國立臺北科技大學
National Taipei University of Technology

數位信號處理實習
實驗報告

電子系三丙

姓名：游鎮遠

學號：107360734

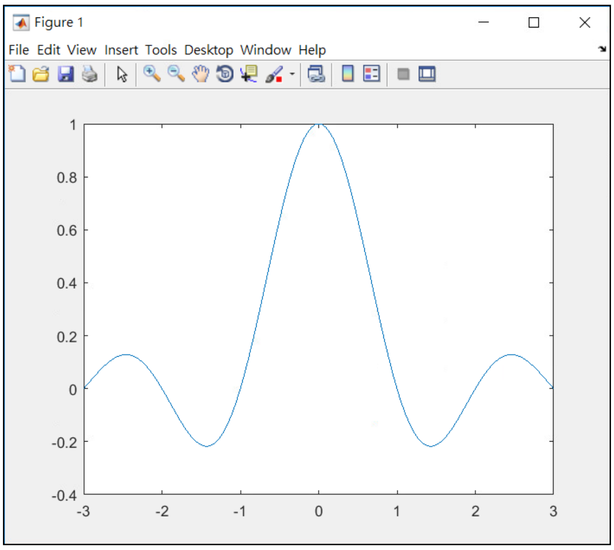
中華民國 110 年 6 月 18 日

實驗一

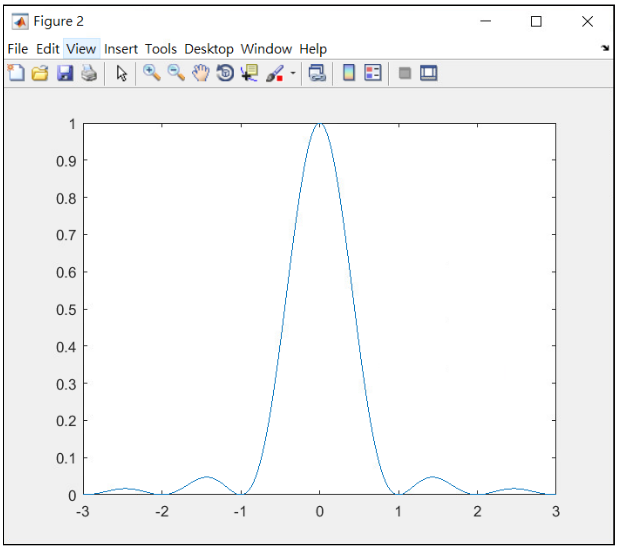
題目：

[Practice 1-1] Sketch signal $\mathrm{sinc}(t)=\dfrac{\sin \pi t}{\pi t}$ and $\mathrm{sinc}^2(t)$ for $-3 \le t \le 3$。

實驗結果：



Sinc(t)



Sinc^2(t)

分析討論：

這題原本我的做法是直接將 Sinc(t)和 Sinc^2(t)的數學公式寫到 matlab 裡，心想應該就可以畫出 Sinc(t)和 Sinc^2(t)的波形圖，但後來發現在 t=0 時 sinc 函數的值是被另外定義為 1，matalb 無法自行判斷，所以我將結果分成兩個 t=0 點和中間可判斷的圖形合併，就可以成功地畫出 Sinc(t)和 Sinc^2(t)的波形圖。

程式碼：

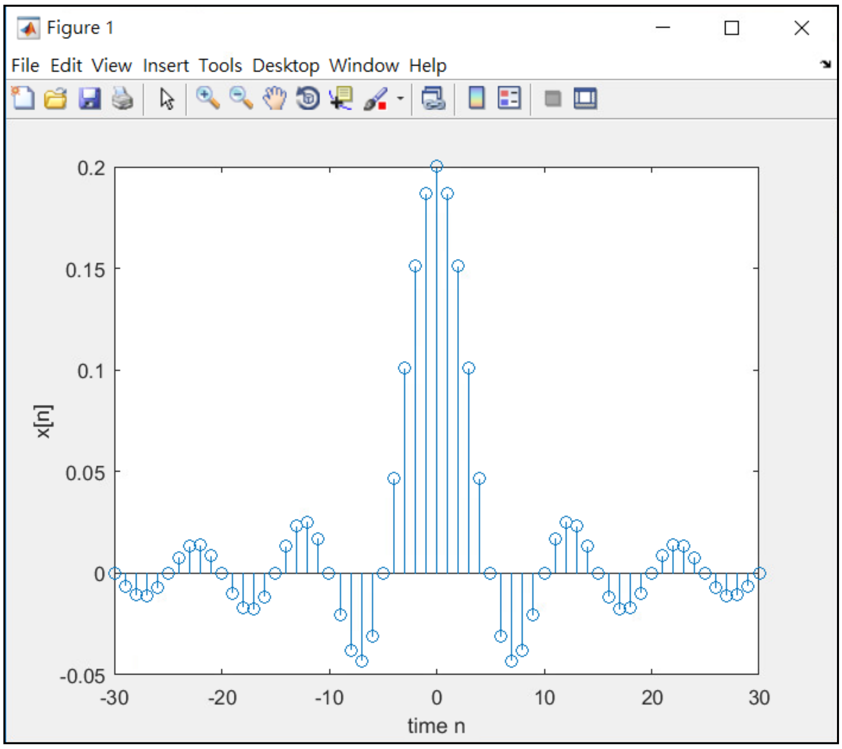
```
clear;
close all;
dt=0.01;

t=dt:dt:3;
xa=sin(pi*t)./(pi*t);

t2=-3:dt:-dt;
xb=sin(pi*t2)./(pi*t2);

xc=1;
xd=[xb xc xa];
t3=-3:dt:3;
figure
plot(t3,xd);
```

```
xe=xd.*xd;
t3=-3:dt:3;
figure
plot(t3,xe);
```

題目：

[Practice 1-2] Sketch signal $x[n]=\dfrac{\sin w_c n}{\pi n}$, where $w_c=0.2\pi,\ -30\le n\le 30$ 。

