%%%%% 握把式圖形與GUI設計

% Example 1

peaks; % 畫出 peaks 3D 圖形

propedit; % 開啟性質編輯器

% Example 2

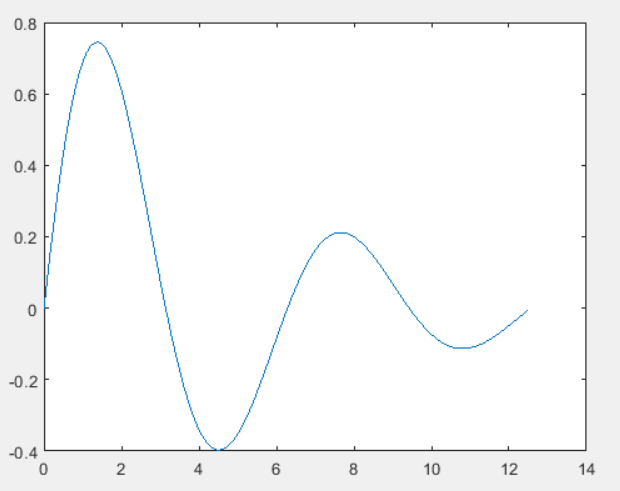
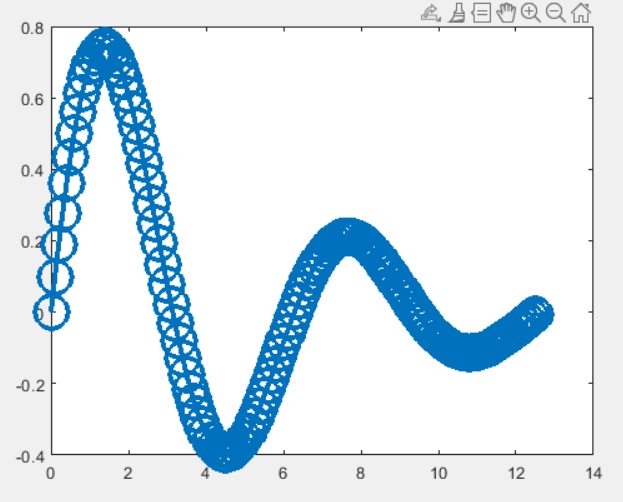
t = 0:0.1:4\*pi;

y = exp(-t/5).\*sin(t);

h = plot(t, y); % h 為曲線的握把

set(h, 'Linewidth', 3); % 將曲線寬度改為 3

set(h, 'Marker', 'o'); % 將曲線的線標改成小圓圈

set(h, 'MarkerSize', 20); % 將線標的大小改成 20

**Example2-2**

**Example2-1**

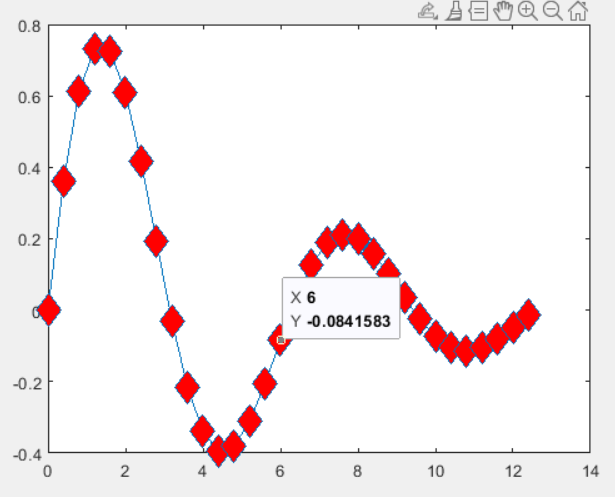
% Example 3

t=0:0.4:4\*pi;

h=plot(t, exp(-t/5).\*sin(t)); % h 為曲線的握把

set(h, 'Marker', 'diamond', 'MarkerSize', 15, 'MarkerFaceColor', 'r'); % 將線標改成菱形、線標大小改成 15、線標的邊色改成紅色

fprintf('MATLAB version = %s\n', version);

get(h)

