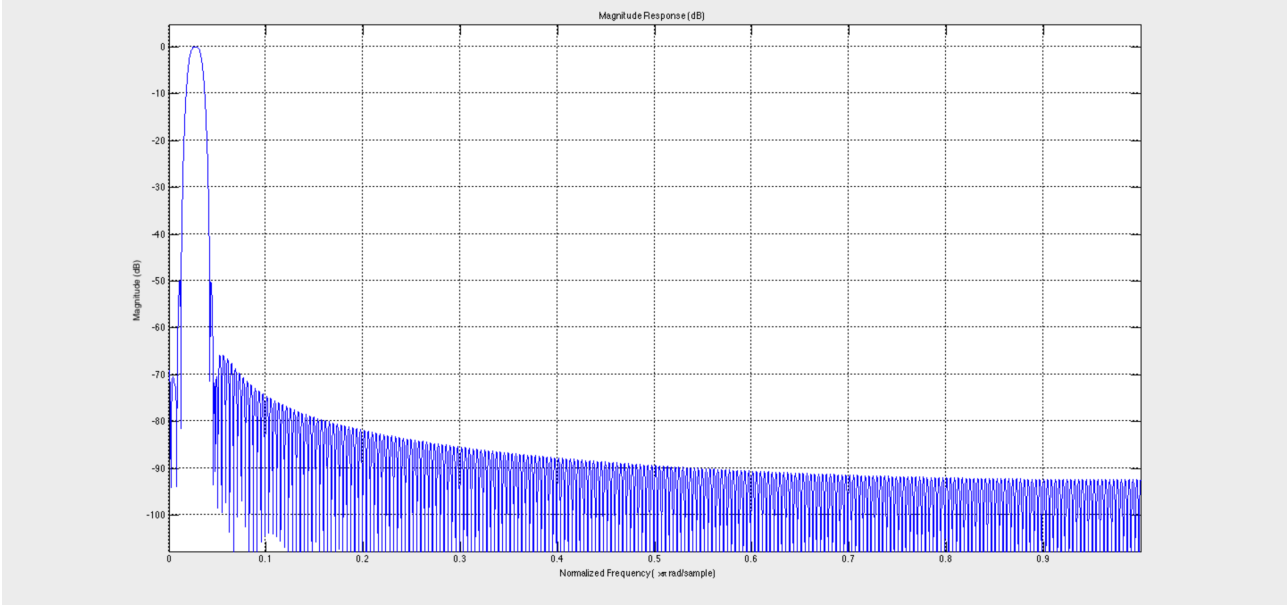
f1, f2 我是看下面的圖形去做選擇的，大概就是 low-pass -> $[\sim, 420]$, band-pass -> $[420, 780]$, high-pass -> $[780, \sim]$ ，這邊的數字是一直改來改去測試最好聽的聲音才定案的。

我總共跑了三個 windowing function (hamming, hanning, blackman)，分別在不同的資料夾下可以聽到。我覺得最好聽的好像是 blackman 聽起來比較順耳。不過下面的報告與截圖都是使用 hamming。

