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

數位信號處理實習

LAB3

電子工程系 蔡偉和 教授

100360318 四子三甲 陳奕璋 學生

2014/3/31

[練習 3-1] 不使用內建函式 exp()、abs()、與 angle()而改以下列的方式來實現 DTFT:

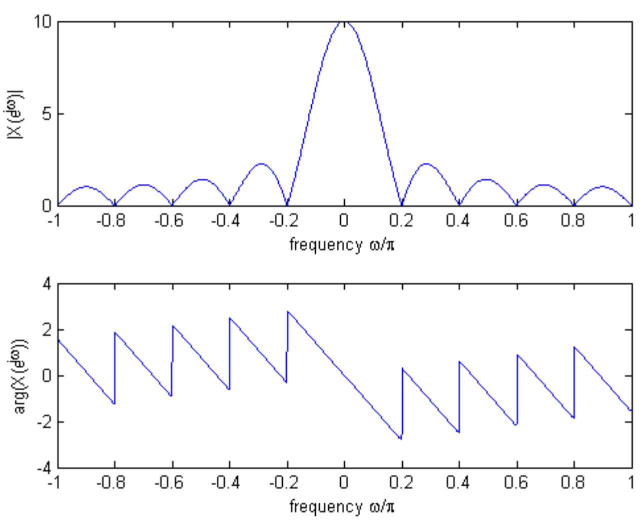
$X(e^{jw}) = \sum_{n=-\infty}^{\infty} x[n]e^{-jwn} = \sum_{n=-\infty}^{\infty} \{x[n]\cos(wn) - jx[n]\sin(wn)\} = X_R(e^{jw}) + jX_I(e^{jw})$ 分成實部與虛部

，則振幅為 $\sqrt{X_R^2(e^{jw}) + X_I^2(e^{jw})}$，相位為 $\tan^{-1}\left[\frac{X_I(e^{jw})}{X_R(e^{jw})}\right]$。

1. 程式碼：

```
% Computing the DTFT of signal x
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 = x*sin(n'*w);
magX = sqrt(xr.^2 + xi.^2);
angX = -atan2(xi, 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))');
```

2. Matlab 波形圖：



3. 實驗結果正確性

此程式執行結果與練習一之波形一致，故可確定程式正確。

4. 心得

撰寫此程式時，一開始很順利，只是造著公式照翻成 Matlab 語法，但在計算角度時，想了很久：「如何把徑度的範圍限制在 1、4 象限?」，因為 atan 的函式只會輸出落在 1、4 象限的角度，因此要處理計算出來的角度，在角度轉換這部分想了很久，最後才知道有 atan2 的函式可以使用，只要給出 x, y 在直角坐標系上的座標，就可計算出四象限的角度了。

在本次實驗中也學習到了下列函式

1. atan 反正切函數，角度範為落在-pi/2 ~ pi/2
2. atan2 反正切函數，角度範為落在-pi ~ pi

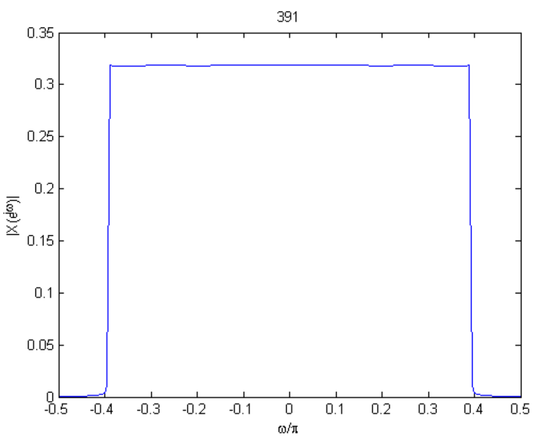
[範例 3-2] 觀察 Gibbs phenomenon。

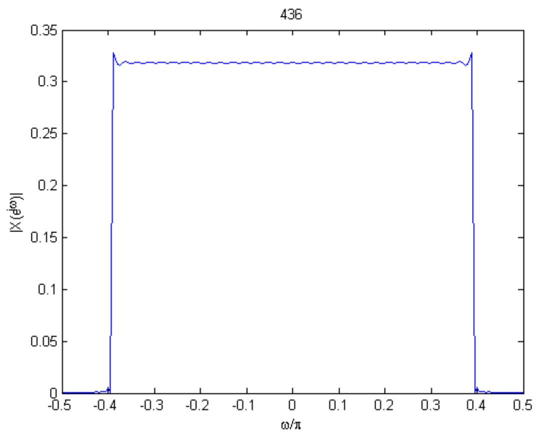
我們知道 $x[n]=\dfrac{\sin w_c n}{\pi n} \quad \leftrightarrow \quad X(e^{jw})=\begin{cases}1, & |w|\le w_c\\ 0, & w_c<|w|<\pi\end{cases}$, $-\infty\le n\le\infty$，但若取有限 n，$-M\le n\le M$，並計算 DTFT，則可發現振幅上有振盪現象，即所謂 Gibbs phenomenon。

