MATLAB- GUI介面的設計

作者： 胡哲銘

國立台灣大學電信工程學研究所

**前言：**

這個講座專區是根據MATLAB介面開發與編譯技巧(第二版，李顯宏編著)這本書來做說明介紹並且用我碩論題目開發的程式來進行實例解說。基本上，GUI設計不難，只要搞清楚GUI內建物件的選項功能以及輸出輸入的關係，你也可以設計出一套不錯的GUI程式介面。

建議已經有MATLAB撰寫基礎的人，可以直接從MATLAB介面開發與編譯技巧這本書的第九章開始研讀，此章節主要介紹如何透過GUIDE將GUI物件建立於介面中，並且建構出彈性化且美觀的人機介面。Fig.1 的GUI介面是我碩論用來實現QBH(Querying By Humming)的程式介面，接下來我會一一說明其中各個方塊的撰寫方式。

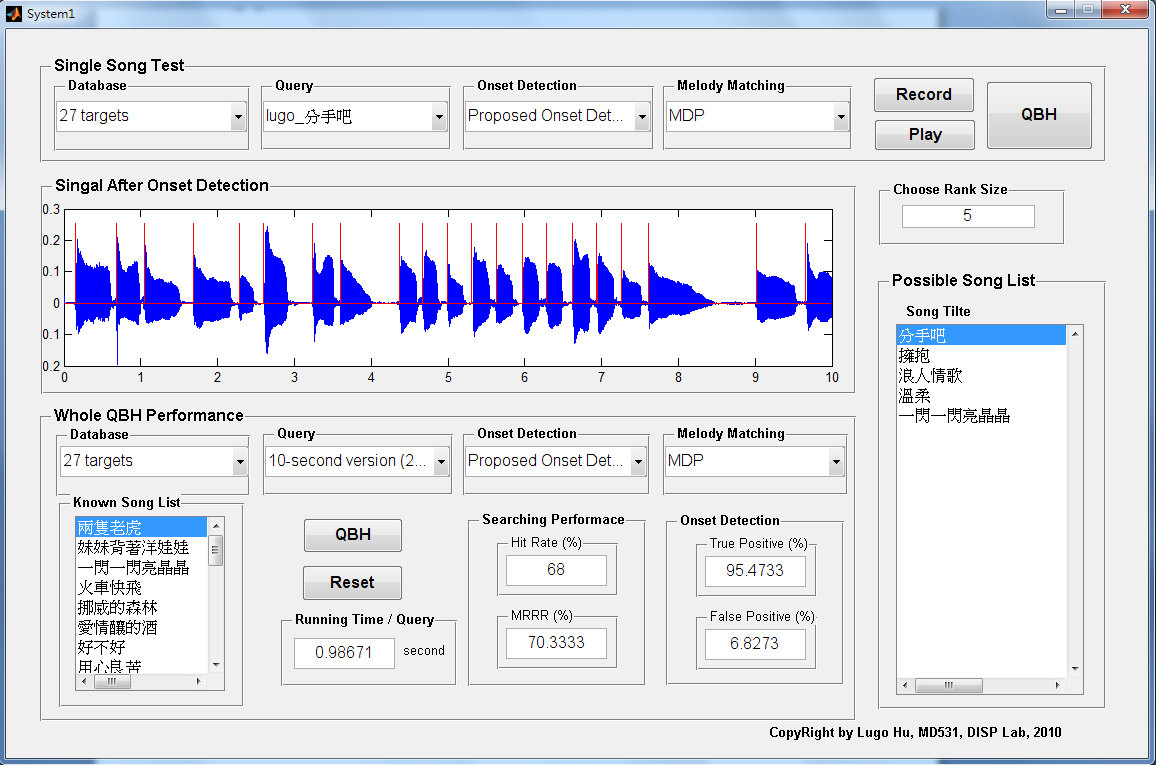
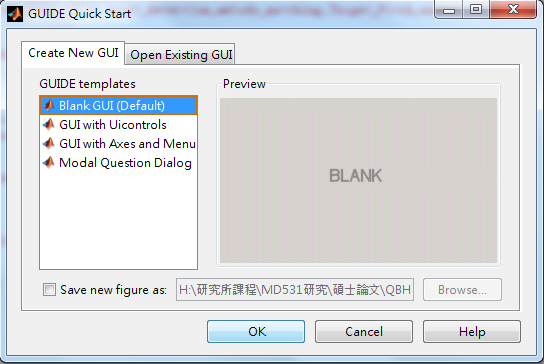


Fig.1

**新手入門：**

打開Matlab，在Command Window打GUIDE，則會出現以下畫面:



若是要開啟一個新的空白介面，選擇Blank GUI (Defualt)，若是已經有舊的GUI存檔(副檔名是 .fig)，則選擇Open Existing GUI。現在以下是一個已經開啟的空白GUI設計介面:



上圖中的功能鍵，每個鍵都有自己的功能：



**功能鍵：**

**Push Button:**

GUI中最常使用也是最簡單的物件，當使用者按下此按鈕時，Matlab就會馬上執行此按鈕相對應的Callback程式來產生動作。

Ex: 設計一個加法器，其動作為兩輸入的的總合，則按下此按鈕，總合則會被記

算出來。

**Slide:**

可允許使用者於一特定範圍內選取設定值。

**Checkbox:**

在GUI中是非常有用的功能選項，當使用者勾選checkbox時，其vlaue的屬性為1，反之為0。Checkbox中的自行設定的

**Radio Button:**

點選Radio Button時，其vlaue的屬性為1，反之為0。用法類似於Checkbox，但是不同於Checkbox能夠點選多個，Radio Button一次只能點選一個，也就是說其餘的Radio Button屬性皆為0，只有點選的Radio Button屬性為1。所以通常數個Radio Button都是編輯在一個Panel底下，意思就是不同Radio Button的功能是必須互斥的。

**Edit Text:**

主要當作一個輸入的介面，以便使用者能夠輸入字串字元以及數字。

**Static Text:**

不同於Edit Text, Static Text主要是用來編輯標題。

**Pop-up Menu:**

下拉式選單，結合switch …case陳述式，當一個程式需要用的可供選擇的函式，可以使用此功能去選擇想要的函式，然後再搭配Push Button進行程式的執行。Popup Menu的內容選項數目必須跟switch case的索引數相同，如選取第一個選項，則value屬性就會回傳1，選取第二個選項，則value屬性就會回傳2，以此類推。

**List Box:**

與Popup Menu很類似，這是一個清單式的選取介面，與Popup Menu最大的不同點在於它可以提供複選的功能。另外，也可以單純只把List Box當作一個文字的輸出清單。

**Toggle Button:**

和Push Button很類似，比較不同的地方在於按下Toggle Button的時候，其Value屬性為1，按鈕不會自行彈起，必須再按一次，Value屬性才會為0，按鈕才會彈起，此功能提供一個比Push Button較安全的機制去避免重複按到。

**Table:**

主要顯示矩陣運算的結果。

**Axes:**

主要顯示運算結果的圖型：影像圖，波型圖。

**Panel:**

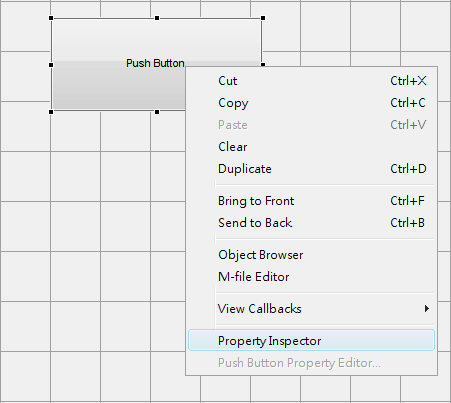
可以包含所有物件(GUI物件，Axes，其他Panel) ，當修改圖型的時候，方便於編輯。

**Button Group:**

和Panel是很像的東西，主要是用來管理排外選取的行為，也就是同一個組群中不能一次選取兩個radio button或toggle button。

**Property:**

對於任一功能選項，點選右鍵，再選擇Property Inspector，則可以進行屬性的編輯。



點選進入後，一些常用的選項如：Tag, String, Font Size, Value (事實上，我寫的QBH介面幾乎只編輯這四個選項，這些已經蠻夠用的)。

**實例說明：**



**Popup Menu說明:**

melody\_matching=get(handles.popupmenu4,'Value');

switch melody\_matching

case 1

melody\_matching = 'DP';

case 2

melody\_matching = 'HMM';

case 3

melody\_matching = 'MHMM';

case 4

melody\_matching = 'MDP';

end

case後面的數字代表popup menu第幾個選項，也是其屬性Value的值，例如：case 1

melody\_matching = 'DP';其中'DP'中的DP代表顯示在Popup menu的輸出選項。

**Edit Text說明:**

rank\_size=str2num(get(handles.edit19,'String'));

其中，edit19代表該edit text的tag，任何輸入裡面的值，輸出都會被判斷為一個字串，因此，str2num的動作是用來將字串轉為數字。

**Push Button 說明:**

[name\_of\_song onsets x Fs]= SingleSearch (query,onset\_detection,melody\_matching,Target\_Pitch,number\_of\_target,rank\_size);

query: 選擇輸入歌曲。

onset\_detection: 選擇何種 onset detection。

melody\_matching: 選擇何種 melody matching。

Target\_Pitch: 由database決定。

number\_of\_target: 由database決定。

rank\_size: 自行輸入選擇。

**List Box 說明:**

songlist=[];

for n=1:rank\_size

songlist=[songlist name\_of\_song{n} '|'];

end

set(handles.listbox1,'String',songlist);

set是將字串顯示出來的功能。

**Axes 說明:**

time\_slot=zeros(1,length(x));

time\_slot(onsets)=max(x)+0.01;

time=(0:length(x)-1)/Fs;

plot(time,x,time,time\_slot,'r');

拉一個Axes視窗，將運算結果繪圖則會自動顯示出來。