

# 國立交通大學

多媒體工程研究所

碩士論文

模擬賽車遊戲中駕駛風格模仿之技術

Techniques of Driving Style Imitation in Simulated  
Car Racing Games

研究生：廖耿德

指導教授：王才沛 教授

中華民國 一百零二年八月

模擬賽車遊戲中駕駛風格模仿之技術

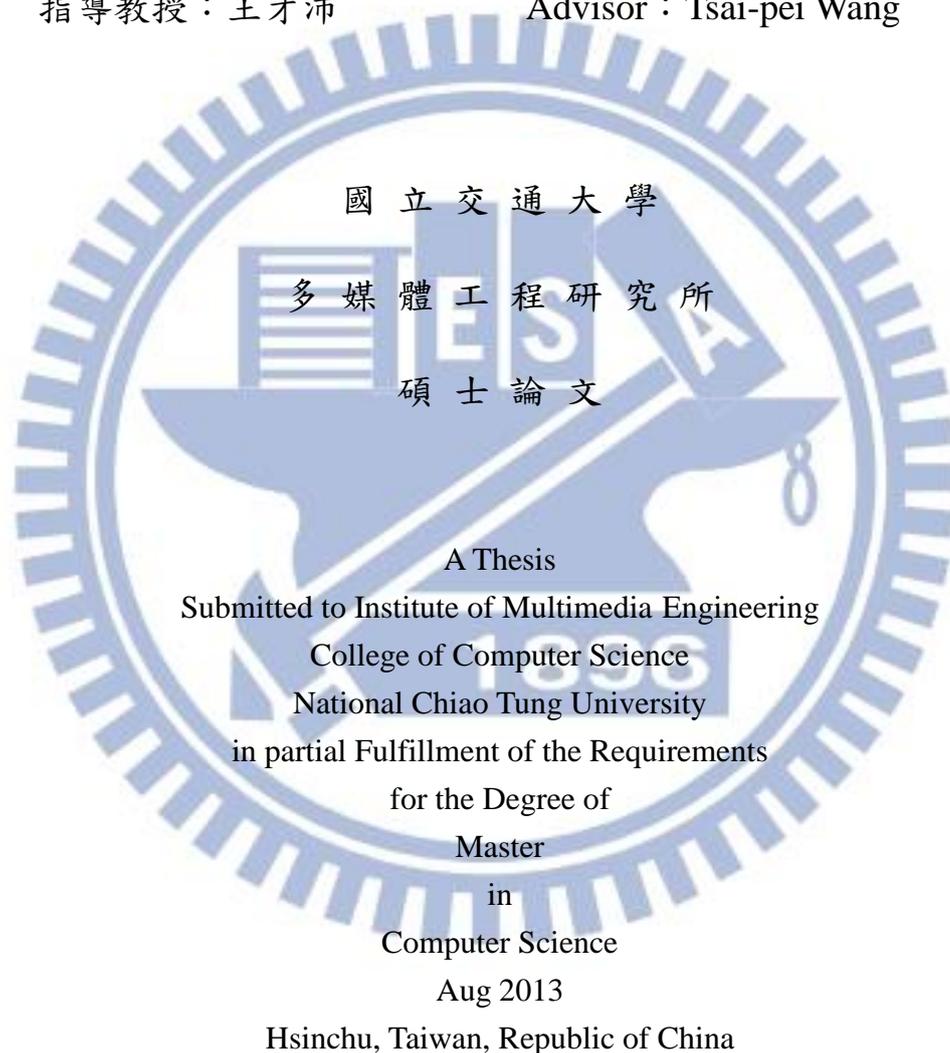
Techniques of Driving Style Imitation in Simulated Car Racing Games

研究生：廖耿德

Student : Keng-te Liao

指導教授：王才沛

Advisor : Tsai-pei Wang



中華民國一百零二年八月

# 模擬賽車遊戲中駕駛風格模仿之技術

學生：廖耿德

指導教授：王才沛

國立交通大學多媒體工程所

## 摘要

本文的研究目的，是在 TORCS 平台上對於人類和非人類玩家進行駕駛風格上的分析及模仿。本文將目前已被提出的控制器架構分為高階和低階控制器，並且透過整合這二類型的控制器達到了更好的模仿效果。

對於模仿效果的評估，本文提出了計算風格相似度的方法。這個方法也有助於讓訓練後的控制器適應未接觸過的賽道。本文中透過使用多目標最佳化演算法，讓控制器在非訓練用的賽道中也能作到自我調整，將駕駛風格及速度表現更加提升。根據實驗，當以一個速度表現接近最佳情況的 NPC 為模仿對象時，即使是在非常困難且未接觸的賽道中也可以表現出相似的軌跡，而且只慢了 4%~14%。

對於人類玩家是否具有獨特的自我駕駛風格，本文中也進行了討論及實驗。透過提出的駕駛風格相似度判斷器，可以分析出玩家們風格的差異，並且作到透過駕駛資料來辨識出玩家。此外，在過去的研究中，如果模仿的對象是人類玩家，那麼訓練出的控制器容易頻繁的發生車禍而有著很低的速度表現。對於這個問題，本文中提出了在具有穩健的駕駛風格的前提下，訓練出的控制器仍然可以表現出模仿對象的駕駛特徵並且幾乎不發生車禍的模仿方法。

# Techniques of Driving Style Imitation in Simulated Car Racing Games

Student : Keng-te Liao

Advisor : Tsai-pei Wang

Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University

## Abstract

This thesis is about the analysis and imitation of human and non-human players' driving styles in the TORCS platform. We classify existing controller architectures as low-level controllers or high-level controllers, and we achieve better results of imitation by integrating these types of controllers.

For evaluating the imitation results, we propose a method to estimate the similarity of driving styles. It is also useful for a trained controller to adapt to a new track. We use multi-objective optimization algorithm as a method for controllers' self-adaptation. According to our experiments, if a NPC with almost optimal speed performance is selected as the imitation target, the trained controller is able to produce similar trajectories and is only slower by 4%~14% than the imitation target in very difficult and unfamiliar tracks.

In this thesis, we also discuss whether different human players have unique driving styles. By using our proposed method for estimating driving style similarities, it is possible to observe the differences of driving styles and we can even recognize a player by analyzing the driving data. Besides, according to some proposed researches, if human players are selected as the imitation targets, the trained controllers usually crash on the tracks frequently. Therefore, the speed performance is quite low. For dealing with this issue, we propose a method to train a controller with robust driving style. Such a trained controller is able to show some driving behaviors of the target human player and seldom crashes.

## 誌謝

能夠完成這篇論文，首先要感謝家人的支持，讓我能夠順利就讀交通大學，並且沒有後顧之憂地進行研究。再來要感謝指導教授王才沛老師，在這二年中盡心盡力的指導。在研究過程中遇到的問題，常常是多虧了老師的指點才能順利解決。在撰寫論文時，老師在百忙之中仍然是逐字的批改草稿，讓我學到了很多，並且讓論文更加的完整。在這二年中，老師也給予了許多論文研究主題以外的知識，讓我過得非常充實。

在這裡還要感謝育任、瀚賢、向德和堡評學長在這二年中的陪伴，讓我在研究之餘可以放鬆心情。在實驗室中的閒聊、看電影以及偶爾的大餐都是很棒的回憶，同時也讓我更有力氣進行研究。

最後要感謝兆祥、台盛、經國和柏淵願意當論文中的實驗者。你們的資料對這篇論文有很重要的影響。



# 目錄

摘要	.....	i
Abstract	.....	ii
誌謝	.....	iii
目錄	.....	iv
表目錄	.....	vi
圖目錄	.....	vii
第一章	研究目的.....	1
第二章	TORCS 介紹與文獻探討.....	4
第三章	研究方法—非人類玩家駕駛風格模仿.....	7
3.1	目標玩家與研究環境.....	7
3.2	方法概觀.....	9
3.3	控制器建構.....	10
3.3.1	控制器架構.....	10
3.3.2	低階控制器.....	11
3.3.3	高階控制器.....	14
3.3.4	指令結合器.....	15
3.3.5	額外加速模組.....	16
3.4	風格相似度判斷器.....	17
3.5	新賽道適應機制.....	19
第四章	實驗結果—非人類玩家駕駛風格模仿.....	21
4.1	控制器訓練結果.....	21
4.2	風格相似度判斷器效果評估.....	25
4.3	模仿實驗結果.....	27
第五章	研究方法—人類玩家駕駛風格模仿.....	34

5.1 目標玩家與研究環境.....	34
5.2 方法概論.....	35
5.3 控制器架構.....	36
5.4 人類玩家駕駛風格分析.....	37
5.5 模仿效果評估方法.....	38
第六章 實驗結果—人類玩家駕駛風格模仿.....	40
6.1 人類玩家駕駛風格分析結果.....	40
6.2 模仿實驗結果.....	43
第七章 結論與未來展望.....	47
參考文獻 .....	49



## 表目錄

表 2-1 TORCS 效用器列表 .....	4
表 4-1 Wheel 2 風格相似度判斷結果 .....	26
表 4-2 E-Road 風格相似度判斷結果 .....	27
表 4-3 風格相似度判斷結果 confusion matrix(橫向) .....	27
表 4-4 風格相似度判斷結果 confusion matrix(縱向) .....	27
表 4-5 Wheel 2 賽道速度表現適應結果 .....	28
表 4-6 Wheel 1 賽道速度表現適應結果 .....	29
表 4-7 E-Road 賽道速度表現適應結果 .....	31
表 4-8 E-Track 6 賽道速度表現適應結果 .....	32
表 6-1 異常狀態比例統計 .....	40
表 6-2 CG Speedway number 1 賽道風格相似度分析結果 .....	41
表 6-3 CG Track 2 賽道風格相似度分析結果 .....	41
表 6-4 Aalborg 賽道風格相似度分析結果 .....	42
表 6-5 Alpine 1 賽道風格相似度分析結果 .....	42
表 6-6 E-Track 4 賽道風格相似度分析結果 .....	42
表 6-7 總平均排名 .....	43
表 6-8(a) 縱向平均排名值統計 .....	43
表 6-8(b) 橫向平均排名值統計 .....	43
表 6-9 人類玩家駕駛風格模仿相似度表現結果一 .....	44
表 6-10 人類玩家駕駛風格模仿相似度表現結果二 .....	45
表 6-11 人類玩家駕駛風格模仿速度表現結果 .....	46

# 圖目錄

圖 3-1 賽道圖 .....	8
圖 3-2 非人類玩家模仿方法流程圖 .....	9
圖 3-3 控制器架構 .....	10
圖 3-4 成員函數參數化的範例 .....	13
圖 3-5 距離探測器 .....	14
圖 3-6 前瞻探測器 .....	14
圖 3-7 軟閥 .....	16
圖 3-8 玩家的輸出指令 .....	18
圖 3-9 判斷器推測的輸出指令 .....	18
圖 3-10 判斷器運作流程 .....	19
圖 3-11 演化結果範例 .....	20
圖 4-1(a) Front 演化結果 .....	21
圖 4-1(b) Max10 演化結果 .....	22
圖 4-1(c) Max20 演化結果 .....	22
圖 4-1(d) Fuzzy System Output 演化結果 .....	22
圖 4-1(e) 軟閥演化結果 .....	23
圖 4-2 油門控制指令比較 .....	23
圖 4-3 轉向控制指令比較 .....	24
圖 4-4 低階控制器的結果 .....	25
圖 4-5 低階+高階控制器的結果 .....	25
圖 4-6 Wheel 2 賽道軌跡比較 .....	28
圖 4-7 Wheel 2 賽道局部軌跡比較 .....	29
圖 4-8 Wheel 1 賽道軌跡比較 .....	30

圖 4-9 Wheel 1 赛道局部軌跡比較.....	30
圖 4-10 E-Road 赛道軌跡比較.....	31
圖 4-11 E-Track 6 赛道局部軌跡比較.....	32
圖 4-12 E-Track 6 赛道軌跡比較.....	32
圖 5-1 方向盤.....	34
圖 5-2 人類玩家模仿方法流程圖.....	36
圖 5-3 控制器架構.....	37



# 第一章 研究目的

在賽車遊戲中，玩家必須對於快速變化的環境進行因應，以達到最短時間內跑完賽道的目標。由於人類玩家往往無法輕易的知道或實行一個“最佳跑法”，因此實務上玩家可能會發展出一套自己能夠應付並且能接受的跑法，此即成為了玩家個人的駕駛風格。如果能夠讓一個 non-player character(NPC)學習一個特定玩家的駕駛風格，遊戲的可玩性可能能夠更高。比如說，有一位玩家 A 想和另一個玩家 B 進行遊戲，但 B 不在遊戲機前，那麼 A 就可以選擇學習了 B 的駕駛風格的 NPC 進行遊戲。本研究的目的就是建構出一個 NPC，它能夠分析並且學習一個特定玩家的駕駛風格，並且能夠一定程度的作到從駕駛資料去辨認出駕駛的玩家。

目前有一個稱為 TORCS 的模擬賽車開源軟體，它能夠讓人自己建構一個賽車 NPC，並且經過修改後可以完整記錄出車子在賽道中所有時間點的狀態資料。本次研究就是以這個軟體作為實驗平台。

模仿駕駛風格這個研究主題目前已經累積了許多成果，其中 AI 的架構在本次研究中大致上分為二類。第一類是低階控制器，這種控制器的輸入是區域性的量測方面的數值訊息而非抽象性的描述，這種架構偏向於模仿玩家直覺性的操作。另一類是高階控制器，輸入是廣域性的描述彎道的緩急程度或形狀等抽象性資訊。這種控制器大多會先決定出賽車應有的目標狀態，再計算出要達到該狀態所需要的操控指令，這近似於模仿玩家的操作策略。在本次研究中，提出了將這二類控制器結合成單獨一個控制器的方法，目的是結合低階和高階這二類控制器的優點，讓模仿的效果和速度表現都可以有新的進展。

模仿的對象在相關的領域中也是一個研究主題。有一些研究中的模仿對象本身就是 NPC，有一些是以一位特定的人類玩家為目標對象，也有單純希望訓練出的 NPC 可以駕駛得像是一般人類玩家就好。本次研究的模仿對象包含了

TORCS 內建的 NPC 以及八位人類玩家，並且針對每一個特定的 NPC 或人類玩家進行風格上的分析及模仿。

對於駕駛風格模仿效果的評估，在已提出的研究中，有只粗略的分為積極或消極性風格的，或在已知賽道中計算 NPC 和單一目標玩家駕駛資料差異等的評估方法。這些現存方法存在著二個問題：第一個是只能在模仿對象駕駛過的賽道上進行風格評估。這個限制使得評估方法不具備太高的實用性，一但遊戲中加入了新賽道，那麼原本訓練出的 NPC 的駕駛風格就無從評估。此外，對於使用演化計算，並且將風格相似度作為目標函數的模仿方法也沒有辦法在新賽道上運作。第二個問題是對於多位玩家的駕駛風格無法作明確的區分和辨識。現存評估方法中，常會透過和單一玩家的駕駛資料比較來判定模仿效果的好壞，但是沒有對於風格的獨特性進行討論，因此無從得知訓練後的 NPC 表現出的是模仿對象獨特的駕駛特徵，還是大多數玩家共通的駕駛習慣。對於上述的二個問題，本研究對於駕駛風格相似度的計算提出了一個數學性的方法，稱為駕駛風格相似度判斷器。它可以在新接觸的賽道中評估模仿的效果，而且根據實驗結果，它同時能一定程度的區分出不同玩家的駕駛風格，並且從駕駛資料辨識出特定玩家。對於人類玩家是否具備獨特的個人駕駛風格這個問題，本研究也使用了判斷器進行實驗和討論。

