



數位信號處理實習

LAB4

電子工程系 蔡偉和 教授

100360318 四子三甲 陳奕璋 學生

2014/4/21

[練習 4-1] 將 Fig. 4-1 之訊號輸入一系統 $y[n] = 0.8y[n-1] + x[n] - x[n-1]$ ，試分別利用 filtering 與 convolution 計算輸出結果，並比較兩者在頻域上的差異。

1. 程式碼：

```
clear();
n = input('How many point of h[n] would like to compute?');

x = [2 3 4 5];

a = [1 -0.8]; % 分母係數
b = [1 -1];   % 分子係數
h(1) = 1;
for i = 2 : n
    h(i) = -1/4*0.8^(i-1);
end

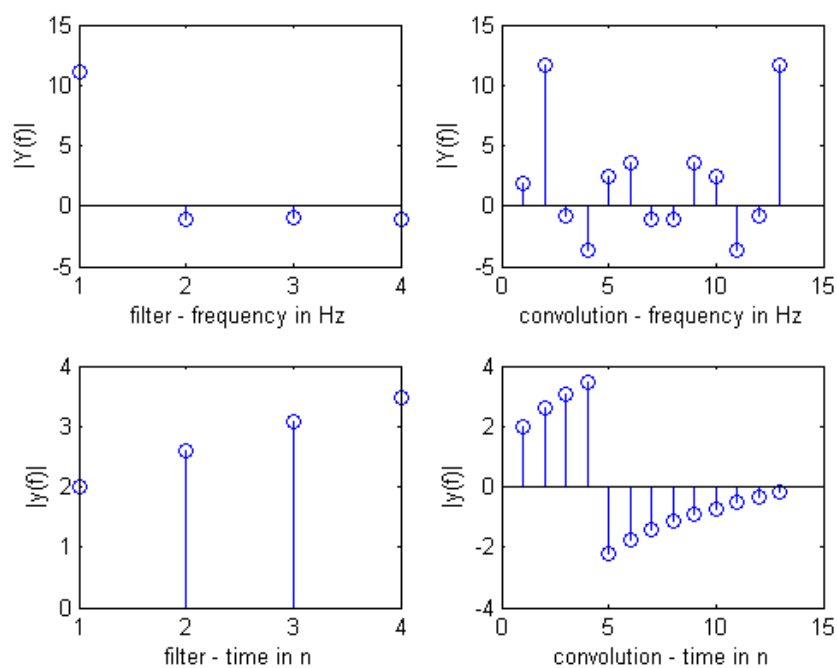
fout = filter(b, a, x);
cout = conv(h, x);
FOUT = fft(fout);
COUT = fft(cout);

% nF=1:length(x);
nC = 1 : n+length(x)-1;

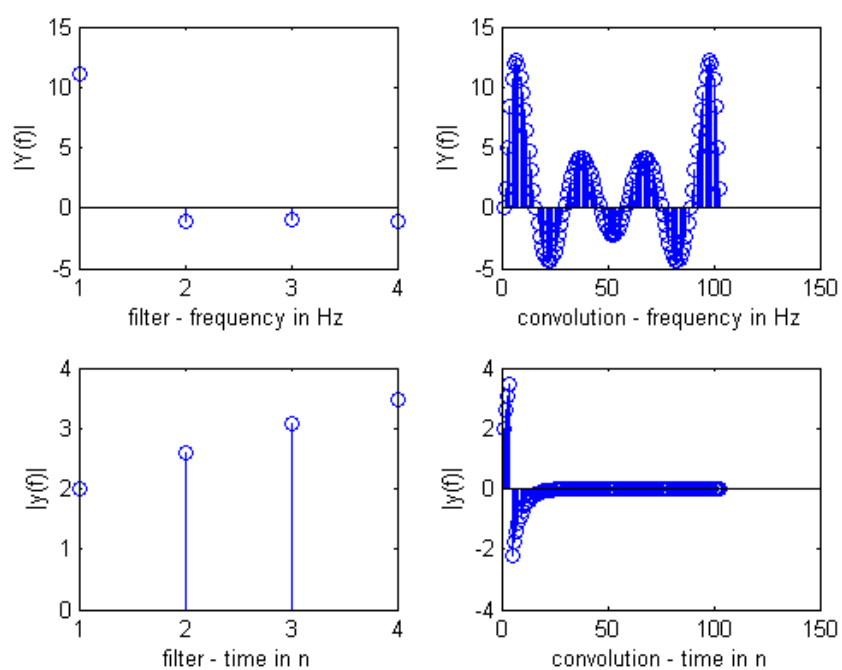
% fill '0' for the filter. Because the length of filter is same as input.
for ex = length(x)+1 : n+length(x)-1
    fout(ex) = 0;
    FOUT(ex) = 0;
end

% freq. domain. (Up)
subplot(2, 2, 1); stem(nC, FOUT);
xlabel('filter - frequency in Hz'); ylabel('|Y(f)|');
subplot(2, 2, 2); stem(nC, COUT);
xlabel('convolution - frequency in Hz'); ylabel('|Y(f)|');
% time domain. (Down)
subplot(2, 2, 3); stem(nC, fout);
xlabel('filter - time in n'); ylabel('|y(f)|');
subplot(2, 2, 4); stem(nC, cout);
xlabel('convolution - time in n'); ylabel('|y(f)|');
```

