EECS2020_HW2

108032053 陳凱揚

1. Convolution sum and echo time estimation – noise remover for echo time estimation

(a)假如我們想從含有雜訊的訊號中找出想要的訊號，則設計出的系統之 impulse response 應為欲得訊號的 reversal。因為在做$x[n]$與$h[n]$的 convolution 時，我們會先將$h[n]$做 reversal，再將$h[n]$向右平移與$x[n]$重疊，如下圖 1-1 所示，則當$h[n]$為$x[n]$的 reversal 時，$h[n]$與$x[n]$可以完全重合，使得乘積的總合最大，如下圖 1-2 所示。因此可透過$h[n]$與含雜訊之訊號做 convolution，在所得訊號之 peak 處，找出被 time delay 後之欲得訊號的位置，再由 time advanced 可得訊號之原始位置。

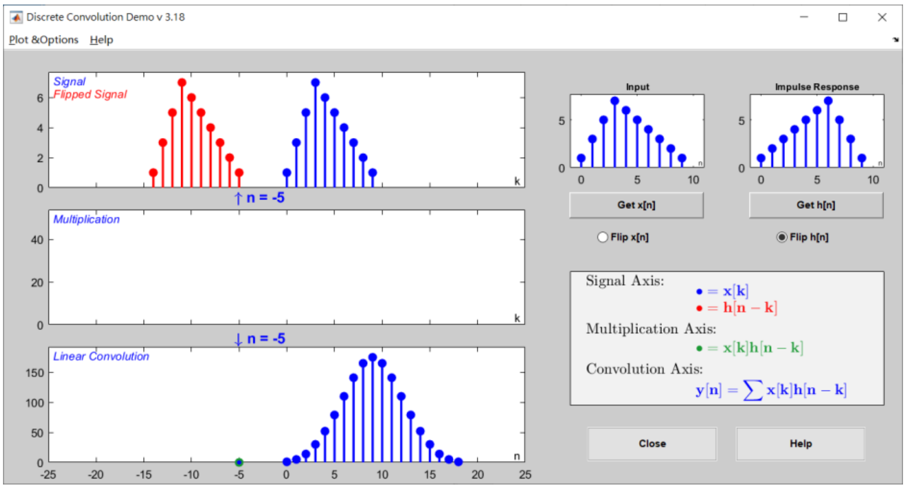

圖 1-1

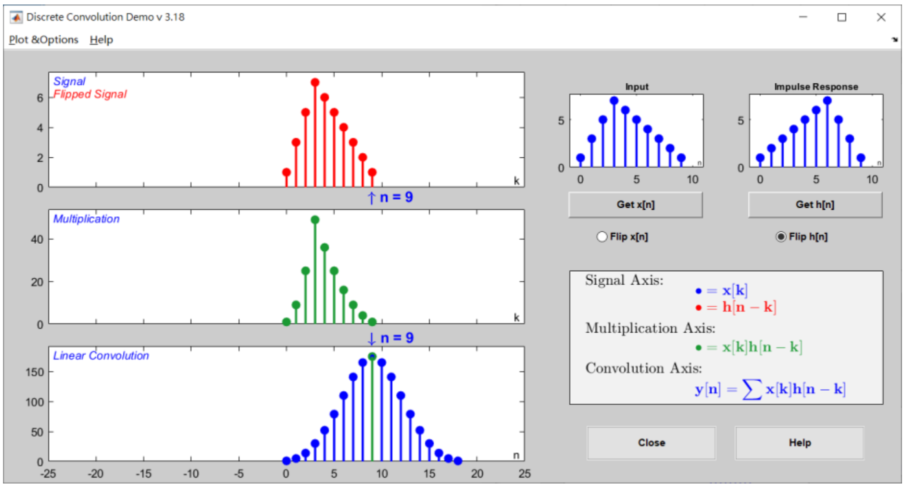

圖 1-2

(b)$MyConv()$函式之實作程式碼如下圖 1-3 所示。且分別使用$MyConv()$與 MATLAB 之內建函式$conv()$計算 Example 2.4，並將其輸出相減，比較其差異，結果如下圖 1-4，可知兩種函式之計算結果完全相同。

```
function [y, support_y] = MyConv(x, support_x, h, support_h)
        support_y = [0 0];
        support_y(1) = support_x(1)+support_h(1);
        support_y(2) = support_x(2)+support_h(2);
        x_len = length(x);
        h_len = length(h);
        y_len = x_len+h_len-1;
        y = zeros(1, y_len);
        for i = 1:x_len
            idx = i;
            for j = 1:h_len
                y(idx) = y(idx)+x(i)*h(j);
                idx = idx+1;
            end
        end
end
```