在目前累積的研究中，對於訓練出的 NPC 如何在新接觸的賽道中調整及適應並沒有提出專門的解決方法。常見的狀況是 NPC 在訓練用的賽道中可以順利行駛，但在測試用的新賽道中卻頻繁地出現車禍而嚴重的影響速度表現。這使得該 NPC 無法呈現出賽車遊戲應有的競速方面的遊戲性。若是針對車禍問題只進行降低進入彎道的速度這種常見於競賽用 NPC 的調整方法，那麼訓練出的 NPC 很有可能在調整的同時失去了模仿對象的駕駛特徵。過去由於在新賽道中無法進行準確的風格評估，使得無法設計出兼顧風格表現和速度表現的目標函數。本研究透過提出的風格相似度判斷器和多目標最佳化(multi-objective optimization)這個方法，設定其中一個目標是速度表現，另一個是駕駛風格的相似度，讓訓練後

的 NPC 可以在過去未接觸過的賽道中進行自我調整。

當以人類玩家為模仿對象時，過去的研究中常會碰到訓練出的 NPC 車禍頻繁。除了造成駕駛風格不易被觀察到外，也嚴重的影響到速度表現。對於以競速為目標的賽車遊戲來說是一個需要解決的問題。本次研究對於這一問題提出了在駕駛風格有一定程度的限制下，訓練出的 NPC 仍然可以保有模仿對象的駕駛特徵並且同時有具競爭性的速度表現。



## 第二章 TORCS 介紹與文獻探討

TORCS 是一款開放源碼的賽車模擬器。它能夠讓玩家親自駕駛模擬賽車，或是自己透過程式設計，創造一個 NPC 來進行遊戲。TORCS 提供了複雜的物理引擎、多樣性的賽道、車輛模組以及多種競賽模式，讓現實賽車中的許多細節都可以被精確地模擬出來。TORCS 在程式架構上是一種 server 對 client 的運作模式。遊戲引擎執行遊戲時，每隔 0.02 秒的間隔(稱為一個 game tick)會將遊戲的環境狀況傳給 client，client 則將計算出的賽車控制指令回傳給 server。在駕駛過程中，每個 game tick 的賽車狀態及環境資訊都可以由修改後的程式記錄下來，讓研究者可以作後續的分析。此外，TORCS 具備快速模擬駕駛情況的功能，讓原本長達數十分鐘的車賽可以在數秒鐘內就模擬完畢，這讓開發者可以更簡單並快速的實現出學習演算法。這些特性讓 TORCS 成為了一個適合人工智慧相關研究者使用的開發及測試平台。

本次研究中使用的 TORCS 偵測器包含了速度偵測、賽道各區段曲率半徑偵測、trackPos 和 Track 偵測器。trackPos 表示車子和賽道中心線的距離，已經用賽道的寬度做正規化，-1 表示賽道右側邊界，+1 表示賽道左側邊界，大於+1 或是小於-1 的值代表車子超出邊界。Track 偵測器包含 19 個距離偵測器，距離探測器能夠回傳它前方距離賽道邊緣的距離，可以探測的最遠距離是 200 公尺。從車子前方的正右方( $-\pi/2$ )到正左方( $\pi/2$ )的範圍之內，每隔  $\pi/18$  有一個偵測器。當車子跑到賽道外或偵測方向超過車子前方 180 度時則會得到不可靠的值。使用的效應器如表 2-1 所列。

表 2-1 TORCS 作用器列表

變數名稱	數值範圍	變數說明
Accel	[0,1]	油門控制，0 為不踩油門，1 為全速。
Brake	[0,1]	剎車控制，0 為不踩剎車，1 為全煞。
Steering	[-1,1]	轉向控制，-1 為最大右轉，1 為最大左轉。

目前在 TORCS 相關的研究中，許多是專注在建構出速度表現佳，而且能對於彎道進行妥善處理的 NPC。[3]中使用了模糊系統(fuzzy system)來處理油門控制問題。模糊系統透過人工定出的成員函數(membership function)和模糊規則(fuzzy rule)計算出當前環境下的目標速度，再將其與當前速度的差值經過軟閾(soft threshold)計算後得出油門控制的指令值。在轉向控制的部分基本上是讓賽車往十九個距離探測器中值最大的探測器方向前進。[2]同樣也用到了模糊系統，但 NPC 會對於成員函數及模糊法則進行調整以達到較好的速度表現以及較少的車禍。調整的方法是使用基因演算法。[8]使用 NEAT(NeuroEvolution of Augmenting Topologies)建構 NPC，[7]、[10]則都是使用了演化計算的方法調整 NPC 的結構。由簡單構造的 NPC 開始演化，之後漸漸演化出較複雜的結構以適應賽道環境。[6]則是研究如何有效的透過探測器計算出賽道的形狀。由於 TORCS 競賽中允許得到的環境資訊有限，因此如果能透過方法取出更多的賽道資訊就有助於讓 NPC 作出更準確的駕駛判斷。[1]是研究讓 NPC 具備超車行為，讓 NPC 在有對手的阻擋下也能發揮出好的速度表現。使用的學習演算法是 Q-learning。

在模仿特定跑法這個主題上有一些已發表的相關研究結果。在[4]的研究中，使用了類神經網路預測軌跡和速度，但學習的目標並不是一個特定的玩家，而是希望 NPC 可以表現得像是人類玩家就好，因此訓練資料混合了許多人類玩家的駕駛資料。在實驗結果方面，由於對於跑法是否像人類這件事並沒有一個衡量標準，因此難以判定效果優劣，此外有一個問題是 NPC 不一定有辦法完全照著預測出的軌跡和速度駕駛。

在[5]的研究裡同樣也使用到了類神經網路，和[4]不同的是它的輸出不是嘗試預測車子應該有的速度和位置，而是直接輸出油門、煞車、轉向和排檔的指令。它的目標是建構出一個駕駛時可以作出仿人類決策的控制器。在實驗的結果上這篇研究並沒有表現得太好，製造出來的 NPC 由於過多的車禍而無法完整的跑完賽道。

[9]的模仿對象是 TORCS 內建速度表現最好的 NPC，而非人類玩家。使用的

方法是 k-nearest neighbor 或演化式類神經網路。NPC 的輸入是較高階的環境資訊，例如輸入一個彎道的緩急程度，而非單純直接將距離探測器的值作為輸入。此篇研究提出了前瞻探測器(look ahead sensor)來探測前方彎道的形狀作為高階資訊。前瞻探測器將賽車前方路段以十公尺為單位劃分成數個區間，並計算出每個區間中的曲率半徑作為區間值，若是直線路段則值設為零。NPC 駕駛時獲得的環境資訊就是前方的數個區間值。之後透過 KNN 或是演化式類神經網路決定出當下的目標速度和目標位置，並且用事先人工定好的操作規則來達到目標狀態。實驗的結果是 NPC 跑出來的路線與速度都十分相似模仿的目標玩家，在速度表現上和最快的內建 NPC 比較起來只有慢了大約 15%。但遇到難度較高的賽道仍然會出現車禍頻繁的現象而無法達到模仿和速度快的效果。

在[11]中使用到多目標最佳化(multi-objective optimization)的演化計算方法。演化的目標是除了希望模仿玩家的駕駛特徵外，還希望 NPC 不要因為一味模仿駕駛習慣而發生車禍，進而失去速度表現。這篇研究中在駕駛風格的判斷上主要從兩個方向討論。第一個是從數字上討論：計算在相同情境下，模仿對象和 NPC 輸出的操控指令值的均方差(Mean Square Error)。在這個部分顯示出在加上模仿駕駛特徵這個演化目標後，均方差有明顯的降低。第二個方向是人為判定：它請多個人來針對 NPC 跑出來的結果進行風格上的辨別。實驗的結果是對於積極或謹慎這二種風格能夠明顯被人辨識出，而且演化後的 NPC 是否具有人類玩家的駕駛風格這點也同樣可以被區分出來。在速度表現上，由於駕駛的穩健程度也是演化目標之一，使得演化後的 NPC 比較不會出現頻繁撞車的情形，也使得速度表現有所提升。不過這篇研究並沒有針對個別玩家風格的差異作分析。

## 第三章 研究方法—非人類玩家駕駛風格模仿

### 3.1 目標玩家與研究環境

非人類玩家和人類玩家的差異在於非人類玩家總是保持著固定的駕駛策略，而且在相同的環境中，非人類玩家能夠準確的進行相同的操作。人類玩家由於駕駛技術上的限制和情緒影響，使得駕駛方式的變異性較高，讓後續的分析及模仿的難度提高。因此在這一章中先以非人類玩家為模仿對象，一方面能夠測試提出的方法在沒有不安定因素影響下的有效程度，另一方面由於人類玩家透過不斷練習後可能可以具備非常精準且有一致性的操作，在這種理想情況下針對非人類玩家的分析及模仿方法就能適用於人類玩家。

非人類玩家駕駛風格模仿的目標玩家是 TORCS 內建的 NPC，Berniw。Berniw 使用的感測器可以直接讀取賽道中各個區段的詳細資料，並且事先計算出可以最快通過彎道的最佳路線和速度，建構出非常接近最佳速度表現的控制器。此外，在 TORCS 內建的賽道中，Berniw 幾乎不會發生嚴重車禍，這使得 Berniw 成為一個適合作為模仿對象的非人類玩家。

本次研究中使用的 TORCS 版本是 1.3.4，並從中挑選了九個賽道進行實驗。九個賽道中有五個是作為模仿實驗時訓練用的賽道，分別是 CG Speedway number 1、CG track 2、Aalborg、Alpine 1 以及 E-Track 4。測試用的賽道是 Wheel 1、Wheel 2、E-Track 6 和 E-Road。賽道圖如圖 3-1 所示。

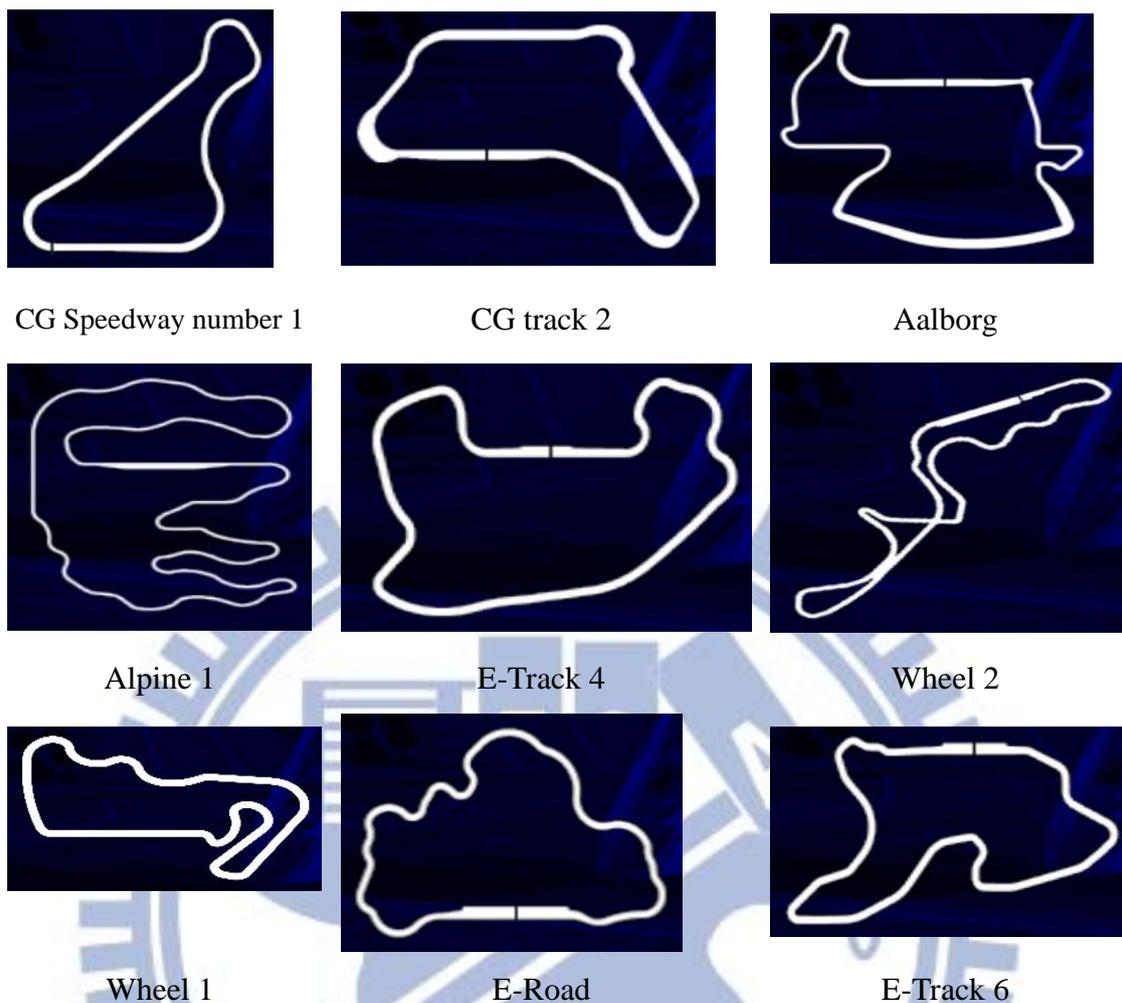


圖 3-1 賽道圖

收集 Berniw 駕駛資料時，會讓 Berniw 在訓練賽道中駕駛二圈，其中第二圈裡每一個 game tick 中賽車的速度、trackPos、Track 偵測器中 19 個距離探測器值以及前方二百公尺內各個賽道區段的曲率半徑會被紀錄下來作為駕駛資料。這些資料在被實際使用前會先經過三個前處理。

1. 左右鏡射:

將每個賽道中的駕駛資料作賽道上左右方向的鏡射，對應的轉向控制指令也會變為等大反向。實際上造成的結果是相當於玩家再駕駛了一個彎道左右相反的賽道，訓練資料的筆數也因此變成二倍。這對於方法中使用到 KNN 的部分有著特別正向的幫助。

2. 異常狀態過濾:

賽車在發生車禍時產生的駕駛資料幾乎不具有被模仿的價值。對於這種異常狀態下的資料會使用一個過濾機制去掉，避免成為訓練資料或是被風格相似度判斷器評估。理想上從車禍確實發生前的超速狀態到車禍結束後回歸正常狀態這段時間都應該被認定成異常狀態，但這部分很難直接讓機器自動辨識出來，因此目前只使用很簡單的過濾機制：當賽車處於賽道外或是處於倒車狀態時就被歸類成異常狀態。

### 3. 儲存結構:

由於收集駕駛資料時是每個 game tick 就記錄一筆資料，因此最後的訓練資料筆數非常龐大。尤其是人類玩家因為駕駛的圈數較多，資料數量會更大，平均來說大約十五萬筆。在演化計算中的個體數和世代數會更加放大 KNN 的時間成本，因此在 KNN 的實作方面是使用 k-d 樹[13]完成。收集的 player 資料都會另外用 k-d 樹的結構儲存。