實驗結果：



$$x[n]=\frac{\sin w_c n}{\pi n},\ \text{where}\ w_c=0.2\pi,\ -30\le n\le 30$$

分析討論：

這題是一個類似 sinc 的離散信號，與連續訊號的自變數是連續的不同，離散訊號是一個序列，即表示其自變數是「離散」的。

在這題的練習中我學會了使用 x(isnan(x))，將缺少的值直接定義，把 n=0 值 0.2 畫出來。

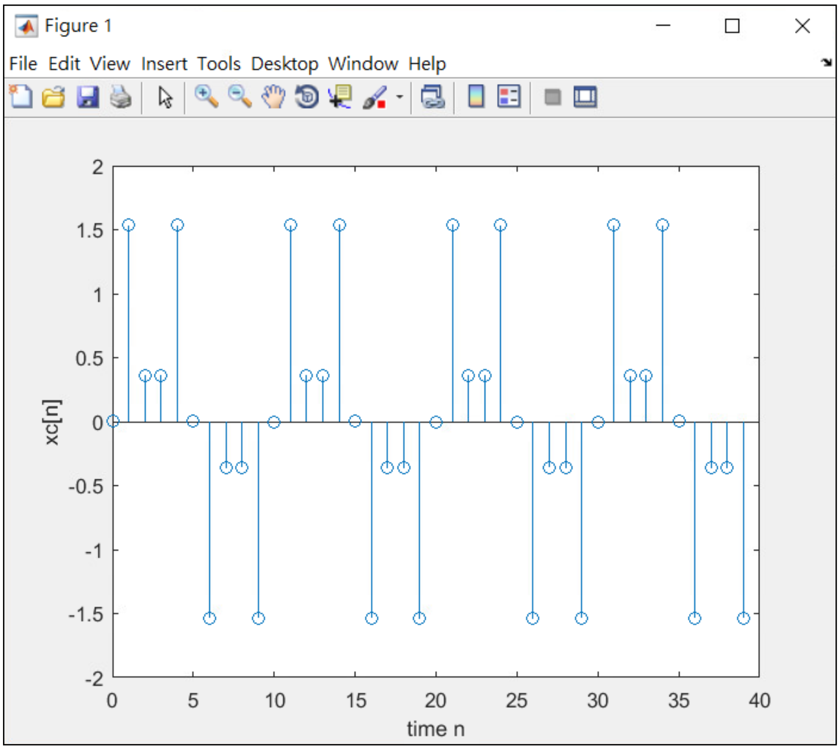
程式碼：

```
clear;
close all;

n=-30:1:30;
x=sin(0.2*pi*n)./(pi*n);
x(isnan(x))=0.2;
stem(n,x);
xlabel('time n'); ylabel('x[n]');
```

題目：

[Practice 1-3] Sketch a discrete-time signal that consists of 10Hz and 30Hz sine components based on a sampling period of 0.01 second.

實驗結果：



discrete-time signal that consists of 10Hz and 30Hz sine components based on a sampling period of 0.01 second.

分析討論：

本題的一開始，程式先設定時間長度(Total length)為 0.4 秒，取樣週期(sampling period)為 0.01 秒，分別將 10Hz 和 30Hz 的 sine 取樣，再將這兩個離散信號加起來並畫出。在觀察實驗結果圖時，已經無法判別出 sine 的形狀，但在詢問老師過後發現這樣是對的。

程式碼：

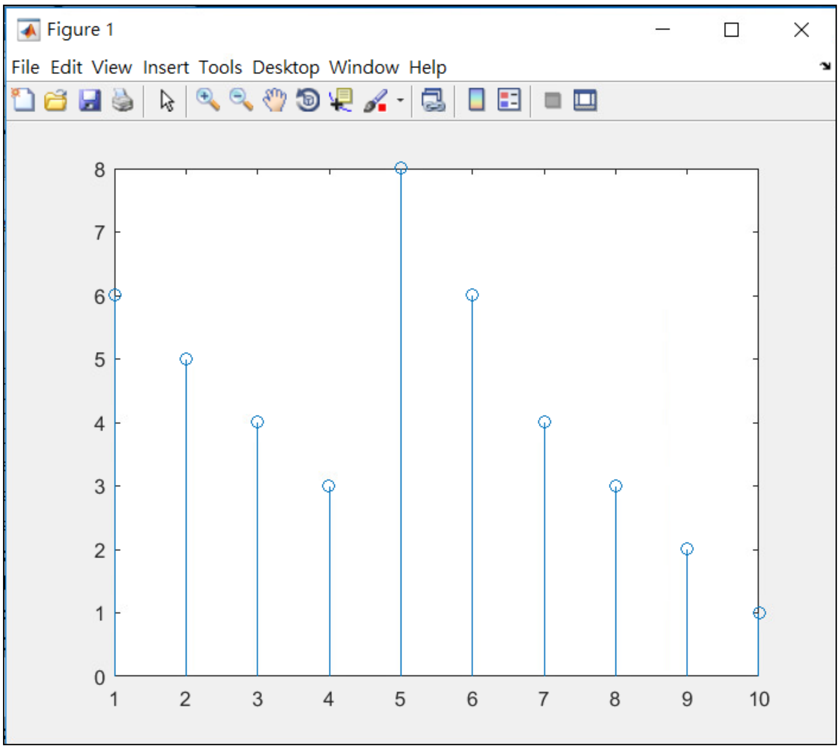
```
clear;
close all;
Length=0.4 % Total length =0.4 sec
T=0.01; % sampling period = 0.01 sec
N=Length/T;
n=0:1:N-1;
x=sin(2*pi*10*n*T);
xb=sin(2*pi*30*n*T);
xc=x+xb;
stem(n,xc);
xlabel('time n'); ylabel('xc[n]');
title('discrete signal x[n]=xa(nT), where T = 0.01 sec');
```

實驗二

題目：

[Practice 2-1] Implement the convolution without using function conv().

實驗結果：



分析討論：

這題是要基於不使用 matlab 的 conv()函數，將兩個離散信號做摺積。利用摺積的定義，以矩陣和兩個 For 迴圈將摺積實現，這題是比較有難度的一題，但還好在和同學討論後有做出來。

程式碼：

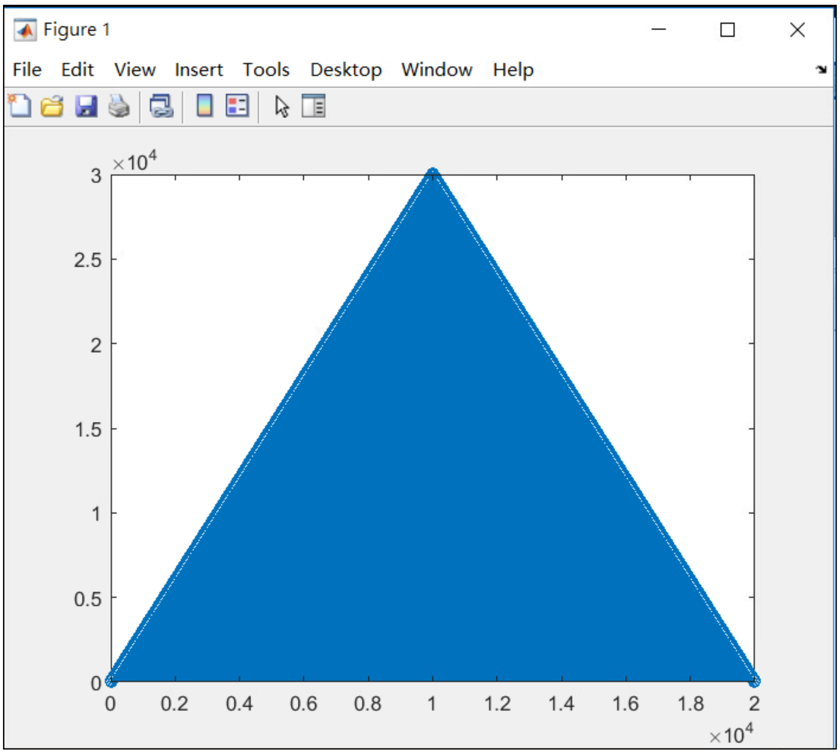
```
clear;
t0=cputime;
x1=[6 5 4 3 2 1];
x2=[1 0 0 0 1];
len1 = length(x1);
len2 = length(x2);
y = zeros(1,length(x1)+length(x2)-1);
x1 = [x1 zeros(1,len2-1)];
x2 = [x2 zeros(1,len1-1)];
t1=cputime-t0;
for n=1:len1+len2
    for k = 1:n-1
        y(n-1) = y(n-1) + x1(k) * x2(n-k);
    end
end
```

```
n=1:length(y);
stem(n,y);
t2=cputime-t0;
```

題目：

[Practice 2-2] There are two signals $x_1[n] = n \% 5,\quad x_2[n] = n \% 4,\quad 1 \le n \le 10000.$
Implement their convolution using the matrix multiplication approach.