2. Matlab 波形圖：



圖一 10 點之 $H[n]$



圖二 100 點之 $H[n]$

3. 計算

根據題目之 $y[n]$ ，計算出 $h[n]$ ，如下

$$y[n] = 0.8y[n-1] + x[n] - x[n-1]$$

$$h[n] = (0.8)^n (u[n] - 0.8^{-1}u[n-1])$$

將 $h[n]$ 依序帶入數值，可得下表

n	$(0.8)^n$	$u[n] - 0.8^{-1}u[n-1]$
0	1	1
1	0.8	$1 - 1.25 = -1/4$
2	0.8^2	-1/4
3	0.8^3	-1/4
...
n	0.8^n	-1/4

根據觀察，可以列出 $h[n]$ 的通式 $h[n] = (-1/4)(0.8^n)$ ，要特別注意的是

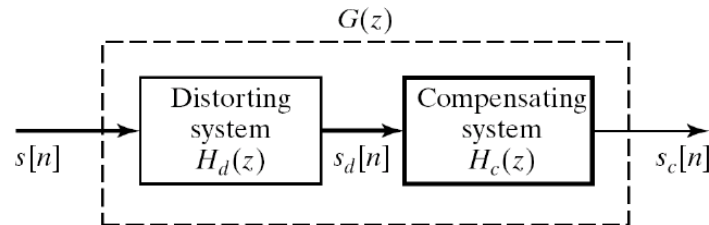
$n=0$ 時的值為 1。

4. 心得

我們可以觀察到，在點數多的狀況下，convolution 出的結果越精準，而 filter 與 convolution 的差異可發現，filter 輸出點數與輸入訊號點數相同，而 convolution 因為是無限多項，由使用者輸入欲求得數量有關。

[練習 4-2] 設計一如下圖之頻率補償系統 $H_d(z)$ (必需為 minimum phase system)，並將一具有 10Hz 與 30Hz 之弦波訊號 $s[n]$ 輸入一 Distorting system

$H_c(z) = \frac{1 - 6.9z^{-1} + 13.4z^{-2} - 7.2z^{-3}}{1 - 1.3z^{-1} + 0.47z^{-2} - 0.035z^{-3}}$ ，繪出 $s_c[n]$ 及其 DFT。