圖 1-3

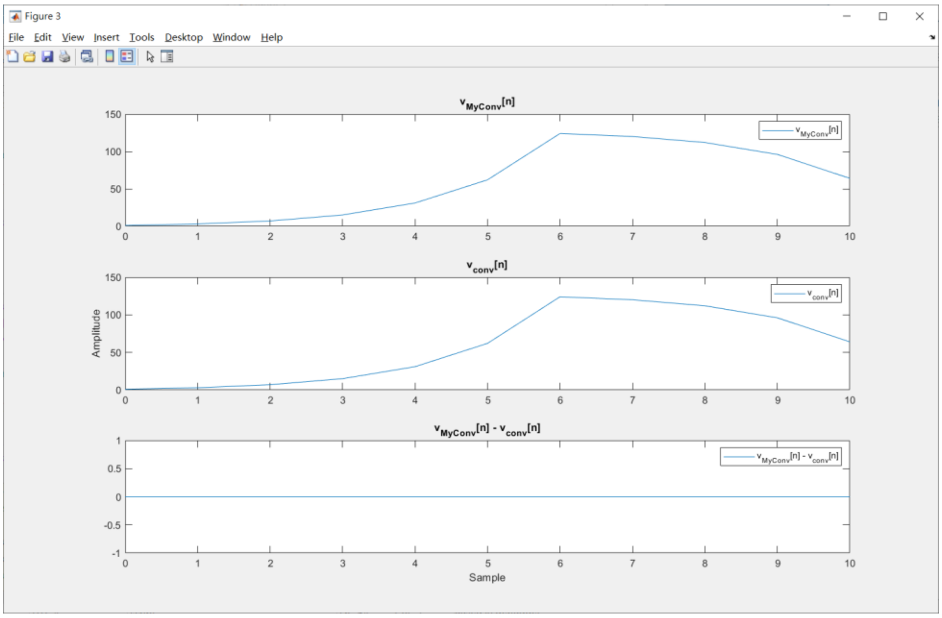

圖 1-4

(c)下圖 1-5、圖 1-6 為含雜訊的訊號經過 noise remover 後的結果，並以 envelope 找出$peak\ index = 2501$，且原始訊號$x[n]$與$h[n]$在 convolve 後，輸出的訊號$y[n]$會有 delay time $n_0 = (length(h) - 1)/2 = 500$，因此$echo\ time = (peak\ index - n_0) * (1/F_s) = 20.01(\sec)$

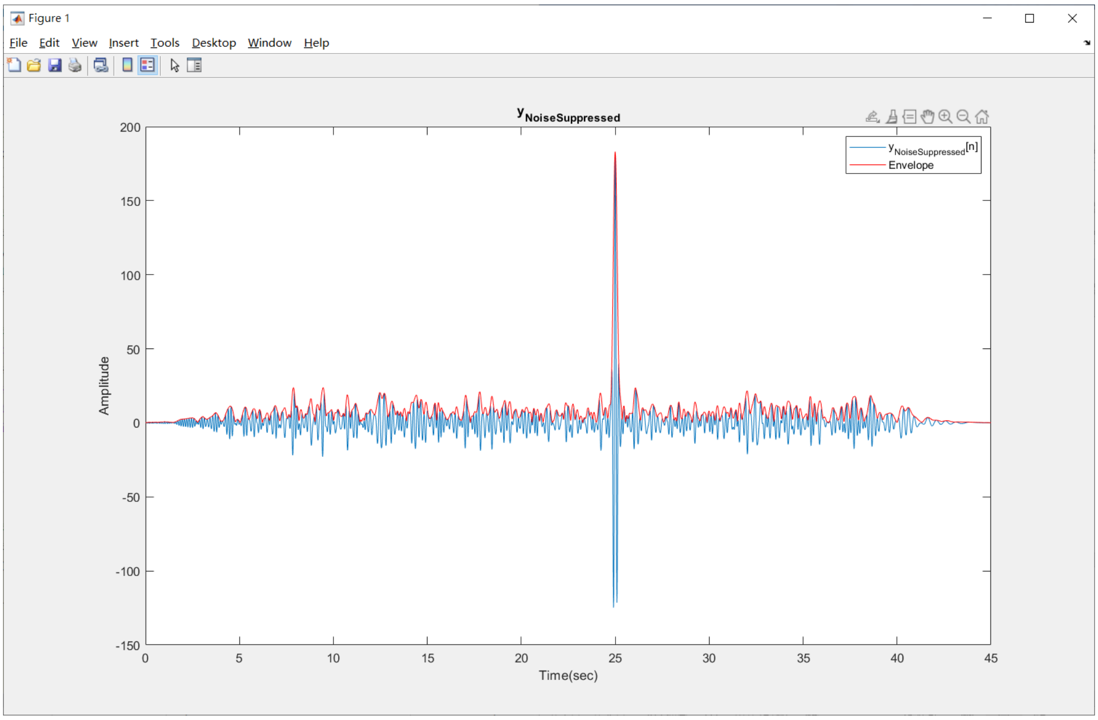

圖 1-5

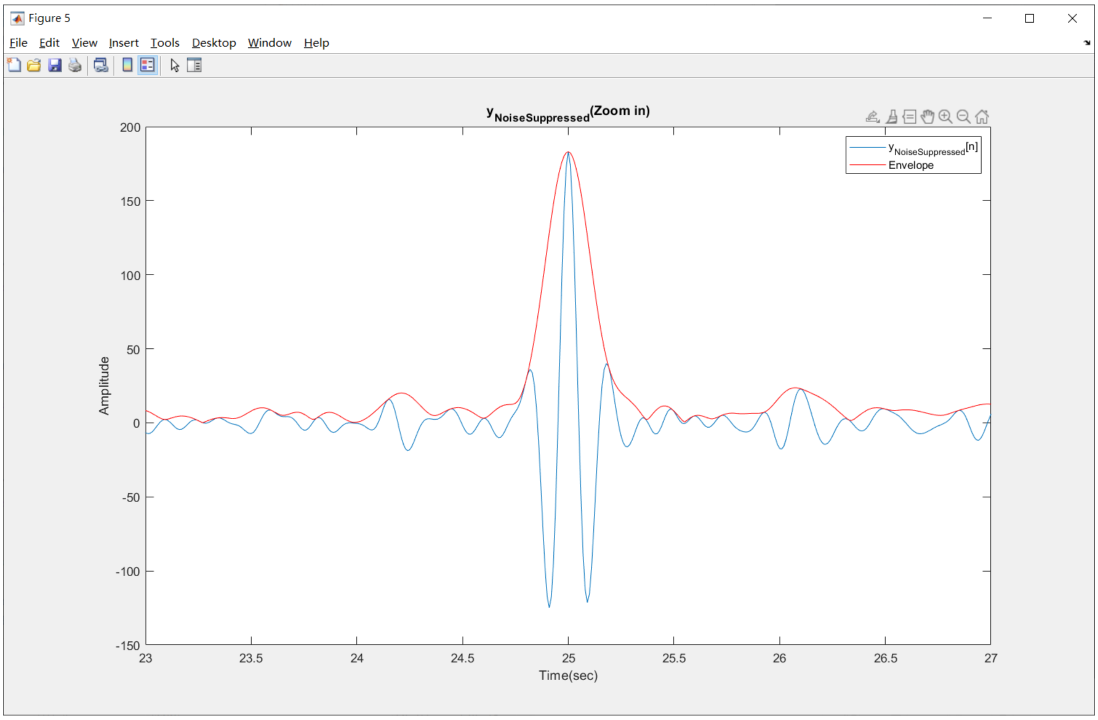

圖 1-6

(d)由於真實訊號為 continuous time，在經過 sampling 後，每個儲存的 sample 的間隔時間皆為$1/F_s$，因此在我們從 discrete time 訊號中找出 echo time 時，與真實訊號的 echo time 相比，其誤差應為$\pm 1/F_s$。