## 3.2 方法概觀

模仿方法的流程如圖 3-2 所示。

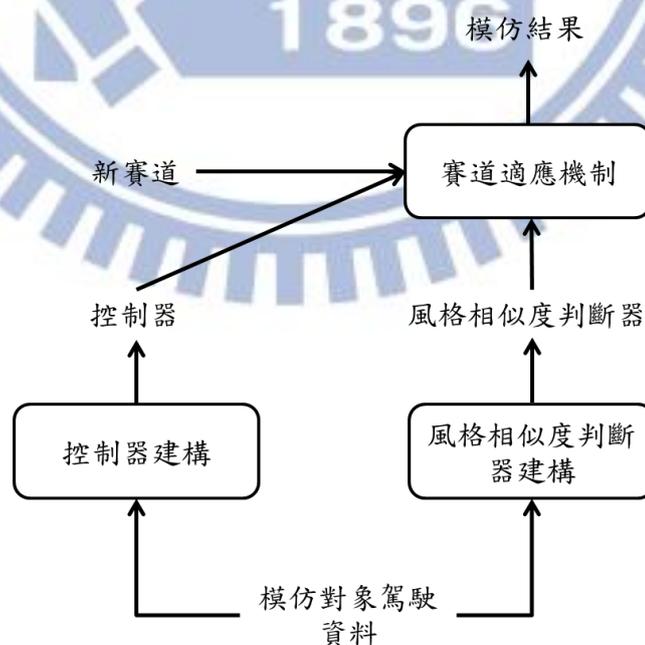


圖 3-2 非人類玩家模仿方法流程圖

收集到的模仿對象駕駛資料會用來建構控制器和風格相似度判斷器。控制器建構的目標是產生出一個具有模仿對象駕駛風格的控制器，風格相似度判斷器則是能夠依據和模仿對象風格的接近程度給予數值。數值越小代表風格越接近，反之則越遠。

適應機制是為了讓控制器在未接觸過的賽道行駛時，可以進行自我調整以達到更好的表現。調整的目標有二個：第一個是速度方面的表現，希望控制器能在最短的時間內行駛完一圈。第二個是駕駛風格的表現，希望控制器能夠盡量表現出模仿對象的駕駛特徵。為了同時對這二個目標作最佳化，使用的方法是多目標演化演算法(multi-objective evolutionary algorithm)。目標函數的輸出是控制器在該賽道中行駛一圈所花的時間，以及風格相似度判斷器給出的數值。

### 3.3 控制器建構

#### 3.3.1 控制器架構

控制器如圖 3-3 所示，主要由三個部分構成：低階控制器、高階控制器以及指令結合器。

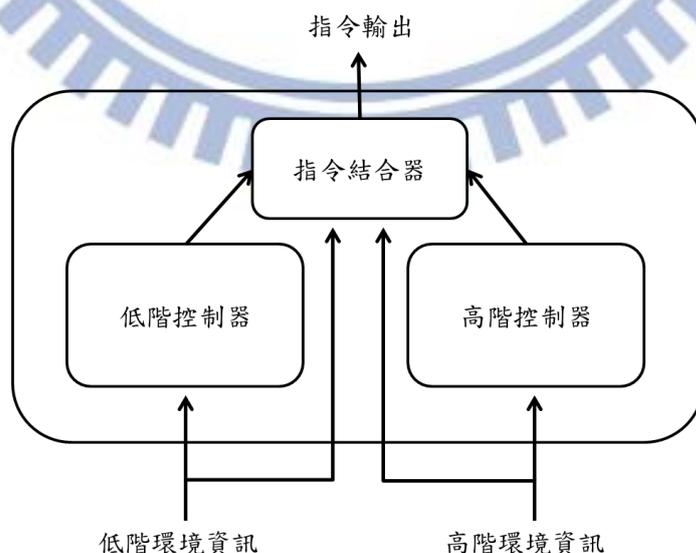


圖 3-3 控制器架構

低階控制器是一種處理低階環境資訊的控制器。低階環境資訊提供的是量測方面的數值訊息而非抽象性的描述。以描述一個彎道為例，低階資訊可能是賽車在彎道中各個距離探測器的值，而高階資訊可能就包含了當前彎道的緩急程度這樣的抽象性描述。低階控制器的優點是對周遭的環境變化比較敏感，因此可以更快速的進行操作上的因應。缺點是比較難作出具有大局觀的操作策略。這使得低階控制器偏向表現出人類玩家在單一彎道中直覺性的環境認知及動作。在 TORCS 的相關研究中，有許多 NPC 是基於低階控制器而設計的。為了克服上述的缺點，常使用的方法是估測出整個賽道環境，再從中取出高階資訊，或是使用演化計算的方法，對 NPC 在小區域內的操作進行調整。

高階控制器處理的是高階環境資訊。高階資訊主要描述彎道的緩急程度或形狀等抽象性資訊。由於高階資訊對周遭環境是一種概括性的描述，因此對於微小的變化難以作到快速的因應。實際上高階控制器的設計方法偏向先決定出一個目標速度或目標位置，之後再計算出要達到該目標狀態所需要的操作。這近似於人類玩家在進入彎道前會先估計適當的入彎速度及入彎位置，是一種具有策略性的駕駛方式。

指令結合器的功能是根據當前的駕駛狀況整合低階控制器和高階控制器的控制指令，並輸出最終的控制指令。如果當前環境較適合低階控制器駕駛，則最終指令就會以低階控制器的輸出為主；反之則以高階控制器的輸出為主。

### 3.3.2 低階控制器

控制器的架構參考了[3]中的設計方法。在[3]中的油門控制模組主要由模糊系統(fuzzy system)構成。[3]的設計目標是建構出一個能夠穩健(robust)行駛，而且可以展現出有競爭力的速度表現。運作的原理是由人工定出的七條模糊規則(fuzzy rule)推測出目標速度，再由一個軟閥(soft threshold)根據速度差輸出油門控制的指令值。

本次研究的設計中仍然保留了[3]使用的七條模糊規則，但對於成員函數(membership function)、模糊規則輸出值以及軟閥進行參數化，並透過基因演算法(genetic algorithm)進行調整。這裡演化的目標是讓油門控制模組輸出的指令值盡量相似於目標玩家會產生的指令值，為此目標函數(objective function)設計成計算在相同輸入下，模仿對象和訓練中的控制器的指令輸出相減的方均根(root mean square)，值越低越好。

[3]使用的七條模糊規則是：

1. IF *Front* is *High* THEN  $V_t$  is  $TS_1$
2. IF *Front* is *Medium* THEN  $V_t$  is  $TS_2$
3. IF *Front* is *Low* and  $Max_{10}$  is *High* THEN  $V_t$  is  $TS_3$
4. IF *Front* is *Low* and  $Max_{10}$  is *Medium* THEN  $V_t$  is  $TS_4$
5. IF *Front* is *Low* and  $Max_{10}$  is *Low* and  $Max_{20}$  is *High* THEN  $V_t$  is  $TS_5$
6. IF *Front* is *Low* and  $Max_{10}$  is *Low* and  $Max_{20}$  is *Medium* THEN  $V_t$  is  $TS_6$
7. IF *Front* is *Low* and  $Max_{10}$  is *Low* and  $Max_{20}$  is *Low* THEN  $V_t$  is  $TS_7$

其中的Front是賽車正前方距離探測器的值，即TORCS的距離探測器Track<sub>0</sub>的值。Max<sub>10</sub>是max(Track<sub>10°</sub>, Track<sub>-10°</sub>)，Max<sub>20</sub>則是max(Track<sub>20°</sub>, Track<sub>-20°</sub>)， $V_t$ 代表目標速度。TS<sub>1</sub>到TS<sub>7</sub>是七條模糊規則的輸出，輸出值是目標速度值。每個成員函數參數化後會有三個可調整參數，分別是low開始下降並且medium開始升高、medium開始下降且low和high皆為0，以及medium降為0且high達到最高這三個距離值。以圖3-4為成員函數參數化的範例，三條鉛錘線代表參數所在的位置。TS<sub>1</sub>到TS<sub>7</sub>的參數化方法就是將7個目標速度都改為可調整的狀態。

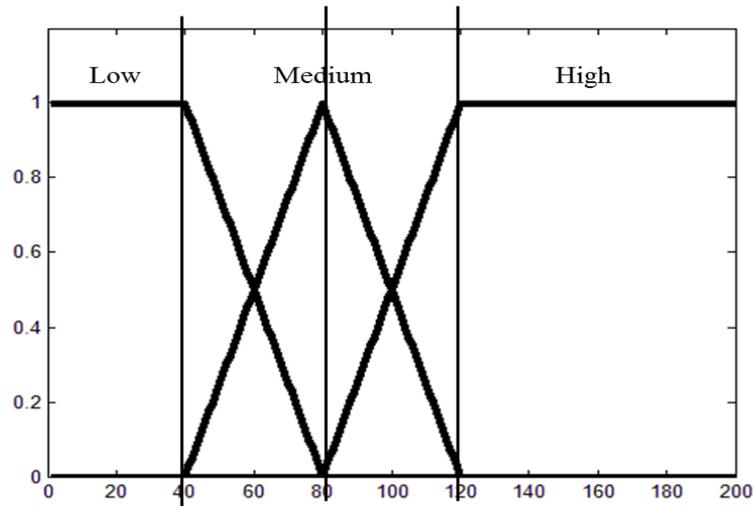


圖 3-4 成員函數參數化的範例

[3]中的用來計算油門輸出值的軟閥是一個 sigmoid function，如方程式(1)所示， $D = \text{當前速度} - \text{目標速度}$ ：

$$\text{油門輸出值} = -1 + \frac{2}{1 + \exp(D)} \quad (1)$$

參數化的部分是將函數定為方程式(2)：

$$\text{油門輸出值} = -1 + \frac{2}{1 + \exp(\alpha D)} \quad (2)$$

$\alpha$  值若越大，訓練出的控制器對速度差反應較劇烈，反之，反應會比較和緩。原本賽車加速和減速的控制都由這個函數負責，即使用的  $\alpha$  值是相同的。但實驗後發現當以模仿玩家為目的時，加速和減速使用不同的  $\alpha$  值會得到低許多的方均根值，因此在這部分決定使用兩個參數各自負責加速和減速的 sigmoid function。

轉向控制模組的設計仍然參考了[3]的作法，並且同樣進行參數化。如方程式(3)所示：

$$\text{轉向輸出值} = S_M + \frac{\text{track}_{M-10} * |S_M - S_{M-10}| - \text{track}_{M+10} * |S_M - S_{M+10}|}{\text{track}_M} \quad (3)$$

$S_M$  中的  $M$  指的是在當前賽車的狀態下，值最大的距離探測器和賽車中線的夾角， $S_M$  就是這個夾角對應的控制指令值， $\text{Track}_M$  是這個距離探測器的值。在本次研究中， $M \leq -50^\circ$  或  $M \geq 50^\circ$  的  $S_M$  各設為 1 和 -1， $M=0$  的  $S_M$  值是 0。意思是當值最

大的距離探測器和中線的夾角過大，即非常需要轉向，方向盤會打到底，而當值最大的距離探測器等於中線，相當於處在直線賽道的正中間，就不進行轉向。

其餘的8個轉向值 $S_{-40}$ 、 $S_{-30}$ 、 $S_{-20}$ 、 $S_{-10}$ 、 $S_{40}$ 、 $S_{30}$ 、 $S_{20}$ 、 $S_{10}$ 都設定為可調整的參數，但正負夾角的對應的轉向值是等大但反向的，例如 $S_{10} = -S_{-10}$ ，所以實際上只有四個參數在調整。調整的方法同樣使用基因演算法，目標仍然是降低控制指令差的方均根。

### 3.3.3 高階控制器

高階控制器的建構參考了[9]中 KNN(k nearest neighbor)的設計方法。[9]的研究目的也是模仿駕駛風格，而且對象也是 TORCS 內建的 NPC。在感測器的部分[9]提出了一種有別於距離探測器(圖 3-5)的方法，稱為前瞻探測器(look ahead sensor)。如圖 3-6 所示，它將前方一定距離內的賽道以十公尺為單位切出多個區段，對每個區段賦予一個值代表彎曲的程度。若區段是直線，值就為零，若是彎道，值會等於曲率半徑。在一定距離內的這些區段的值就構成一筆感測資料，例如圖中的 NPC 當下從感測器得到的輸入是 $[0,60,90,90,60,0,0]$ ，之後 KNN 進行比較時使用的就是找出最相近當前這些區段值的感測資料，然後計算這些感測資料當時對應的速度和位置的平均作為現在的目標狀態。這樣設計感測器的好處是可以很容易的表達出前方路段的形狀或彎道緩急等高階資訊。

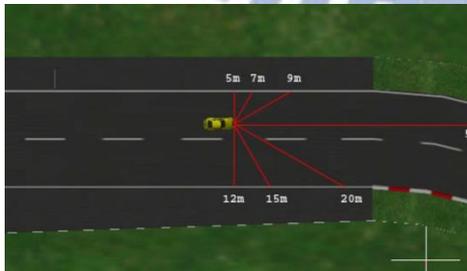


圖 3-5 距離探測器 來源[9]



圖 3-6 前瞻探測器 來源[9]

本次研究中對於[9]提出的探測方法中的區段值給法稍作修改，將原本是彎

道的區段值改為曲率半徑的倒數。以[0,60,90,90,60,0,0]這個例子來說，在這裡就會被改為[0,1/60,1/90,1/90,1/60,0,0]。這樣做的原因是可以讓越趨近於直線的區段值越接近零，而彎曲程度越大的區段值越大於零。在油門控制的實現上和[9]中相同，計算方法如式(4)和式(5)所示， $V_t$  和  $V_c$  分別是估計出的目標速度和賽車當前的速度， $accel$  是輸出的油門指令值， $brake$  是輸出的煞車指令值。

$$accel = \begin{cases} 1 & \text{if } V_t - V_c \geq 20, \\ (V_t - V_c)/20 & \text{if } 20 > V_t - V_c > 0, \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$brake = \begin{cases} 1 & \text{if } V_t - V_c \leq -20, \\ -(V_t - V_c)/20 & \text{if } -20 < V_t - V_c < 0, \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

轉向控制中決定目標位置的方法和油門控制一樣使用了 KNN。計算方式如式(6)所示， $steer$  是輸出的轉向指令值， $\theta$  是賽車和賽道中軸線的夾角， $y_t$ 和 $y_c$ 分別是估計出的目標橫向位置和賽車當前的橫向位置， $S$  代表車輪轉向的最大值。 $\varepsilon$  是用來調控轉向控制的操作風格。 $\varepsilon$  值越大，控制器對環境變化的反應就越劇烈。

$$steer = (\theta + \varepsilon*(y_t - y_c))/S \quad (6)$$

### 3.3.4 指令結合器

指令結合器的功能是将低階控制器和高階控制器的輸出依當前的環境和賽車狀態整合成控制器的總輸出。整合方法的核心概念是當賽車處於比較需要快速反應的狀態(例如前方有障礙物、即將衝出賽道或正在通過彎曲度大的彎道)時，控制器會偏向低階控制器，而在處於比較單純且安全的環境(例如離彎道很遠的長直線)時，會表現的像高階控制器。

判斷狀態的方法上，本研究用了  $Front$ 、 $Max_{10}$ 和 $Max_{20}$  這三個值的平均。值越大就判斷為越單純的環境，反之則判斷為越需要即時反應的環境。值大小的分