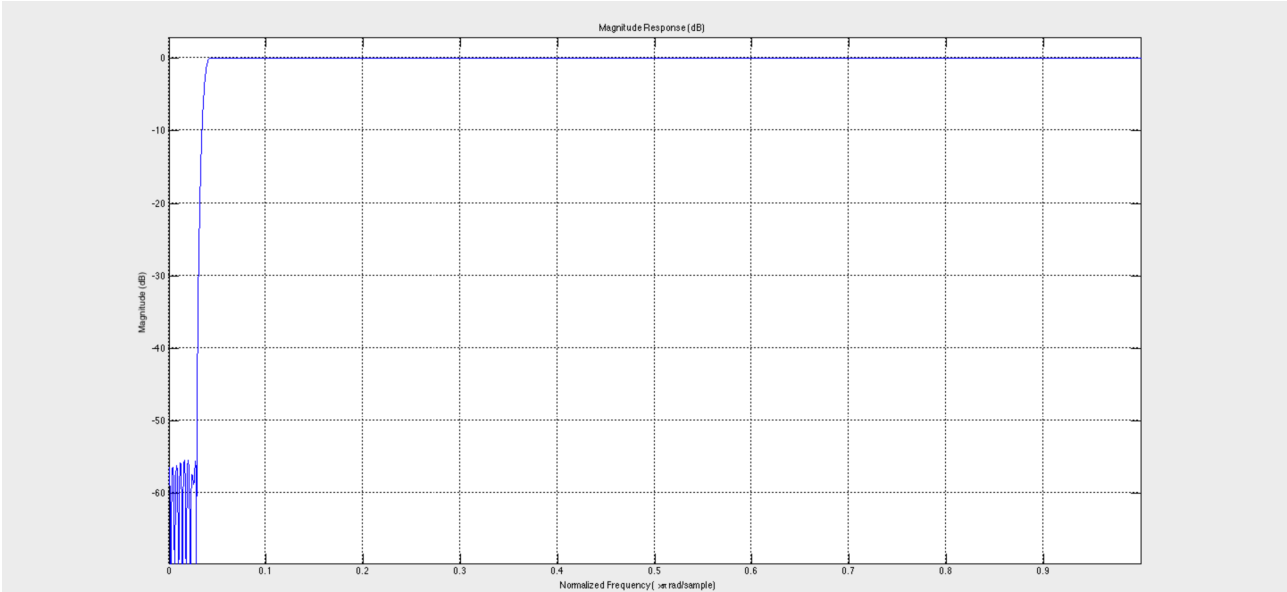
Low-pass 420 hz （以下三張圖是使用 fvtool 所產生）



Band-pass 420 - 780 hz

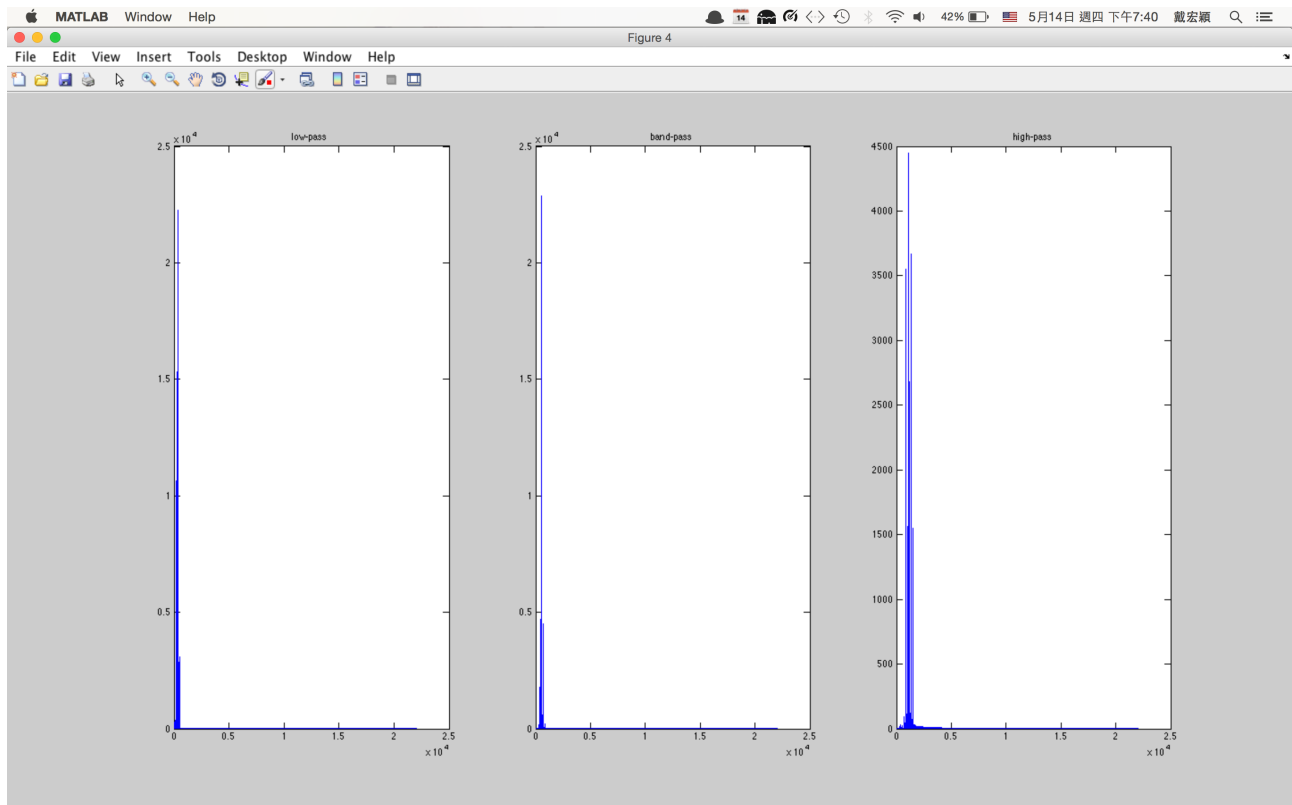


High-pass 780 hz

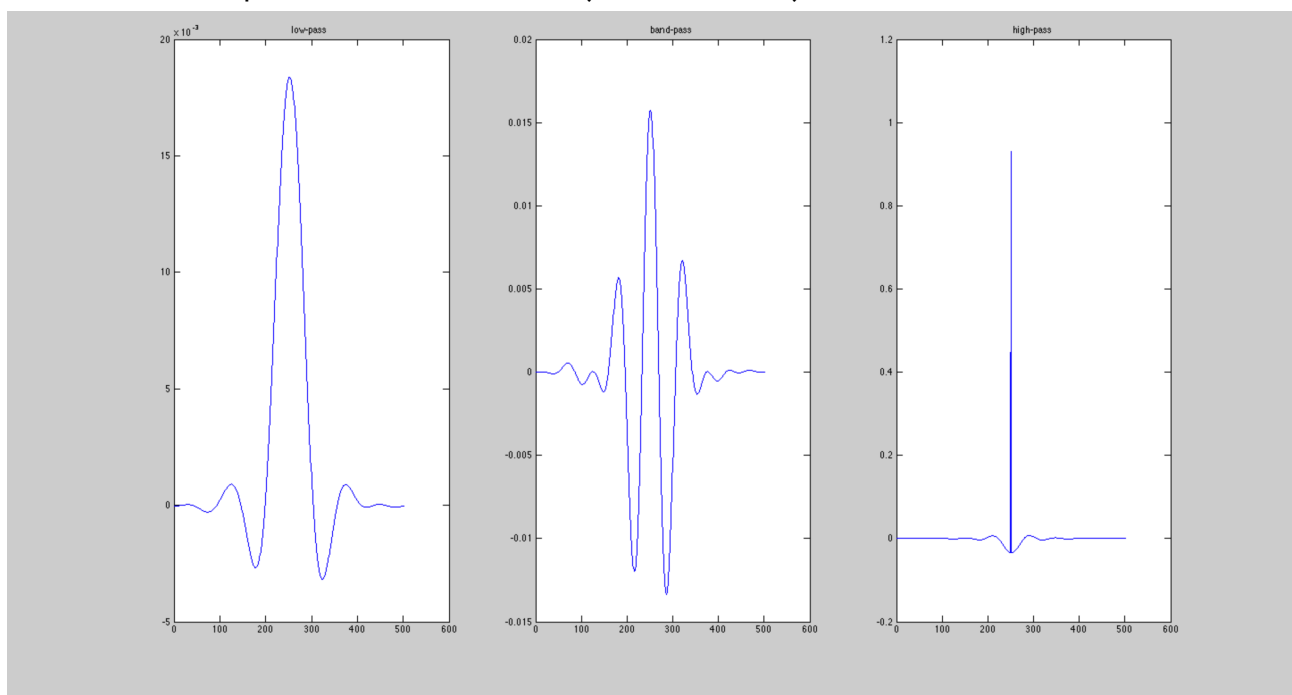


從上面可以看到 low-pass 的時候會留下最低的那部分，雖然在中高區還是會有部分雜訊留下來。
band-pass 雖然過濾出中間部分，但是其他區域還是有明顯雜訊。
high-pass 看起來比較乾淨的原因，我猜是因為在 780up 之後那整塊區域被留下來，所以沒當然雜訊就剩下中低區域了。

下面是 The spectrums of the filters 的結果，使用助教給的 sample code 產生



以下是 The shapes of the filters (time domain) 的結果：



在仔細聆聽分離出來的音檔中，其實我們可以很明顯聽見雜訊，在距離感比較遠的地方依然可以聽見一些其他區域的聲音。

Part2.

