

1.程式：

構.wav 檔的標頭檔格式，再來就輸出八個Cosine 波的.wav檔。下面是 Sample Rate分別為 8k 和16k 的標頭檔：

8k：

```
ChunkID：RIFF
ChunkSize：16008    //SampleRate*2+8
Format：WAVE
SubChunk1ID：fmt
SubChunk1Size：16
AudioFormat：1
NumChannels：1
SampleRate：8000
ByteRate：16000      //SampleRate*2
BlockAlign：2
BitPerSample：16
SubChunk2ID：data
SubChunk2Size：16000  //SampleRate*2
```

16k：

```
ChunkID：RIFF
ChunkSize：32008    //SampleRate*2+8
Format：WAVE
SubChunk1ID：fmt
SubChunk1Size：16
AudioFormat：1
NumChannels：1
SampleRate：16000
ByteRate：32000      //SampleRate*2
BlockAlign：2
BitPerSample：16
SubChunk2ID：data
SubChunk2Size：32000  //SampleRate*2
```

標頭檔寫入後就開始寫data，而data 就是  $x[t] = 10000(2\pi ft)w(t)$ 。

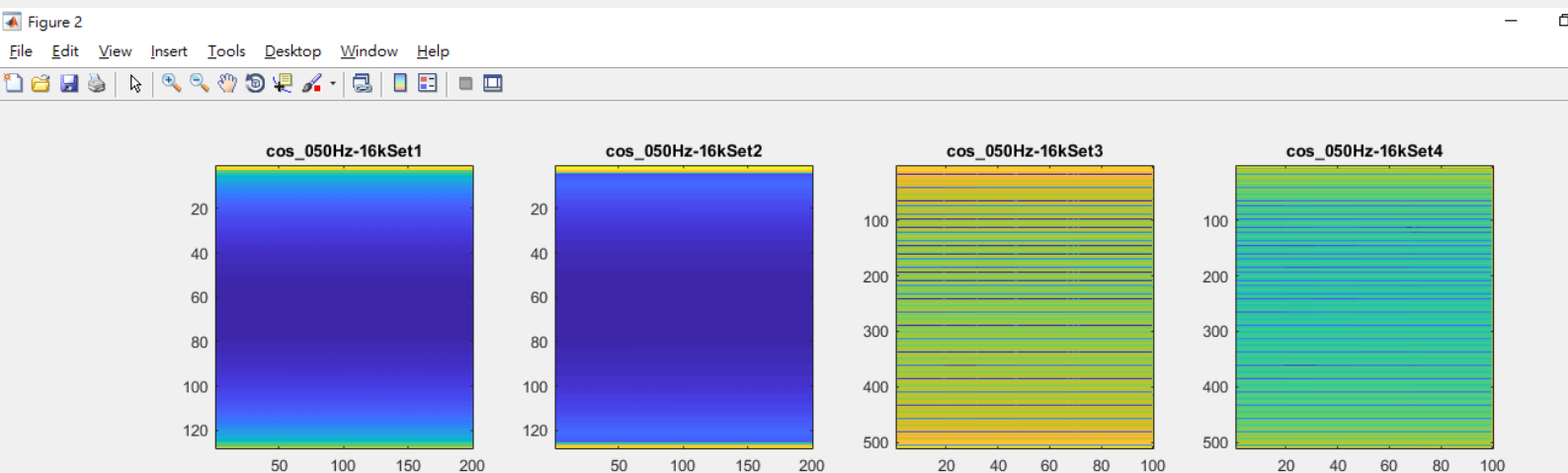
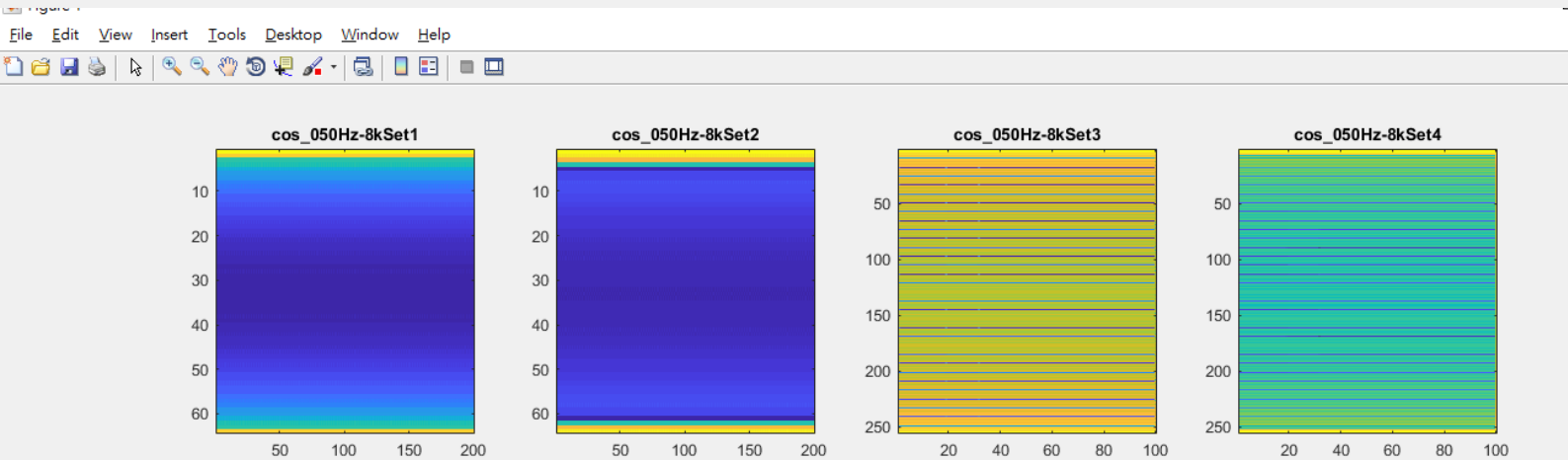
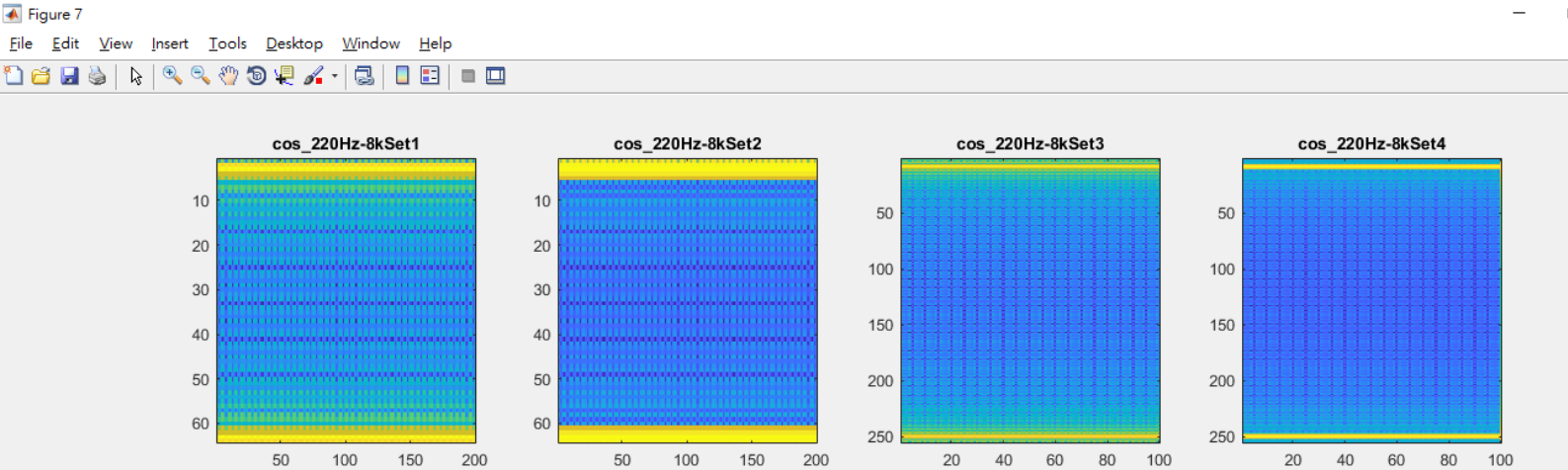
這邊我是用for 迴圈來一次完成八個.wav 檔，所以迴圈是跑 0~7，然後我在每次跑的時候用if 加他的條件，再用sprintf 存入我要寫的.wav 檔。例如：在跑第0次時我的 Hz = 50，k =8000 所以我要開的檔就是 cos\_050Hz-8k.wav。

接下來就要先把SampleRate、DataSize 和Data 取出來然後再開始跑Setting 的部分，這邊我也是用for 迴圈來一次跑10 個音檔，一樣是用if 加入條件只是這次要寫入的要換成.txt 檔。

