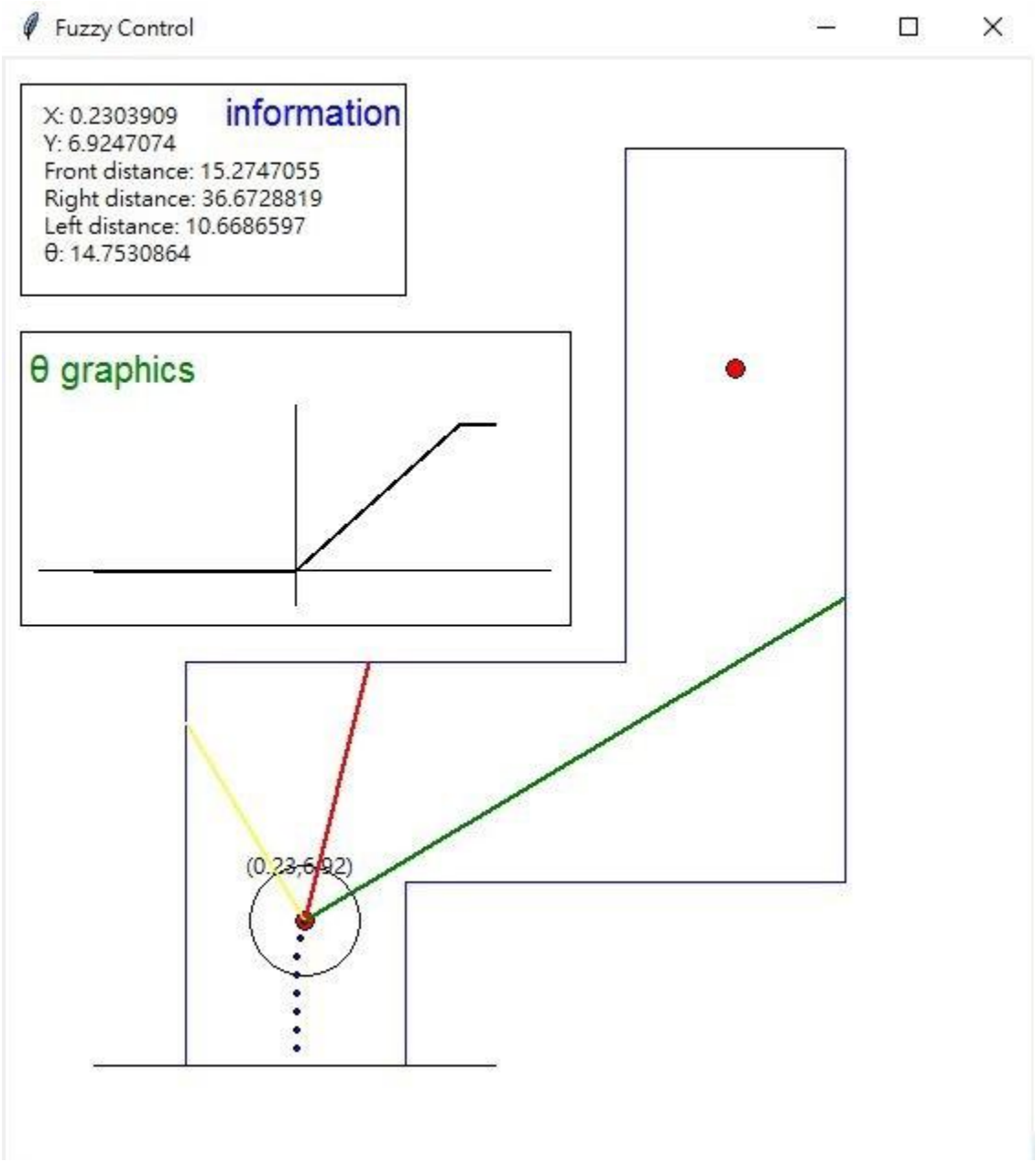
HomeWork1_Report

資工 4A_102502511_黃宇謙

前言：

本次實驗是以程式模擬車行駛的情形，使車子能在遇到轉彎處，順利過彎，並且由起點走到終點線。



實驗步驟：

1. 由於一開始並沒有相關數據，因此必須先自行移動一次，使車子安全抵達中點，並將轉方向盤的角度記錄下來，以供接下來要建模組的時候使用。
2. 將感測器偵測到的距離，帶入模組中，進行模糊控制運算。
3. 模糊運算的結果再藉由去模糊化，得到車子方向盤轉幾度。