實驗結果：



分析討論：

這題是將兩個 n=10000 的規律性離散信號，以 matlab 的 circshift 來實現摺積。在程式執行時，因為這一題的計算量太大了，所以實驗結果圖執行很久才出來，其結果是一個完美的正三角形。

程式碼：

```
clear;
t0=cputime;
x1 = zeros(1,10000);
x2 = zeros(1,10000);
for i=1:10000
    x1(i) = mod(i,5);
    x2(i) = mod(i,4);
end

x1 = fliplr(x1);
len1 = length(x1);
len2 = length(x2);
```

```
x1 = [x1 zeros(1,len2-1)];
x2 = [zeros(1,len1-1) x2];
mat = zeros(length(x1),length(x2));
t1=cputime-t0;
y = zeros(1,length(x1));

for i=1:length(x1)
    mat(i,:) = circshift(x1,i-1);
    y(i) = circshift(x1,i-1) * x2';
end
t2=cputime-t0;

n=1:length(y);
stem(n,y);
t3=cputime-t0;
```

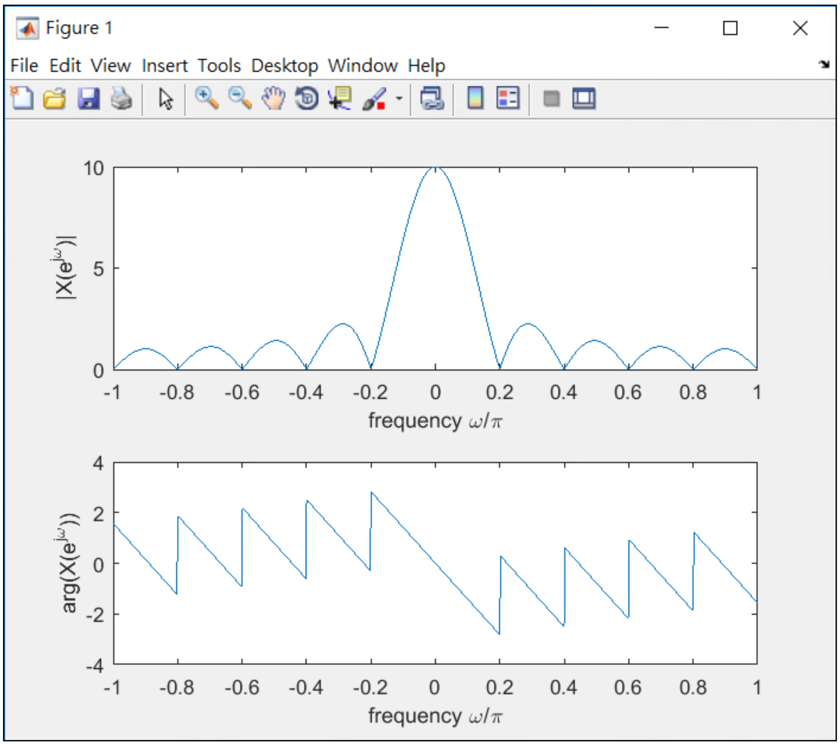
實驗三

題目：

[Practice 3-1] Implement the DTFT without using functions exp(), abs(), and angle(), but instead, using the following expressions.

$X(e^{jw}) = \sum_{n=-\infty}^{\infty} x[n]e^{-jwn} = \sum_{n=-\infty}^{\infty}\{x[n]\cos(wn) - jx[n]\sin(wn)\} = X_R(e^{jw}) + jX_I(e^{jw})$, i.e., real and imaginary parts, and its magnitude and phase are $\sqrt{X_R^2(e^{jw}) + X_I^2(e^{jw})}$ and $\tan^{-1}\left[\frac{X_I(e^{jw})}{X_R(e^{jw})}\right]$, respectively.

實驗結果：



分析討論：

　　這題是利用 DTFT 的定義和數學計算公式，將一個方波的離散信號做離散時間的傅立葉傳換，並觀察實驗結果圖。方波的離散信號經過 DTFT 後，呈現出 sinc 形式的頻譜圖，和老師上課時講的 DTFT 理論相符合，代表我們寫的程式是正確的，讚。

程式碼：

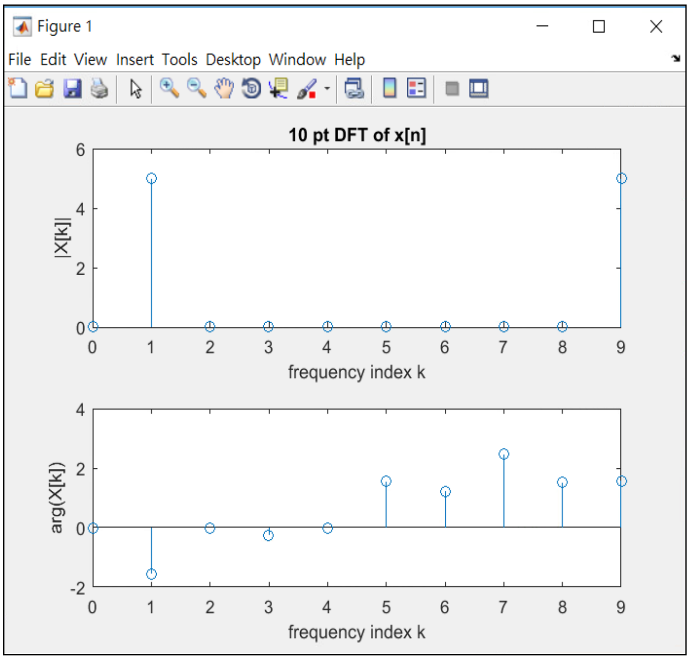
```
clear;
x=[1 1 1 1 1 1 1 1 1 1];
n=0:length(x)-1;
K=500;
k=-K:K;
w=pi*k/K;
XR=(x*cos(n'*w));
XI=(-j*x*sin(n'*w));
```

