

電腦模擬車的運動方程式如下：

$$x(t+1) = x(t) + \cos[\phi(t) + \theta(t)] + \sin[\theta(t)]\sin[\phi(t)] \quad (10.18)$$

$$y(t+1) = y(t) + \sin[\phi(t) + \theta(t)] - \sin[\theta(t)]\cos[\phi(t)] \quad (10.19)$$

$$\phi(t+1) = \phi(t) - \arcsin\left[\frac{2\sin[\theta(t)]}{b}\right] \quad (10.20)$$

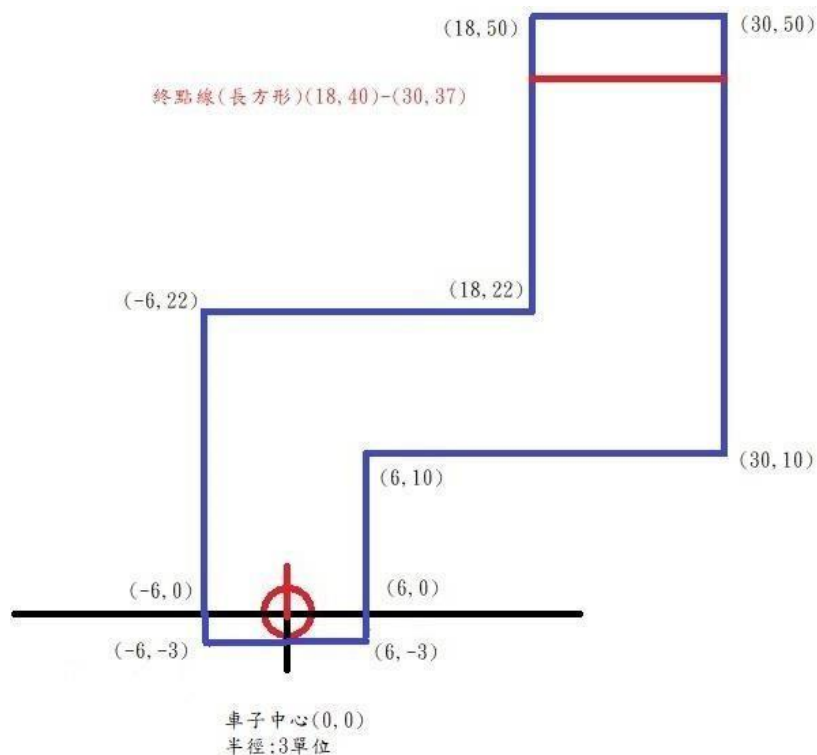
其中 $\phi(t)$ 是模型車與水平軸的角度， b 是模型車的長度， x 與 y 是模型車的座標位置， $\theta(t)$ 是模型車方向盤所打的角度，我們對模擬的輸入輸出變數限制如下：

$$\begin{cases} \phi(t) \in [-90^\circ, 270^\circ] \\ \theta(t) \in [-40^\circ, 40^\circ] \end{cases}$$

(Notice：模型只需輸出“方向盤角度”，再根據公式改變 x 、 y 、 $\phi(t)$)

車體大小設定為直徑 6 單位，初始角度+90 度

車體中心設有感測器，可偵測正前方與左右各 45 度之距離。根據前左右三個感測器的數值，輸入RBFN網路並輸出方向盤角度，並透過改變方向盤角度（注意：方向盤的角度右轉為正）讓車輛能在不碰壁的狀況下到達終點，畫出過程並記錄各項數值。



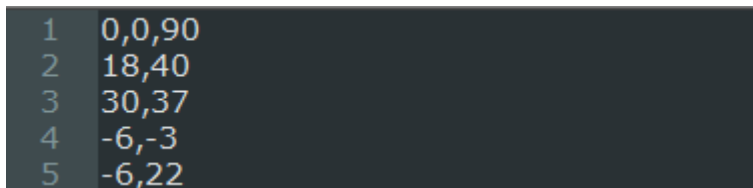
程式要求:

附檔說明：

- 訓練資料集有兩種格式的移動紀錄：train4DA11.txt、train6DA11.txt，請利用這兩種訓練集分別設計兩個網路。

(兩者可為同一模型僅差在輸入維度不同，亦可為兩個完全不同模型，端看同學設計)

- train4D.txt 格式:前方距離、右方距離、左方距離、方向盤得出角度(右轉為正)
- train6D.txt 格式:X 座標、Y 座標、前方距離、右方距離、左方距離、方向盤得出角度(右轉為正)
- 軌跡座標點(文件為 unix 格式，建議以 notepad++等編輯器瀏覽):



```
1 0,0,90
2 18,40
3 30,37
4 -6,-3
5 -6,22
```

第一行為車體中心起始的 (x, y, ϕ degree)

二， 三行標示出終點區域位置

第二行為區域左上角 (x, y)

第三行為區域右下角 (x, y)

(終點為一個矩形區域)

第四行(含)以後為軌道邊界節點

(x, y) 直到最後一行

最後一行與第四行數值相同

形成一個封閉的跑道

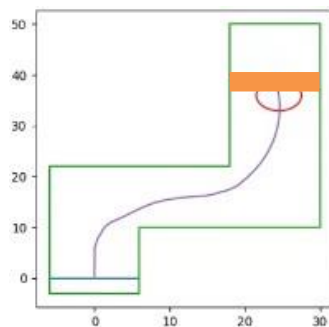
軌道於起點線右下角為 (-6, -3); 左下角為 (6, -3)

起點線為 (-6, 0) -> (6, 0)

程式要求:

1. GUI 介面:
 - 甲、讀取軌道並根據軌道座標點繪製軌道及自走車
 - 乙、以動畫顯示自走車每一時刻的位置及行走方向
 - 丙、顯示左、前、右 3 個測距 sensor 測得的距離
2. 紀錄自走車行駛路徑，紀錄格式下面會說明
3. 讀取行進路徑記錄檔讓自走車根據紀錄檔中的路徑行走碰撞偵測，自走車碰到軌道及終點須能自動停止
4. 將左、前、右 3 個測距 sensor 測得的距離當作模型輸入，利用 RBFN 訓練出一個模型可以使車子順利抵達終點。

(附圖為 軌跡範例圖)



作業繳交:

- A. 可執行檔
 1. 必須包含 UI，並能顯示模擬結果
 2. 演算法不得使用現有的 AI 框架（如 caffe, tensorflow, pytorch 等）

(建議只使用numpy等低階程式庫，GUI方面則不設限。有疑慮可以來信詢問)
- B. 程式源始碼
- C. 說明文件(PDF檔):

內容包含: (1)程式介面說明、(2)程式碼說明、(3)實驗結果(包含移動軌跡截圖)與(5)分析。

實驗內容需能以繳交檔案重現，請於文件內說明如何操作。
- D. 兩種格式的移動紀錄(成功走到終點的版本): **track4D.txt**、**track6D.txt**

track4D.txt 格式: 前方距離、右方距離、左方距離、方向盤得出角度(右轉為正)

track6D.txt 格式: X 座標、Y 座標、前方距離、右方距離、左方距離、方向盤得出角度(右轉為正)

角度以 degree 輸出，**rad 視為無效檔案**，請以空格分隔項目

範例如下:

```
0.0000000 0.0000000 22.0000000 8.4852814 8.4852814 2.3702363
0.0000000 0.9991444 21.0028513 8.3706824 8.6047198 2.1288341
0.0137760 1.9983593 20.0084911 8.2526888 8.7363457 1.8747606
```