界用了 sigmoid function 作為軟閾，低階控制器和高階控制器的輸出值依軟閾值的大小作加權平均。實驗中使用的 sigmoid function 如方程式(7)和圖 3-7 所示，其中  $x = \text{mean}(\text{Front}, \text{Max}_{10}, \text{Max}_{20})$ 。結合成最終輸出指令的方式透過方程式(8)計算完成：

$$w = \frac{1}{1 + \exp(-0.1 * (x - 100))} \quad (7)$$

$$\text{總輸出指令} = w * \text{高階控制器指令} + (1-w) * \text{低階控制器指令} \quad (8)$$

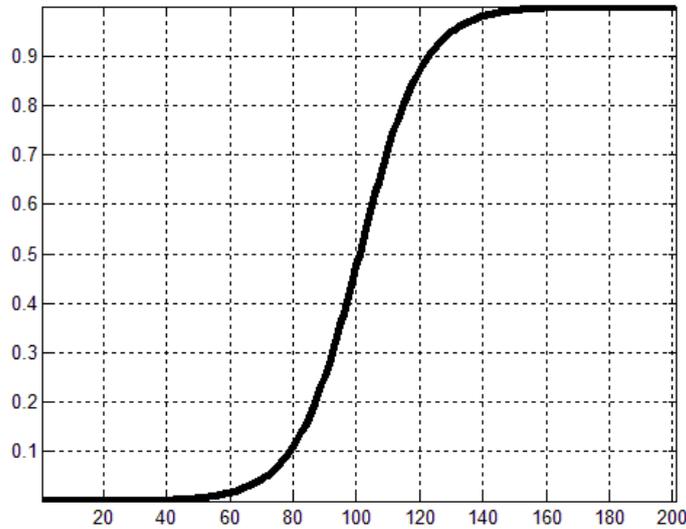


圖 3-7 軟閾

### 3.3.5 額外加速模組

在本次研究的高階控制器使用了 KNN，這在控制器的速度表現上會有潛在的問題，在[9]中也有敘述到。以一個長直線路段來看，一般來說不論是玩家還是 NPC 都會在這個路段中定出一個較高的目標速度，但是由於 KNN 給出的目標速度經過平均而使得估計的目標速度容易過低，造成的影響是訓練出的控制器會在顯然十分安全的路段中進行煞車而影響到速度表現。額外加速模組就是為了解決這個問題而提出的。

額外加速模組的核心概念是當賽車位於長直線路段中，油門就會全開。判斷

長直線路段的方法仍然使用到了  $\text{Front}$ ， $\text{Max}_{10}$ 和 $\text{Max}_{20}$ 的平均。當這個平均值很大時，可以使用一個油門總是全開的控制器在駕駛。當接近彎道時，由於估計的目標速度較準確，因此轉為以原本高階控制器為主的駕駛方式。最後很接近彎道時就由低階控制器負責。在額外加速模組和高階控制器之間的切換仍然使用 sigmoid function 作為軟閥，但這裡會設定為比較接近硬閥以避免高階控制器的影響力過小。

### 3.4 風格相似度判斷器

在本研究中，風格相似度判斷器的主要功用有三個。第一個是幫助建構出的 NPC 能夠在新賽道中進行自我調整。由於實際上沒有模仿對象在新賽道中的駕駛資料，因此需要有一個機制可以在新賽道上對於建構出的 NPC 和模仿對象的風格相似度進行判斷。第二個功用是對於本研究中使用的模仿方法的效果進行評估。第三個是對於玩家的駕駛風格進行分析。目的在觀察玩家是否具備獨特的自我駕駛風格。

相似度判斷方法的基本概念是計算出賽車在同樣的狀態下，模仿對象和控制器輸出指令值的差異。為此對於每個被模仿的對象都會建構一個風格判斷器。判斷器在實作上使用了 KNN 這個方法，KNN 中的資料是模仿對象過去駕駛時，各個時間點的環境狀態、賽車狀態以及當時輸出的控制指令值。判斷器運作時會從中找出最相近當前環境狀態和賽車狀態的資料，並將這些資料中對應到的指令輸出值的平均作為推測的模仿對象輸出指令，之後與控制器的輸出指令相減以計算差異。圖 3-8 是一位玩家在一個賽道中輸出的轉向控制指令值的紀錄，圖 3-9 則是透過 KNN，從該位玩家在別的賽道中的駕駛資料推測應該輸出的轉向控制指令值。

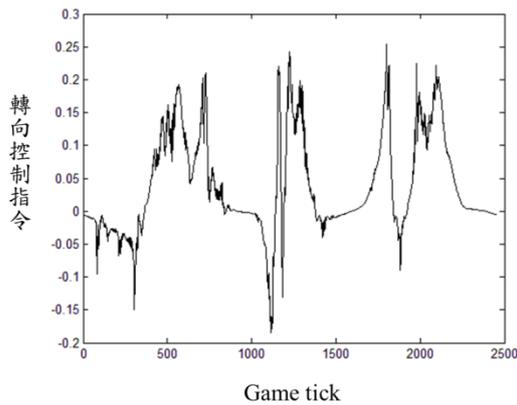


圖 3-8 玩家的輸出指令

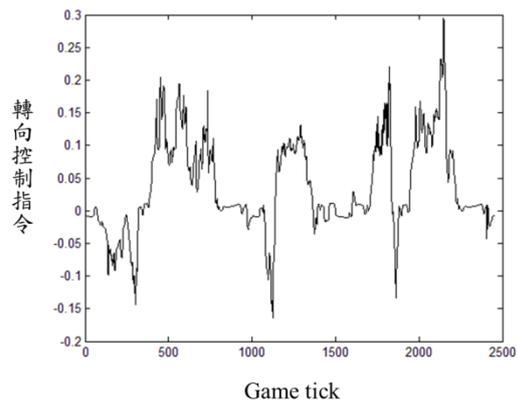


圖 3-9 判斷器推測的輸出指令

當控制器跑完賽道後，相似度判斷器會對每個 game tick 記錄下來的資料進行上述方法的計算，最後輸出這些指令值差的方均根。由於指令有油門控制和轉向控制這二種，因此會先將這二種指令作正規化。正規化的方法是將二種指令的平均值調成 0，標準差調成 1。

環境和賽車的狀態主要包含了低階控制器和高階控制器用到的低階與高階資訊。總共是 19 個距離探测器和 20 個區段值，20 個區段值同 3.3.3 節中所述，是將前方 200 公尺的路段以 10 公尺為單位劃分為 20 個區段，並記錄下 20 個區段中平均曲率半徑的倒數，若區段是直線則值設為 0。此外，再另外加上速度與第二章中提到的 trackPos 偵測器的值共 41 個輸入值來描述環境和賽車狀態。在這裡使用了主成分分析(principal component analysis)去除訓練資料中這 41 種資訊中的冗餘，並保留了 10 個主成分作為 KNN 的輸入。圖 3-10 是判斷器在每個 game tick 記錄下的資料中實行的動作，其中 KNN 及控制器輸出的指令值都已經作過正規化。

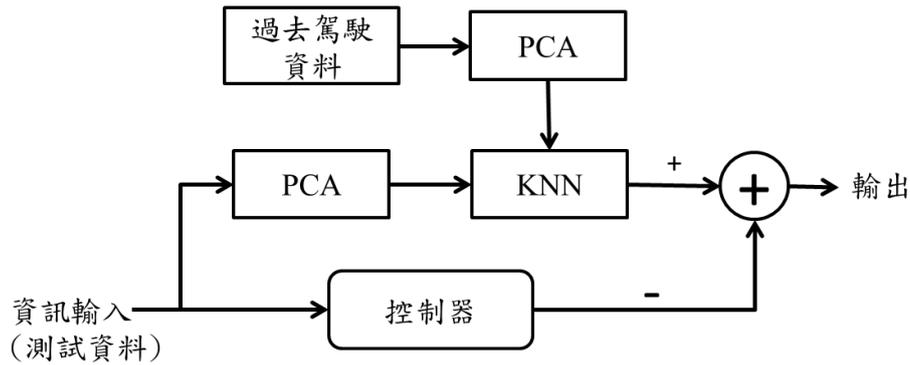


圖 3-10 判斷器運作流程

在本研究中使用 20NN，並且將 20NN 輸出的結果與前 9 個 game tick 的 20NN 結果作平均後才成為該 game tick 的 20NN 輸出。作平均的原因是實際上玩家在駕駛時基本上無法在極短的時間內大幅度的變更操控指令，而 KNN 的輸出根據訓練資料的狀況和 K 值有可能會在短時間內出現小幅度的震盪，這點會無意義的增加了方均根值，因此才將短時間內預估的模仿對象操控指令值作平滑化。

### 3.5 新賽道適應機制

訓練出的控制器在新接觸的賽道中進行調整及適應的目標有兩個。第一個是在該賽道的速度表現，這點由控制器跑完一圈所花費的時間作為衡量基準。第二個是與模仿對象的駕駛風格相似度，這部分由另外設計出的風格相似度判斷器完成。為了同時對於這二個目標進行最佳化所使用的方法是 NSGA-II[14]這個多目標演化演算法(multi-objective evolutionary algorithm)。圖 3-11 是一個演化的例子。圖中橫軸是控制器跑完一圈賽道的秒數，縱軸是風格相似度判斷器給出的值，值越低越好。

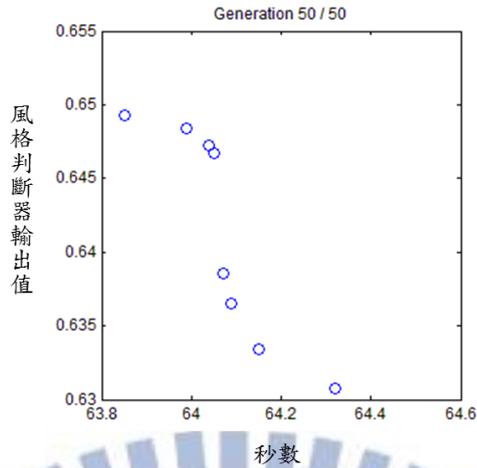


圖 3-11 演化結果範例

控制器調整的方式是將從 3.3 節中所述方法訓練出的參數作隨機增減。增減的幅度不大於原本參數值的 20%。在適應機制中設定一個世代的個體數是 50，經過 50 個世代演化。演化時的第一世代是將 3.3 節中建構出的控制器裡的參數進行隨機增減而產生出的控制器。

多目標最佳化方法給出的解並不唯一。以 3-11 圖來看，有些控制器展現出比較好的速度表現，但在風格相似度上卻不如另一些速度表現較差，卻有著更相似模仿對象駕駛風格的優點。在尋找解的過程中，比較容易被剔除的是速度表現差而且相似度也差的解。在最後留下來的解中，則依情況由人工盡量挑選接近左下角的解作為適應賽道的結果。

## 第四章 實驗結果—非人類玩家駕駛風格模仿

### 4.1 控制器訓練結果

圖 4-1 是低階控制器油門模組的訓練結果。在這裡要模仿的對象是 Berniw。在訓練賽道中每個 game tick 的指令輸出，油門控制模組演化時所用的訓練資料只佔了約總資料量的 20%。這是由於 Berniw 在訓練賽道中僅有約 10% 的時間進行剎車，為了避免演化出的控制器為了降低方均根值而總是輸出加速的指令，因此只另外從加速的指令中取樣出和減速指令大約相同數量的資料。訓練完成後，使用 Berniw 所有的駕駛資料進行測試。在這裡將油門和煞車的指令結合成一種指令，值介於-1 到+1 之間。值大於 0 時代表踩油門，小於 0 是踩煞車。圖 4-1(a)~(d) 是模糊系統的成員函數和輸出演化的結果。4-1(e) 是軟閥的演化結果，從中可以看出控制器在加速和減速時的操作風格有明顯的差異。最後計算出誤差的方均根值是 0.5824。

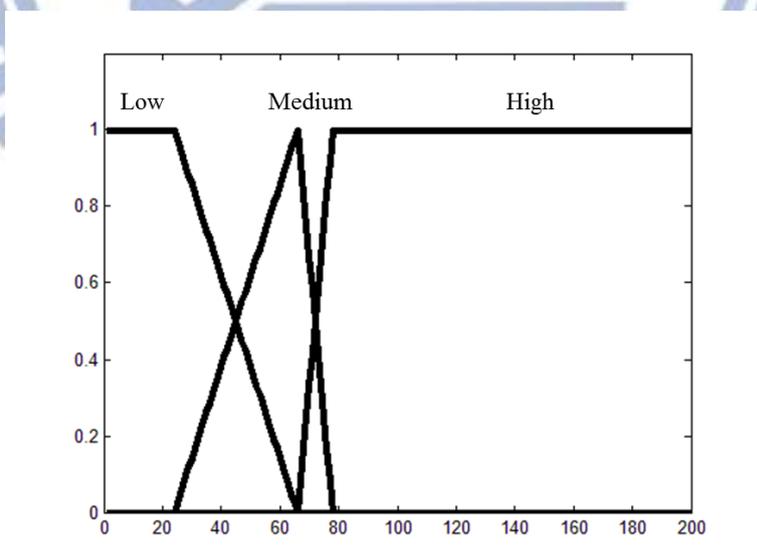


圖 4-1(a) Front 演化結果

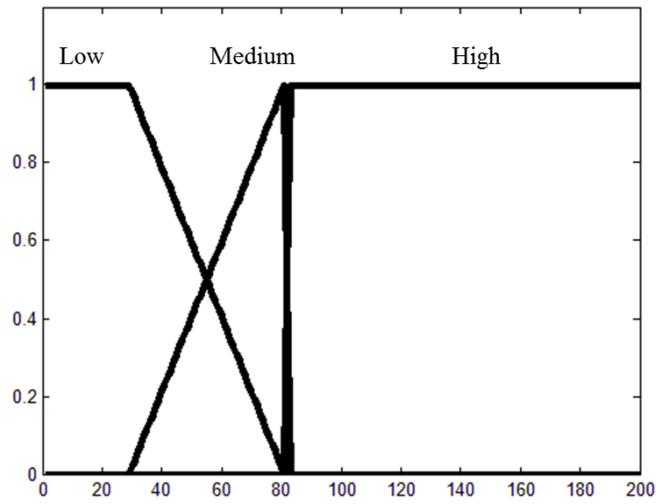


圖 4-1(b) Max10 演化結果

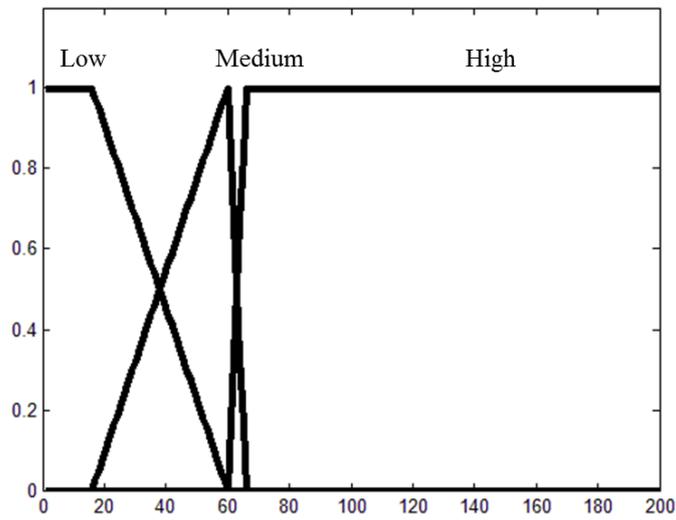


圖 4-1(c) Max20 演化結果

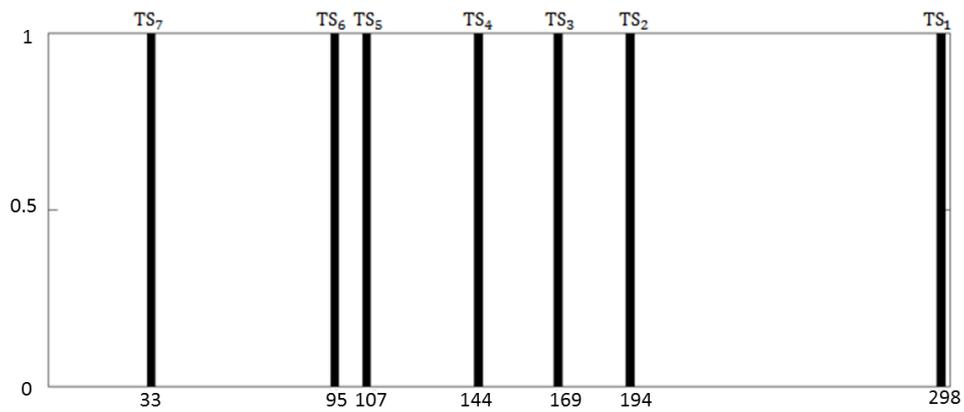


圖 4-1(d) Fuzzy System Output 演化結果

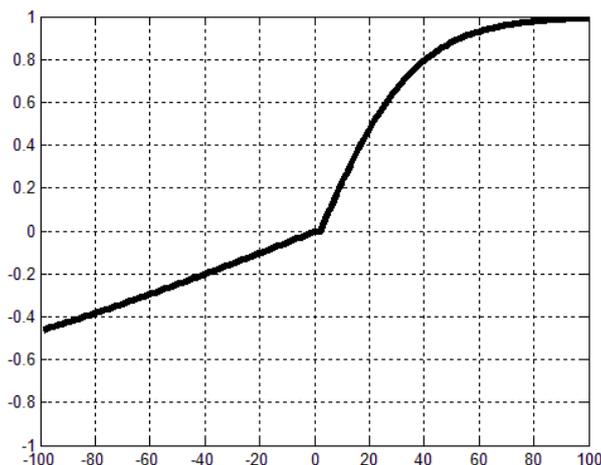


圖 4-1(e) 軟閥演化結果

圖 4-1(a)~(e) 低階控制器油門模組的訓練結果

低階控制器轉向控制模組的訓練資料是 Berniw 所有的駕駛資料。轉向控制指令的值介於 -1 到 +1 之間，訓練後得到的結果是  $S_{40} = 0.4075$ 、 $S_{30} = 0.2751$ 、 $S_{20} = 0.1845$ 、 $S_{10} = 0.0993$ 。誤差的方均根值是 0.0784。