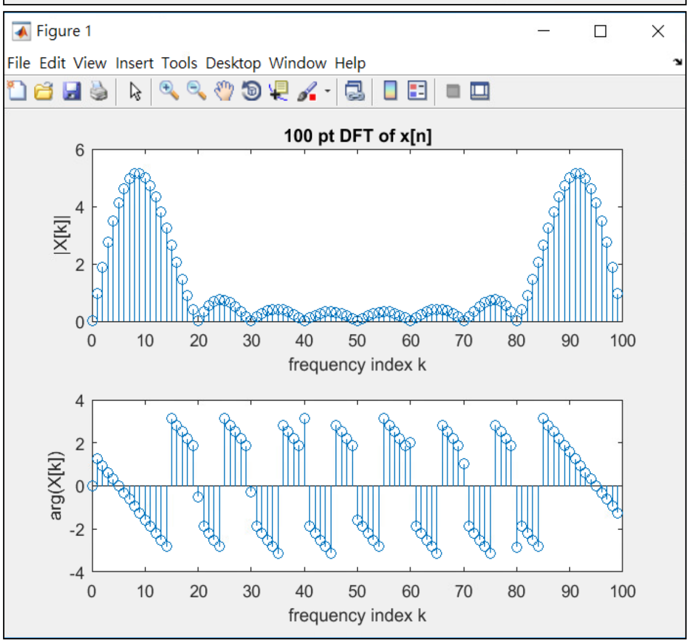
```
magX=sqrt((XR.^2)+((XI/j).^2));
angX=atan2(XI/j,XR);
title('DTFT of x[n]');
subplot(2,1,1); plot(w/pi,magX);
xlabel('frequency \omega/\pi'); ylabel('|X(e^j^\omega)|');
subplot(2,1,2); plot(w/pi,angX);
xlabel('frequency \omega/\pi'); ylabel('arg(X(e^j^\omega))');
```

題目：

[Practice 3-2] Take one period of the sine signal in Fig. 1-2, and then sketch its 10-pt DFT and 100-pt DFT, respectively.

實驗結果：





分析討論：

這題是利用一個 10Hz 的 sine 信號，取出一個週期做 DFT，並觀察轉換後的頻譜圖，結果 0 到 100Hz 的頻率都有值。聽完老師的講解後，我們瞭解到一個週期的 sine 信號，會擁有很多種頻率的值，不會只存在一種頻率的值，除非 sine 信號是無限長的。

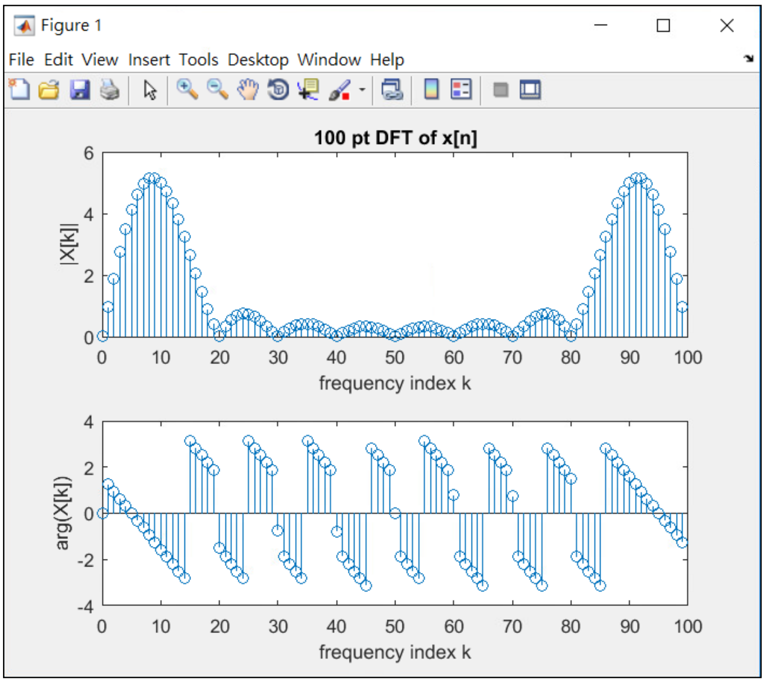
程式碼：

```
f0=10; % 10 Hz sine wave
Length=0.1; % Total length =0.1 sec
T=0.01; % sampling period = 0.01 sec
q=Length/T;
n=0:1:q-1;
x=sin(2*pi*f0*n*T);
n=0:length(x)-1;
N=100;
k=0:N-1;
X=x*exp(-j*2*pi/N*n'*k);
magX=abs(X);
angX=angle(X);
stem(k,magX);
subplot(2,1,1); stem(k,magX); xlabel('frequency index k');
ylabel('|X[k]|');
title('100 pt DFT of x[n]');
subplot(2,1,2); stem(k,angX); xlabel('frequency index k');
ylabel('arg(X[k])');
```

題目：

[Practice 3-3] Use function fft() to compute the DFT in Practice 3-2 and sketch the DFT.

實驗結果：



分析討論：

做完這題的實驗，雖然結果和 Practice 3-2 的結果一樣，但我學習到了使用 matlab 內建的程式 fft()做 FFT，而不用使用像上一題的公式做法(X=x*exp(-j*2*pi/N*n'*k);)，且藉由這一個內建程式，我們便可以更快速地求得信號的 DFT 結果。

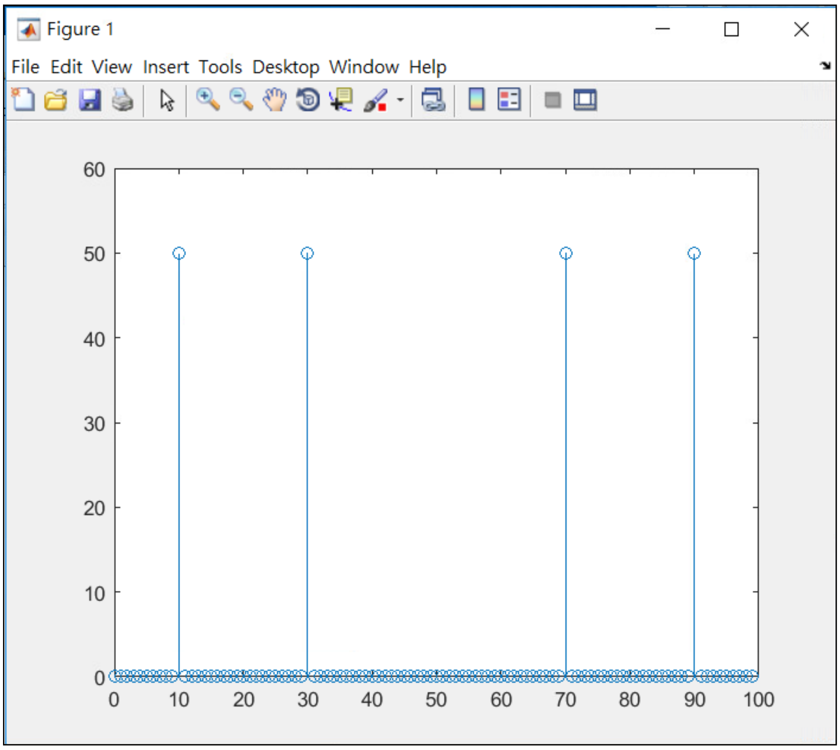
程式碼：

```
f0=10; % 10 Hz sine wave
Length=0.1; % Total length =0.1 sec
T=0.01; % sampling period = 0.01 sec
q=Length/T;
n=0:1:q-1;
x=sin(2*pi*f0*n*T);
n=0:length(x)-1;
N=100;
k=0:N-1;
X=fft(x,N);
magX=abs(X);
angX=angle(X);
stem(k,magX);
subplot(2,1,1); stem(k,magX); xlabel('frequency index k');
ylabel('|X[k]|');
title('100 pt DFT of x[n]');
subplot(2,1,2); stem(k,angX); xlabel('frequency index k');
ylabel('arg(X[k])');
```

題目：

[Practice 3-4] Use function fft() to compute the DFT of the signal in Practice 1-3, i.e., a discrete-time signal that consists of 10Hz and 30Hz sine components based on a sampling period of 0.01 second.