2. Reverberator implementation

(a)以下為推導 reverberator 之 impulse response 的過程：

$$y[n] = x[n] + ay[n-D]$$
$$= x[n] + a(x[n-D] + ay[n-2D])$$
$$= x[n] + ax[n-D] + a^2(x[n-2D] + ay[n-3D])$$
$$= x[n] + ax[n-D] + a^2x[n-2D] + a^3x[n-3D] + \cdots$$

$$x[n] = \delta[n]$$
$$\longrightarrow y[n] = \delta[n] + a\delta[n-D] + a^2\delta[n-2D] + a^3\delta[n-3D] + \cdots$$
$$\rightarrow h[n] = [1\ 0 \ldots 0\ a\ 0 \ldots 0\ a^2\ 0 \ldots 0\ a^3 0 \ldots], n_0 = 0$$
$$When\ a = 0.7, D = 5$$
$$\rightarrow h[n] = [1\ 0\ 0\ 0\ 0\ 0.7\ 0\ 0\ 0\ 0\ 0.49\ 0\ 0\ 0\ 0\ 0.343 \ldots], n_0 = 0$$

下圖 2-1 為使用 MATLAB 的$filter()$函式所作出的 impulse response，與我們的推導過程相符。

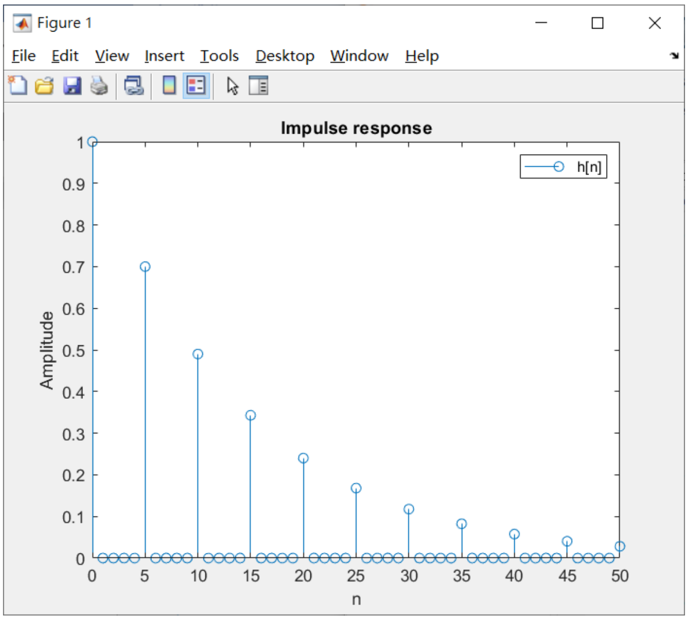

圖 2-1

(b)$y[n] = ay[n-D] + x[n]$，此為一 IIR 系統，無法實際操作此系統，但是當$-1 < a < 1$時，可將此系統近似為一 FIR 系統並實際運算：

$$y[n] = x[n] + ax[n-D] + a^2x[n-2D] + \cdots \cong \sum_{k=0}^{m} a^k x[n-kD]$$

若要在 real time 實施此系統的話，其必須為一 causal 系統，因此條件為$n \geq n - kD \rightarrow D \geq 0$，且$h[n] = 0,\ when\ n < 0$。

另外，使系統為 stable 的條件為$\sum|h[n]| < \infty$，此 reverberator 之

$$\sum_{k=0}^{\infty}|h[k]| = |1| + |a| + |a^2| + |a^3| + \cdots = \lim_{k\to\infty}\frac{1(1-|a|^k)}{1-|a|}$$

當$-1 < a < 1$，$\lim_{k\to\infty}\frac{1(1-|a|^k)}{1-|a|} = \frac{1}{1-|a|} < \infty$，否則$\lim_{k\to\infty}\frac{1(1-|a|^k)}{1-|a|} = \infty$。
因此當$-1 < a < 1$，此 reverberator 為一 stable 系統，而D的值並不影響是否為 stable 系統，只代表取樣的間隔距離而已。

(c)下圖 2-2 顯示了分別以$Myconv()$與$conv()$函式對音檔作 convolution 後的結果完全相同。而在$y[n] \cong \sum_{k=0}^{M} a^k x[n-kD]$的 FIR 近似過程中，我決定$M$的方法為計算$a^M < 0.001$之$M$的最小正整數解，此計算意義為當$x[n-kD]$的比重小於千分之一時，則忽略不記。此系統的$M = 20$，因此當$k \geq M$，則$a^k x[n-kD] \cong 0$，我們即可以$y[n] \cong \sum_{k=0}^{19} a^k x[n-kD]$，實際計算此系統。

