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**。

2. 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

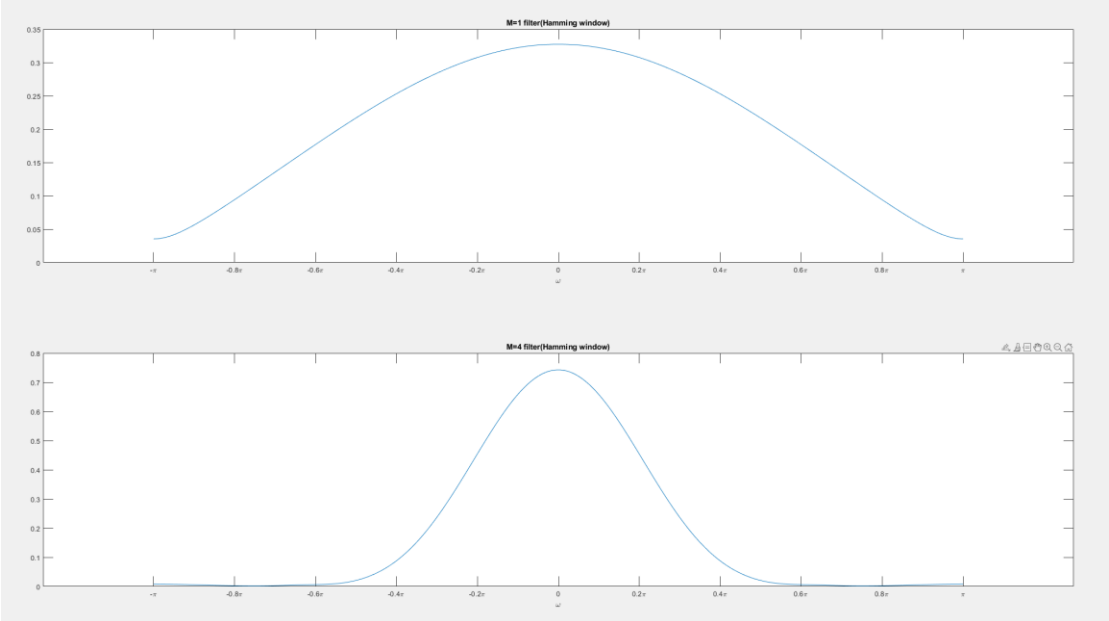
檔案名稱	內容物	對應使用 Matlab 程式
M_1_LPF.txt M_4_LPF.txt M_16_LPF.txt M_64_LPF.txt M_256_LPF.txt M_512_LPF.txt M_1024_LPF.txt M_2048_LPF.txt	儲存不同 M 值的 LPF 經過 DTFT 所得到的數值	使用 plot_LPF_DTFT.m 以觀察 LPF DTFT 後的圖形
sin3500Hz_M1.wav sin3500Hz_M4.wav sin3500Hz_M16.wav sin3500Hz_M64.wav sin3500Hz_M256.wav sin3500Hz_M512.wav 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	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

sin5000Hz_M256.txt sin5000Hz_M512.txt sin5000Hz_M1024.txt sin5000Hz_M2048.txt		
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