細節：

感測器：

目的：

感測器用來偵測車體與牆壁之間的距離，並將距離提供給模糊控制進行運算。

實作方式：

本程式是以向量來算車體與牆壁的距離，車體主要有三條向量分別為前距離$<\cos(\varphi) * t , \sin(\varphi) * t>$、右距離$<\cos(\varphi-45) * t , \sin(\varphi-45) * t >$、左距離$<\cos(\varphi+45) * t , \sin(\varphi+45) * t >$，這三條向量都會通過車體$(X_0 , Y_0)$，因此線性函數為$<X_0 + \cos(\varphi) * t , Y_0 +\sin(\varphi) * t>$.....等，其中 $t > 0$，再者牆壁的線性函數為$<X_1 + (X_2 - X_1) * k , Y_1 + (Y_2 - Y_1) * k>$，$0<= k <= 1$，當車體方向之線性函數和牆壁支線性函數有交點，此交點和車體的距離即為車體與牆壁的距離，依序對各個牆壁求交點和距離，其中距離最短的便是感測器偵測該方向的距離。

模糊控制：

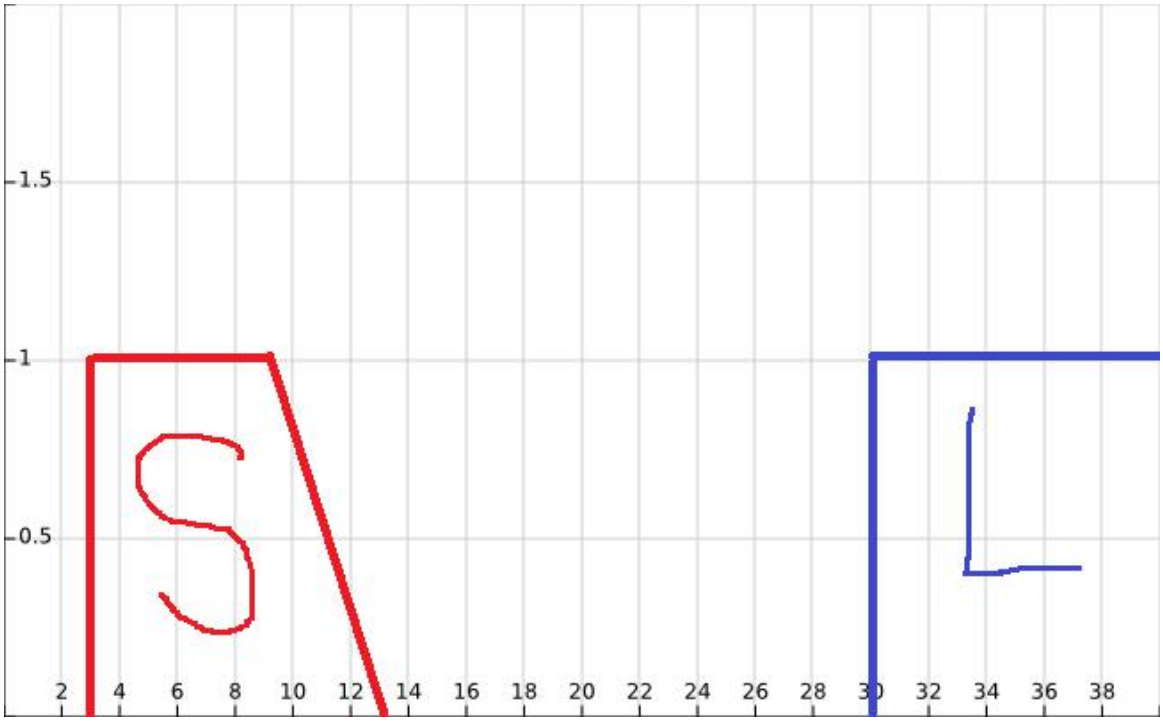
目的：

計算出車體該情況下，方向盤必須旋轉幾度。

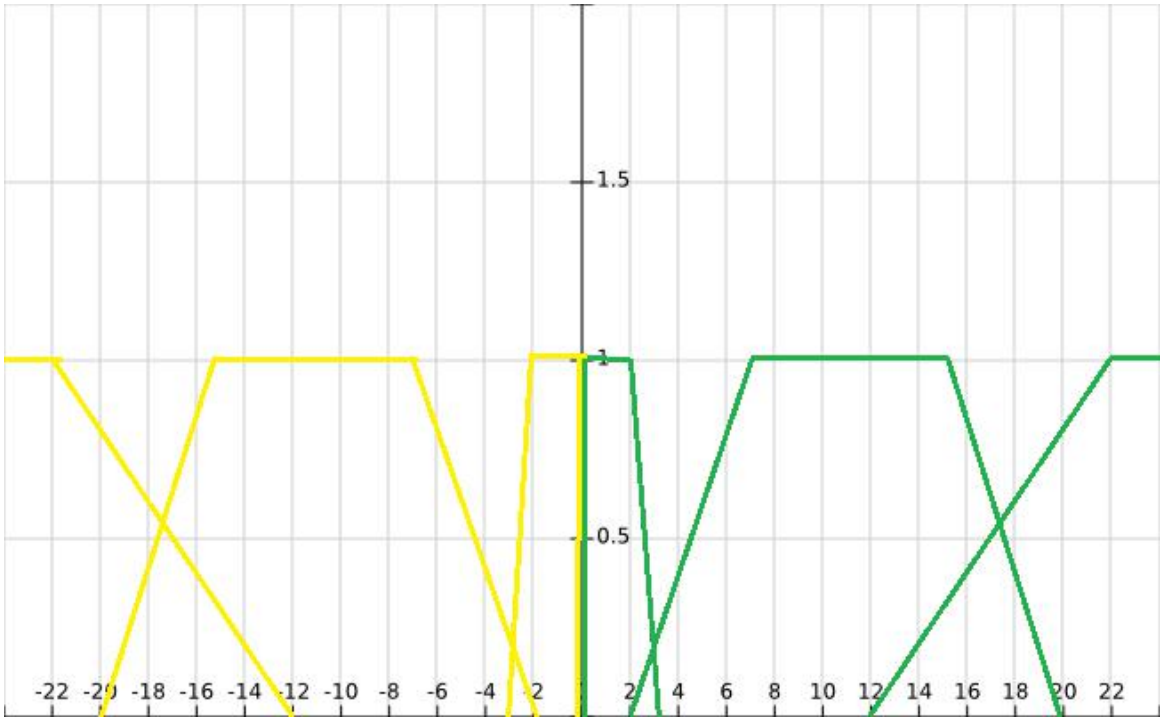
模組和規則：

取得模組之後，並可開始定義規則(如：If X = A then Y = B)