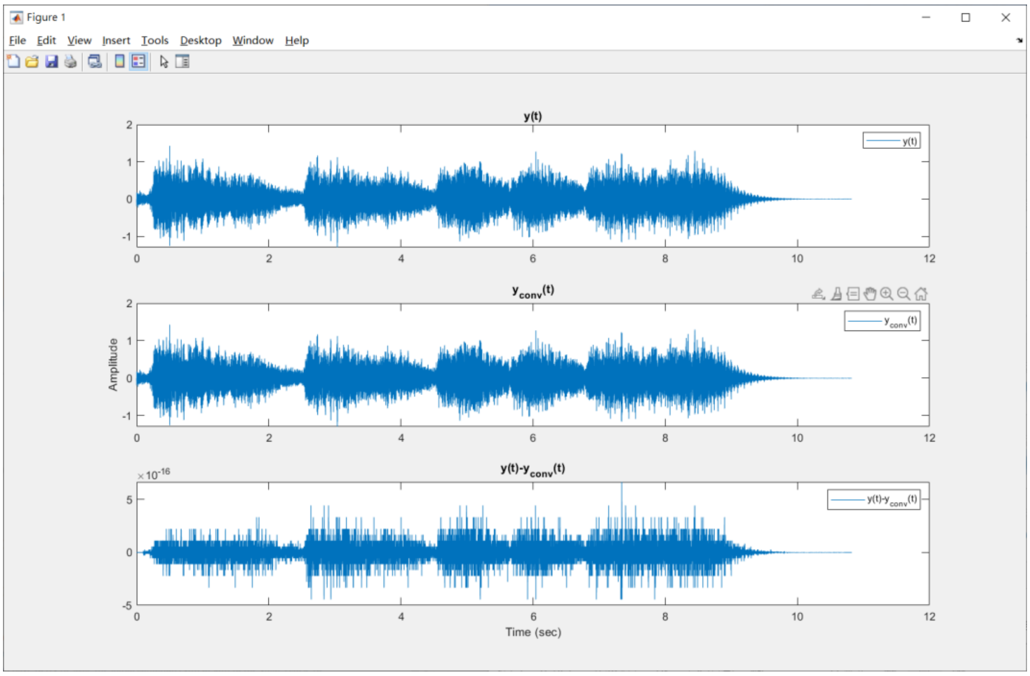

圖 2-2

下圖 2-3 為$x(t)$和$y(t)$的比較圖，可以發現$y(t)$的振幅大多大於$x(t)$的振幅，而$x(t)$的聲音聽起來較為單薄，而在經過 reverberator 的$y(t)$則聽起來有迴音感、較為飽滿。另外，由下圖 2-4 的放大比較圖，可以發現當$t < 0.1$時，$x(t) = y(t)$，這是因為我們使用$MyConv()$的 initial condition 為$y[n] = 0, x[n] = 0,\ when\ n < 0$，所以在$t < \tau$時，$x(t) = y(t)$。而在大約九秒之後的$x(t)$與$y(t)$的差異也漸漸變小，這是因為作為 input 的$x(t)$已經結束，output 的$y(t)$只剩下迴音留存。