圖 4-2 和圖 4-3 是訓練後控制器和 Berniw 在油門控制和轉向控制的一部分比較結果。圖中的實線是 Berniw 的控制指令，虛線是訓練後控制器的控制指令。從圖可以看出油門控制指令變化較劇烈，因此方均根值也較大。轉向控制指令則是震盪幅度較小，誤差的方均根值也較低。

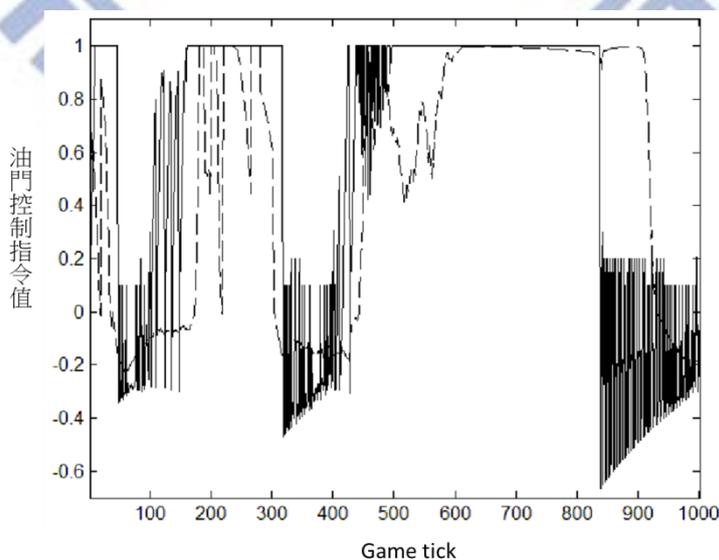


圖 4-2 油門控制指令比較：實線是 Berniw，虛線是訓練後判斷器

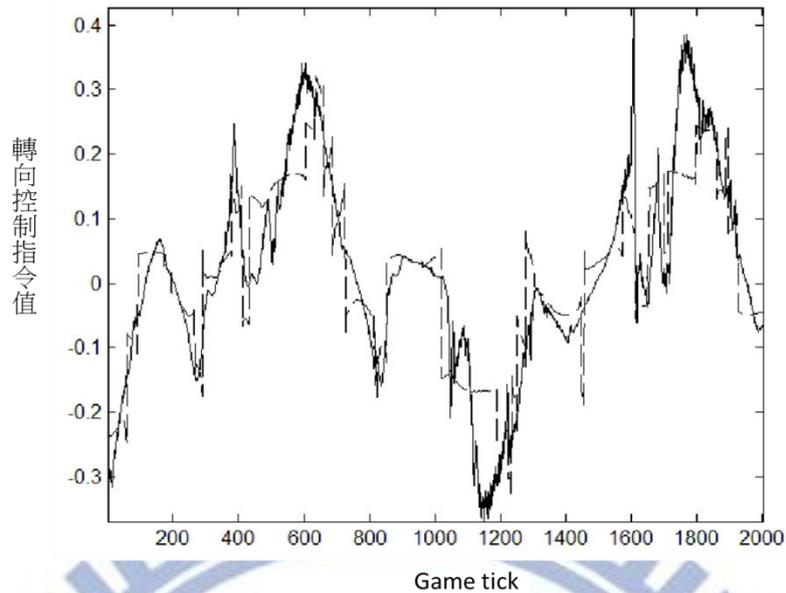


圖 4-3 轉向控制指令比較：實線是 Berniw，虛線是訓練後判斷器

透過指令結合器將低階和高階控制器整合後，可以發現到除了直觀上比較偏向一般人類玩家的思考模式外，對於實際的模仿效果也有重要的影響。圖 4-4 和圖 4-5 是一小段模仿結果的展示與比較。圖 4-4 是只使用低階控制器的結果，其中實線軌跡是 Berniw 跑出的，而虛線是訓練後的控制器跑出的結果。圖 4-5 是指令結合器依上述方法結合低階控制器和高階控制器後的結果。而單純使用高階控制器會在彎道處出車禍。從比較圖中可以明顯看出低階控制器並沒有辦法對於前方的彎道進行位置上的規劃，使得控制器在軌跡上跑出了與 Berniw 差異很大的結果。在結合高階控制器後，軌跡明顯相似許多。控制器在彎道前出現了和 Berniw 相仿的規劃，也進一步的讓低階控制器在進入彎道後可以展現出相似的軌跡。



圖 4-4 低階控制器的結果

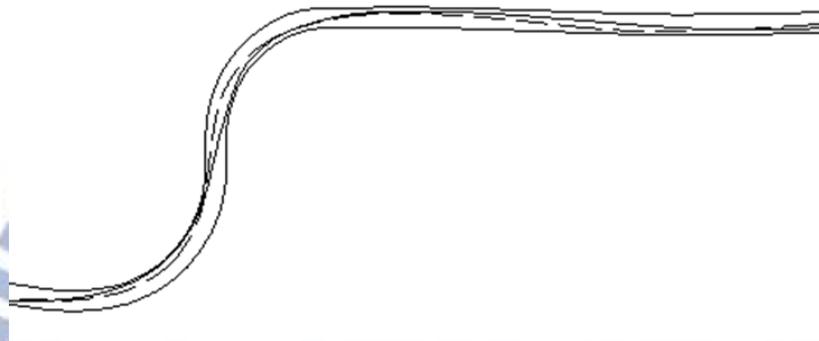


圖 4-5 低階+高階控制器的結果

## 4.2 風格相似度判斷器效果評估

相似度判斷器在多目標演化演算法中是作為目標函數使用，理想上對於相似於模仿玩家的駕駛資料必須給出低的方均根值，反之則要有較高的值。這一節就是對於判斷器是否具備這樣的能力進行評估。

為了評估判斷器的能力，在這裡使用到了三個在駕駛行為上存在差異的 NPC，分別是 Berniw、Olethros 以及[12]中建構的基底控制器，在這裡命名為 BtLin。之後從實驗中觀察是否能夠透過判斷器給出的數值，從駕駛資料上區分出這三個 NPC。

Berniw 和 Olethros 都是 TORCS 內建的 NPC，Berniw 在轉向和油門控制方面有著細膩的事前計算，使得 Berniw 在這三個控制器中的速度表現和安全性都是最好的。Olethros 使用的是較簡單的預測機制，在轉向的部分類似 Berniw，都

會在彎道中嘗試緊貼跑道內側，但在油門控制方面，在部分路段會出現超速的現象，而在某些路段可能又出現過於保守的低速跑法。BtLin 是一個比較重視安全性而忽略速度表現的 NPC。它的駕駛原則是盡量讓賽車處在賽道中線以避免衝出賽道的車禍，因此在轉向和油門的控制上，都和 Berniw 及 Olethros 這種以競速為目標的 NPC 有著比較大的區別。

實驗使用了 3.1 中所述的 9 個賽道，每次實驗會將三位 NPC 在其中 8 個賽道中的駕駛資料建成各自的風格相似度判斷器，並將剩餘的一個賽道作為測試賽道，測試時是讓風格判斷器判斷三位 NPC 在測試賽道上的駕駛資料。共作 9 次實驗。

表 4-1 和表 4-2 是其中 2 次的實驗結果。表格中的縱向是判斷器，橫向是玩家。以表 4-1 中的結果為例，以第一列來看，Berniw 風格相似度判斷器計算出在該賽道中 Berniw 駕駛資料的方均根值是 0.354，Olethros 是 0.494，而 BtLin 是 0.789。這代表 Berniw 風格判斷器成功辨識出在該賽道中 Berniw 的駕駛行為最像 Berniw，因為方均根值是最底的，而 BtLin 是最不相似的。表 4-2 中的第一列則是判斷錯誤的例子。Berniw 判斷器對於 Olethros 的駕駛資料給出了比 Berniw 的 0.526 更低的方均根值，0.491。

以橫向來進行數值上的比較，是本次研究中目標函數的運作方式，即觀察判斷器是否能在多個駕駛結果中，給予最相似模仿對象風格的駕駛資料最低的數值。若以縱向來看，則是一個分類問題，即給予一筆駕駛資料，然後辨識出是哪一位玩家的駕駛記錄。

表 4-1 Wheel 2 風格相似度判斷結果

	Berniw	Olethros	BtLin
Judger <sub>Berniw</sub>	0.354	0.494	0.789
Judger <sub>Olethros</sub>	0.594	0.323	0.572
Judger <sub>BtLin</sub>	8.769	1.481	0.181

表 4-2 E-Road 風格相似度判斷結果

	Berniw	Olethros	BtLin
Judger <sub>Berniw</sub>	0.526	0.491	0.791
Judger <sub>Olethros</sub>	0.619	0.409	0.591
Judger <sub>BtLin</sub>	8.605	3.057	0.180

總和所有的實驗結果，每個賽道有 3 個判斷器運作，作 9 次實驗共 27 次運作。就橫向的比較來說，其中判斷錯誤的次數是 4 次，即正確率是  $23/27 = 85.19\%$ 。若只以 Berniw 和 BtLin 這二種風格差異很大的 NPC 進行判斷，則 27 次的判斷結果全部正確，即正確率 100%。以縱向來說，判斷錯誤的次數是 2 次，正確率 92.59%，若只以 Berniw 和 BtLin 這二種 NPC 作比較，則出現一次錯誤，正確率  $17/18 = 94.44\%$ 。表 4-3 和表 4-4 分別是橫向和縱向判斷結果的 confusion matrix。

表 4-3 風格相似度判斷結果 confusion matrix(橫向)

	Berniw	Olethros	BtLin
Judger <sub>Berniw</sub>	8	1	0
Judger <sub>Olethros</sub>	1	6	2
Judger <sub>BtLin</sub>	0	0	9

表 4-4 風格相似度判斷結果 confusion matrix(縱向)

	Berniw	Olethros	BtLin
Judger <sub>Berniw</sub>	9	1	0
Judger <sub>Olethros</sub>	0	8	1
Judger <sub>BtLin</sub>	0	0	8

### 4.3 模仿實驗結果

實驗使用的測試賽道是 Wheel 2、Wheel 1、E-Road 以及 E-Track 6，其餘的五個賽道作為訓練賽道用以訓練控制器和風格判斷器。在適應機制運作完畢後，會從中人工挑選一個速度表現和風格相似度都有良好表現的控制器與 Berniw 進

行比較。

下面會針對四個賽道的實驗結果進行比較和探討。主要的討論項目是適應前後的速度和相似度表現、和目標玩家的差異，及其它觀察到的現象。

## Wheel2 賽道實驗結果

Wheel 2 是一個長賽道，而且其中包含了許多不同特性的彎道，屬於難度偏高的賽道。速度表現的比較如表 4-5 所示。Berniw 花費了 120.13 秒，適應前(未經多目標演化演算法調整)控制器 148.49 秒，相較於 Berniw 慢了 23.61%。適應後(經多目標演化演算法調整)控制器是 136.27 秒，相較於 Berniw 慢了 13.44%。

表 4-5 Wheel 2 賽道適應結果

玩家	Berniw	適應前控制器	適應後控制器
時間(s)	120.13	148.49(-23.61%)	136.27(-13.44%)
風格判斷器輸出值	0.5557	0.9155	0.7535

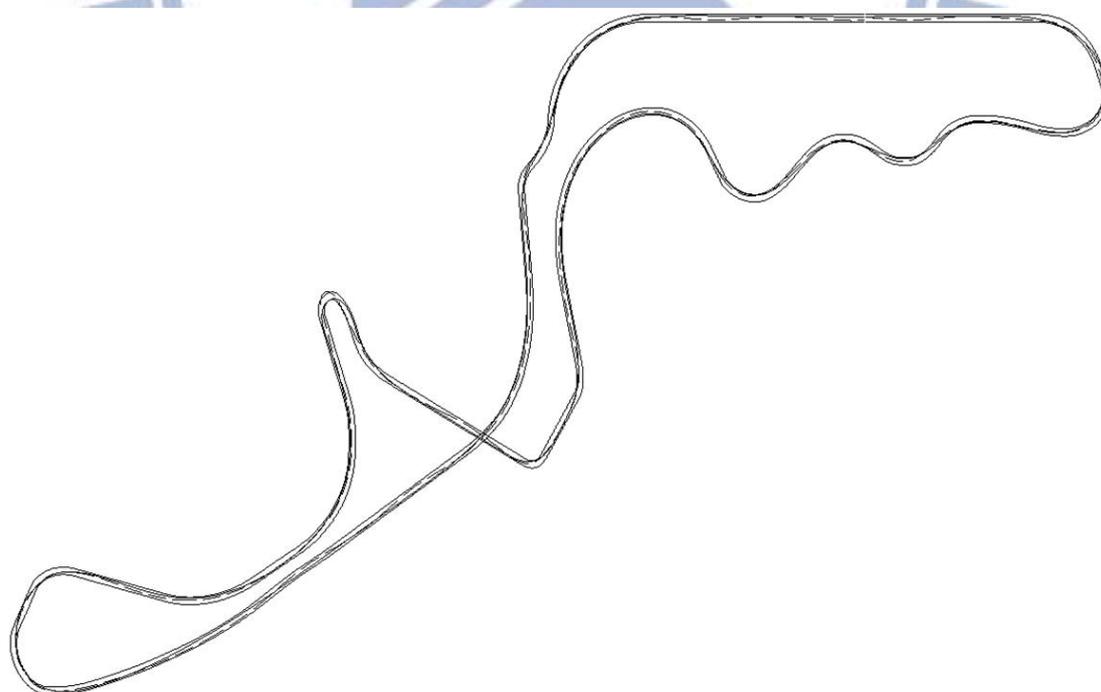


圖 4-6 Wheel 2 賽道軌跡比較

圖 4-6 是駕駛軌跡的比較。實線是 Berniw 的駕駛軌跡，而虛線是適應後控制器的軌跡。由結果圖可以看出在大部分的彎道中兩個控制器的軌跡十分相似，差距較大的主要在於圖中最尖銳的彎道處，如圖 4-7 所示：