這題主要是把一個原本是 8bit 雙聲道的音檔弄成 4bit 希望我們能還原回去 8bit 或者讓他減少明顯的雜訊聲音。

Step 1. Audio dithering 把原本的聲音加入些隨機的雜訊

我是這樣寫的：`noise = y+rand(size(y))-0.5; % y = origin wav`

做完 add random noise 以後才把他們 remapping 到 $[-128, 128]$ 的 domain 中。

Step 2. Noise shaping 把雜訊推到高頻處以便等等用 low-pass 把他們一次幹掉
這邊直接用講義公式來處理， c 的部分我取 0.9 感覺比較好聽。

Step 3. Hanning windowing function 用 part1 的 filter 來處理

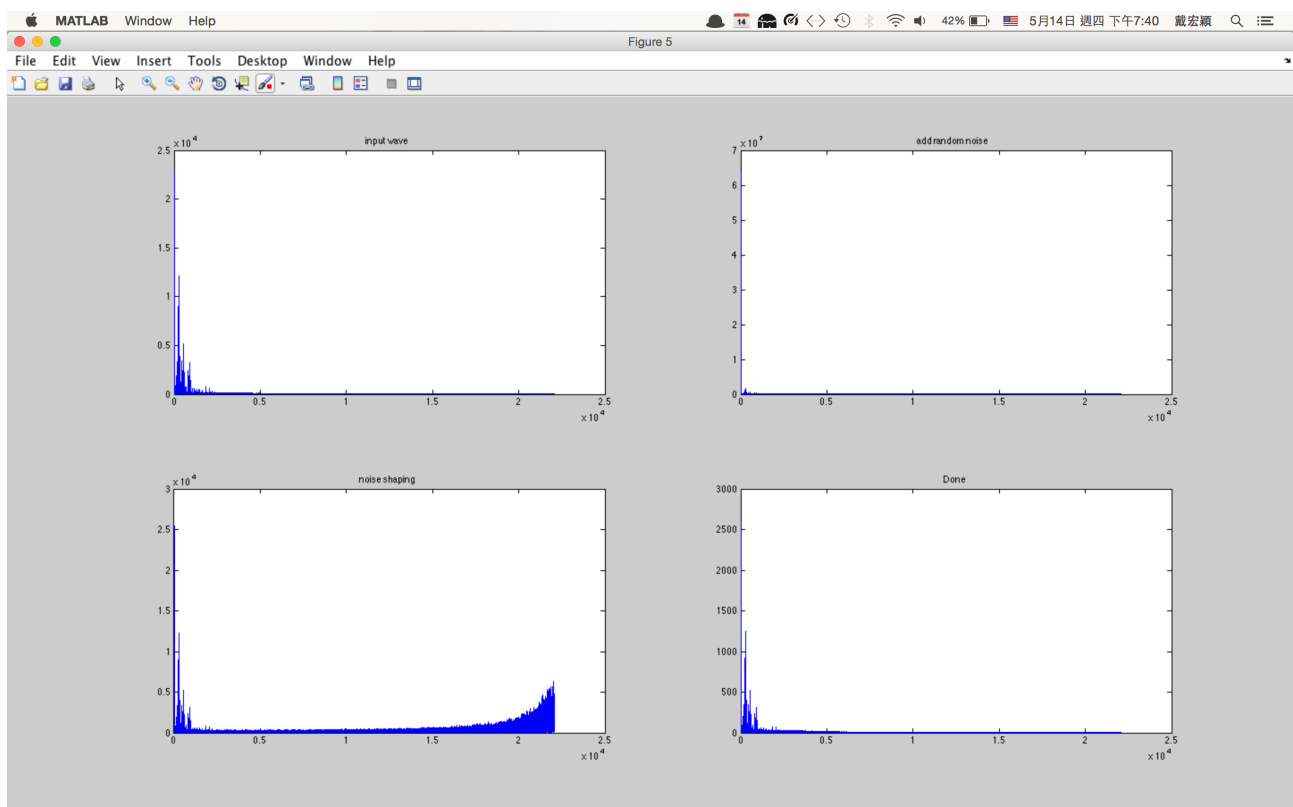
過濾我們需要的部分，也是看頻率可以發現在大概 500 hz 就差不多可以完成了。

左上：原本的音樂

右上：加入雜訊

左下：做完 noise shaping

右下：最後處理完的音檔



隨機加入雜訊的好處是，在經過 noise shaping 以後，會把我們最想聽到的部分濾到。感覺上會變成是好像有點模糊的音樂，而不是像原本的音檔那麼刺耳。因為把雜訊逼到高頻率區，又是透過 low-pass 來過濾，最後聽到的結果會比較低。在雜訊的籠罩下，也會讓人覺得好像隔了一層紗在聽音樂。