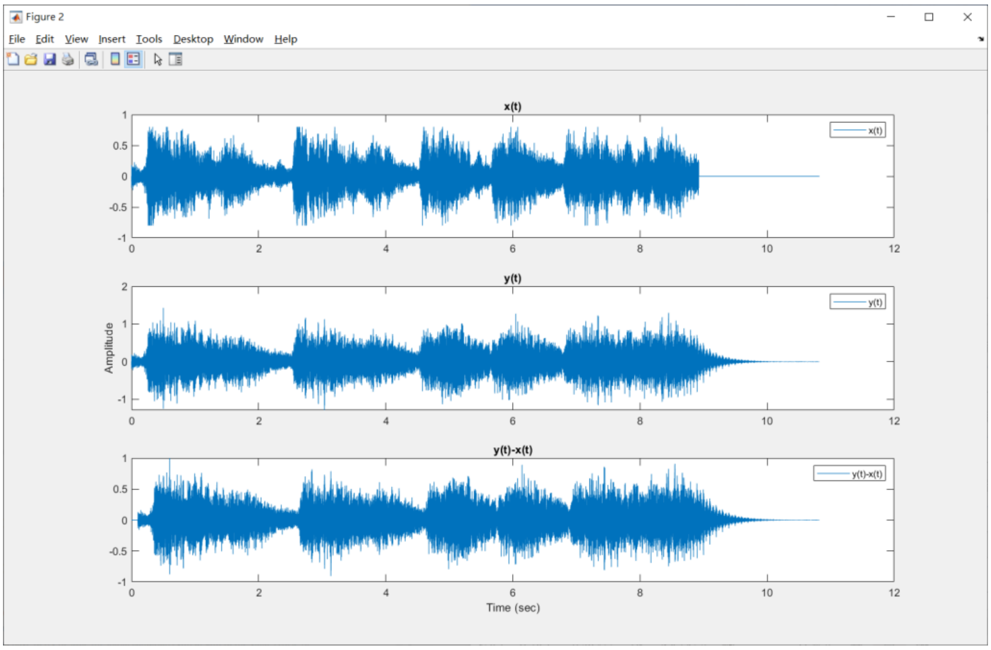

圖 2-3

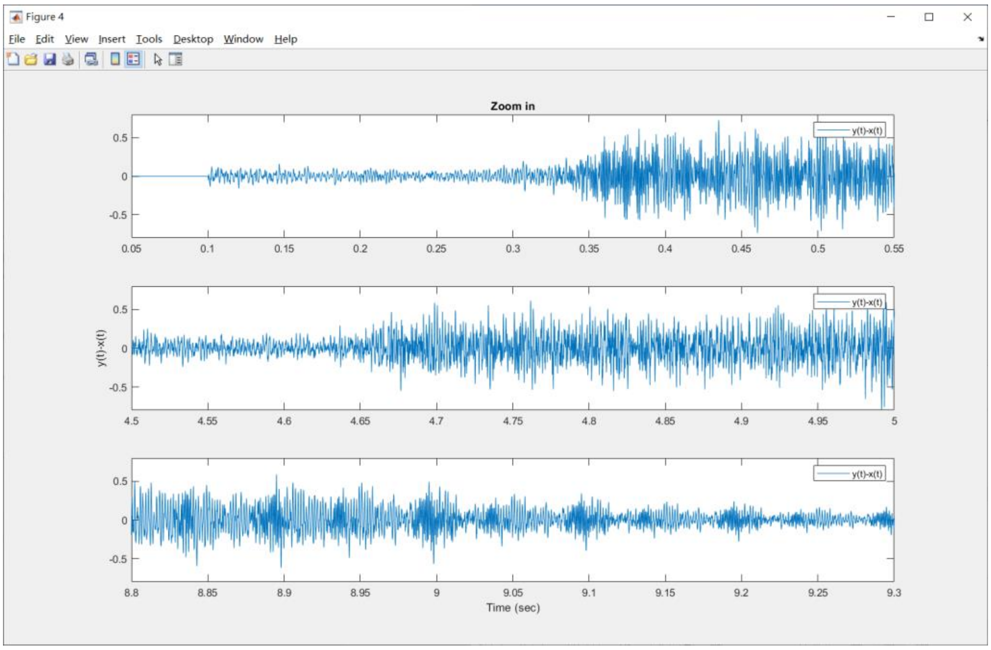

圖 2-4

(d)由下圖 2-5 可看出我們以 LCCDE 做出的結果與利用$filter()$函式做出的結果完全相同。在下圖 2-6 的$x(t)$與$y(t)$的比較圖，用肉眼看起來與(c)小題的比較圖幾乎相同，而在下圖 2-7 即顯示了以(c)小題的 FIR 系統與以(d)小題的 IIR 系統所得到的 output 幾乎相同，證明了此兩種方法所得出的結果可視為相同。但在下圖 2-8 可看出其仍有約10^{-3}的些微差距，這是因為將 IIR 系統近似於 FIR 系統的過程時，捨棄項所造成的結果。另外，在此題我使用的 initial condition 與(c)小題所使用的相同，都為$y[n] = 0, x[n] = 0,\ when\ n < 0$。

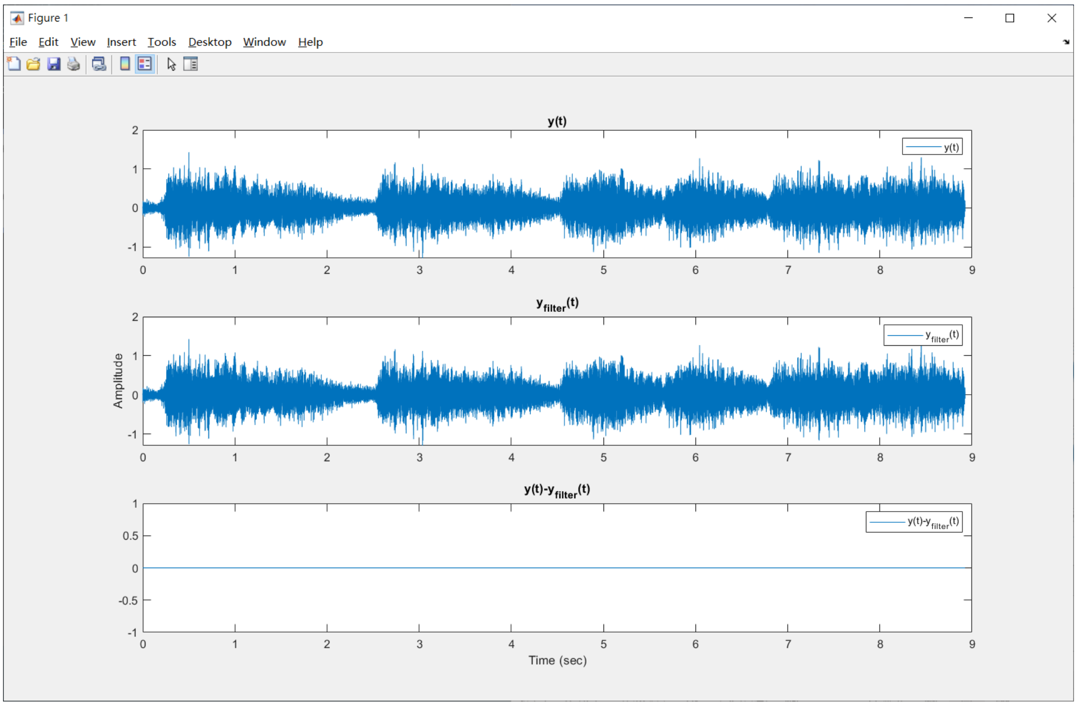
Figure 1
File Edit View Insert Tools Desktop Window Help
y(t)
y(t)
y_{filter}(t)
y_{filter}(t)
Amplitude
y(t)-y_{filter}(t)
y(t)-y_{filter}(t)
Time (sec)

圖 2-5

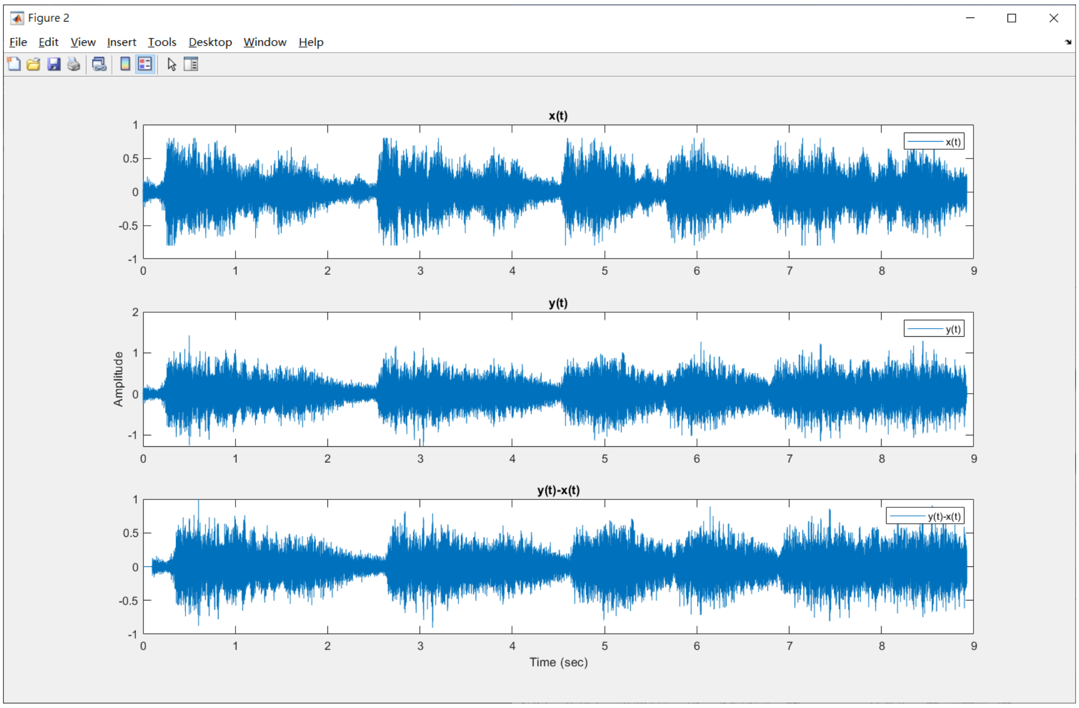
Figure 2
File Edit View Insert Tools Desktop Window Help
x(t)
x(t)
y(t)
y(t)
Amplitude
y(t)-x(t)
y(t)-x(t)
Time (sec)