再來要開始做Setting，因為Setting1 和Setting2 的設定是一樣的，所以我主要就分成一、二一塊，三、四一塊。每一塊再開始乘Window Function 之前要先算好我的設定。有需要的有：

M：是每區塊包含得點的個數 = SampleRate \* Frame Interval

Frame：是總共有切成幾個區塊 = ( DataSize / 2 ) \* M



P：是乘以Window Function的點的個數 =  $\text{SampleRate} * \text{Analysis Window Size}$   
N：是要做DFT的點的個數 = DFT Window Size

接下來開始計算，根據老師的講義，我的算法是每個N 為一個區塊，區塊內小於P 的乘以我們窗函數，大於P 的設為0，再用乘完的結果做DFT，做完再算出震幅，最後再存入我要的.txt檔。

最後的最後就用MathLab 畫圖，下面為輸出結果。

Figure 3

File Edit View Insert Tools Desktop Window Help

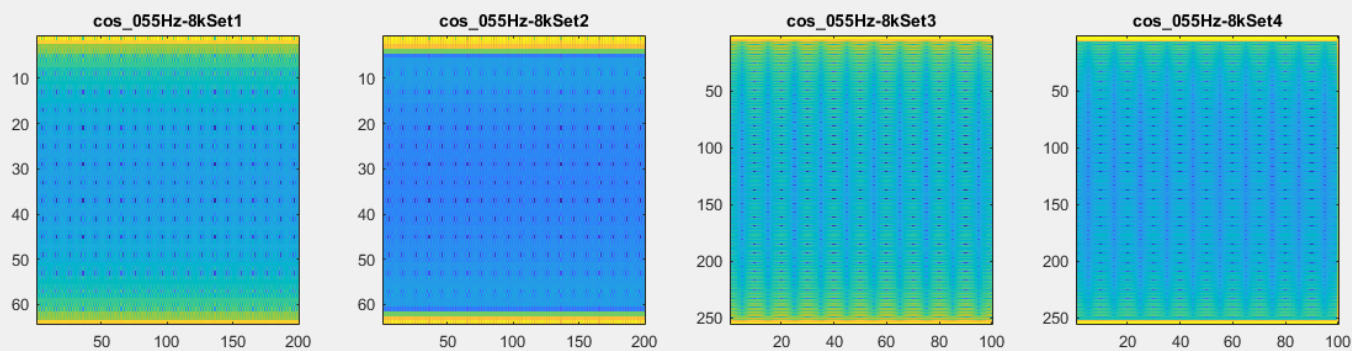


Figure 4

File Edit View Insert Tools Desktop Window Help

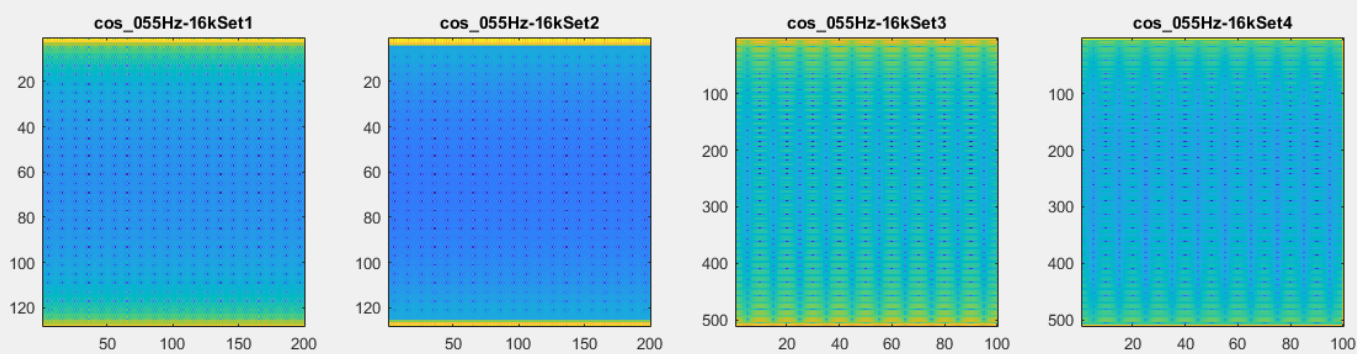


Figure 5

File Edit View Insert Tools Desktop Window Help

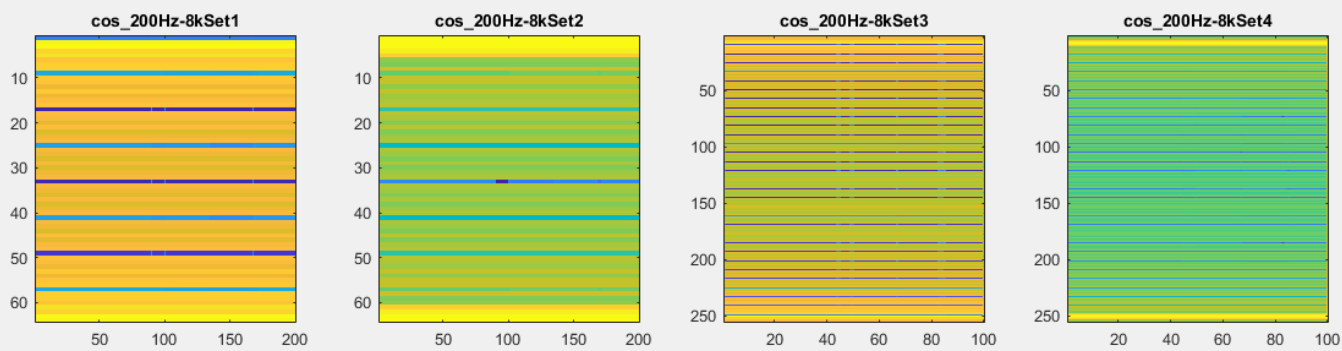
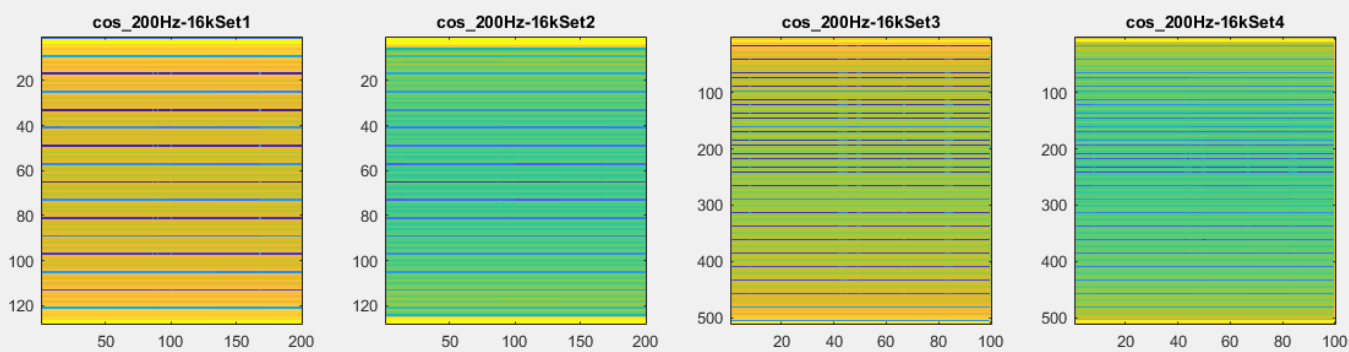
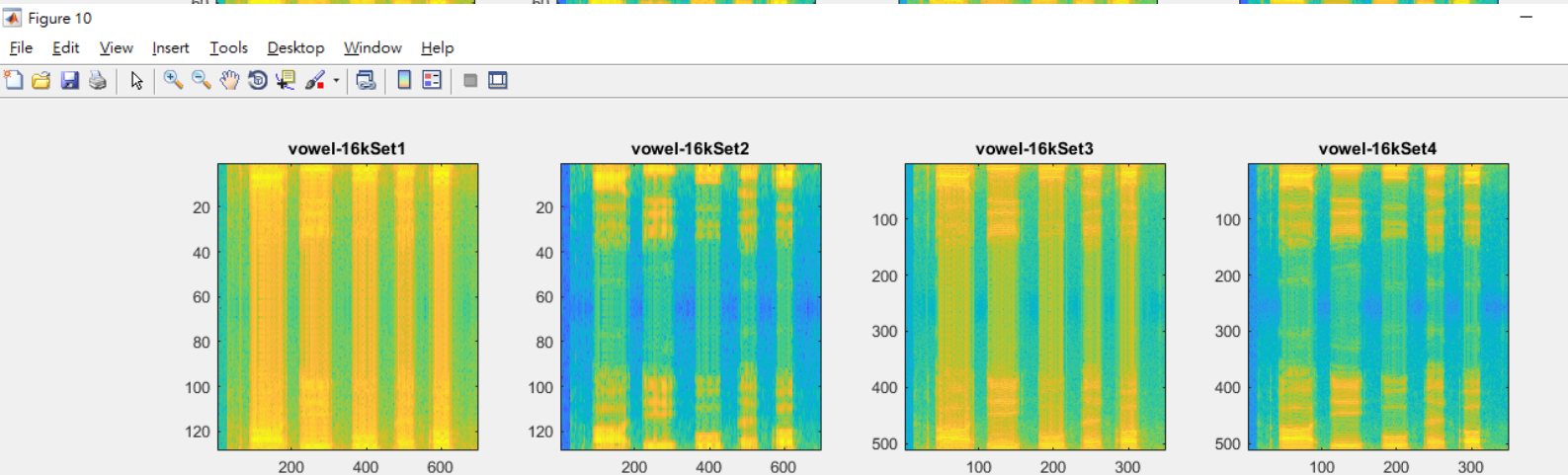
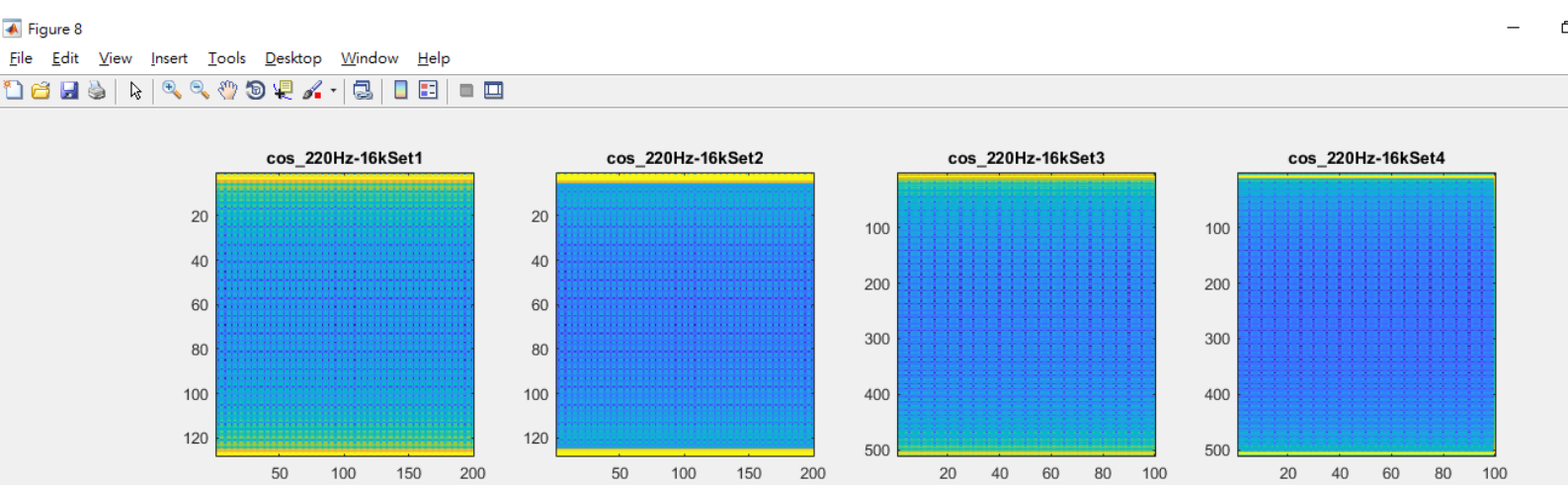


Figure 6

File Edit View Insert Tools Desktop Window Help



1.觀察輸出結果圖片時我發現在同樣的訊號和同樣的取樣頻率下，使用 Rectangular Window Function 會比使用 Hamming Window Function 來的黃，其代表的是做DFT時造成的能量洩漏，而原因是做完DFT之後Rectangular的頻譜相較於Hamming 是比較鬆散的，且他的兩邊比較高，所以才會造成這樣效果。但在觀察Vowel 兩個音檔時發現使用Rectangular時的頻譜對於在該出現能量的頻率時其顏色比Hamming 來的更深，所以Rectangular Window Function 的頻率識別率是比較高的。



2.Setting 1 & 2 和 Setting 3 & 4 的差別是取的 Window 大小不同，Setting 3 & 4 的 Window 比較大，且 Setting 1 & 2 的  $M = P$ ，然而 Setting 3 & 4 的是  $M < P$ ，這意味著 3 & 4 每個窗取得點比 1 & 2 來得多，重疊的部分也比較多，所以做出來的結果會比較精確。

3.取樣頻率 (SampleRate) 比較高的輸出結果會比取樣頻率低的來得精確。