DSP Assignment03 Report

1. **程式使用說明**

執行程式時，**需先跑過c code(生成所需txt、wav)，才能跑Matlab程式**。

程式總共有1個Makefile、3個Matlab檔案、4個c檔案、1個標頭(.h)檔案：

Makefile: 利用cmd輸入make以compile所有.c和.h檔案。輸入 make test以執行程式。

plot\_LPF\_DTFT.m: 畫出不同M底下的LPF(頻域軸)

test\_diffM\_transient.m: 畫出3500Hz和5000Hz經過不同M所產生出的暫態(transient)。

minimum\_phase.m: 算出minimum phase版本的LPF(M=4,16,64)並畫出與原本的LPF相對圖形。

function.h: function標頭檔。

main.c: 主程式，執行function並輸出資料的地方。

function.c: 裡面function包含hamming、low\_pass、generateSin、gen\_lowpass\_DTFT、result\_txt、lowpass\_coef\_txt等函式。

gen\_wav.c: 此檔案只包含一個函式，主要為生成wav音檔使用。

through\_lpf.c: 此檔案只包含一個函式，主要為計算弦波進入LPF之後的數值使用。

執行程式時，**需先跑過c code(生成所需txt、wav)，才能跑Matlab程式**。

1. **C code程式生成檔案說明**

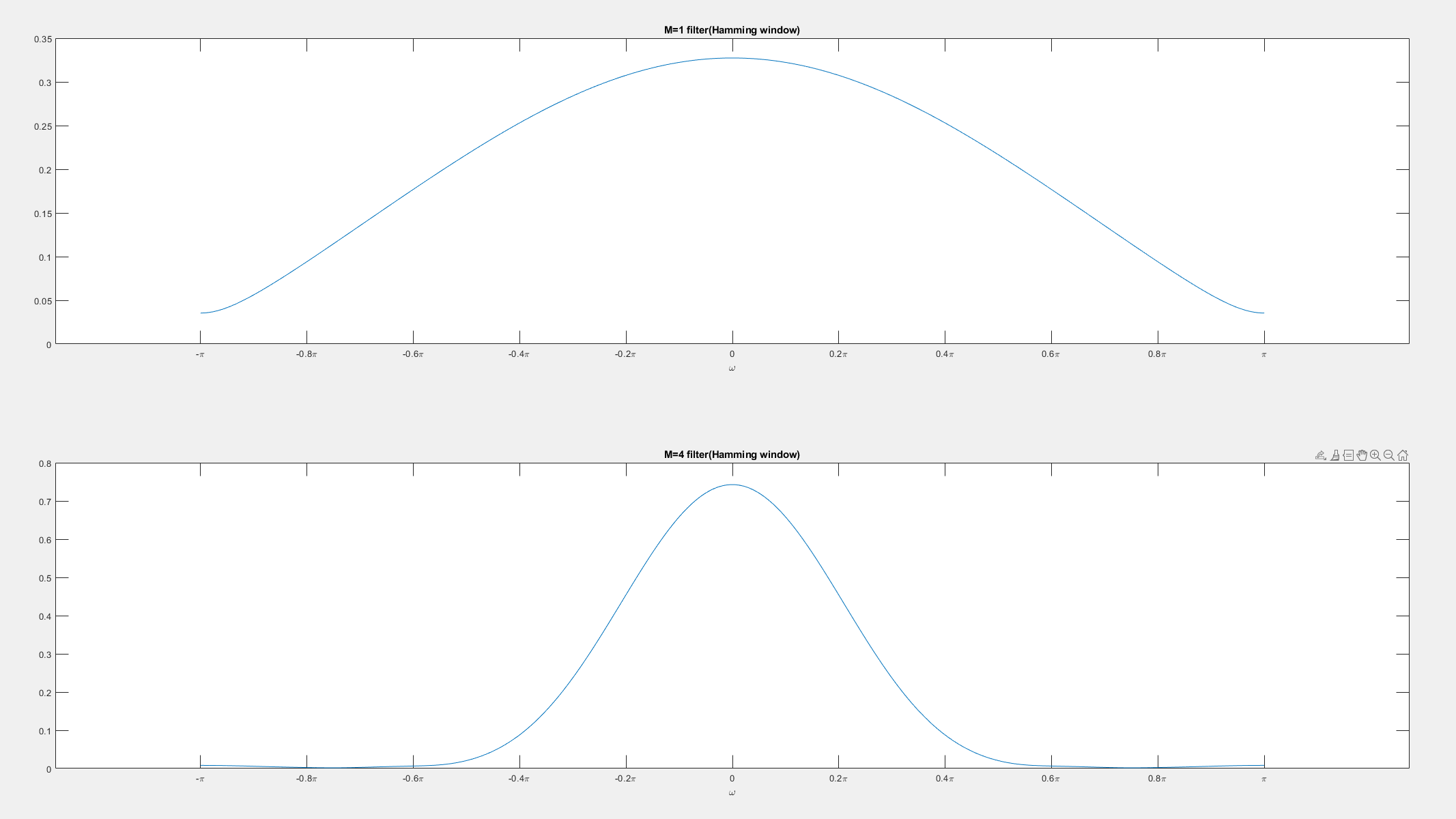
首先，透過make和make test後，會執行C code的運行，運行時間由於有許多資料須輸出，所以耗時大約要1分鐘。跑完之後會生成的檔案如下表所示(依照程式生成的先後順序)：