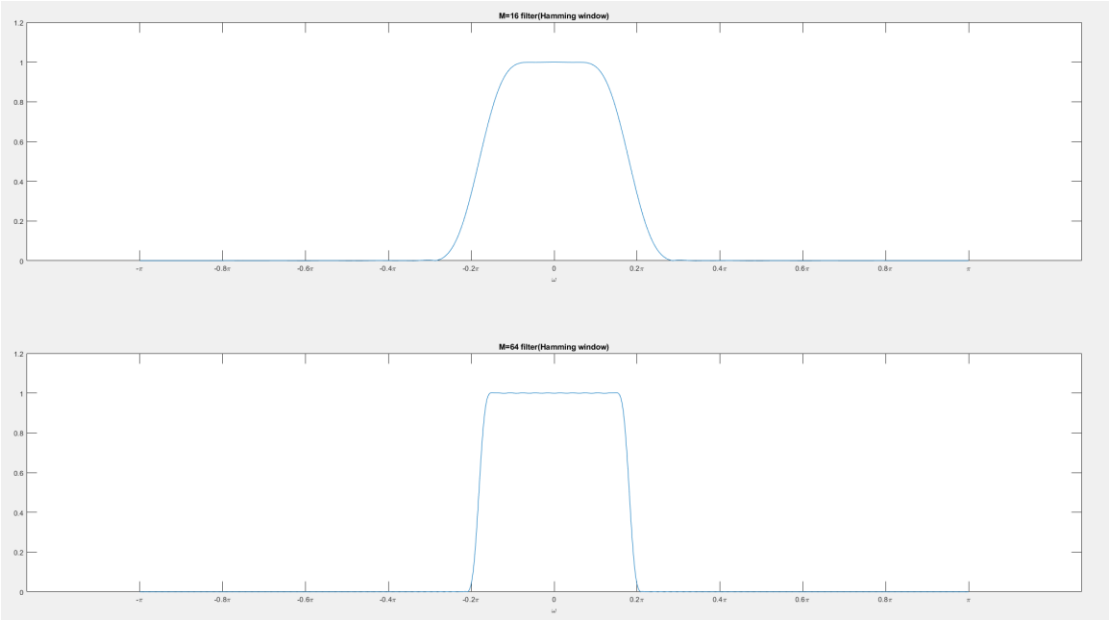
3. Matlab 程式結果、生成檔案說明

3.1. plot_LPF_DTFT.m

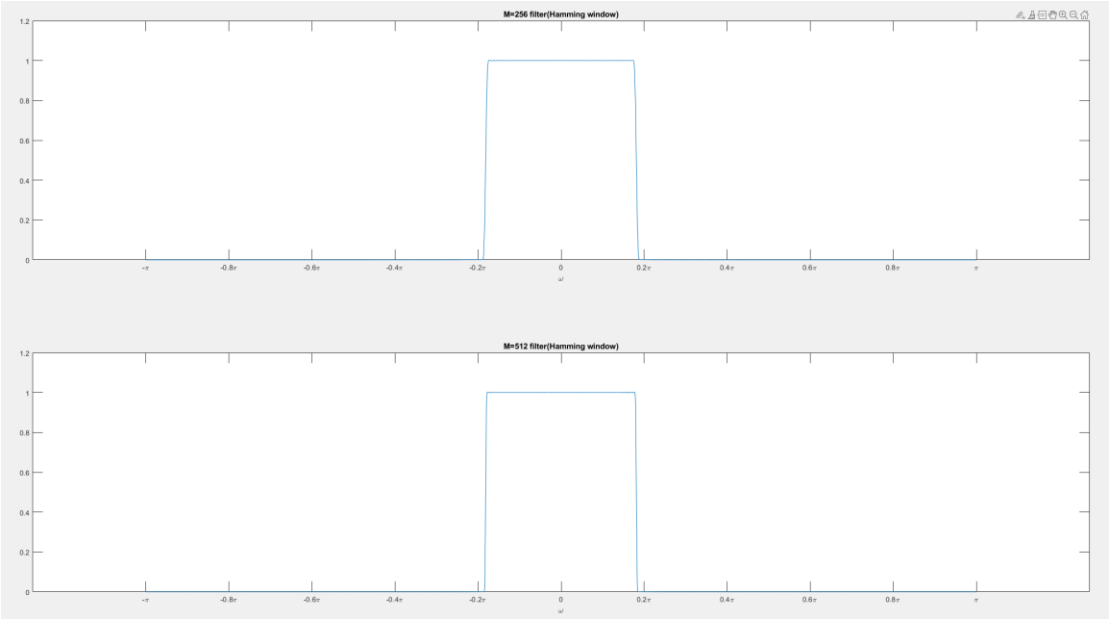
執行完 C code 並執行此檔案，可以觀察到不同 order 數下的 LPF DTFT 後的圖形。可以發現當 Order 數越高的情況下，DTFT 後的圖形越像是理想的低通濾波器(Ideal LPF)。