1. 程式碼：

```
f = linspace(-0.5,0.5,200); % [-0.5pi, 0.5pi]
wc = 0.25*pi;               % cutoff frequency

for M=1:5:2000
    n = (-M:M);
    for j=1:length(f)
        X(j) = sum(wc/pi*sinc(wc*n).*exp(-i*2*pi*n*abs(f(j))));
    end
    plot(f,abs(X));
    xlabel('\omega/\pi');
    ylabel('|X(e^j^\omega)|');
    title(M);
    pause;
end
```

2. Matlab 波形圖：





3. 心得

雖然方波是由奇次諧波組成，但觀察 Gibbs 現象時，發現不論 M 再怎樣增加，都不可能得到一完整的方波波形，如上圖，在 M=391 時已經接近方波，但 M 繼續往上增時到了 M=436，又發現兩旁頻率變化較大的部分又有震盪現象出現，說明了不能利用合成方式得到完美的方波波形，還是會有些許的尖點出現。

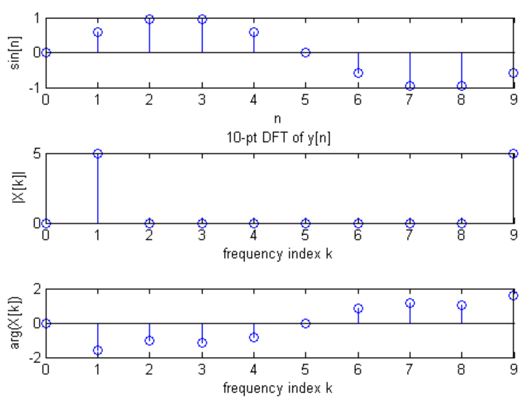
在此次實驗中學習到`linspace(i, f, p)`函式，i 為初值，f 為終值，p 為需要的點數。

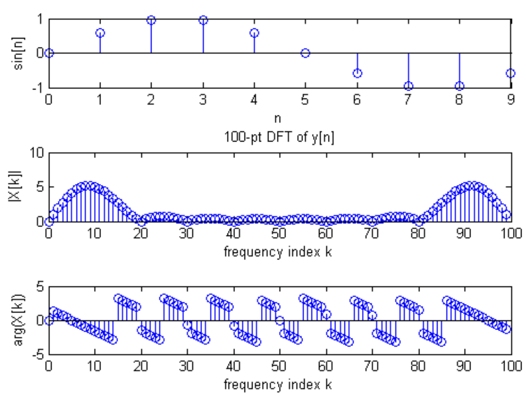
[練習 3-2] 取 Fig. 1-2 之離散弦波訊號的一個週期(10 points)進行 DFT，分別繪出 10-point 與 100-point DFT，討論兩者差異。

1. 程式碼：

```
clear;
f0 = 10;            % 10 Hz sine wave
Length = 0.1;       % Total length = 0.1 sec
T = 0.01;           % sampling period = 0.01 sec
N = Length/T;
n = 0 : 1 : N-1;
x = sin(2*pi*f0*n*T);

% Print original signal
subplot(3,1,1); stem(n, x);
xlabel('n'); ylabel('sin[n]');

% 100-point DFT
N = 100;
k = 0 : N-1;
X = x*exp(-j*2*pi/N*n'*k);
magX = abs(X);
angX = angle(X);

% Print DFT signal
subplot(3, 1, 2); stem(k, magX);
xlabel('frequency index k'); ylabel('|X[k]|');
title('100-pt DFT of y[n]');
% Print ARG
subplot(3, 1, 3); stem(k, angX);
xlabel('frequency index k'); ylabel('arg(X[k])');
```