表（一）：C code生成檔案對應之使用方式

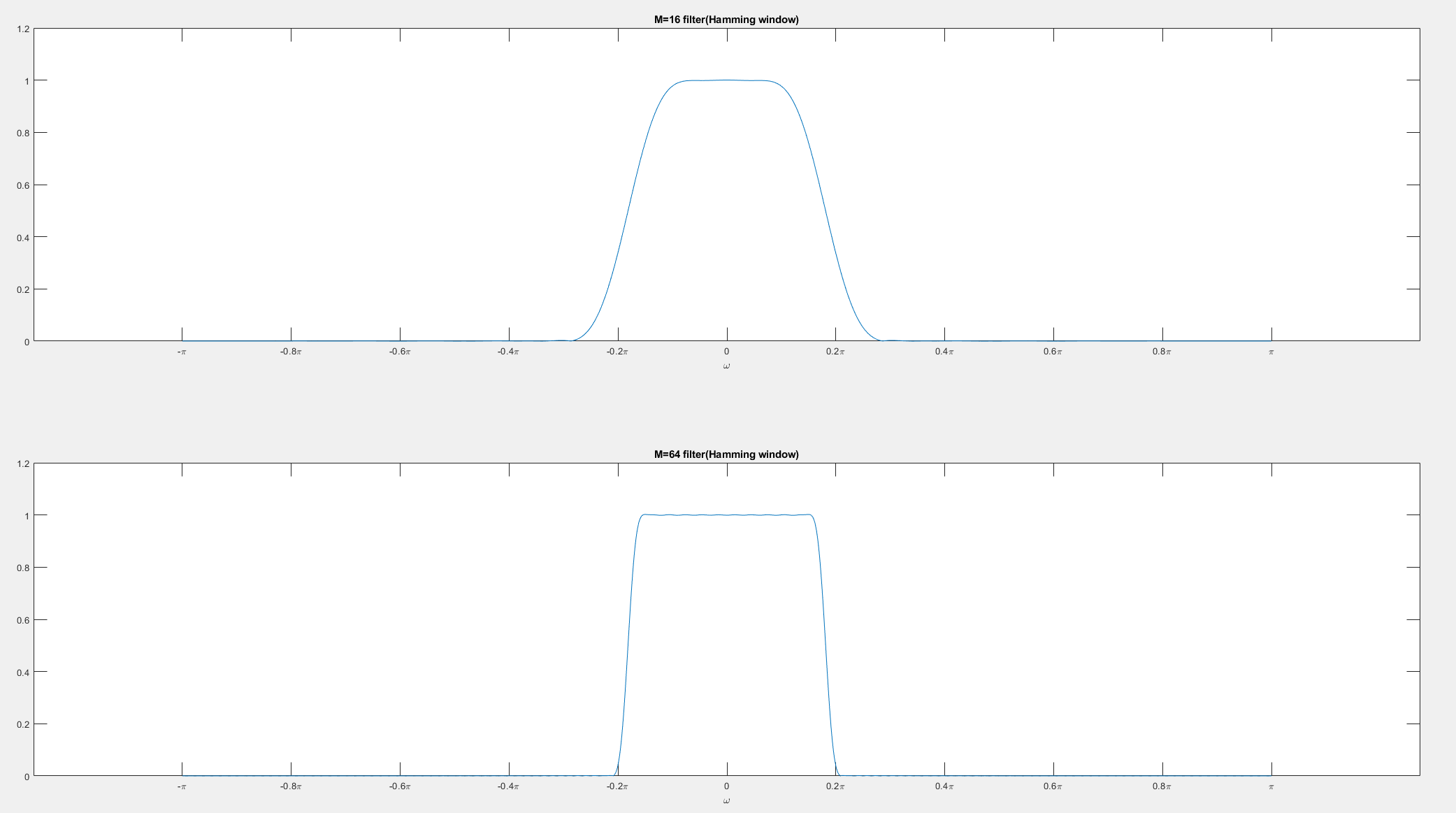
|  |  |  |
| --- | --- | --- |
| 檔案名稱 | 內容物 | 對應使用Matlab程式 |
| M\_1\_LPF.txt  M\_4\_LPF.txt  M\_16\_LPF.txt  M\_64\_LPF.txt  M\_256\_LPF.txt  M\_512\_LPF.txt  M\_1024\_LPF.txt  M\_2048\_LPF.txt | 儲存不同M值的LPF經過DTFT所得到的數值 | 使用plot\_LPF\_DTFT.m以觀察LPF DTFT後的圖形 |
| sin3500Hz\_M1.wav  sin3500Hz\_M4.wav  sin3500Hz\_M16.wav  sin3500Hz\_M64.wav  sin3500Hz\_M256.wav  sin3500Hz\_M512.wav  sin3500Hz\_M1024.wav  sin3500Hz\_M2048.wav  sin5000Hz\_M1.wav  sin5000Hz\_M4.wav  sin5000Hz\_M16.wav  sin5000Hz\_M64.wav  sin5000Hz\_M256.wav  sin5000Hz\_M512.wav  sin5000Hz\_M1024.wav  sin5000Hz\_M2048.wav | 生成兩個不同頻率的弦波經過不同order數的LPF之後的音檔 | 直接使用並聆聽即可 |
| sin3500Hz\_M1.txt  sin3500Hz\_M4.txt  sin3500Hz\_M16.txt  sin3500Hz\_M64.txt  sin3500Hz\_M256.txt  sin3500Hz\_M512.txt  sin3500Hz\_M1024.txt  sin3500Hz\_M2048.txt  sin5000Hz\_M1.txt  sin5000Hz\_M4.txt  sin5000Hz\_M16.txt  sin5000Hz\_M64.txt  sin5000Hz\_M256.txt  sin5000Hz\_M512.txt  sin5000Hz\_M1024.txt  sin5000Hz\_M2048.txt | 儲存兩個不同頻率的弦波經過不同order數的LPF之後的數值 | 使用test\_diffM\_transient.m以觀察不同order數的LPF所造成的暫態(transient)現象 |
| sin3500Hz.txt  sin5000Hz.txt | 儲存兩個不同頻率的弦波數值 | 用於minimum\_phase.m並生成經過minimum phase LPF之後的數值 |
| h\_M\_4.txt  h\_M\_16.txt  h\_M\_64.txt | 儲存M=4,16,64 LPF的時域(未經過DTFT)的數值 | 用於minimum\_phase.m並製作minimum phase LPF |

