國立中山大學 電機工程學系

碩士論文

應用強化式學習建構模糊類神經控制系統

Constructing Neuro-Fuzzy Control Systems Based on Reinforcement Learning Scheme

研究生：裴善成 撰

指導教授：李錫智 博士

中華民國 九十六年七月

摘要

早期的模糊控制器依賴專家知識建立規則庫，無法使用輸出入的訓練資料，這是因為受控體反應有延遲現象的緣故。本論文提出一種新的模糊控制器設計方法，以強化學習演算法建立模糊控制器，目的是為了從延遲的反應中發掘可以達控制目標的最佳控制訊號順序。

此系統使用一延時類神經網路來預測可能的控制效果，在學習中也同時增加新的模糊規則。Q-learning 網路和模糊控制器網路也同時使用梯降法修正。實驗結果顯示所得之模糊規則可有效進行控制。

關鍵字：模糊類神經, 強化式學習, 控制系統

Abstract

Traditionally, the fuzzy rules for a fuzzy controller are provided by experts. They cannot be trained from a set of input-output training examples because the correct response of the plant being controlled is delayed and cannot be obtained immediately. In this paper, we propose a novel approach to construct fuzzy rules for a fuzzy controller based on reinforcement learning. Our task is to learn from the delayed reward to choose sequences of actions that result in the best control. A neural network with delays is used to model the evaluation function Q. Fuzzy rules are constructed and added as the learning proceeds. Both the weights of the Q-learning network and the parameters of the fuzzy rules are tuned by gradient descent. Experimental results have shown that the fuzzy rules obtained perform effectively for control.

Keywords:

control system, neuro-fuzzy, reinforcement learning

目錄

第一章　簡介

控制問題是指對系統發出控制信號，以期該系統達到所要求的反應。舉凡車輛的行走、飛機的飛行、冰箱的溫度等均可列入此問題的範疇中。隨著時代的進步，人類面對更加複雜的問題時，單靠人工發出控制命令已是不合效率或不切實際的事，因此近代控制的研究主題便著墨於『自動控制』方面。自動控制便如字面一般，在不需要或是極少人工操作員下進行系統的控制工作。迄今自動控制也已經有相當卓越的研究成果，廣泛運用於生活及工業各種應用上。

自動控制仍有許多挑戰需要面對，傳統控制方法的設計是需要分析系統的特性，根據系統特性找出正確的控制法則，便可使控制器根據這些法則進行自動控制的工作。但這其中有兩項最主要的問題，在現實問題上不一定能找出正確的控制法則，基於現實問題中存在許多隨機性和不確定性，這同時也造成了另一個問題，系統特性無法被正確的掌握。在傳統控制方法中，大部份是以約略化方式解決這些問題，亦即是將這些不確定性儘量減小，從而讓系統大多數的時間內大致保持在所欲控制的目標上，例如常用的線性化（linearization）技巧，就是將系統中較不可預期的非線性（nonlinear）行為影響減低至最

小，便可以線性控制規則設計控制系統。

當問題中不可確定性不可忽略時，這些約略方法便難以適用，為了突破這項瓶頸，許多專家便將許多人工智慧領域的技術應用到控制系統上，進行智慧型控制的研究，諸如類神經網路（neural network）、模糊系統（fuzzy system）、貝氏網路（Baysian network）等均有應用成功的例子，相關的研究可參考以下文獻。智慧型控制的主要特點是具有學習能力，能對專家提供的控制範例做歸納或推論出正確的控制規則，也可以隨著環境的改變而修正本身的控制規則，改良了傳統控制方法對於不確定性等問題。

人工智慧的方法主要分成兩大類，監督式學習（supervised learning）和非監督式學習（unsupervised learning）。監督式學習如前所述是歸納推論專家提供的訓練樣本得來。非監督式學習不需要訓練樣本，本身便具有將數據歸納分群的能力，但卻無法得出確定的控制規則，需要搭配其他的推論方法才能應用於控制系統中，所以常作為前置處理用。例如搭配模糊系統，將輸入資料群聚以提升模糊規則的歸屬度等。

然而在控制系統設計中，監督式學習有一個很大的問題，那就是無法取得所需的訓練資料。對於監督式學習來說，必須有輸入－輸出成對的訓練資料才能做調整，對於控制系統的設計，輸入的訓練資料

就代表了『環境的狀況』，輸出的訓練資料就代表了『對於這種狀況，應該做如何的動作才能達到控制目標』。有些控制問題可以適用這樣的情形，例如對於某些非線性控制系統，傳統線性控制系統設計無法達成我們的要求時，便可將系統分析的結果套用在人工智慧方法上，例如模糊系統或是類神經網路等，藉由這些方法設計智慧型非線性控制器，同時也經由系統分析的結果，設計能使這些控制器適應環境的調變方法。