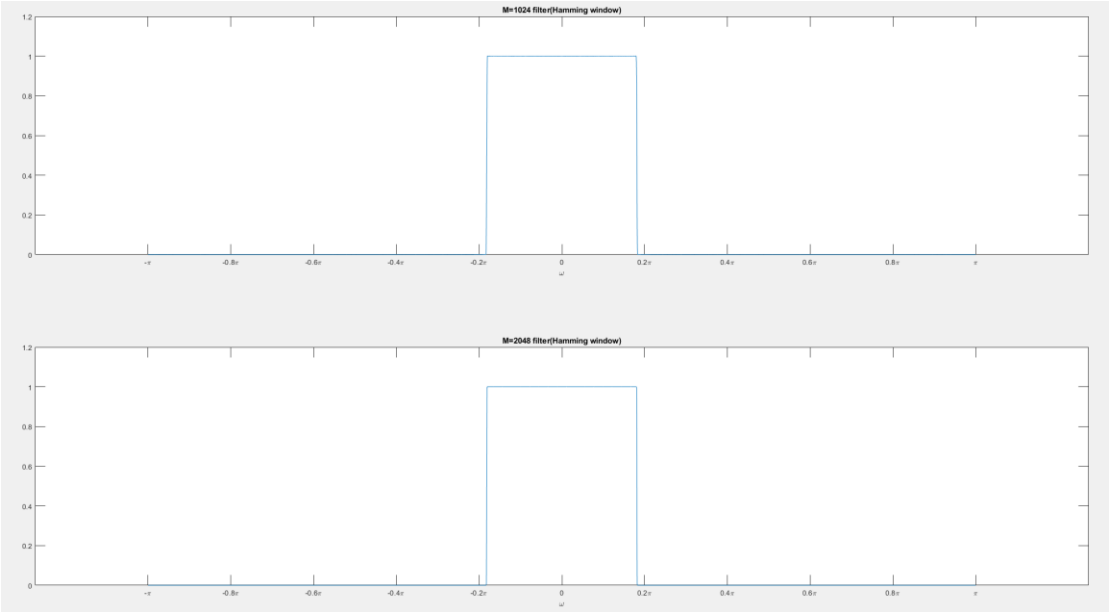
圖（一）：LPF DTFT, order = 1,4



圖（二）：LPF DTFT, order = 16,64



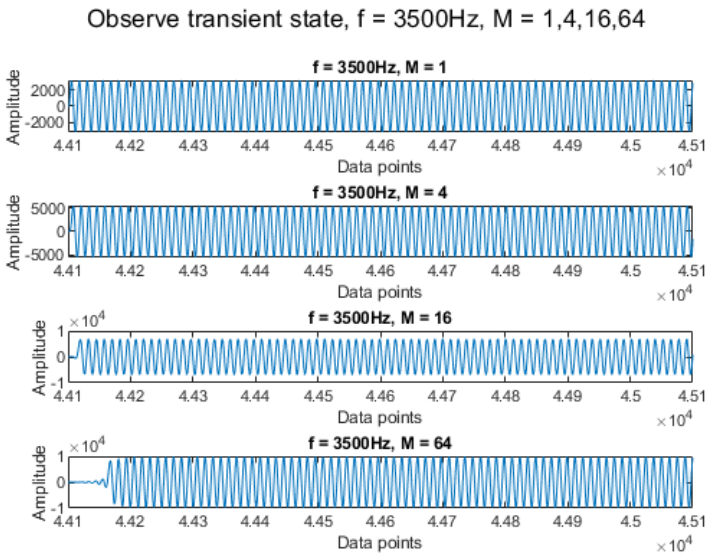
圖（三）：LPF DTFT, order = 256,512



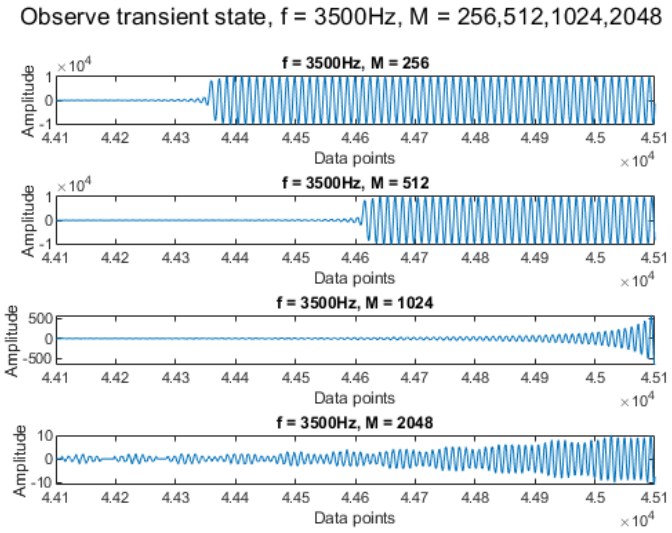
圖（四）：LPF DTFT, order = 1024,2048

3.2. test_diffM_transient.m

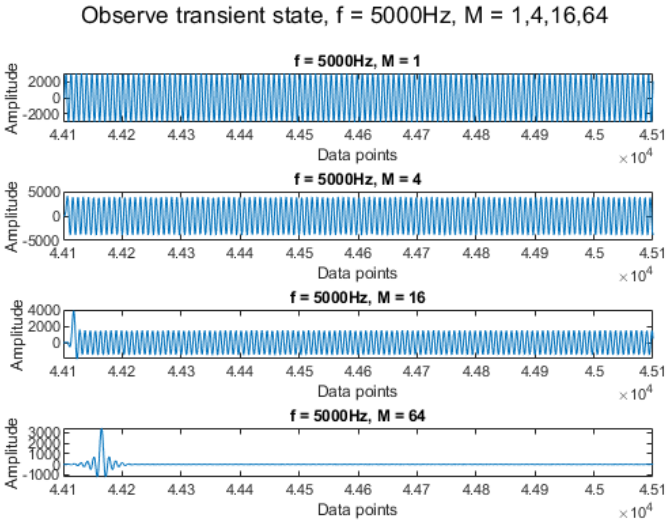
執行完 C code 並執行此檔案，可以觀察到兩種不同頻率的 SIN 波在經過不同 order 數下的 LPF 後的圖形。