2. Matlab 波形圖：





3. 實驗結果正確性

將 `X = x*exp(-j*2*pi/N*n'*k);` 換為 `X = fft(x, N);`，發現結果一致，證明了 DFT 的程式是正確的。

4. 心得

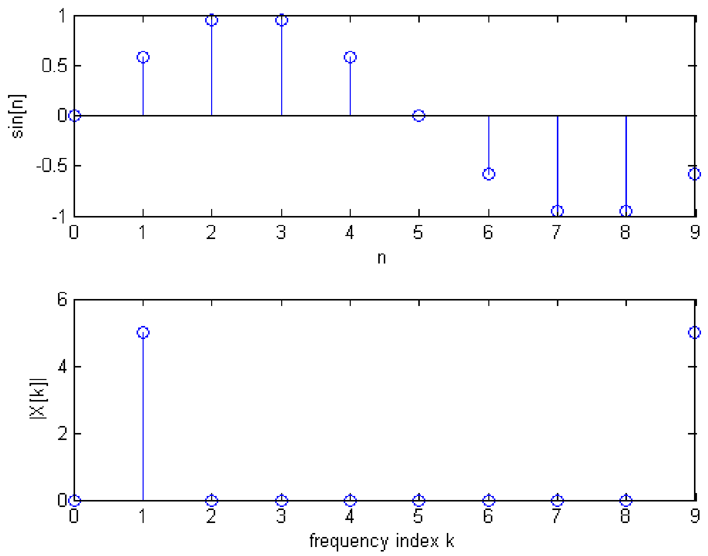
使用較多點數的 DFT 可以發現在頻域的解析度上，點數越多，解析度越好，在時域變動較大的部分，在頻譜上就可以顯現的出來。

[練習 3-3] 使用函式 fft()計算練習 3-2 的 DFT，並繪出結果。

1. 程式碼：

```
clear;
f0 = 10;            % 10 Hz sine wave
Length = 0.1;       % Total length = 0.1 sec
T = 0.01;           % sampling period = 0.01 sec
N = Length/T;
n = 0 : 1 : N-1;
x = sin(2*pi*f0*n*T);

% Print original signal
subplot(2,1,1); stem(n, x);
xlabel('n'); ylabel('sin[n]');

% FFT
X = fft(x);
magX = abs(X);

% Print DFT signal
k = 0 : N-1;
subplot(2, 1, 2); stem(k, magX);
xlabel('frequency index k'); ylabel('|X[k]|');
```

2. Matlab 波形圖：



3. 心得

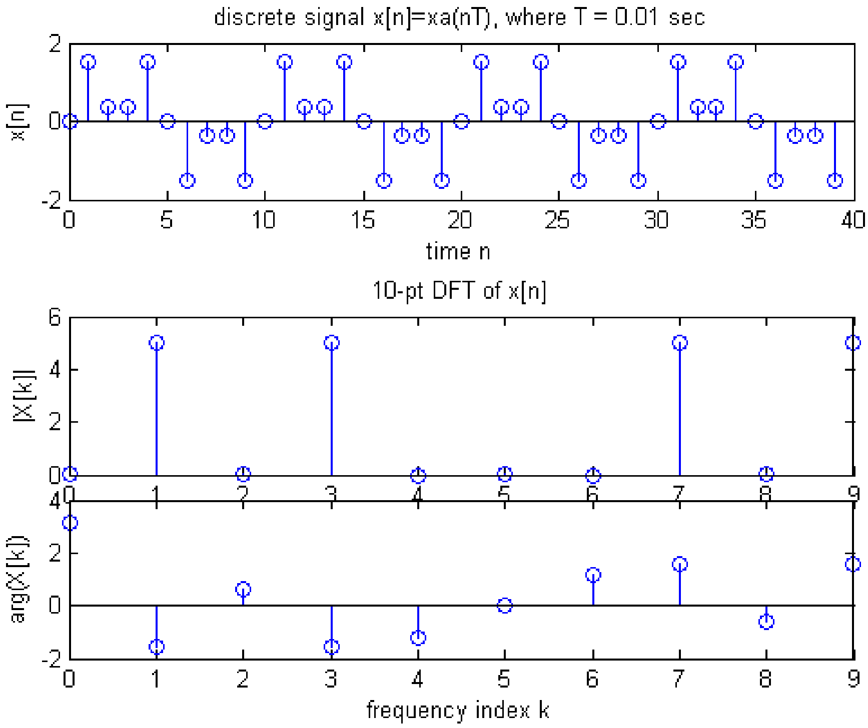
實現同樣的結果，使用 fft 函式發現簡單許多，不用擔心數學式打錯造成結果不對的問題，相信在程式執行效率上，使用 fft 函式也有可能有優化的動作，速度較快。在頻域輸出的部分，我一樣做了絕對值的處理，較好觀察主頻域。

[練習 3-4] 在練習 1-3 中曾繪出一包含 10Hz 與 30Hz 之弦波離散訊號，其中取樣週期為 0.01 second，試利用函式 fft()計算此訊號前 10 points 之 DFT。