圖 4-7 Wheel 2 賽道局部軌跡比較

該彎道是一個彎曲程度很大的急轉彎，而且彎道前有很長的路段接近直線，一方面是這樣的急轉彎在訓練資料中比較稀少，在高階控制器中 KNN 的方法架構下占的比重就下降，使得高階控制器可能估出過高的目標速度；另一方面是彎道前的長直線路段也容易讓低階控制器持續的輸出加速指令，最終導致的結果是在該彎道中幾乎衝出賽道。

### Wheel 1 賽道實驗結果

Wheel 1 賽道在接近終點時有連續幾個大型彎道，這部分也是屬於駕駛難度偏高的路段。速度表現的比較如表 4-6 所示。相較於 Berniw，控制器在適應前與適應後分別慢了 13.97% 和 4.75%。後者的速度表現已十分優異。

表 4-6 Wheel 1 賽道適應結果

玩家	Berniw	適應前控制器	適應後控制器
時間(s)	94.48	107.68(-13.97%)	98.97(-4.75%)
風格判斷器輸出值	0.5000	0.7184	0.5343

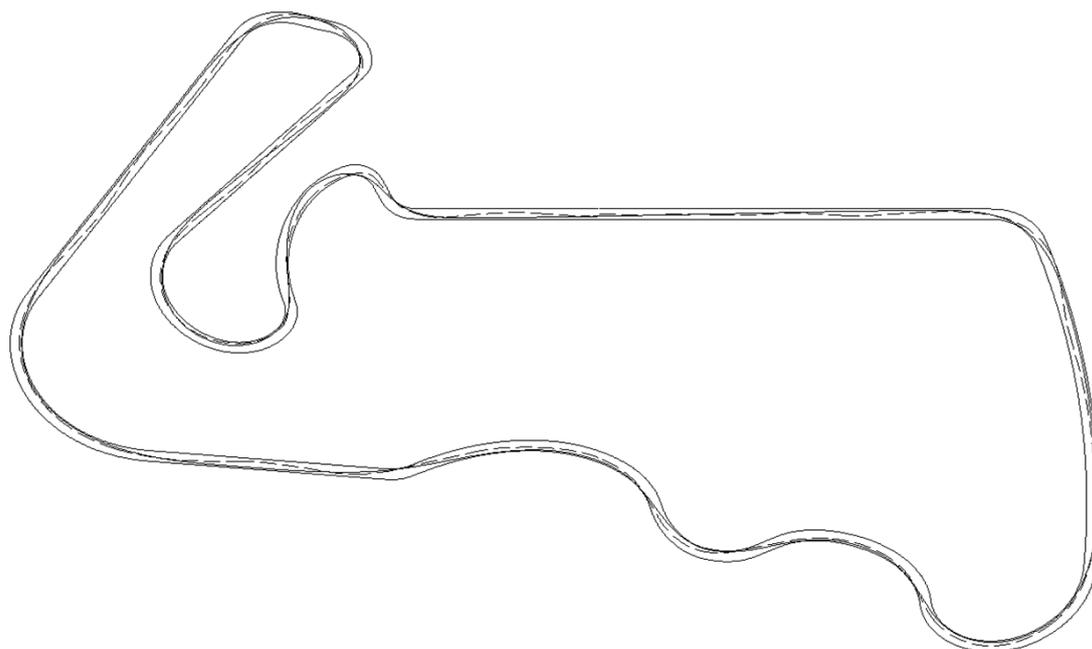


圖 4-8 Wheel 1 賽道軌跡比較

軌跡比較部分如圖 4-8 所示，其中比較有趣的部分是在終點前的連續彎道中，  
如圖 4-9：



圖 4-9 Wheel 1 賽道局部軌跡比較

適應後控制器在駕駛時緊貼內側，但 Berniw 卻選擇了向外側移動。這應該是由於 Berniw 已對下一個彎道的入彎點進行了規劃，因此才在彎道中向賽道外側移動。但控制器在該彎道中主要由低階控制器負責，由於 Berniw 在大部分的彎道中都是位於賽道內側，因此低階控制器只會預測出應該保持在內側賽道而沒有辦法作到如高階控制器般對於接下來的路段進行規劃。

## E-Road 賽道實驗結果

E-Road 賽道幾乎都是由連續的彎道構成。在這個賽道中，適應前控制器發生了車禍，因此在表 4-7 中可以看到速度表現比 Berniw 慢了 58.56%。但適應後可以作到只慢了 8.2%。軌跡比較的部分如圖 4-10 所示。

表 4-7 E-Road 賽道適應結果

玩家	Berniw	適應前控制器	適應後控制器
時間(s)	73.98	117.30(-58.56%)	80.05(-8.20%)
風格判斷器輸出值	0.6467	5.9683	0.6078

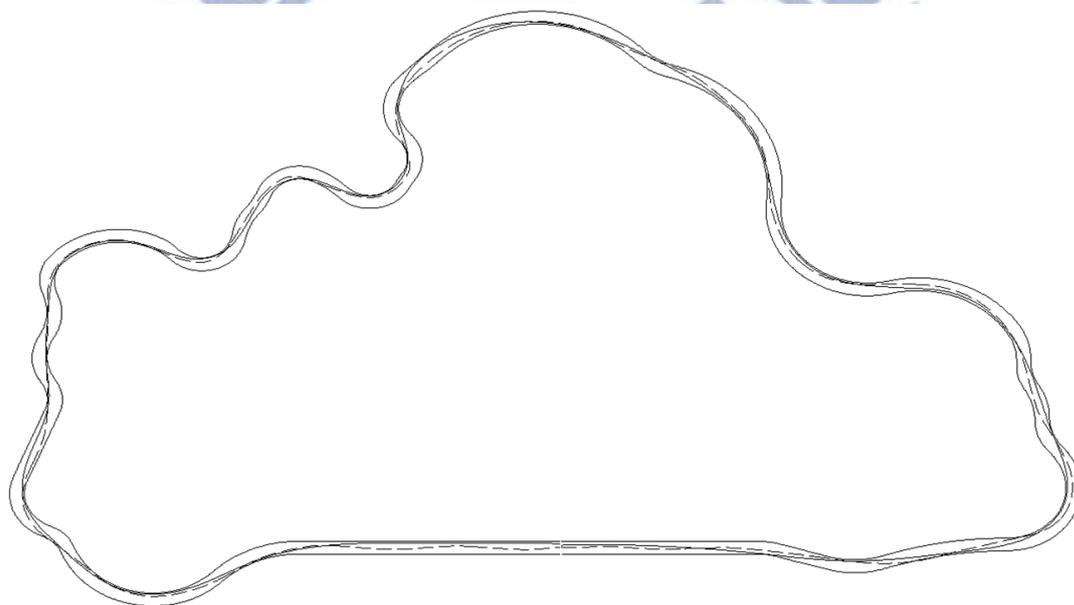


圖 4-10 E-Road 賽道軌跡比較

## E-Track 6 賽道實驗結果

E-Track 6 是一個比較容易駕駛的賽道，主要原因是其中的彎道比較和緩。唯一比較困難的彎道在圖 4-12 左下角處，如圖 4-11 所示：



圖 4-11 E-Track 6 賽道局部軌跡比較

這個情形基本上和控制器在 Wheel 2 出現的問題是一樣的，控制器幾乎是衝出賽道。速度表現的部分，從表 4-8 中可以發現到適應前後的改善幅度相對較小，但表現的並不差，適應前慢了 12.48%，而適應後只慢了 8.1%。

表 4-8 E-Track 6 賽道適應結果

玩家	Berniw	適應前控制器	適應後控制器
時間(s)	93.82	105.53(-12.48%)	101.42(-8.10%)
風格判斷器輸出值	0.3747	0.8156	0.5668

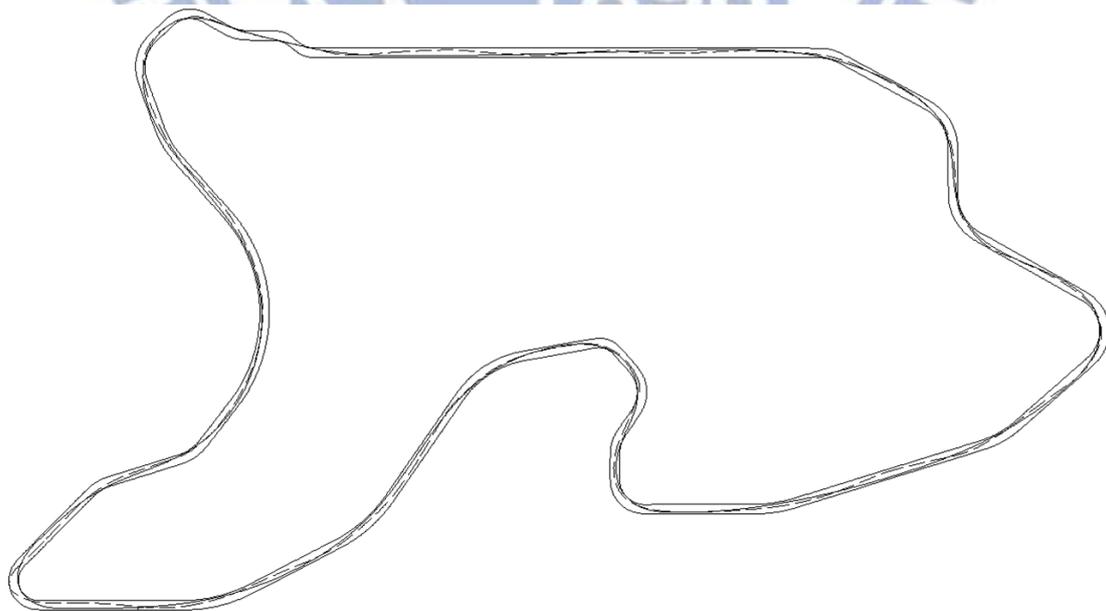


圖 4-12 E-Track 6 賽道軌跡比較

以四個測試賽道的實驗結果來看，可以發現到適應後的 NPC 在接近入彎點處幾乎都表現出和 Berniw 十分相似的軌跡。軌跡相差較多的路段主要有兩個特性：第一個是長直線路段。推測的原因是由於直線屬於安全路段，Berniw 可能允許賽車在路段中處在任何位置，因此無法準確預測出軌跡。第二個是連續的彎道。例如 Wheel 1 赛道中就出現了適應後的 NPC 無法表現出 Berniw 的事前規劃能力。主因是由於目前的指令結合器基本上是透過前方距離探測器的值來決定輸出，在連續的彎道中指令結合器會傾向輸出低階控制器的指令，使得高階控制器的規劃能力沒有辦法被表現出來。



## 第五章 研究方法—人類玩家駕駛風格模仿

### 5.1 目標玩家與研究環境

模仿的人類玩家有八位：KT、YR、HA、SD、HB、JG、ZS、TS。八位玩家都是介於 24 到 26 歲的男性。每位人類玩家在每一個實驗用賽道中連續駕駛五圈，並進行二次。第一次的五圈是為了讓玩家熟悉賽道，避免因過多的車禍而影響到資料品質。只有第二次的五圈駕駛記錄才會作為研究用資料。由於完成所有賽道的駕駛時間過長，容易造成疲勞，因此玩家不一定會一次跑完所有的實驗用賽道，大部分玩家會需要多天的時間分次完成。玩家駕駛時使用實體類比方向盤作轉向控制，油門和煞車透過類比搖桿進行控制，圖 5-1 是實驗時使用的方向盤。排檔設置為自動排檔。



圖 5-1 方向盤

使用的 TORCS 版本是 1.3.4，駕駛難度設定成新手級別。在新手級別中，賽車並不會因為毀損情況嚴重而被迫中止比賽。人類玩家駕駛的賽道有 5 個：CG Speedway number 1、CG track 2、Aalborg、Alpine 1 以及 E-Track 4。收集的駕駛資料同樣會進行 3.1 節中所述的三個前處理。

## 5.2 方法概論

第三章使用的非人類玩家駕駛風格模仿的方法沒有辦法直接套用。根據實驗，建構出的控制器有以下問題：

1. 訓練低階控制器時，油門控制部分的方均根值過大。
2. 組合上高階控制器後，控制器在接近彎道處頻繁的發生車禍。
3. 適應機制中，控制器無法順利的持續朝目標演化，可能的原因是車禍過於頻繁，使得理想的特徵沒有辦法被保留。

目前推測的可能原因有二點：

1. 人類玩家的駕駛資料中車禍的頻率過高：

雖然賽車衝出車道或是倒車這種明顯的異常狀態可以簡單的從資料中濾掉，但人類玩家往往在車禍發生前就會先預測到，並且採取應急措施。應急措施大部份伴隨著短時間內的劇烈操作，而這樣的操作幾乎沒有辦法反應出個人的駕駛風格，實際上造成的影響是控制器的訓練資料品質受到影響。以本篇研究使用的低階控制器來說，由於結構上是使用模糊系統，因此更難作到短時間內劇烈的操作改動。此外，在車禍發生前的超速狀態也會被高階控制器學習到，因此在接近彎道時控制器可能只是嘗試重現出會發生車禍的狀態。

2. 人類玩家個人駕駛風格一致性不高：

由於一個人類玩家在相同的環境中每次駕駛的方式都有一定的差異，這會直接影響到低階控制器的訓練，並且會使得高階控制器在一段賽道中頻繁的更換目標狀態，造成賽車容易打滑。原本高階控制器應該要具備的入彎前規劃能力也因此不容易表現出來。

第三章中使用的方法目的是建構出一個已經具備模仿對像特色的控制器，適應機制只是作為特定賽道上兼顧速度和風格的微調方法。但在以人類玩家為模仿對像時，為了處理上面提到的問題，將原本方法進行更改。

首先目標不再是嘗試直接建構一個具備模仿對象風格且適用未知賽道的控制器。現在的目標是針對各個賽道去建構個別適用的控制器。每當遇到一個陌生賽道時，都會由一個共通的基本控制器演化，演化的目標是速度表現以及重現出目標玩家的風格。而這個基本控制器至少有穩健的駕駛能力以及可以調整這二個特性。穩健駕駛能力是為了在實務上產生出一個具備足夠競爭力的 NPC 與其它玩家進行遊戲，並且讓之後的演化過程可以在較短的時間和空間成本下順利進行。