1. 程式碼：

```

clear;
clc;
num = [1 -6.9 13.4 -7.2];
den = [1 -1.3 0.47 -0.035];
pole = roots(num);
zero = roots(den);

ca = 1;
cb = 1;
for i = 1 : length(pole)
    if (abs(pole(i)) > 1)
        ca = ca * (-1) * pole(i);
        pole(i) = 1 / pole(i);
    end
end
for i = 1 : length(zero)
    if abs(zero(i)) > 1
        cb = cb * (-1) * zero(i);
        zero(i) = 1 / zero(i);
    end
end

mina = ca * poly(pole);
minb = cb * poly(zero);

f0 = 10;           % 10 Hz sine wave
f1 = 30;           % 30 Hz sine wave
T = 0.01;          % sampling period = 0.01 sec
N = 100;
n = 0 : 1 : N-1;
a = sin(2*pi*f0*n*T);
b = sin(2*pi*f1*n*T*0.1); % increase the 30Hz sampling rate.
x = a + b;

figure(1);
  
```

```

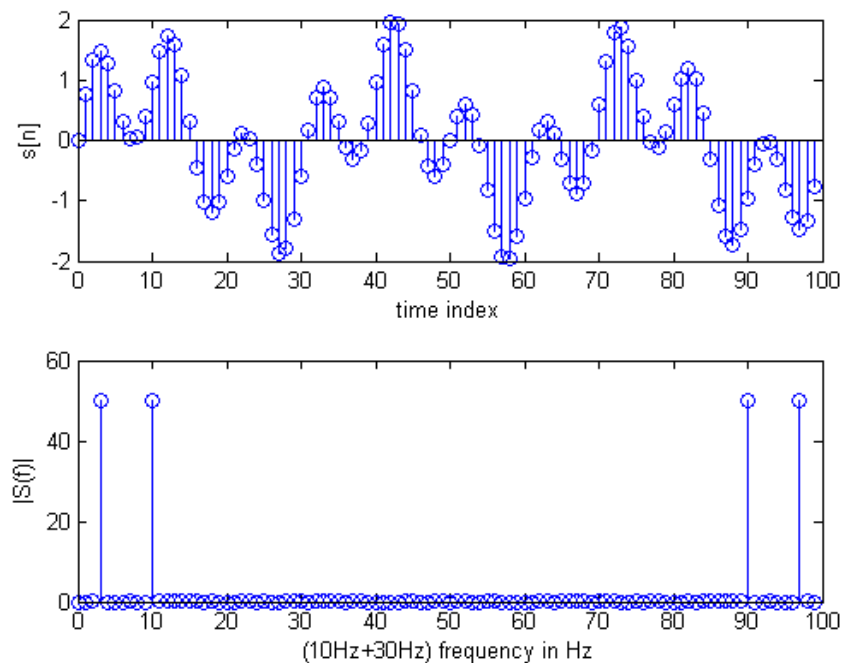
subplot(2, 1, 1); stem(n, x);
xlabel('time index'); ylabel('s[n]');
f = n / T / N;
subplot(2, 1, 2); stem(f, abs(fft(x)));
xlabel('(10Hz+30Hz) frequency in Hz'); ylabel('|S(f)|');

figure(2);
y = filter(num, den, x);
subplot(2, 1, 1); stem(y);
xlabel('time index'); ylabel('sd[n]');
subplot(2, 1, 2); stem(f, abs(fft(y)));
xlabel('(Distorting) frequency in Hz'); ylabel('|Sd(f)|');

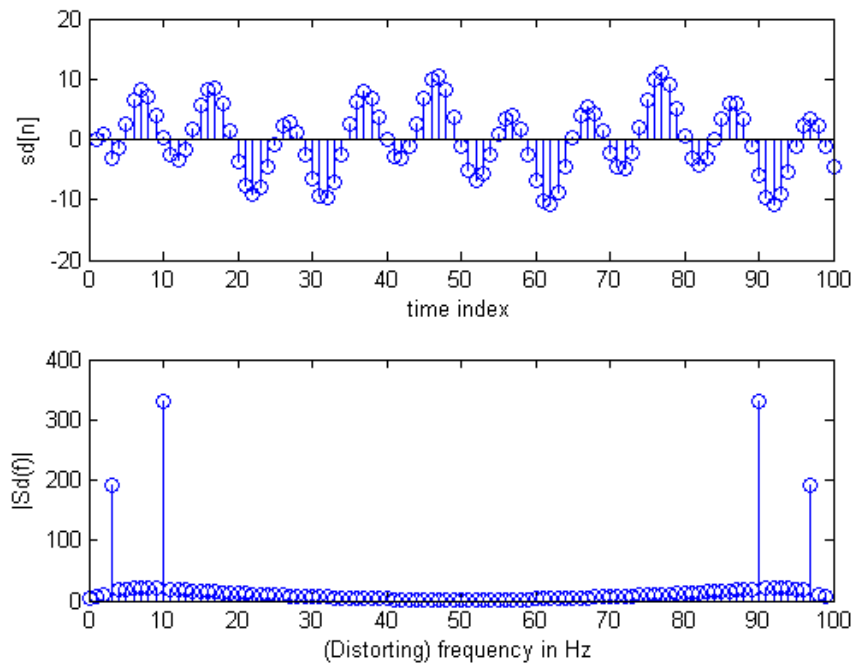
figure(3);
y2 = filter(minb, mina, y);
subplot(2, 1, 1); stem(y2);
xlabel('time index'); ylabel('sc[n]');
subplot(2, 1, 2); stem(f, abs(fft(y2)));
xlabel('(Compensation) frequency in Hz'); ylabel('|Sc(f)|');