**Example3**

% Example 4

plot(rand(10, 2)); % 畫出兩條曲線 **因為沒有設變數**

h=findobj(0, 'type', 'line') % 找出曲線的握把 **但一般不會這麼做**

set(h, 'LineWidth', 3); % 經由握把將曲線寬度改為 3

% Example 5

h = uicontrol; % 產生按鈕

set(h, 'String', '請按我！'); % 在按鈕表面加入文字「請按我！」

cmd = 'fprintf(''有人按我一下喔！\n'');'; % 定義按鈕被按後的反應指令

set(h, 'Callback', cmd); % 設定按鈕的反應指令



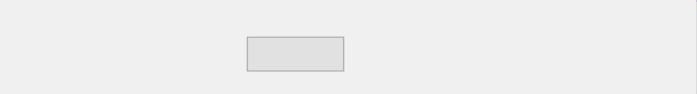
**Example5**

% Example 6

close all % 關閉所有圖形視窗

uicontrol('style', 'push', 'position', [200 20 80 30]);

UI control的類型



**Slide**

**Push**

30

80

20

200

uicontrol('style', 'slide', 'position', [200 70 80 30]);

uicontrol('style', 'radio', 'position', [200 120 80 30]);

**Frame**

**Radio**

uicontrol('style', 'frame', 'position', [200 170 80 30]);

uicontrol('style', 'check', 'position', [200 220 80 30]);

uicontrol('style', 'edit', 'position', [200 270 80 30]);

uicontrol('style', 'list', 'position', [200 320 80 30], 'string', '1|2|3|4');

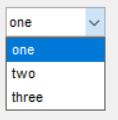
uicontrol('style', 'popup', 'position', [200 370 80 30], 'string', 'one|two|three');

**Popup**

**List**

**Edit**

**Check**



% Example 7

% 產生新圖形視窗，其左下角之座標為[30, 30]，

% 長度為300，高度為200（均以Pixel為單位）

figure('position', [30 30 300 200]);

% 在圖形視窗內產生一個圖軸，其左下角之座標為[0.1, 0.2],

% 長度為0.8，高度為0.8（使用標準化的單位，即圖形的左下角為[0, 0]，

% 長度及高度都是1。）

axes('position', [0.1 0.2 0.8 0.8]);

% 視窗上的第一個圖形，為三度空間的peaks函數。

pointNum = 20;

[xx, yy, zz] = peaks(pointNum);

surf(xx, yy, zz);

colormap hsv

axis tight

% 第一個UI控制物件，用以控制背景格線的顯示。

h1 = uicontrol('style', 'checkbox', 'string', 'Grid on', ...

'position', [10, 10, 60, 20], 'value', 1);

% 第二個UI控制物件，用以指定X軸及Y軸的格子點數目。

h2 = uicontrol('style', 'edit', 'string', int2str(pointNum), ...

'position', [90, 10, 60, 20]);

% 第三個UI控制物件，用以指定顯示曲面所用到的色盤矩陣。

h3 = uicontrol('style', 'popupmenu', ...

'string', 'hsv|hot|cool', ...

'position', [170, 10, 60, 20]);

% 第一個UI控制物件的反應指令為「grid」。

set(h1, 'callback', 'grid');

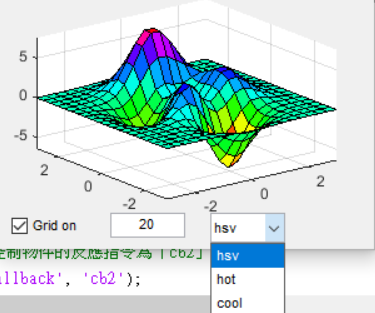
**按鍵的外觀，也稱握把**

**所有指令的控制都用callback**

% 第二個UI控制物件的反應指令為「cb2」。

set(h2, 'callback', 'cb2');