1. **Matlab程式結果、生成檔案說明**
   1. **plot\_LPF\_DTFT.m**

執行完C code並執行此檔案，可以觀察到不同order數下的LPF DTFT後的圖形。可以發現當Order數越高的情況下，DTFT後的圖形越像是理想的低通濾波器(Ideal LPF)。



圖（一）：LPF DTFT, order = 1,4



圖（二）：LPF DTFT, order = 16,64

一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

自動產生的描述

圖（三）：LPF DTFT, order = 256,512

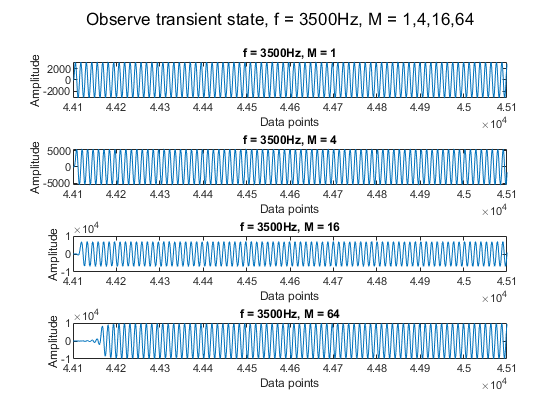
一張含有 文字, Rectangle, 螢幕擷取畫面, 圖表 的圖片

自動產生的描述

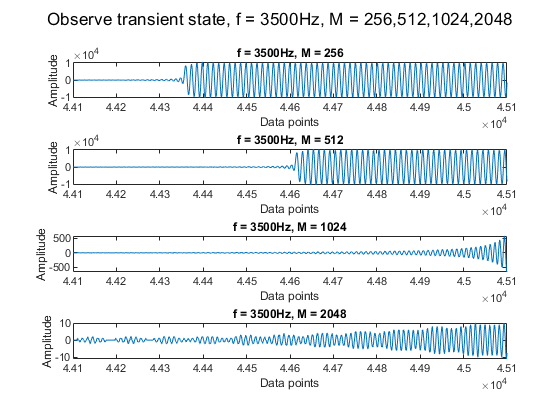
圖（四）：LPF DTFT, order = 1024,2048

* 1. **test\_diffM\_transient.m**

執行完C code並執行此檔案，可以觀察到兩種不同頻率的SIN波在經過不同order數下的LPF 後的圖形。可以發現當Order數越高的情況下，Transient state的時間就會越長，也就造成波形進入Steady State的時間越延後。由於此次實驗取樣率為44100Hz，並且我讓音檔的前一秒和最後一秒為靜音(方便觀察transient)，所以觀察時我從第44100個點(音檔開始位置)開始觀察。



圖（五）：觀察暫態，f = 3500Hz, Order = 1,4,16,64

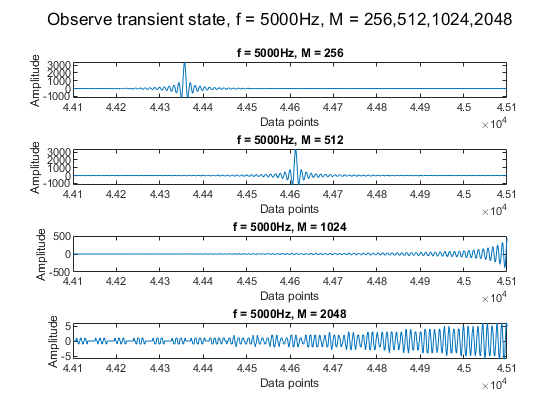


圖（六）：觀察暫態，f = 3500Hz, Order = 256,512,1024,2048

一張含有 文字, 螢幕擷取畫面, 行, 字型 的圖片

自動產生的描述

圖（七）：觀察暫態，f = 5000Hz, Order = 1,4,16,64



圖（八）：觀察暫態，f = 5000Hz, Order = 256,512,1024,2048

* 1. **minimum\_phase.m**

執行完C code並執行此檔案，可以觀察到