可以發現當 Order 數越高的情況下，Transient state 的時間就會越長，也就造成波形進入 Steady State 的時間越延後。由於此次實驗取樣率為 44100Hz，並且我讓音檔的前一秒和最後一秒為靜音(方便觀察 transient)，所以觀察時我從第 44100 個點(音檔開始位置)開始觀察。



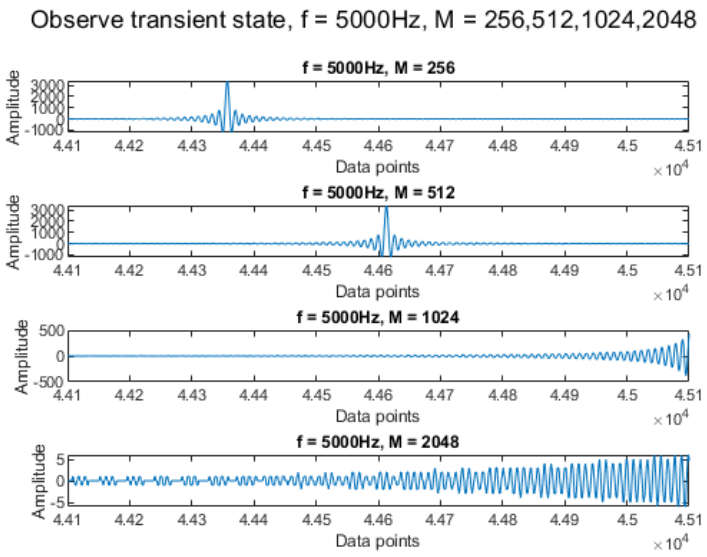
圖（五）：觀察暫態，f = 3500Hz, Order = 1,4,16,64



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



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



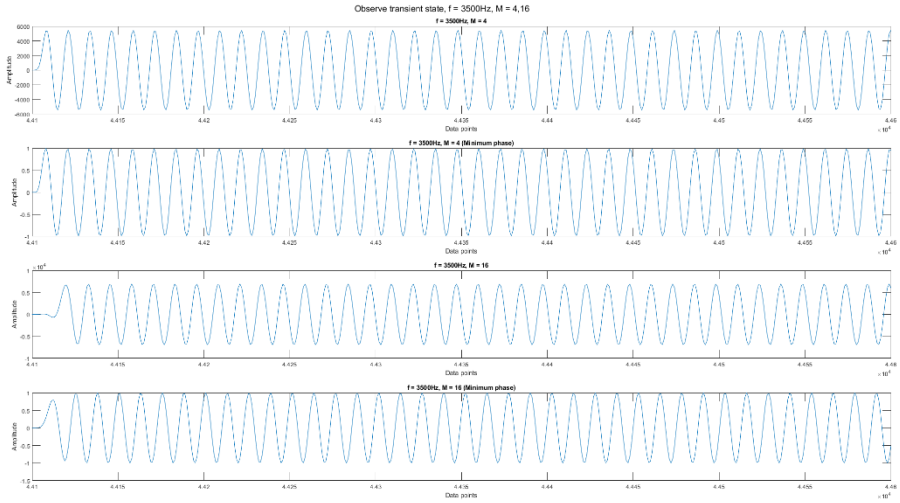
圖（八）：觀察暫態，f = 5000Hz, Order = 256,512,1024,2048

