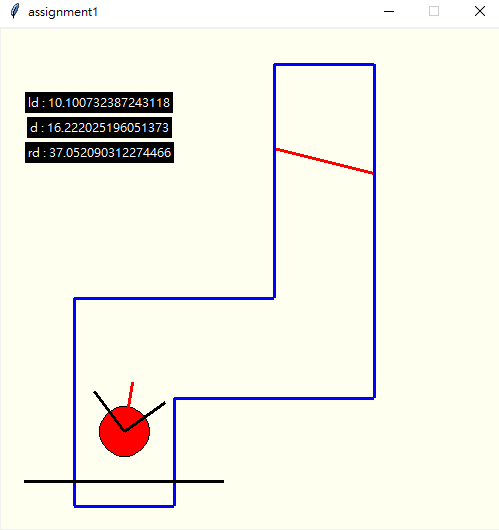
智慧型計算 作業一 說明文件:

1. 程式介面說明:

這次作業的實作程式為python, 並使用 tkinter作為 UI 的產生工具.



上圖為 UI 畫面的截圖, 外框和線的部分, 為依照case0.txt 來產生的 車道場地. 紅色怎代表車子, 紅色線為目前車子行進的方向, 黑色則為 另外兩個sensor 的方向. 左上方則是顯示目前 三個sensor 所感測到何 邊界的距離. 車子會根據這三sensor 所蒐集到的數據. 來決定下一個時間點所行進的方向theta 來進行自動駕駛的動作.

1. 程式碼說明:

程式碼是以python 來進行實作, 並沒有參考其他程式碼.說明的部分在原始碼 car.py中有使用註解來進行說明. 而模糊演算法的設置 在下頭會有詳述.

1. 模糊規則設定:

模糊化機構.

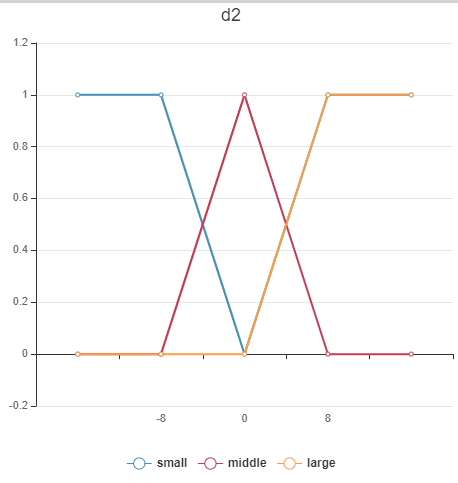
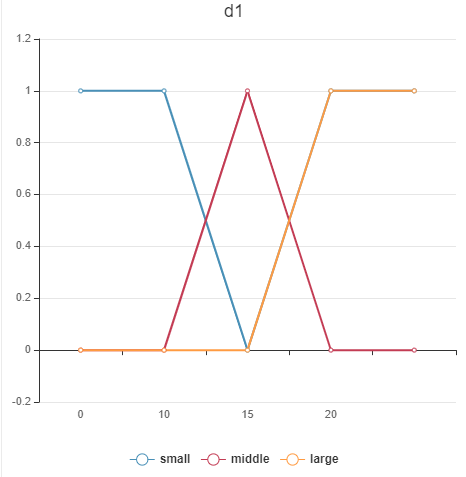
對輸入的模糊化機構處理使用 模糊單點的形式.



模糊推論引擎, 模糊規則庫:

此處分別針對 輸入 d (前方偵測距離), ld (左方偵測距離), rd (右方偵測距離) 先進行整理常 d1 = d, d2 = ld – rd.

針對此兩個數值d1, d2 做出模糊集合, 和最終的 輸出的集合(c) 三個模糊集合.

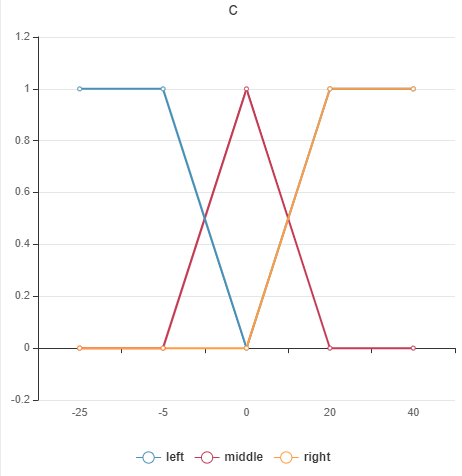


上面為我對 d1, 和 d2 模糊集合做出的定義,

d1 : < 15 is small, >= 10 and < 20 is middle, >=15 is large

d2 : < 0 is small, >= -8 and < 8 is middle, >=0 is large

輸出集合 c 則為



c : < 0 is left, >= -5 and < 20 is middle,>=0 is right

此處產生 九組模糊規則:

If d1 is large and d2 is large. and c is left

If d1 is large and d2 is middle. and c is middle

If d1 is large and d2 is small. and c is right

If d1 is middle and d2 is large. and c is left

If d1 is middle and d2 is middle. and c is middle

If d1 is middle and d2 is small. and c is middle

If d1 is small and d2 is large. and c is left

If d1 is small and d2 is middle. and c is middle

If d1 is small and d2 is small. and c is right

在這邊則是採用最小最大合成和 模糊蘊含RM 的方式. 會先由輸入 d1, d2 在分別的 模糊集合上 d1, d2 上 找出分別的啟動強度 a1, a2 取最低的 啟動強度 a 最後 在 模糊集合 c 上 投射, 最後再將 9 條規則產生的 模糊及蘊含做出 集合的動作.

去模糊化機構:

利用權重平均法找出去模糊化後的輸出角度.

Z =

1. 實驗結果:

經果多次調變規則, 和對於模糊函數的定義, 讓自走車 順利通過 作業要求的軌道 .

1. 分析:

這裡所產生的可調變 變數 和規則, 是我規則寫完 再去 經由 反覆實驗 後, 所找出 可以順利通過變數的組合以及規則. 那基本上 這個規則 就是照著 左邊大走左邊. 右邊多走右邊 . 沒事走中間.這種概念, 在經由模糊集合的定義 和使用, 讓 車子的行走軌跡可以更加的有彈性, 以及對於地形的適應力!