實驗結果：



分析討論：

這題的程式一開始先將 10Hz 和 30Hz 的 sine 信號做取樣，取樣週期為 0.01 秒，將取樣好的兩種信號加起來後，使用 fft()做 DFT，然後將實驗結果畫出，發現頻譜圖中 10Hz 和 30Hz 有值，代表程式是正確的。

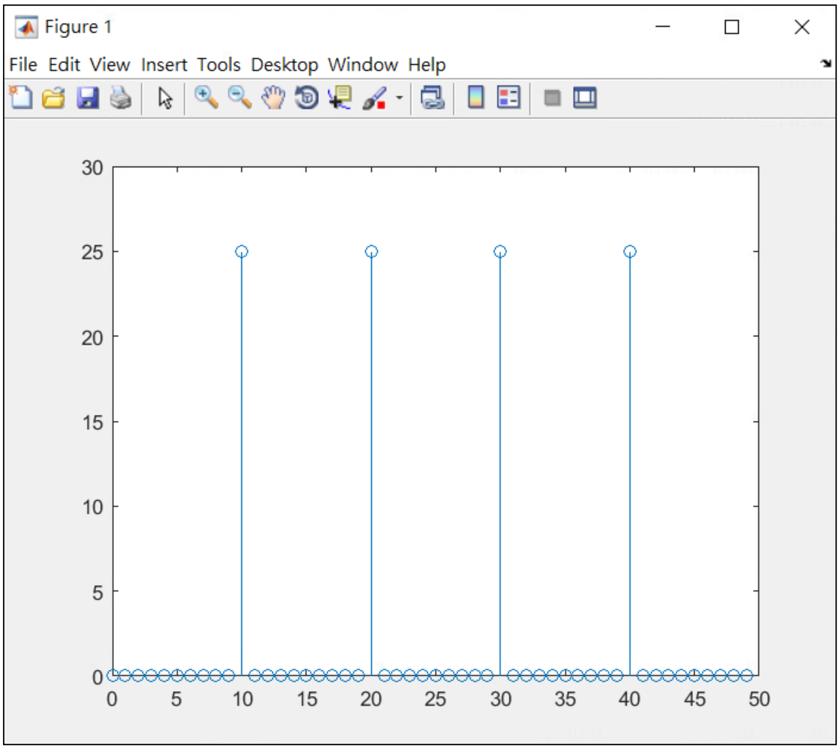
程式碼：

```
clear all;
f0 = 10;
f1 = 30;
Length=1;
T=0.01;
N=Length/T;
n=0:1:N-1;
x0=sin(2*pi*f0*n*T);
x1=sin(2*pi*f1*n*T);
x2 = x1 + x0;
xa = fft(x2);
magxa = abs(xa);
stem(n, magxa);
```

題目：

[Practice 3-5] Sketch a discrete-time signal that consists of 10Hz and 30Hz sine components based on a sampling period of 0.02 second. Then, use function fft() to compute its DFT, and compare the differences between this result and that in Practice 3-4.

實驗結果：



分析討論：

這題和 Practice 3-4 的結果互相比較後，發現頻譜圖中 10、20、30、40Hz 都有值，這是因為取樣頻率太低造成的失真，根據理論，取樣頻率要大於兩倍信號頻率才可以。

程式碼：

```
clear all;
f0 = 10;
f1 = 30;
Length=1;
T=0.02;
N=Length/T;
n=0:1:N-1;
x0=sin(2*pi*f0*n*T);
x1=sin(2*pi*f1*n*T);
x2 = x1 + x0;
xa = fft(x2);
magxa = abs(xa);
stem(n, magxa);
```

實驗四

題目：

[Practice 4-1] Input the signal in Fig. 4-1 to system y[n] = 0.8y[n-1] + x[n] - x[n-1]. Sketch the response based on filtering and convolution, respectively. Observe the differences between filtering and convolution.

實驗結果：

w

1x8 double

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2	2.6000	3.0800	3.4640	-2.2288	-1.6192	-1.0496	-0.5120					
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													

y

1x4 double

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2	2.6000	3.0800	3.4640									
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													

分析討論：

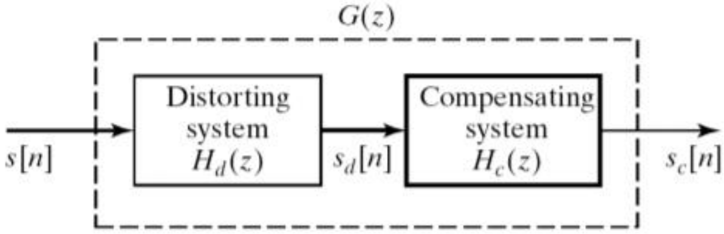
這題是用 impulse response h[n]和輸入信號 x[n]做摺積求 y[n]，用 matlab 求輸出信號可以用 Conv()、Filter()兩種函式，觀察實驗結果後，發現 Conv()、Filter()的結果前四點的值相同，其他點的值不同，詢問老師後，才知道我們通常只求前四個值當作結果，表示兩種方式求 y[n]都是對的。