但是在我們之前的討論中有提到，使用智慧型控制系統設計的主要原因，常常是不能有效率地做系統分析。某些環境中，輸出入資料取得困難，要取得足以做系統分析的輸出入資料所花費的時間金錢等成本太高，例如在倒單擺（inverted pendulum）實驗中，要測試每一種傾斜角度下，台車前後位移的各種操作順序等，其所花費的時間是不切實際的。因此監督式學習經常並非用來解決控制系統設計問題，而是解決其中某部份問題，例如針對倒單擺系統，使用模糊類神經網路來取代效率不佳的線性控制器等。

以上所述表達了智慧型控制的瓶頸，對於複雜的控制系統問題來說，無論是監督式學習或是非監督式學習都沒辦法有效解決問題，因此考慮第三類的強化式學習（reinforcement learning)（又可稱之為增強式學習）。強化式學習可算是介於監督式學習及非監督式學習間

的第三種學習方式，具有自我驗證輸出值正確與否的能力，能發掘驗證輸出值正確與否這項特點是非監督式學習做不到的，而也不需要像監督式學習一樣依賴訓練資料。對於增強式學習而言，整個學習過程是在找尋『與環境互動的正確方法』，亦即在系統的各種狀態下嘗試不同的動作、或是輸出訊號，以控制目標所定立的某種評價標準評判此項動作的價值，並將具有最高價值的動作記憶起來，便可以找到最有價值的訓練方法。

在控制系統問題上，強化式學習可以自行摸索最佳的控制法則，即使是事先沒有經過系統分析，也能夠調整適應系統的性質，根據需要控制的目標，評價所發出的控制訊號，進而找到適當的控制規則等。已有許多著作探討強化式學習在控制系統的應用，如上例的倒單擺系統、機器人行走的學習、或是自動車輛駕駛等，這些都是屬於難以做系統分析或是無法有效率地收集訓練資料的控制系統問題，便是強化式學習可以應用的領域。

強化式學習有兩件主要工作，第一是嘗試找尋最佳控制訊號，第二是發出最佳控制訊號使系統達到控制目標，這兩件工作本應是先後進行，但在無法確定何時才能找到最佳控制器的情況下，只能同時進行這兩件工作，造成強化式學習有著初期效率極糟的問題，對於某些工業上的應用，這可能會導致重大的損失，若能找到適當的初始值，

便可改善初期效果太差，以及增加訓練速度。

這篇論文中提出一個架構，以簡單的控制系統做出控制器的初始值，結合類神經與模糊類神經網路架構強化式學習機制，在學習中修正模糊類神經網路控制器的參數與架構。測試中本系統有達到控制的目標與學習的功能。

第二章　文獻探討

2.1　強化式學習

強化式學習是種在與環境互動中，不斷嘗試不同的行動，找尋最佳行動策略的學習方法。強化式演算法中主要有兩個角色：一個是『環境』（environment），環境是所欲解決問題之一切外在因子的總稱，例如在旅行銷售員問題（traveling salesman problem）中是指可行走的城市數、路徑的連接與距離等，用在倒單擺問題上是指台車重量、單擺長度、磨擦係數等。另一個角色是『代理人』（agent），代理人負責與環境互動並學習互動的結果，與環境互動是代理人送出『行動』（action）給環境，致使環境改變目前的『狀態』（state），狀態改變後對代理人的目標有何影響，是表現在『獎賞』（reward）上的，正面影響越大，獎賞自然也越高，代理人根據所收到的狀態改變狀況與獎賞，可以建立或修正『價值函數』（value function），價值函數代表某狀態對所欲達成目標的影響程度，與獎賞不同的是，價值函數還考慮了該狀態未來的影響，所以結合了立即影響的獎賞與未來影響的價值函數，能提供代理人有前瞻性的判斷依據。當代理人多次與

環境互動後，愈趨正確的價值函數便可提供代理人正確的參考依據，代理人根據價值函數做出判斷行動的『策略』（policy）也會越來越好，最終將會找到符合目標的策略，完成學習的目的。

為了考慮未來影響，價值函數除了紀錄目前可得獎賞外，還紀錄了未來狀態的獎賞，未來會是怎樣的狀態本屬不可知之事，但唯一可以肯定的是，代理人會儘量選擇能達成目標的行動，也就是做出能移往最高價值的狀態的選擇，所以可以假設未來每次狀態最大可得獎賞的總和，便是該狀態的未來價值。