圖 5-2 是模仿方法的流程。

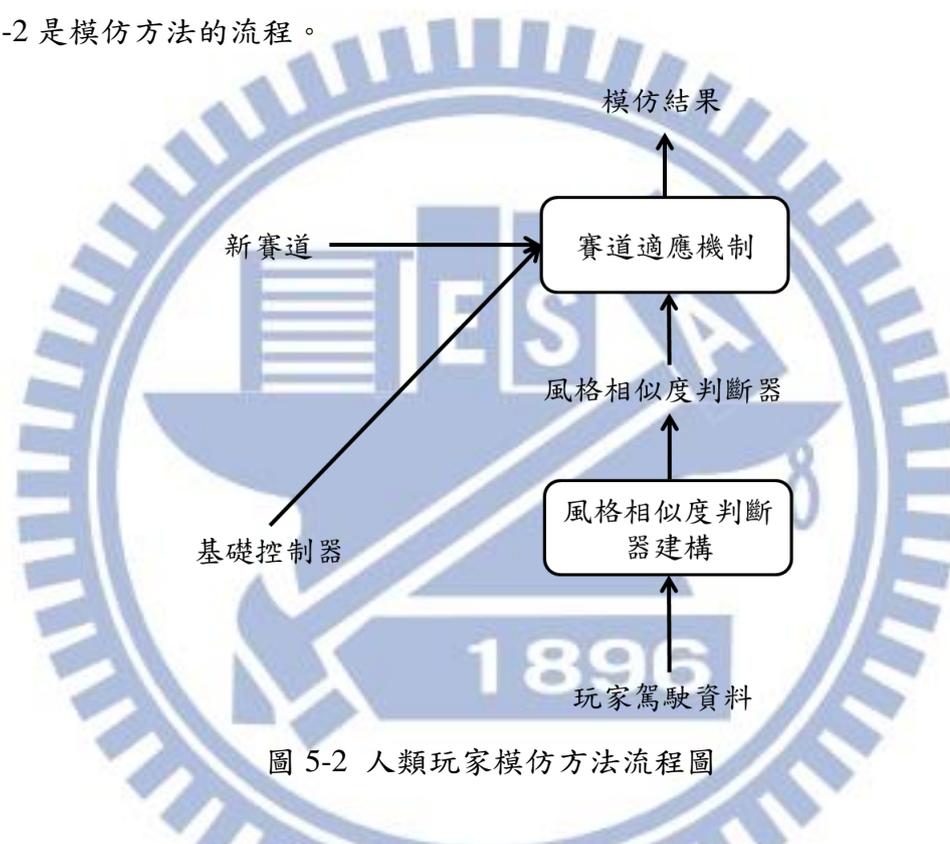


圖 5-2 人類玩家模仿方法流程圖

演化的部分跟第三章中的適應機制是相同的。針對每個人類玩家建出各自的風格判斷器，之後作為目標函數中的風格相似度判斷依據。速度表現方面仍然是以完成一圈賽道的時間作為判斷基準。

### 5.3 控制器架構

控制器的結構和第三章中所提的是十分相似的，如圖 5-3 所示。差別在於將高階控制器改為簡單的中線導向控制器。中線導向控制器只輸出轉向控制指令，

控制的目標是讓賽車位於賽道的中線上，可以視為一種簡化版的高階控制器。

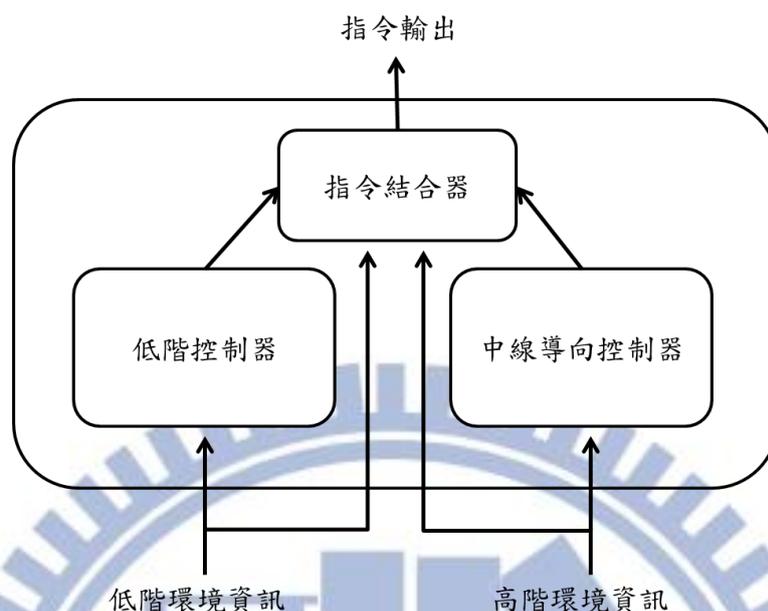


圖 5-3 控制器架構

這樣設計的原因是為了避免圖 4-4 中所示，單純只有低階控制器會發生的問題。在指令結合器方面，首先油門控制指令全由低階控制器負責。而轉向控制的部分，仍然是整合兩種控制器的輸出，但是將式(7)更改為式(9)：

$$w = \frac{1}{1 + \exp(-0.1 * (x - 150))} \quad (9)$$

用意是希望只有在賽車處於較長的安全路段時，才讓中線導向控制器發揮影響力。在接近彎道時的入彎速度和位置則期望讓低階控制器可以在演化時自行學到一些目標玩家的風格。

## 5.4 人類玩家駕駛風格分析

非人類玩家大部分從內部的演算法就能知道駕駛策略和行為方面的差異，但以人類玩家為實驗對象時，並不能確定玩家們是否確實存在著自我風格。因此對於這次研究的八位玩家的駕駛資料進行了分析，目標是觀察玩家是否具有著自我

駕駛風格，而且與其它玩家的風格存在著差異。

分析方法和 4.3 節中的風格相似度判斷器評估法是相同的。每次實驗會挑一個賽道作為測試環境，並將其餘四個作為訓練賽道，共作五次實驗。

## 5.5 模仿效果評估方法

實驗時會從 5 個賽道中取 1 個作為測試用賽道，剩餘的 4 個作為訓練用賽道，建構 8 位玩家個別的風格相似度判斷器。在 5 個賽道上模仿 8 個玩家，共有 40 個模仿結果。

由於人類玩家即使在單一賽道上駕駛，每一次產生的軌跡和速度都有一定差異，因此在這裡使用的風格相似度評估偏向數學上的分析。評估方法的步驟如下：

1. 選定一個測試賽道，讓在其餘 4 個賽道中訓練出的 8 個控制器在這個測試賽道中各駕駛一次。
2. 選定一個人類玩家，使用該玩家在測試賽道上的駕駛資料建構出風格相似度判斷器。
3. 使用風格相似度判斷器判斷步驟 1 中 8 個控制器在測試賽道上的駕駛資料，會得到 8 個判斷結果。
4. 8 個控制器中有一個是以步驟 2 中選定的人類玩家為目標訓練而成的，因此理想狀態下該控制器的判斷結果的值會是 8 個中最低的。

這個評估方法的概念是希望觀察出在穩健駕駛的前提下，控制器是否確實能夠演化出目標玩家的駕駛特徵。由於在這個評估方法中，訓練控制器時並沒有用到人類玩家在測試賽道上的駕駛資料，而建構實驗評估用風格相似度判斷器時也沒有用到訓練賽道上的資料，因此如果從評估的結果觀察到步驟 4 中理想狀態下的結果，那麼可以知道有兩件事是成立的：

1. 基礎控制器在演化後至少表現出目標玩家部分的駕駛特徵，而且與其它人類玩家的駕駛特徵存在著較大的差異。
2. 人類玩家在測試賽道中表現出的駕駛特徵和在其它 4 個賽道中表現出的特徵相似。也就是即使單從軌跡和速度上很難直觀的看出共同的特徵，但實際上仍然存在著風格上的一致性。

實務上風格一致性的高低、模仿方法和評估方法的適切性都會影響到實驗的結果，使得步驟 4 中的理想狀態並不容易成立。因此實際上在步驟 4 中會將 8 個判斷器值作大小上的排名，值越低排名越前面。實驗觀察的重點就在於排名是否都落在前端。



## 第六章 實驗結果—人類玩家駕駛風格模仿

### 6.1 人類玩家駕駛風格分析結果

首先列出的結果是玩家在各個賽道中關於異常狀態所占比例的統計。表 6-1 是 8 位玩家在 5 個賽道中發生衝出賽道和倒車情形的統計。其中的百分率是該位玩家在該賽道的駕駛資料中，發生上述異常狀態的比例。其中 Alpine 1 實際上是駕駛難度很高的賽道，異常狀態的比率會很低是由於賽道的邊緣幾乎都設置著牆壁。因此儘管人類玩家頻繁的在該賽道出現擦撞牆壁的車禍，但不會有衝出賽道的情形。

表 6-1 異常狀態比例統計

	CG1(%)	CG2(%)	Aa(%)	A1(%)	E4(%)	平均(%)
KT	10.55	10.17	3.02	17.17	0.1	6.63
YR	40.54	14.89	6.34	19.75	0	11.61
HA	2.98	2.21	1.14	13.04	0.47	3.80
SD	17.44	9.74	8.04	29.07	1.44	11.59
HB	40.53	12.45	18.77	26.64	2.79	16.55
JG	48.58	49.13	43.36	48.86	3.07	35.67
ZS	3.95	4.07	17.82	21.61	0.82	10.84
TS	43.85	53.07	37.66	59.34	1.18	35.62

表 6-2 是在 CG Speedway number 1 賽道上風格相似度的分析結果。以表 6-2 中 Judger<sub>KT</sub>-KT 這一格為例。意思是以 KT 在 CG track 2、Aalborg、Alpine 1 以及 E-Track 4 中的駕駛資料建成的風格相似度判斷器，判斷 KT 在 CG Speedway number 1 的駕駛資料後得到一個數值 0.462。

為了更清楚風格分析的結果，這裡將表格橫向和縱向的數值進行排名。名次的排法是將每行或每列中的數值由小到大進行排序，其中原本在理想狀況下會是最底的數值在序列中的排名。因此理想上每個排名值都是 1。以第一橫列為例，

Judger<sub>KT</sub>-KT 的值應該要是最低的，但Judger<sub>KT</sub>-ZS 的值 0.431 更小，因此排名第 2。表格的右下角列的是縱向平均排名/橫向平均排名，如果表格中的排名值是隨機產生的，則平均排名應該會大約等於 4.5，代表辨識不出風格上的異同。如果平均排名越接近 1，則可以認為玩家具有著越明顯的、具獨特性的自我駕駛風格。

表 6-2 CG Speedway number 1 賽道風格相似度分析結果

	KT	YR	HA	SD	HB	JG	ZS	TS	排名
Judger <sub>KT</sub>	0.685	0.827	0.804	0.726	1.158	0.666	0.756	0.971	2
Judger <sub>YR</sub>	0.861	0.820	0.824	0.557	1.277	0.698	0.738	0.719	5
Judger <sub>HA</sub>	0.843	0.637	0.595	0.655	0.986	0.670	0.974	1.280	1
Judger <sub>SD</sub>	1.074	1.107	1.147	0.797	1.456	0.718	1.055	0.739	3
Judger <sub>HB</sub>	0.827	0.951	1.090	0.697	0.915	0.571	0.575	1.429	5
Judger <sub>JG</sub>	1.688	2.007	1.894	1.559	2.089	0.668	1.700	1.231	1
Judger <sub>ZS</sub>	0.939	1.101	1.125	0.879	1.402	0.654	0.708	1.245	2
Judger <sub>TS</sub>	1.396	1.631	1.248	1.096	1.695	0.832	1.528	0.643	1
排名	1	2	1	5	1	4	2	1	2.125/2.5

表 6-3 CG track 2 賽道風格相似度分析結果

	KT	YR	HA	SD	HB	JG	ZS	TS	排名
Judger <sub>KT</sub>	0.462	0.491	0.624	0.923	0.812	0.920	0.431	1.373	2
Judger <sub>YR</sub>	0.719	0.475	0.552	0.598	0.762	1.172	0.408	1.301	2
Judger <sub>HA</sub>	0.492	0.605	0.420	0.770	0.837	0.991	0.449	1.135	1
Judger <sub>SD</sub>	0.746	0.739	0.700	0.753	0.962	1.049	0.495	1.138	5
Judger <sub>HB</sub>	0.644	0.853	0.669	0.771	0.526	1.161	0.433	1.630	2
Judger <sub>JG</sub>	1.526	1.537	1.346	1.405	1.659	1.197	1.174	1.607	2
Judger <sub>ZS</sub>	0.954	0.924	0.825	0.913	0.840	1.200	0.451	1.644	1
Judger <sub>TS</sub>	1.229	1.189	0.976	1.182	1.447	1.378	1.002	0.977	2
排名	1	1	1	2	1	6	5	1	2.25/2.125

表 6-4 Aalborg 賽道風格相似度分析結果

	KT	YR	HA	SD	HB	JG	ZS	TS	排名
Judger <sub>KT</sub>	1.687	1.894	2.066	1.458	2.142	1.617	1.629	1.343	5
Judger <sub>YR</sub>	1.751	1.851	1.920	1.381	1.821	1.693	1.402	1.523	7
Judger <sub>HA</sub>	1.556	1.9063	1.666	1.251	1.968	1.426	1.467	1.566	6
Judger <sub>SD</sub>	2.097	1.704	2.377	1.313	2.179	1.492	1.757	1.602	1
Judger <sub>HB</sub>	1.836	1.759	1.702	1.317	1.569	2.249	1.369	2.072	3
Judger <sub>JG</sub>	3.436	3.174	3.100	2.144	3.261	2.079	3.037	1.726	2
Judger <sub>ZS</sub>	2.167	1.877	2.625	1.558	2.259	2.024	1.309	1.732	1
Judger <sub>TS</sub>	2.463	2.233	2.802	1.706	2.916	1.501	2.324	1.104	1
排名	2	3	1	2	1	7	1	1	2.25/3.25

表 6-5 Alpine 1 賽道風格相似度分析結果

	KT	YR	HA	SD	HB	JG	ZS	TS	排名
Judger <sub>KT</sub>	0.786	0.853	0.958	0.715	1.332	0.915	0.811	0.834	2
Judger <sub>YR</sub>	0.867	0.777	1.050	0.752	1.315	0.948	0.788	0.917	2
Judger <sub>HA</sub>	0.919	0.837	0.979	0.688	1.642	0.910	0.847	0.904	7
Judger <sub>SD</sub>	1.073	0.998	1.340	0.892	1.333	1.080	0.994	1.043	1
Judger <sub>HB</sub>	0.912	0.923	1.163	0.853	1.038	0.920	0.832	0.935	7
Judger <sub>JG</sub>	1.737	1.809	2.148	1.407	2.100	0.997	1.814	1.284	1
Judger <sub>ZS</sub>	1.085	0.998	1.549	0.996	1.358	1.074	0.827	1.098	1
Judger <sub>TS</sub>	1.169	1.155	1.473	0.896	1.634	0.724	1.163	0.694	1
排名	1	1	2	5	1	6	3	1	2.5/2.75

表 6-6 E-Track 4 賽道風格相似度分析結果

	KT	YR	HA	SD	HB	JG	ZS	TS	排名
Judger <sub>KT</sub>	0.337	0.399	0.416	0.620	0.790	0.919	0.563	0.546	1
Judger <sub>YR</sub>	0.457	0.524	0.422	0.545	1.023	0.971	0.560	0.489	4
Judger <sub>HA</sub>	0.372	0.382	0.348	0.600	0.809	0.865	0.558	0.572	1
Judger <sub>SD</sub>	0.477	0.538	0.521	0.590	1.032	1.077	0.644	0.642	4
Judger <sub>HB</sub>	0.777	0.710	0.708	0.535	0.771	0.848	0.745	0.402	6
Judger <sub>JG</sub>	1.277	1.560	1.450	1.370	2.136	1.389	1.849	0.995	4
Judger <sub>ZS</sub>	0.557	0.462	0.718	0.536	0.822	0.986	0.422	0.511	1
Judger <sub>TS</sub>	0.720	0.838	0.831	1.045	1.444	1.464	1.043	0.891	4
排名	1	4	4	4	1	7	1	7	3.25/3.125

表 6-7 總平均排名

縱向平均排名	橫向平均排名
2.475	2.75

將表 6-2~6-6 中的排名值平均後的結果如表 6-7 所示。表 6-8 則是針對排名值出現的次數進行統計，其中列出的是次數及累計機率。從表 6-8(a)中可發現透過單筆駕駛記錄去辨識出玩家這個問題可以有著 52.5% 的正確率，比隨機猜測的 12.5% 高出許多。