程式碼：

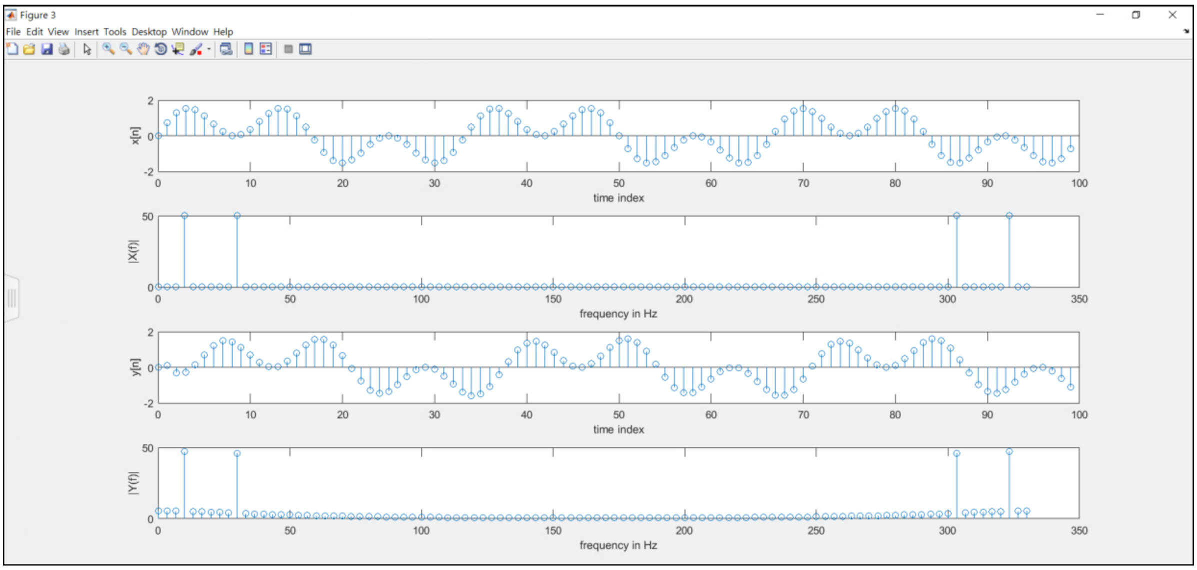
```
x=[2 3 4 5];
a=[1,-0.8];
b=[1,-1];
for n=1:20
    h(n)=0.8.^(n-1)-0.8^(n-2);
    h(1)=1;
end
w=conv(h,x)
y=filter(b,a,x)
```

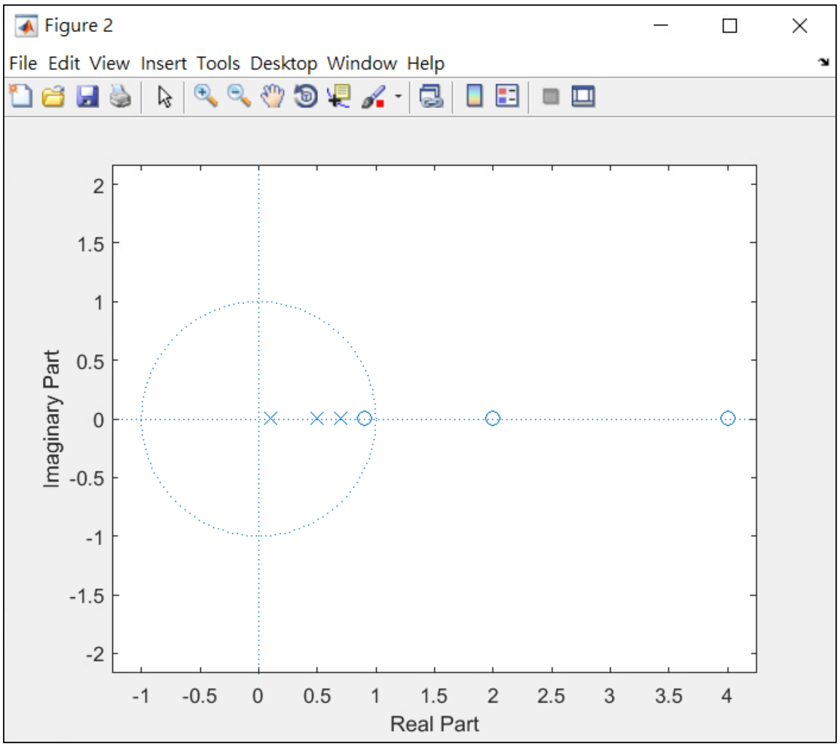
題目：

[Practice 4-2] Design a compensating system $H_c(z)$ as shown below (must be a minimum phase system). Suppose that a sine signal $s[n]$ consisting of 10Hz and 30Hz is input to a distorting system $H_d(z)=\dfrac{1-6.9z^{-1}+13.4z^{-2}-7.2z^{-3}}{1-1.3z^{-1}+0.47z^{-2}-0.035z^{-3}}$. Plot the output $s_c[n]$ and its DFT.



實驗結果：





分析討論：

　　這題實驗我們是先在計算紙上將 Hd[z]拆成兩個部分，一個是 Minimum phase，另一個 All Pass，之後再令 Hc(z)為 1 / Minimum phase，觀察實驗的結果後，發現輸入信號與輸出信號是相同的，代表計算結果和程式都是正確的。

程式碼：

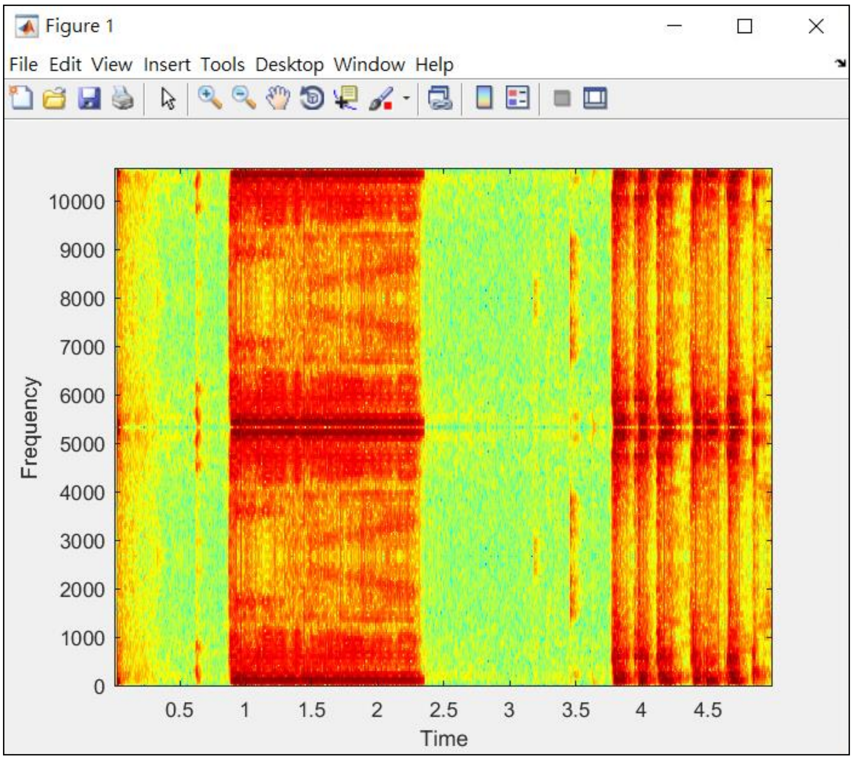
```
num0 =[1 ,-6.9,13.4,-7.2];
den0 =[1,-1.3,0.47,-0.035];
num1 =[1,-1.3,0.47,-0.035];
den1 =[8,-13.2,6.4,-0.9];

freqz(num0,den0,200,100);
figure(2);
zplane(num0,den0);
figure(3);

f0=10;
f1=30;
T=0.003;
N=100;
n=0:1:N-1;
x0=sin(2*pi*f0*n*T);
x1=sin(2*pi*f1*n*T);
x2=x0+x1;
f=n/T/N;
y=filter(num0,den0,x2);
y1=filter(num1,den1,y);

subplot(4,1,1); stem(n,x2);
xlabel('time index'); ylabel('x[n]');
subplot(4,1,2); stem(f,abs(fft(x2)));
xlabel('frequency in Hz'); ylabel('|X(f)|');
subplot(4,1,3); stem(n,y1);
xlabel('time index'); ylabel('y[n]');
subplot(4,1,4); stem(f,abs(fft(y1)));
xlabel('frequency in Hz'); ylabel('|Y(f)|');
```