```

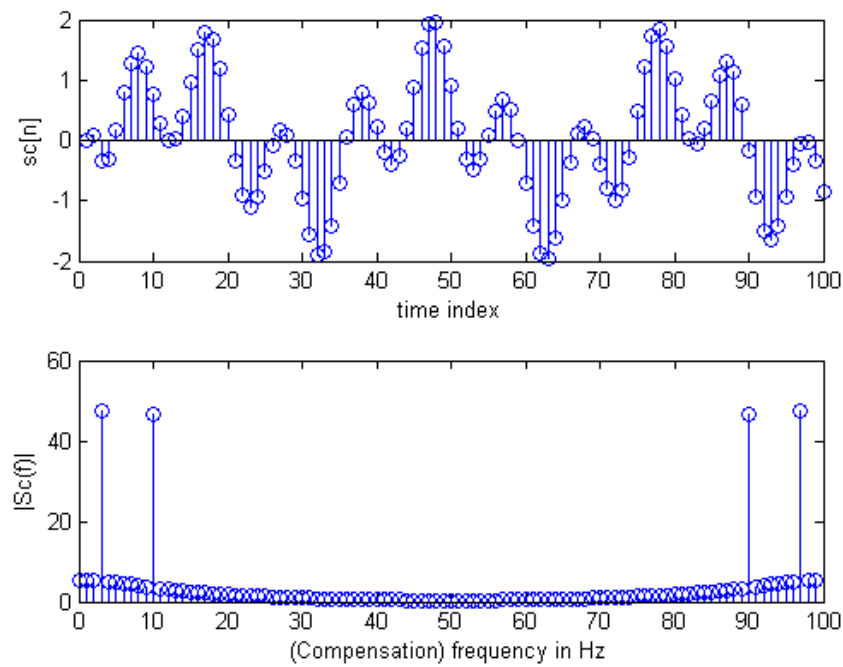
2. Matlab 波形圖：



圖三 10Hz+30Hz 之離散弦波



圖三 經過 Distorting System 後之波形



圖三 經過補償後之波形

3. 心得

因為所有系統都可以拆為 All Pass System 與 Minimum phase System，而 Minimum phase System 是 Stable 的，因此可用此特性補償。

我們可以發現補償後的結果和原訊號相比有相位移，這是因為當 All Pass System 相位移等於 0 時，由 Minimum Phase System 所產生的相位移。

經過這題的練習，我覺得這樣的應用在通訊系統的實務上應該很常運用，在此題中的 Distorting 就像是實際系統中的干擾與訊號在空間反射所造成的失真，而接收端也許就可以透過此方法校正。