1. 程式碼：

```
% Discrete sine wave 10Hz and 30Hz
clear;
f0 = 10;              % 10 Hz sine wave
f1 = 30;              % 30 Hz sine wave
Length = 0.4;         % Total length = 0.4 sec
T = 0.01;             % sampling period = 0.01 sec
N = Length / T;
n = 0:1:N-1;
x1 = sin(2*pi*f0*n*T);
x2 = sin(2*pi*f1*n*T);
subplot(7, 1, [1, 2]); stem(n, x1+x2);
xlabel('time n'); ylabel('x[n]');
title('discrete signal x[n]=xa(nT), where T = 0.01 sec');

% Computing the DFT of signal x
N = 10;
k = 0 : N-1;
x = x1 + x2;
X = fft(x, N);
magX = abs(X);
angX = angle(X);

subplot(7, 1, [4, 5]); stem(k, magX); xlabel('frequency index k');
ylabel('|X[k]|');
title('10-pt DFT of x[n]');
subplot(7, 1, [6, 7]); stem(k, angX); xlabel('frequency index k');
ylabel('arg(X[k])');
```

2. Matlab 波形圖：



3. 心得

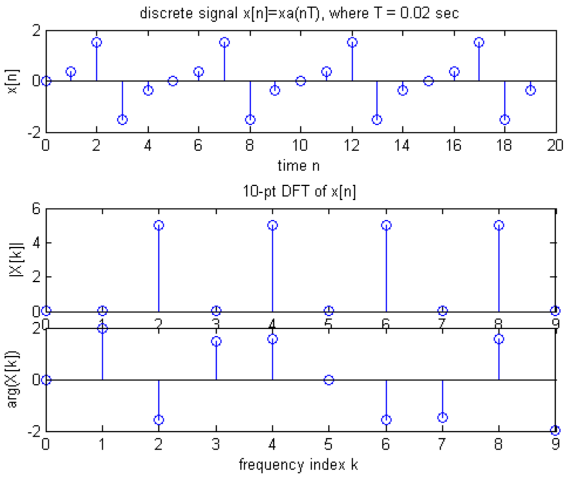
先前的實驗曾經畫出 10Hz 與 30Hz 的離散訊號，現在要計算 DFT，過程中沒有遇到大困難，將先前的程式再帶入 fft 函式就求得了。

[練習 3-5] 若上述包含 10Hz 與 30Hz 之弦波離散訊號是透過取樣週期為 0.02 second 所產生，試利用函式 fft()計算此訊號前 10 points 之 DFT，討論此結果與練習 3-4 有何差異。

1. 程式碼：

```
% Discrete sine wave 10Hz and 30Hz
clear;
f0 = 10;              % 10 Hz sine wave
f1 = 30;              % 30 Hz sine wave
Length = 0.4;         % Total length = 0.4 sec
T = 0.02;             % sampling period = 0.02 sec
N = Length / T;
n = 0:1:N-1;
x1 = sin(2*pi*f0*n*T);
x2 = sin(2*pi*f1*n*T);

subplot(7, 1, [1, 2]); stem(n, x1+x2);
xlabel('time n'); ylabel('x[n]');
title('discrete signal x[n]=xa(nT), where T = 0.02 sec');

% Computing the DFT of signal x
N = 10;
k = 0 : N-1;
x = x1 + x2;
X = fft(x, N);
magX = abs(X);
angX = angle(X);

subplot(7, 1, [4, 5]); stem(k, magX); xlabel('frequency index k');
ylabel('|X[k]|');
title('10-pt DFT of x[n]');

subplot(7, 1, [6, 7]); stem(k, angX); xlabel('frequency index k');
ylabel('arg(X[k])');
```

2. Matlab 波形圖：



3. 心得

發現同樣訊號但取樣周期變為 0.02s 與上題實驗有所差異，相較起來 0.02s 的取樣周期有失真的狀況出現，因為取樣周期變長，在長度不便的狀況下，相對取樣得點數會變低，因此取樣出來的訊號與真實的訊號差異變大，若我們改用 0.001s 取樣，相對來說會取到 0.4/0.001 點，如下圖，可見取樣訊號更密集，更接近原始訊號的長相。

為了避免 Aliasing Phenomenon 出現，根據 Nyquist Theorem，取樣頻率至少要輸入最大訊號的兩倍。而雖然點數越多越好，但是越多點數在硬體上的成本越高，相對也需要更多空間來儲存取樣的訊號，更重要的是，較多的取樣點是否對於電路有實質上的幫助，是否會讓電路輸出更精確，這是需要取捨的部分。