圖 2-6

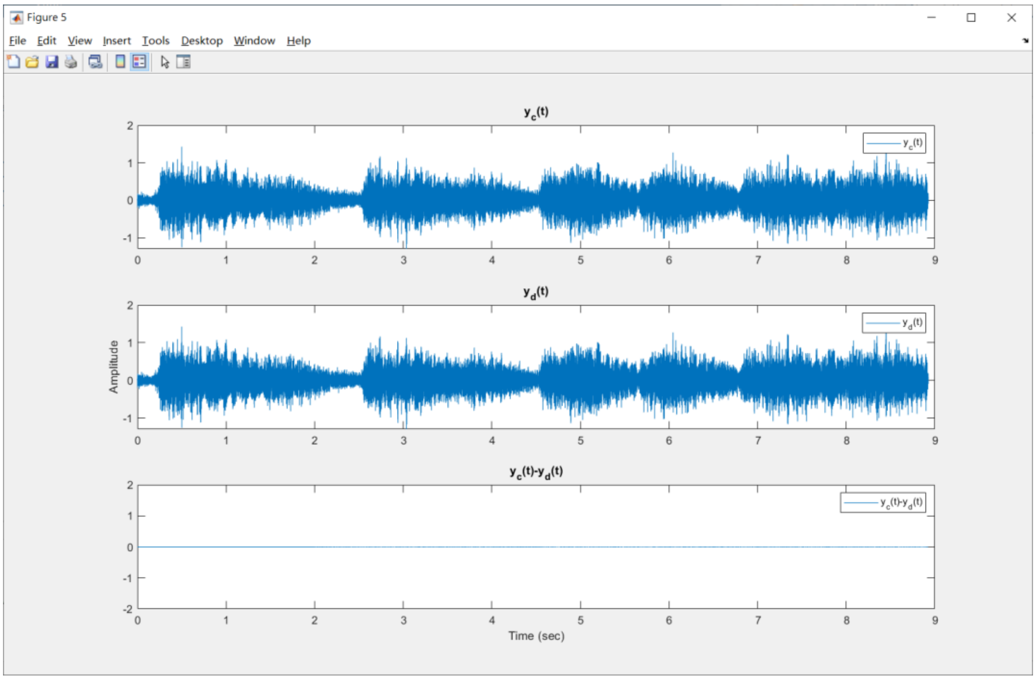

圖 2-7

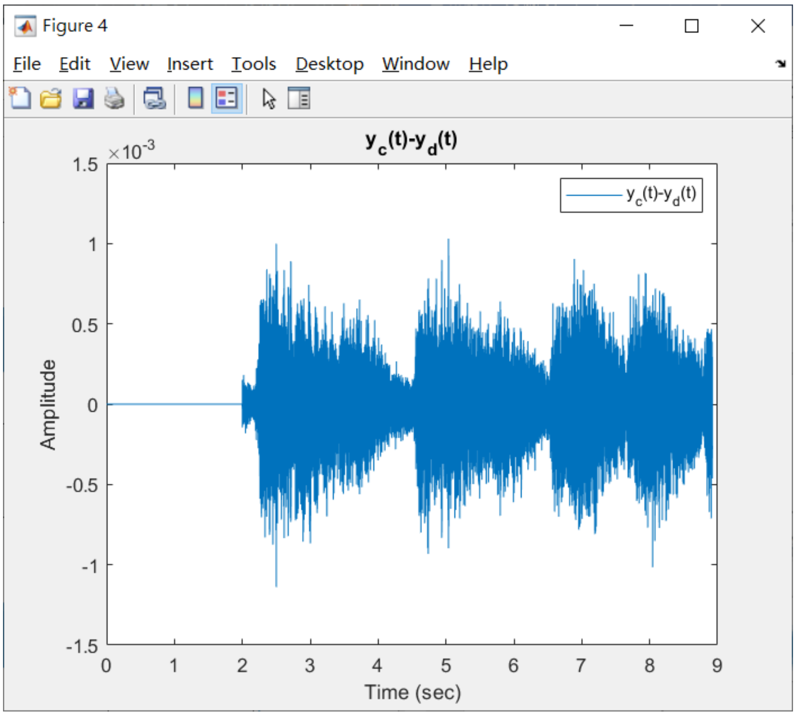

圖 2-8

(e)在(b)小題時，我們得出一個 stable 的 reverberator 系統的條件為 $-1 < a < 1$，因此在此題我將a分別設為1.05和−1.05，即可得到一個 unstable 的系統，其波形如下圖 2-9 所示，數值會隨著時間，以指數成長。此時的音檔 output 在一開始的短暫時刻還能聽出正常的聲音，但隨著時間的增加，$y(t)$振福也隨著指數上升，數值很快的就超越了電腦硬體所能播出的最大音量，變成了當機的聲音，聽起來像是一種槍聲在不斷的重複撥放。