% 第三個UI控制物件的反應指令為「cb3」。

set(h3, 'callback', 'cb3');

% **Cb2**

**四捨五入的函數**

% 取得第二個控制物件的數值

pointNum = round(str2num(get(h2, 'string')));

% 若數字太大或太小，則設定為10

if pointNum <= 1 | pointNum > 100,

pointNum = 10;

set(h2, 'string', int2str(pointNum));

end

% 根據所得數字，重畫peaks曲面

[xx, yy, zz] = peaks(pointNum);

surf(xx, yy, zz);

axis tight;

% 根據第一個UI控制物件，決定是否要畫格線

if get(h1, 'value')==1,

**1就是打勾**

grid on;

else

grid off;

end

**% CB3**

% 根據第三個UI控制物件來決定使用的色盤矩陣。

switch get(h3, 'value')

case 1

colormap(hsv);

case 2

colormap(hot);

**暖色系**

case 3

colormap(cool);

otherwise

**冷色系**

disp('Unknown option');

end

% Example 8

function ui02(action)

if nargin<1, action='initialize'; end

switch(action)

case 'initialize' % 圖形視窗及UI控制物件的初始化。

% 產生新圖形視窗，其左下角之座標為[30, 30]，

% 長度為300，高度為200（均以Pixel為單位）

figH = figure('position', [30 30 300 200]);

% 在圖形視窗內產生一個圖軸，其左下角之座標為[0.1, 0.2],

% 長度為0.8，高度為0.8（使用標準化的單位，即圖形的左下角為[0, 0]，

% 長度及高度都是1。）

axes('position', [0.1 0.2 0.8 0.8]);

% 視窗上的第一個圖形，為三度空間的peaks函數。

pointNum = 20;

[xx, yy, zz] = peaks(pointNum);

surf(xx, yy, zz);

colormap hsv

axis tight

% 第一個UI控制物件，用以控制背景格線的顯示。

h1 = uicontrol('style', 'checkbox', ...

'tag', 'ui4grid', ...

'string', 'Grid on', ...

'position', [10, 10, 60, 20], 'value', 1);

% 第二個UI控制物件，用以指定X軸及Y軸的格子點數目。

h2 = uicontrol('style', 'edit', ...

'tag', 'ui4pointNum', ...

'string', int2str(pointNum), ...

'position', [90, 10, 60, 20]);

% 第三個UI控制物件，用以指定顯示曲面所用到的調色盤。

h3 = uicontrol('style', 'popupmenu', ...

'tag', 'ui4colorMap', ...

'string', 'hsv|hot|cool', ...

'position', [170, 10, 60, 20]);

% 第一個UI控制物件的反應指令為「grid」。

set(h1, 'callback', 'grid');

% 第二個UI控制物件的反應指令為「ui02('setPointNum')」。

set(h2, 'callback', 'ui02(''setPointNum'')');

% 第三個UI控制物件的反應指令為「ui02('setColorMap')」。

set(h3, 'callback', 'ui02(''setColorMap'')');

case 'setPointNum' % 第二個UI控制物件的callback。

% 找出第一及第二個UI控制物件的握把。

h1 = findobj(0, 'tag', 'ui4grid');

h2 = findobj(0, 'tag', 'ui4pointNum');

% 取得第二個UI控制物件的數值。

pointNum = round(str2num(get(h2, 'string')));

% 若數字太大或太小，則設定為10。

if pointNum <= 1 | pointNum > 100,

pointNum = 10;

set(h2, 'string', int2str(pointNum));

end

% 根據所得的數字，重畫peaks曲面。

[xx, yy, zz] = peaks(pointNum);

surf(xx, yy, zz);

axis tight;

% 根據第一個UI控制物件，決定是否要畫格線。

if get(h1, 'value')==1,

grid on;

else

grid off;

end

case 'setColorMap' % 第三個UI控制物件的callback。

% 找出第三個UI控制物件的握把。

h3 = findobj(0, 'tag', 'ui4colorMap');