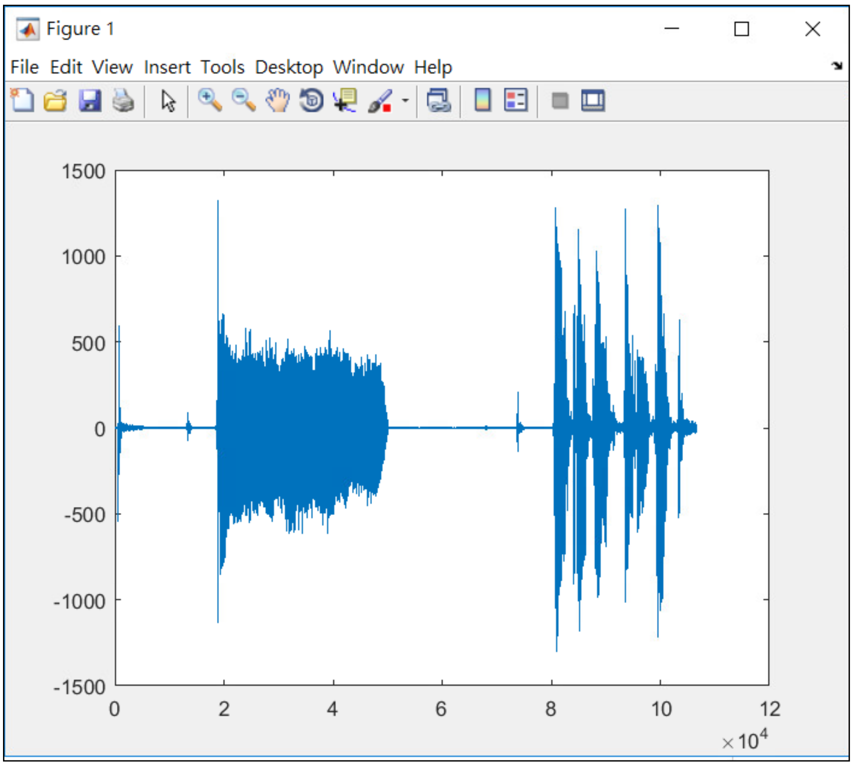
<u>實驗五</u>

題目：

[Practice 5-1] Perform ↑ 4 and then ↓ 3 of an audio signal without using the Matlab functions. Plot the spectrogram of the resulting signal.

實驗結果：





分析討論：

這題以 Code Ex_5_1 錄一個聲音檔 16kHz.pcm，並先用一個 For 迴圈做 4 的升取樣，再用一個 For 迴圈做 3 的降取樣，以此做到 4/3 的升取樣。做完這題的實驗，讓我學會用 matlab 錄音，也學會不以 matlab 的內建程式，做到升取樣和降取樣。

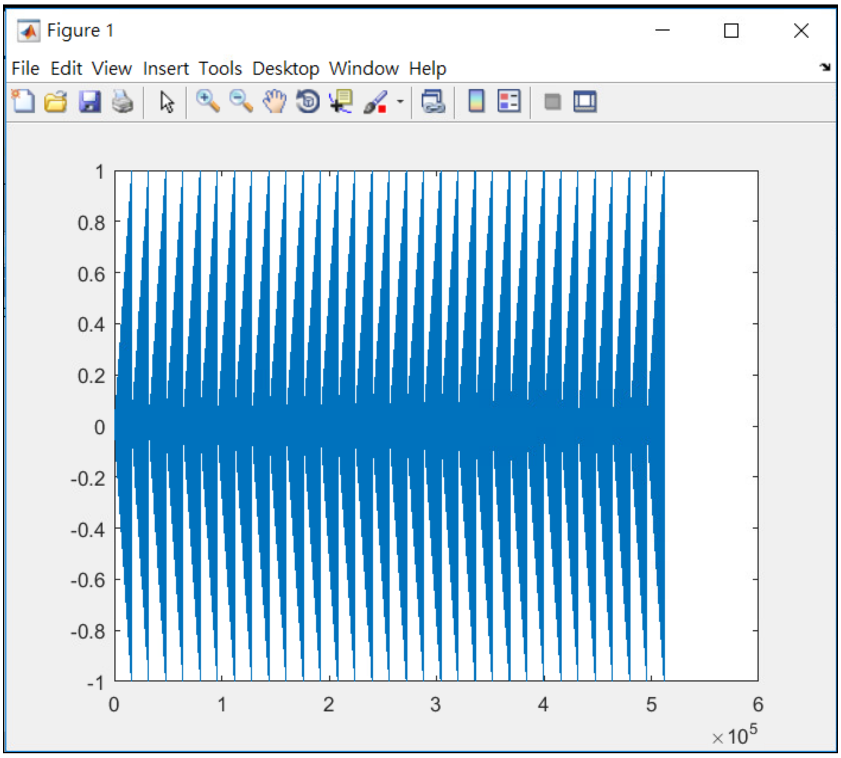
程式碼：

```
% Read a pcm file.
clear; fp=fopen('16kHz.pcm','rb');
x=fread(fp,'short');
fclose(fp);
Fs=16000;
y=zeros(320000,1);
for i=1:1:80000
y(4*i)=x(i);
end
sound(y,160000);
y1=zeros(106667,1);
for i=1:1:106666
y1(i)=y(3*i);
end
sound(y1,106666);
% Plot the waveform.
plot(y1);
figure(2);
specgram(y1,512,Fs*4/3,320);
```

題目：

[Practice 5-2] Generate a music with meoldy: So Mi Mi Fa Re Re Do Re Mi Fa So So So ; So Mi Mi Fa Re Re Do Mi So So Do, via Matlab function sound.m at sampling frequency of 8000Hz

實驗結果：



分析討論：

這題是利用各音高的頻率不同，以 matlab 的程式編排頻率，將一首歌曲用 matlab 播放出來，我編排的這首歌為兩隻老虎，後來我也將實驗的波形改成各種波形，發現歌曲的各種波形有不同的特色。