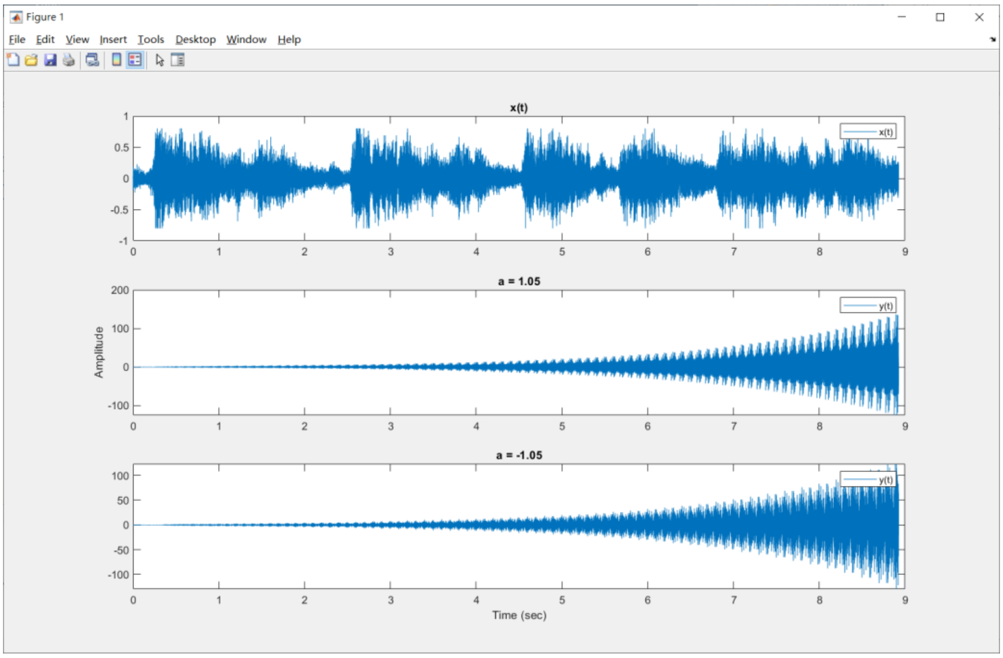

圖 2-9

(f)在之前的小題中，我們都將 initial condition 設為$y(t) = 0, x(t) = 0, when\ t < 0$，而在這個小題，我們在$t < 0$時將$x(t)$分別設為1, -1和一個介於$(-1, 1)$的一個隨機數，並比較不同的 initial condition 對輸出的影響。由下圖 2-10 和放大後的圖 2-11 可以看出 initial condition 主要影響在一開始的輸出，大約在$0 \leq t \leq 0.2$，為 2 個τ的範圍，這是因為$a^k = (0.7)^k$，當$k > 2$時，其影響力就越來越小了，越難越看得出差別。這也可以由$y(t) = y_{tr}(t) + y_{ss}(t)$來解釋，initial condition 主要影響的是 transient response，也就是一開始的輸出。