3.3. minimum_phase.m

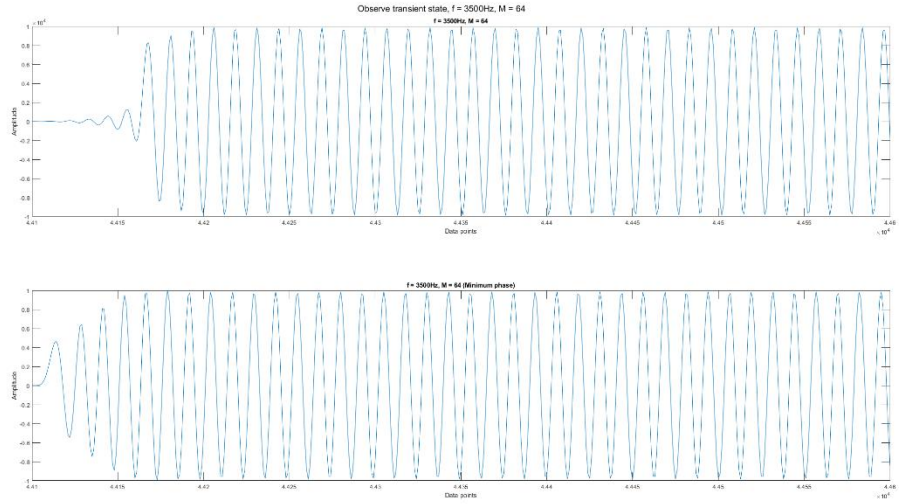
執行完 C code 並執行此檔案，此檔案透過 Matlab 的 function 來將 z-transform 超過 unit circle 的根找出來(poly2sym 將 coefficient 轉成 polynomial、root 求得 polynomial 的根、vpa 轉成高精準度的數字)。

找出來這些根之後，將這些根做共軛倒數(Conjugate Reciprocal)，使這些根能夠翻轉至 unit circle 內。

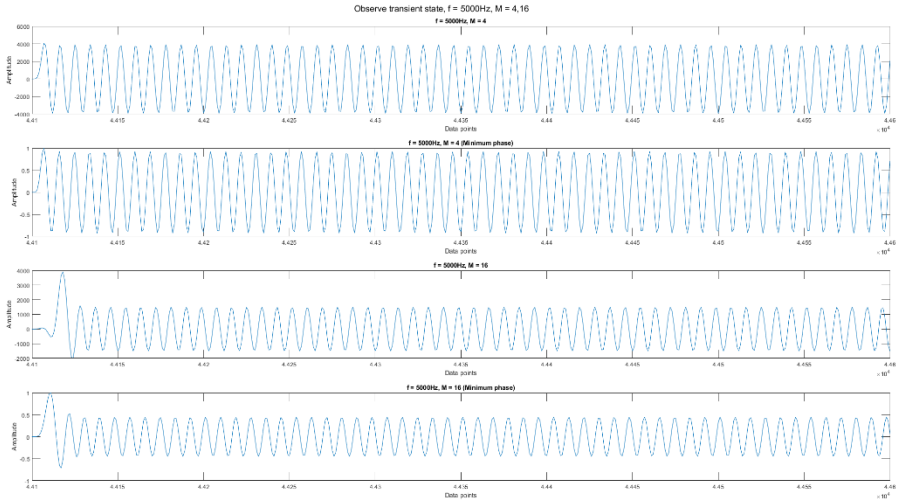
使所有根都在 unit circle 內，此低通濾波器即為 minimum phase LPF。特性就是訊號能量都集中在前半部，並且會使原本 transient state 所造成的 delay 時間縮小。但原本 linear phase 所有頻率的 delay 時間都相同，minimum phase 則是某些頻率的 delay 時間會不相同，造成訊號若包含該頻率的訊號會延後不同時間最後可能破壞訊號原本的樣子。