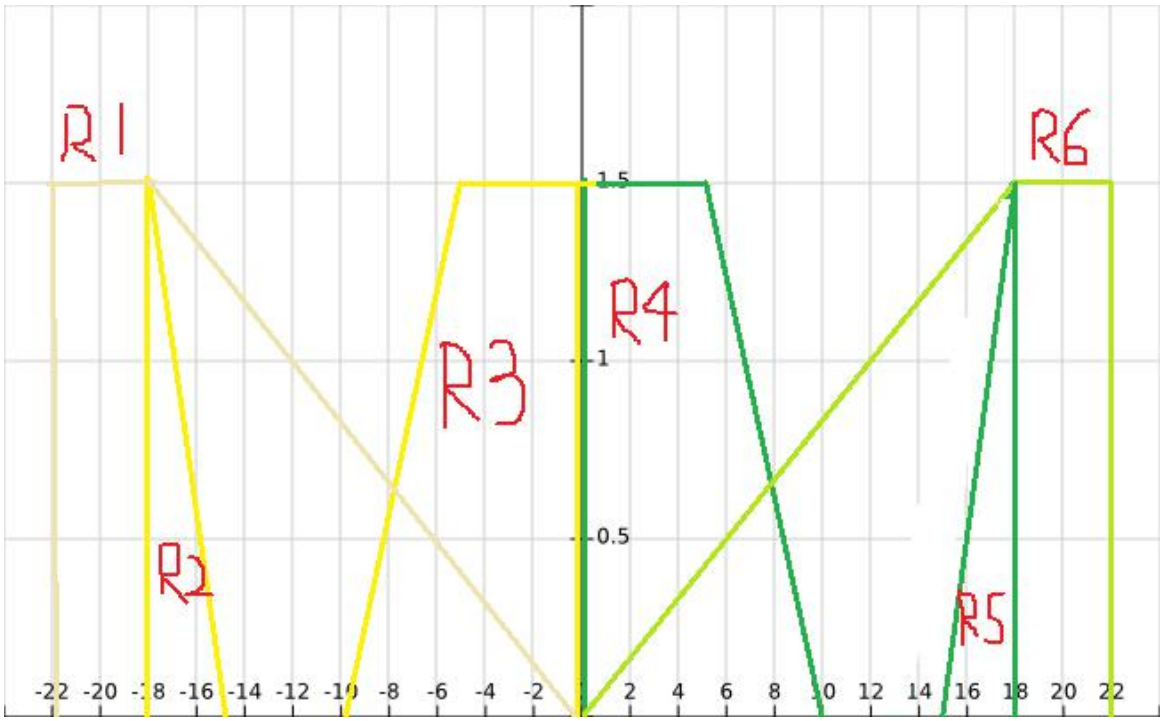
本程式主要為兩個變數做輸入(前方距離 和 右方距離 - 左方距離)



(圖 1) 前方距離模組(d1)



(圖二) 右方距離-左方距離(d2)



(圖三) 方向盤角度(c)

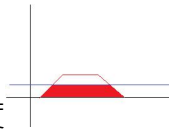
If d2 負數(大) then c1 If d2 負數(中) then c2
If d2 正數(大) then c6 If d2 正數(中) then c5
If d1(大) and d2 正數(小) then c4
If d1(大) and d2 負數(小) then c3

計算：

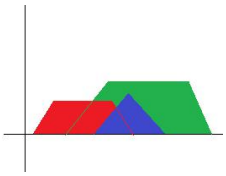
將感測器偵測出的距離，帶入各歸屬函數中(依規則)，取得啟動強度，再由啟動強度算出 c，最後再將 c 做去模糊化的動作。

本程式採取重心法，離散區間為 1 帶入計算。

實作重心法：

對每一個點(角度)依序帶入其原歸屬函數求其歸屬度，再與其啟動強度比較大小，較小的為該點在計算完之歸屬函數的歸屬度  ，必再

與其他規則的歸屬度比較大小，較大的為該點最終結果的歸屬度



，再將所有點的角度與歸屬度相乘再相加除以歸屬度的累加。

$$y^* = \frac{\sum_{i=1}^{L} \mu_C(y_i) \cdot y_i}{\sum_{i=1}^{L} \mu_C(y_i)}$$

心得：

本次程式我覺得最花時間的地方便是調模組參數和設定規則，再調參數時，真的花不少時間，不停地修正，不停撞牆，慢慢修改後，才逐漸讓車子可以順利可以轉彎而不觸碰牆壁，雖然期間花很多心力下去思考，但是看到自己的車可以自動過彎讓我覺得一切值得。

參考資料：

陳禹齊 同學 之 Python GUI 使用介面

http://morris821028.github.io/2014/05/19/lesson/hw-computational-intelligence/ 之 規則思考方向