% 根據第三個UI控制物件來決定使用的色盤矩陣。

switch get(h3, 'value')

case 1

colormap(hsv);

case 2

colormap(hot);

case 3

colormap(cool);

otherwise

disp('Unknown option');

end

otherwise

error('Unknown action string!');

end

% Example 9

function mouse01(action)

% mouse01: 本例展示如何設定滑鼠事件的反應指令, based on "switchyard programming"

% Roger Jang, 20040405

if nargin<1, action='start'; end

switch(action)

case 'start' % 開啟圖形視窗

axis([0 1 0 1]); % 設定圖軸範圍

box on; % 將圖軸加上圖框

title('Click and drag your mouse in this window!');

% 設定滑鼠按鈕被按下時的反應指令為「mouse01 down」

set(gcf, 'WindowButtonDownFcn', 'mouse01 down');

case 'down' % 滑鼠按鈕被按下時的反應指令

% 設定滑鼠移動時的反應指令為「mouse01 move」

set(gcf, 'WindowButtonMotionFcn', 'mouse01 move');

% 設定滑鼠按鈕被釋放時的反應指令為「mouse01 up」

set(gcf, 'WindowButtonUpFcn', 'mouse01 up');

% 列印「Mouse down!」訊息

fprintf('Mouse down!\n');

case 'move' % 滑鼠移動時的反應指令

currPt = get(gca, 'CurrentPoint');

x = currPt(1,1);

y = currPt(1,2);

line(x, y, 'marker', '.');

% 列印「Mouse is moving!」訊息及滑鼠現在位置

fprintf('Mouse is moving! Current location = (%g, %g)\n', x, y);

case 'up' % 滑鼠按鈕被釋放時的反應指令

% 清除滑鼠移動時的反應指令

set(gcf, 'WindowButtonMotionFcn', '');

% 清除滑鼠按鈕被釋放時的反應指令

set(gcf, 'WindowButtonUpFcn', '');

% 列印「Mouse up!」訊息

fprintf('Mouse up!\n');

end

% Example 10

function mouse02(action)

% mouse02: 本例展示如何以來設定滑鼠事件的反應指令

% Roger Jang, 20040406

if nargin == 0, action = 'start'; end

switch(action) % 開啟圖形視窗

case 'start'

axis([0 1 0 1]); % 設定圖軸範圍

box on; % 將圖軸加上圖框

title('Click and drag your mouse in this window!');

% 設定滑鼠按鈕被按下時的反應指令

set(gcf, 'WindowButtonDownFcn', sprintf('%s %s', mfilename, 'down'));

case 'down' % 滑鼠按鈕被按下時的反應指令

% 設定滑鼠移動時的反應指令

set(gcf, 'WindowButtonMotionFcn', sprintf('%s %s', mfilename, 'move'));

% 設定滑鼠按鈕被釋放時的反應指令為

set(gcf, 'WindowButtonUpFcn', sprintf('%s %s', mfilename, 'up'));

% 列印「Mouse down!」訊息

fprintf('Mouse down!\n');

case 'move' % 滑鼠移動時的反應指令

fprintf('Mouse is moving! ');

feval(mfilename, 'print');

% 列印「Mouse is moving!」訊息及滑鼠現在位置

case 'up' % 滑鼠按鈕被釋放時的反應指令

feval(mfilename, 'print');

% 清除滑鼠移動時的反應指令

set(gcf, 'WindowButtonMotionFcn', '');

% 清除滑鼠按鈕被釋放時的反應指令

set(gcf, 'WindowButtonUpFcn', '');

% 列印「Mouse up!」訊息

fprintf('Mouse up!\n');

case 'print'

currPt = get(gca, 'CurrentPoint');

x = currPt(1,1); y = currPt(1,2);

line(x, y, 'marker', '.', 'EraseMode', 'xor');

fprintf('Current location = (%g, %g)\n', currPt(1,1), currPt(1,2));

end