表 6-8(a) 縱向平均排名值統計

排名	1	2	3	4	5	6	7	8
次數	21	6	2	3	3	2	3	0
累計百分比(%)	52.5	67.5	72.5	80	87.5	92.5	100	100

表 6-8(b) 橫向平均排名值統計

排名	1	2	3	4	5	6	7	8
次數	15	10	2	4	4	2	3	0
累計百分比(%)	37.5	62.5	67.5	77.5	87.5	92.5	100	100

## 6.2 模仿實驗結果

表 6-9 是風格模仿的評估結果。以 KT-Aa 這一格來說，意思是評估方法中選定的人類玩家是 KT，而測試用賽道是 Aalborg。“2”的意思是評估方法中步驟 4 裡原本在理想狀況下要是最低的值，實驗的結果是排名第二低。

表 6-9 人類玩家駕駛風格模仿相似度表現結果一

	CG1	CG2	Aa	A1	E4
KT	6	1	2	3	1
YR	6	4	6	3	7
HA	3	2	3	1	3
SD	1	4	2	4	2
HB	1	1	1	1	1
JG	2	4	7	8	1
ZS	2	2	6	3	1
TS	2	2	2	6	4
平均值	2.875	2.5	3.625	3.625	2.5

就整體的實驗結果來看，如果 40 個實驗結果都是隨機產生的 1 到 8 之間的排名值，那麼平均的結果大約是 4.5。實驗出來的總平均排名值 3.025，可以觀察出模仿和評估方法仍然是有效果的。

如果分成 5 個賽道個別觀察，可以發現到難度很高的 Aalborg 和 Alpine1 的平均排名值較大。推測可能的原因是由於賽道難度大，造成玩家為了避免車禍而頻繁的採取應急操作，使得個人駕駛風格表現的機會變少。此外，目前的控制器在要求速度表現的情況下，會自然的演化出極少發生車禍這個特性，這和在 Aalborg 和 Alpine1 中多次發生車禍的人類玩家們還是有著本質上的風格差異。

CG Speedway number 1、CG track 2 以及 E-Track 4 比較容易駕駛，可能讓人類玩家可以較無壓力的表現出個人駕駛風格。在實驗結果上，這三個賽道的平均排名值也明顯的比 Aalborg 和 Alpine1 低。

模仿相似度的結果同之前風格相似度分析，可以由橫向和縱向來觀察。表 6-9 的結果是橫向結果，也就是在控制器表現出穩健風格的前提下，觀察透過模仿對象的駕駛資料訓練出的控制器，是否存在有比其他 7 個透過別的玩家資料訓練出的控制器具有更多模仿對象的特徵。這是 5.5 節中提出的評估方法。

表 6-10 列的是縱向結果，我們的目的是分析透過模仿對象的駕駛資料訓練出的控制器和所有玩家的紀錄的相似度。總平均排名是 4.35，而各賽道的平均排

名值都很接近 4.5，也就是隨機排列時的值。

我們對這樣的結果有一個可能的解釋：由於模仿各玩家的控制器是從基礎控制器演化而來，因此已經具備了相對穩健的駕駛風格。表 6-10 最右欄中的平均排名顯示各個人類玩家和基礎控制器間已有相似度的差異，而這個差異也反映到演化後的控制器。這樣做的理由是，由於不是每一個人的駕駛風格都具有穩健性，假設控制器並不具備預設的風格，如果模仿對象的風格具有強烈積極性，那麼控制器也必須以這樣的風格為目標進行演化，而這樣的模仿方式最終訓練出的控制器會車禍太過頻繁而沒有速度上的競爭力。

表 6-10 人類玩家駕駛風格模仿相似度表現結果二

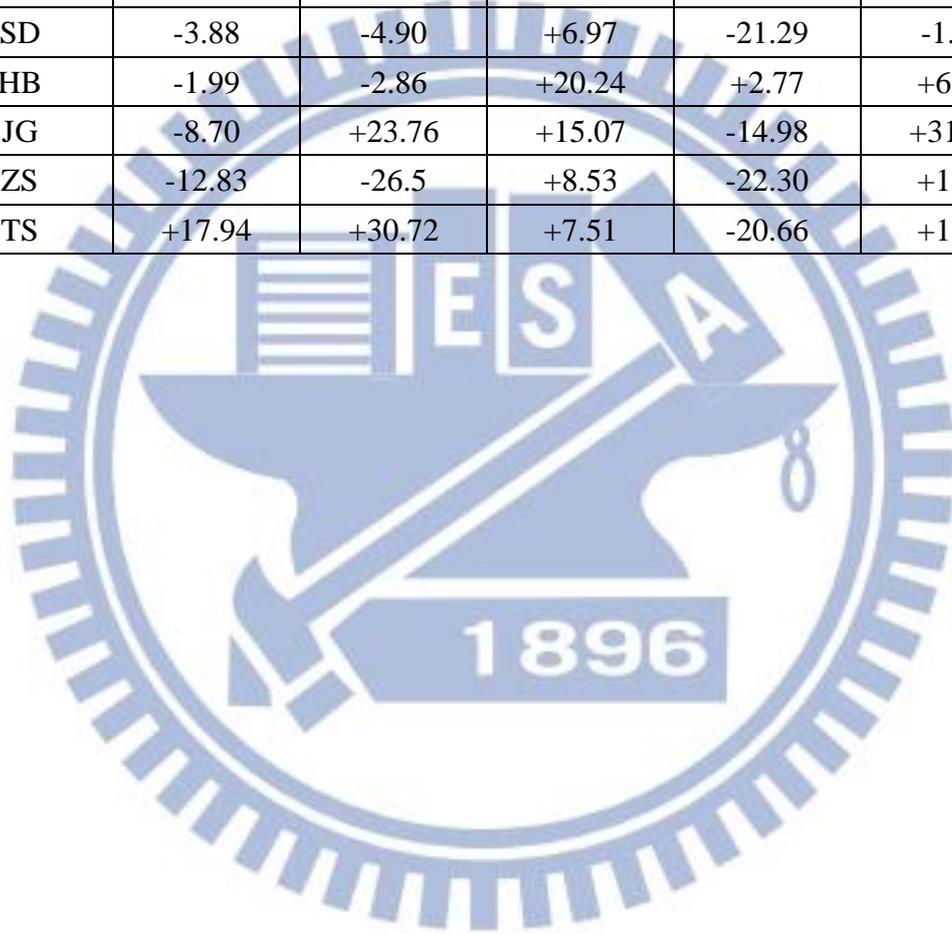
	CG1	CG2	Aa	A1	E4	平均值
KT	3	5	1	5	5	3.8
YR	2	2	2	5	4	3
HA	5	7	4	3	6	5
SD	5	5	6	6	3	5
HB	1	1	3	1	1	1.4
JG	8	8	8	8	7	7.8
ZS	2	2	5	2	2	2.6
TS	7	3	7	7	7	6.2
平均值	4.125	4.125	4.5	4.625	4.375	4.35/4.35

表 6-11 是 40 個模仿結果中，控制器和人類玩家在速度表現方面的比較。負號代表人類玩家速度表現較好，正號代表控制器速度表現較好。以 KT-CG1 為例，意思是在 CG Speedway number 1 上，KT 比以 KT 為模仿對象訓練出的控制器快了 23.76%。KT-Aa 則是在 Aalborg 上，以 KT 為模仿對象訓練出的控制器比 KT 快了 1.5%。從表中可觀察出在比較簡單的賽道中，例如 CG Speedway number 1 和 CG track 2，人類玩家幾乎都表現的比 NPC 好。在稍難一點的 E-Track 4 中，則是只有一半的人類玩家勝過 NPC。在相當難的 Aalborg 和 Alpine1 裡，NPC 在 Aalborg 中全部勝出，在 Alpine 1 則是只贏過一位人類玩家。推測的原因是 Alpine

1 中的賽道邊緣幾乎都有牆壁圍繞，因此人類玩家在發生車禍後，透過牆壁碰撞反而能以相當快的速度通過路段。

表 6-11 人類玩家駕駛風格模仿速度表現結果

	CG1(%)	CG2(%)	Aa(%)	A1(%)	E4(%)
KT	-23.76	-12.12	+1.50	-17.98	-17.71
YR	-18.65	-7.58	+9.25	-16.42	-14.06
HA	-12.25	-19.93	+12.49	-12.80	-15.36
SD	-3.88	-4.90	+6.97	-21.29	-1.72
HB	-1.99	-2.86	+20.24	+2.77	+6.83
JG	-8.70	+23.76	+15.07	-14.98	+31.14
ZS	-12.83	-26.5	+8.53	-22.30	+1.97
TS	+17.94	+30.72	+7.51	-20.66	+1.60



## 第七章 結論與未來展望

這次研究的目的是對於玩家的駕駛風格進行分析，並且讓 NPC 學習特定玩家的駕駛風格。本次實驗中使用到的控制器透過了提出的結合機制整合了低階控制器和高階控制器，在模仿 TORCS 內建的 NPC，Berniw 時，實驗結果顯示出在未接觸過的賽道中，訓練後的 NPC 和 Berniw 駕駛出的軌跡十分相似，在接近彎道處尤其明顯。速度表現方面，在測試用的賽道難度偏高的情況下，NPC 只比 Berniw 慢了 4%~14%。這說明了這樣的控制器架構發揮了預期的功能。

這次研究為了實現接觸新賽道時的適應機制及後續的效果評估而提出了風格相似度判斷器。從三個 NPC 駕駛風格分析的結果來看，判斷器確實有著效果。除了本身作為目標函數用途的功能可以被觀察出外，透過駕駛資料來辨識 NPC 也有著高準確率的結果。在四個測試賽道中，實驗結果顯示出在速度表現有明顯改善時，仍然能保有與模仿目標相似的駕駛軌跡。這代表了風格相似度判斷器和使用多目標最佳化的適應機制是有作用的。

以人類玩家為分析及模仿對象時，為了讓訓練出的 NPC 避免因頻繁的車禍造成的低速度表現，對於模仿方法和控制器的架構進行了簡化及限制，使得訓練出的 NPC 的駕駛風格至少必須表現出穩健性。在穩健性風格的前提下，根據實驗結果，NPC 可以相對性的表現出更接近模仿對象的駕駛特徵。這個結果建立在二個事實上：第一個是人類玩家在不同的賽道中確實表現出了獨特的自我駕駛風格。第二個是基礎控制器可以透過適應機制演化出模仿對象部分的駕駛特徵。第一個事實從風格分析的實驗中可以觀察到其存在程度。除了平均排名值皆明顯低於 4.5 這個隨機結果，透過駕駛資料來辨識玩家的正確率也達到了 52.5%。第二個事實從最後的模仿實驗結果中也可以被觀察到，在難度偏高的賽道 Aalborg 和 Alpine 1 中，平均排名值偏高到 3.625，但在其餘三個賽道中的平均排名值則皆低於 3。

在未來可改進的部分，首先是高階控制器可以嘗試使用更強韌的方式預測目標狀態。本次實驗使用的 KNN 方法的問題是當模仿的對象車禍頻繁時，高階控制器會連帶模仿到車禍前狀態。在本次研究中為了處理這個問題而對模仿方法和控制器作了限制，實際上可能可以透過更優秀的高階控制器解決問題。此外，單純改進指令結合器可能也有助於解決上述問題。目前指令結合器透過穩健且直觀的方法整合了二種控制器的輸出，如果可以作到更精確的判斷當前狀態適用的整合方法，除了可以降低車禍的出現頻率，對於目前方法在連續彎道中較弱的規劃能力也可能有幫助。

在以人類玩家為分析及模仿對象時，駕駛資料中的異常狀態過濾機制也有很大的改進空間。目前的方法是將賽車處於赛道外和倒車時的駕駛資料濾掉。以實際情況來說，賽車在確實發生車禍前可能就已經處在超速狀態，而且這段期間人類玩家可能進行的是危機處理的操作，使得個人風格不容易表現出來。這些不利於分析及模仿的駕駛資料可能會在赛道中產生而不會被目前的過濾機制去除。此外，車禍發生後會有一段回復到正常駕駛狀態的操作。這樣的操作可能會由於速度較低而影響到訓練出的 NPC 的速度表現。如果可以作到自動的辨識出這類異常狀態，駕駛風格的分析和模仿應該可以有更好的結果。

在本論文中，玩家在赛道中的駕駛行為被假定為在各個時間點中都是彼此獨立的。在未來的發展上，可能可以對於多個時間點中行為的相關性進行分析，以處理更複雜的風格分析及模仿問題。

## 參考文獻

- [1] D. Loiacono, A. Prete, P. L. Lanzi, L. Cardamone, “Learning to Overtake in TORCS Using Simple Reinforcement Learning,” *In 2010 IEEE Congress on Evolutionary Computation (CEC)*, 2010, pp. 1-8
- [2] D. Perez, G. Recio, Y. Saez, “Evolving a Fuzzy Controller for a Car Racing Competition,” in *Proc. IEEE Symposium on Computational Intelligence and Games*, 2009, pp. 263 – 270
- [3] E. Onieva, D. A. Pelta, V. Milanés, J. Pérez, “A fuzzy-rule-based driving architecture for non-player characters in a car racing game,” *Soft Computing - A Fusion of Foundations, Methodologies and Applications* vol. 15, no. 8, pp. 1617-1629, 2011
- [4] J. Muñoz, G. Gutierrez, A. Sanchis, “A human-like TORCS controller for the simulated car racing championship,” in *Proc. IEEE Symposium on Computational Intelligence and Games*, 2010, 473 –480
- [5] J. Muñoz, G. Gutierrez, A. Sanchis, “Controller for TORCS created by imitation,” in *Proc. IEEE Symposium on Computational Intelligence and Games*, 2009, pp. 271 – 278
- [6] J. Quadflieg, M. Preuss, O. Kramer, G. Rudolph, “Learning the Track and Planning Ahead in a Car Racing Controller,” in *Proc. IEEE Symposium on Computational Intelligence and Games*, 2010, pp. 395 – 402
- [7] L. Cardamone, D. Loiacono and P. L. Lanzi, “Applying Cooperative Coevolution to Compete in the 2009 TORCS Endurance World Championship,” *In 2010 IEEE Congress on Evolutionary Computation (CEC)*, 2010, pp.1 – 8
- [8] L. Cardamone, D. Loiacono, P. L. Lanzi, “Evolving Competitive Car Controllers

for Racing Games with Neuroevolution,” in *Proc. the 11th Annual conference on Genetic and evolutionary computation*, 2009, pp. 1179-1186

[9] L. Cardamone, D. Loiacono and P. L. Lanzi, “Learning drivers for TORCS through imitation using supervised methods,” in *Proc. IEEE Symposium on Computational Intelligence and Games*, 2009, pp. 148-155

[10] M. Ebner and T. Tiede, “Evolving Driving Controllers using Genetic Programming,” in *Proc. IEEE Symposium on Computational Intelligence and Games*, 2009, pp. 279 – 286

[11] N. van Hoorn, J. Togelius, D. Wierstra and J. Schmidhuber, “Robust player imitation using multiobjective evolution,” In *IEEE Congress on Evolutionary Computation*, 2009, pp.652-659

[12] 林俞丞, “使用近似的高階資訊於基於模糊邏輯的 TORCS 模擬器車控制器之研究,” 碩士論文, 國立交通大學, 2011

[13] [http://en.wikipedia.org/wiki/K-d\\_tree](http://en.wikipedia.org/wiki/K-d_tree)

[14] <http://www.mathworks.com/matlabcentral/fileexchange/31166-ngpm-a-nsga-ii-program-in-matlab-v1-4>