程式碼：

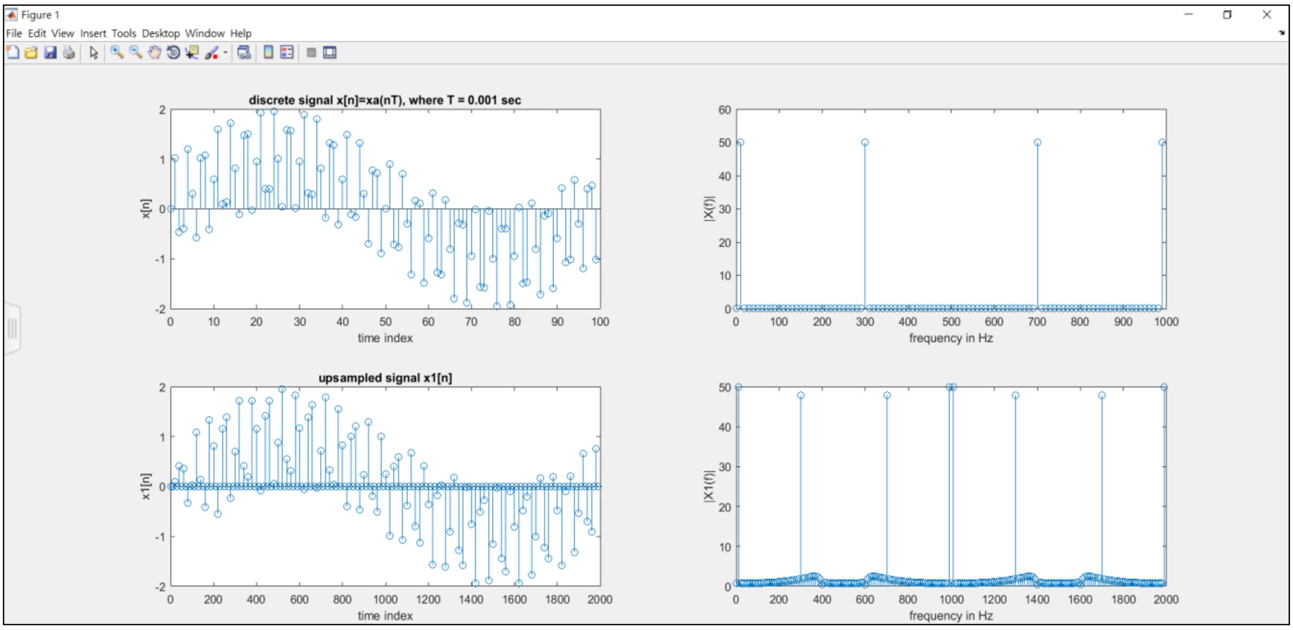
```
clear all;
clear all;

Fs = 16000
t = linspace(0,2*pi, 16000)

freqs = [261 293 329 349 392 440 493]
x1 = square(freqs(1) * t) .* ( t / (2 * pi));
x2 = square(freqs(2) * t) .* ( t / (2 * pi));
x3 = square(freqs(3) * t) .* ( t / (2 * pi));
x4 = square(freqs(1) * t) .* ( t/ (2 * pi));
x5 = square(freqs(1) * t) .* (t/ (2 * pi));
x6 = square(freqs(2) * t) .* ( t / (2 * pi));
x7 = square(freqs(3) * t) .* ( t / (2 * pi));
x8 = square(freqs(1) * t) .* ( t / (2 * pi));
x9 = square(freqs(3) * t) .* ( t / (2 * pi));
x10 = square(freqs(4) * t) .* (t / (2 * pi));
x11 = square(freqs(5) * t) .* ( t / (2 * pi));
x12 = square(freqs(3) * t) .* ( t / (2 * pi));
x13 = square(freqs(4) * t) .* ( t / (2 * pi));
x14 = square(freqs(5) * t) .* ( t / (2 * pi));
x15 = square(freqs(5) * t) .* ( t / (2 * pi));
x16 = square(freqs(6) * t) .* ( t / (2 * pi));
x17 = square(freqs(5) * t) .* ( t / (2 * pi));
x18 = square(freqs(4) * t) .* ( t / (2 * pi));
x19 = square(freqs(3) * t) .* (t / (2 * pi));
x20 = square(freqs(1) * t) .* ( t / (2 * pi));
x21 = square(freqs(5) * t) .* ( t / (2 * pi));
x22 = square(freqs(6) * t) .* ( t / (2 * pi));
x23 = square(freqs(5) * t) .* ( t / (2 * pi));
x24 = square(freqs(4) * t) .* ( t / (2 * pi));
x25 = square(freqs(3) * t) .* ( t / (2 * pi));
x26 = square(freqs(1) * t) .* ( t / (2 * pi));
x27 = square(freqs(1) * t) .* ( t / (2 * pi));
x28 = square(freqs(5) * t) .* ( t / (2 * pi));
x29 = square(freqs(1) * t) .* ( t / (2 * pi));
x30 = square(freqs(1) * t) .* ( t / (2 * pi));
x31 = square(freqs(5) * t) .* ( t / (2 * pi));
x32 = square(freqs(1) * t) .* ( t / (2 * pi));
y = [x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15 x16 x17 x18 x19
x20 x21 x22 x23 x24 x25 x26 x27 x28 x29 x30 x31 x32 ];
plot(y)
sound(y, Fs, 16);
```

實驗六

題目：

[Practice 6-1] Design a Chebyshev lowpass digital filter using Matlab function upsample.m to perform upsampling of the signal in Example 6-2 by a factor 2. Sketch the resulting waveform and spectrogram.

實驗結果：



分析討論：

這題設計的 Chebyshev 低通數位濾波器，它的截止頻率設在 0.8 rad/sec，如果取樣頻率為 1000Hz 時，截止頻率在 400Hz，因為我將信號以 upsample()，為信號升取樣 2，如此一來取樣頻率變為 2000Hz，大於信號最高頻 300Hz 兩倍，就不會產生 aliasing，在 400Hz 以下的頻率都可通過 Chebyshev 低通數位濾波器，所以信號 10Hz 和 300Hz 都會被留下。做完這題我學到使用 matlab 的內建程式 upsample()做升取樣，也學習到 Chebyshev 低通數位濾波器的特性。

程式碼：

```
% Chebyshev lowpass filter
[b,a] = cheby1(9,0.05,0.8);
% cut-off freq. = 0.8 pi = 400 Hz
% signal x
f1=10;
% 10 Hz sine wave
f2=300;
% 300 Hz sine wave
T=0.001;
% sampling freq. = 1000 Hz
N=100;
n=0:1:N-1;
x=sin(2*pi*f1*n*T)+sin(2*pi*f2*n*T);
subplot(2,2,1); stem(n,x);
xlabel('time index'); ylabel('x[n]');
title('discrete signal x[n]=xa(nT), where T = 0.001 sec');
% DFT of x
f=n/T/N;
subplot(2,2,2); stem(f,abs(fft(x)));
xlabel('frequency in Hz'); ylabel('|X(f)|');
% lowpass filtering & Decimation & DFT

y=filter(b,a,x);
z=upsample(y,2);
n2=0:1:N*2-1;
f=n2/(T/2)/(N*2);
subplot(2,2,3); stem(f,z);
xlabel('time index'); ylabel('x1[n]');
title('upsampled signal x1[n]');
subplot(2,2,4); stem(f,abs(fft(z)));
xlabel('frequency in Hz'); ylabel('|X1(f)|');
```