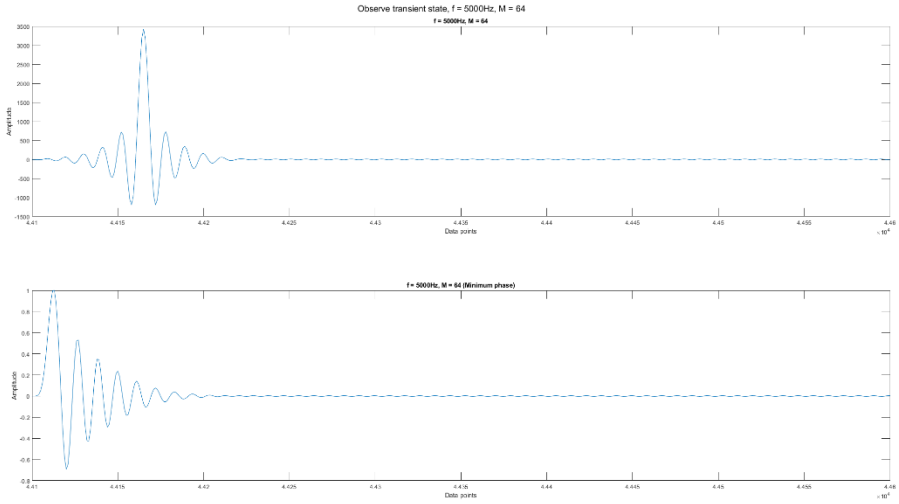
圖（九）：觀察經過 Minimum phase LPF 訊號圖型，f = 3500Hz, Order = 4,16



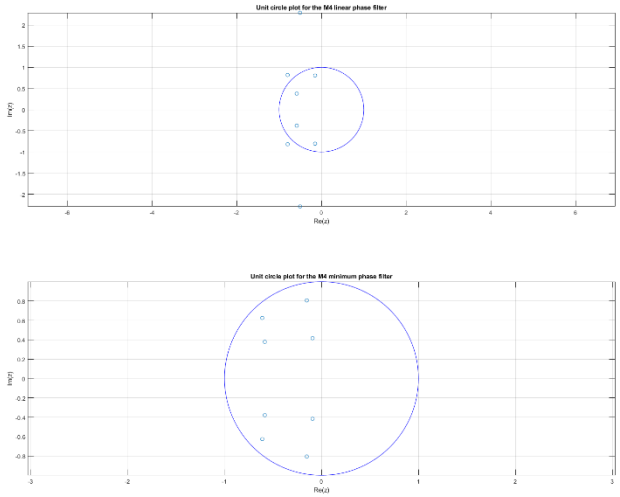
圖（十）：觀察經過 Minimum phase LPF 訊號圖型，f = 3500Hz, Order = 64



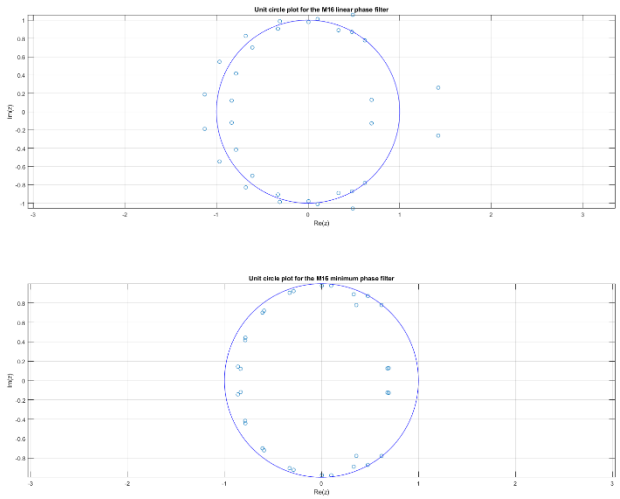
圖（十一）：觀察經過 Minimum phase LPF 訊號圖型，f = 5000Hz, Order = 4,16



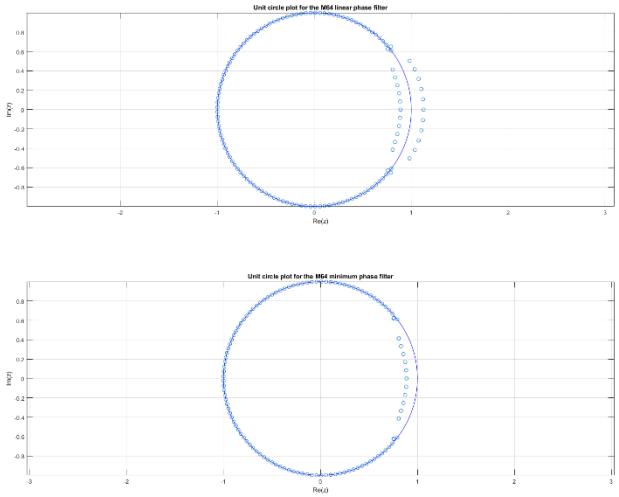
圖（十二）：觀察經過 Minimum phase LPF 訊號圖型，f = 5000Hz, Order = 64



圖（十三）：Unit Circle M=4



圖（十四）：Unit Circle M=16



圖（十五）：Unit Circle M=64