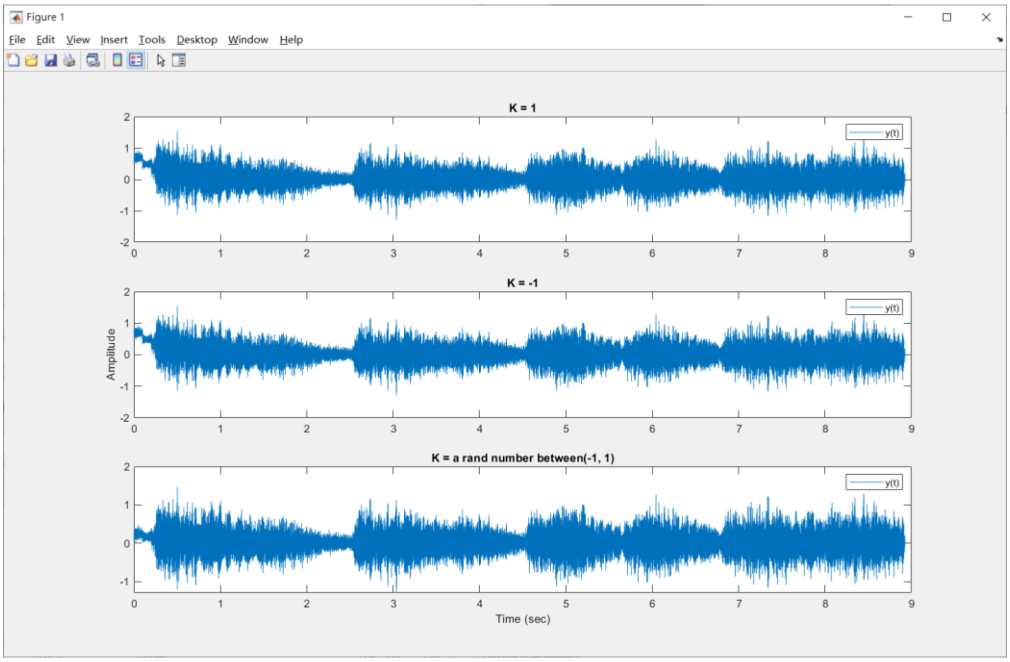

圖 2-10

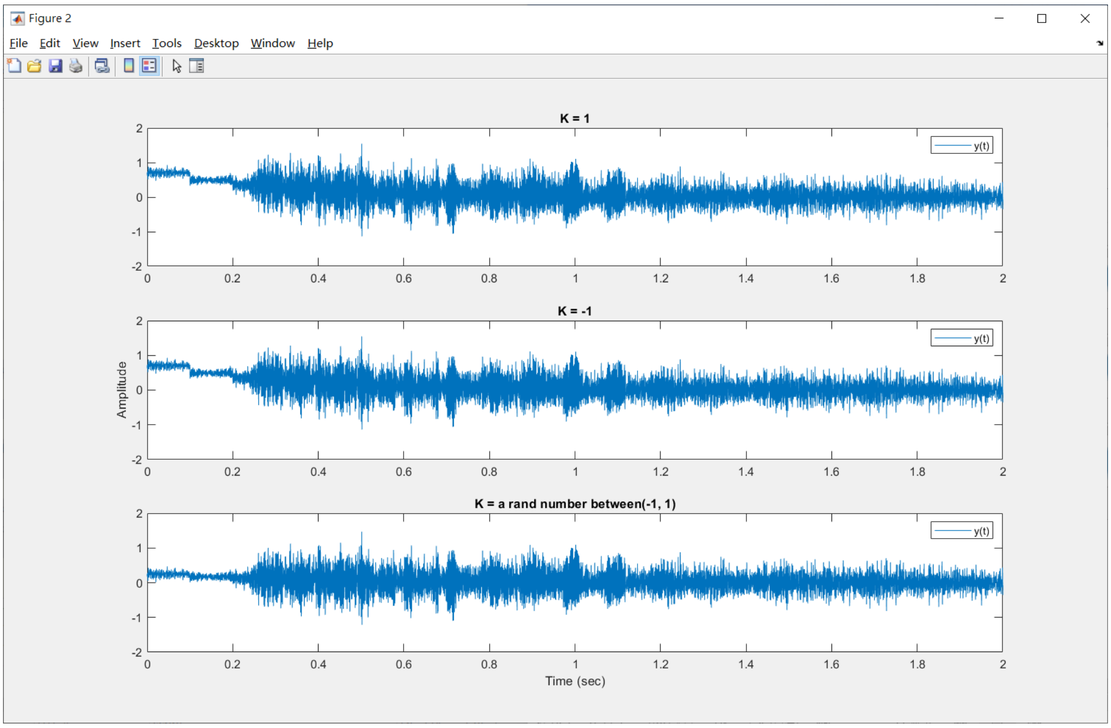

圖 2-11

(g)在這個小題我選擇了 3 個不同的F_s來做 magnitude response，並由於$D=\tau F_s$，因此不同的F_s就有不同的D值，即代表不同的 reverberator，藉此找出不同系統的 magnitude impulse 差異。由圖 2-12 可看出圖形於$(0,F_s/2)$具週期性，而此週期T與D有關，關係式為$TD=1$，T的單位為F_s，D沒有單位，例：當$F_s=50\rightarrow D=\tau F_s=5\rightarrow T=1/D=0.2(F_s)$。

透過此關係式，我們可以知道當頻率$f_0=k*\frac{1}{D}$，$k\in N,\ f_0\in(0,\frac{1}{2})$時，此頻率會被系統最放大；當頻率$f_0=(k+\frac{1}{2})*\frac{1}{D}$，$k\in N,\ f_0\in(0,\frac{1}{2})$時，此頻率會被系統最縮小，這邊頻率$f_0$所使用的單位為$F_s$。

Bonus：在(g)小題我們只考慮區間$(0,F_s/2)$的頻率，這是因為 DT 訊號在頻率上具有週期性，也就是$x[\omega,n]=x[\omega+\omega_0,n],\ where\ \omega_0=2\pi$。以$\omega\in(0,2\pi)$作為討論，此時最高的頻率出現在$\omega=\pi$，最低的頻率出現在$\omega=0,\ 2\pi$，頻率以$\omega=\pi$為對稱，若要考慮到所有的頻率只需要取區間$\omega\in(0,\pi)$即可，對應到 CT 訊號的頻率則為$f_0\in(0,F_s/2)$。

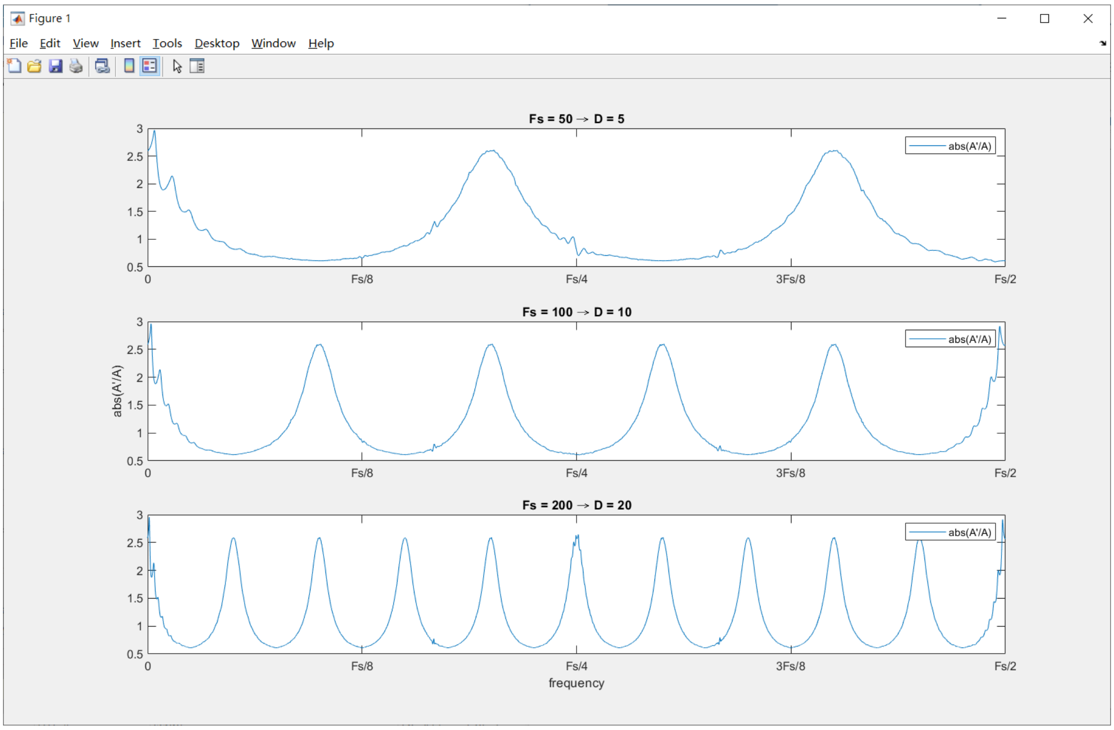
Figure 1
File Edit View Insert Tools Desktop Window Help
Fs = 50 → D = 5
abs(A'/A)
Fs = 100 → D = 10
abs(A'/A)
Fs = 200 → D = 20
abs(A'/A)
3
2.5
2
1.5
1
0.5
0
Fs/8
Fs/4
3Fs/8
Fs/2
frequency

圖 2-12