價值函數可以正式寫成以下形式：

$$V(S_{t_0}) = r(S_{t_0}) + r(S_{t_1}) + r(S_{t_2}) + \cdots + r(S_{t_\tau})$$

其中S_t代表各時間的狀態，$r(S)$代表在該狀態所得之獎賞，$V(S)$代表該狀態的價值，若未來所有狀態均假設是以最佳狀態來選擇，則可以寫成以下遞迴型式：

$$V^*(S_{t_0}) = r(S_{t_0}) + V^*(S_{t_1})$$
$$V^*(S_{t_\tau}) = r(S_{t_\tau})$$

上式中t_τ代表終止時間。

價值函數除了純粹以狀態函數的方法來表示外，還有另一種表示方式，以狀態與動作來表示，這兩個表示法是等價的，只是型式不同而已。狀態動作函數可寫成以下形式：

$$Q(S_{t_0}, A_{t_0}) = r_{t_0} + \max_A \left(Q(S_{t_1}, A_{t_1}) \right)$$

代表所採取的行動。使用狀態動作函數的優點是較明顯表達某行動在某狀態所造成的影響。

目前常用的強化式學習演算法包括了 Q-learning 、 Sarsa 、 Actor-critic 等，其中 Q-learning 因其容易實作等性質，廣在各項問題中應用。在 Q-learning 中，代理人紀錄了每個狀態下，所有可能採取的行動的價值，而代理人執行的策略便是在每一狀態時，從所有行動的價值中，找出最大價值的行動並執行之。所用的學習方法，是在每次行動後，比對原本預期的最大價值，與實際所得的獎賞、與實際上環境改變至的新狀態、可得的未來價值等，做比較，更新原有的預期價值。如果單看一次更新的話可以寫成下式：

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_t + \gamma \max_A Q(S_{t+1}, A) - Q(S_t, A_t) \right]$$

$Q(S_t, A_t)$ 代表在狀態 S_t 時進行 A_t 所得的預期價值，R_t 是代理人執行動作 A_t 後實際得到的獎賞，S_{t+1} 是環境接受代理人所執行的動作後，實際轉變成的狀態，$\max_A Q(S_{t+1}, A)$ 是在 S_{t+1} 狀態時、在所有能進行的行動中、預期所能得到的最大價值。γ 是未來獎賞的折扣率，通常是一個小於 1 的常數，代表離現在時間越久的未來所預期的價值越不受重視，原因多數是為了對儘快達成目標的策略價值加分等。α 是更新速率，用以控制學習收斂速度，一般通常一開始會將此值設為

1 ，在學習過程中逐漸減小趨至於零，在學習達成目標後便設為零、停止學習而不再更新已有的價值函數。

強化式學習的基礎是找到每個狀態下最佳的行動，為了達成這個目標必須要多方嘗試，但做隨機嘗試可能與『代理人每次都會選擇最佳行動』的假設相違背，亦即不去選擇已知的最佳行動，反而去挑選隨機的任意行動，這可稱是『解決問題和探索新知』（exploitation and exploration）的兩難問題。在 Q-learning 演算法中，通常運用 ε-greedy 演算法來解決這個問題。基本上這也可以算是 greedy 演算法的一種，亦即會找最佳的解答，符合我們解決問題的需求，不過每次在尋找解答時，會有的機率會做隨機選擇，試圖找到更好的行動。在訓練初期對價值函數所知甚少時，此值便會設定較大以增加探索各種行動的機會，待至後期已大致探索完畢時，此值便會設定較小以增加做出正確選擇的機會。

Q-learning 演算法的步驟如下：

步驟一、初始化所有的 $Q(S,A)$ 函數

步驟二、取得系統的初始狀態 S

步驟三、根據目前的狀態 S 使用 ε-greedy 選出行動 A

步驟四、代理人對環境執行行動A、環境傳回新的狀態S'及獎賞R，若該回合結束，環境並無傳回新的狀態，則跳至步驟七

步驟五、更新原有的期望價值函數

$$Q(S, A) \leftarrow Q(S,A)+\alpha\left[R + \gamma \max_{A} Q(S',A) - Q(S,A) \right]$$

步驟六、代換$S \leftarrow S'$，回到步驟三

步驟七、更新原有的期望價值函數$Q(S, A) \leftarrow Q(S,A)+\alpha[R]$，若訓練目標尚未達成，回到步驟二

Q-learning 相較於其他強化式演算法的優勢在於：1）只需要較少的儲存空間，相較於某些將預期價值函數與選取行動策略分開儲存的強化式演算法，例如 actor-critic 等。2）其預期價值函數較直覺，相較於其他使用狀態價值函數的演算法，使用狀態動作價值函數比較能直覺地了解在某狀態執行某行動的價值。 3）因為以上兩個特點，使得 Q-learning 較易實作。

強化式學習近年來有相當多的研究專注其上，不論是基本的尋求最佳解的問題、實務面的控制問題、以及對局競賽類問題等，都有不少的研究成果。Q-learning 演算法是其中的代表作，提供了一個直覺而有效的解決問題的規劃，因此可用已有的人工智慧演算法實作

Q-learning ，從最簡單的表格查詢到模糊類神經網路等，強化式學習都能結合已有的研究成果為問題的解決帶來新的導引。

2.2 模糊類神經系統

模糊系統（fuzzy system）自發明以來，已有不計其數的理論研究，除了已經在實務上應用外，也進入了日常家電中，成為新科技的代名詞。

模糊理論的出發點是重新定義傳統的邏輯集合概念，傳統的邏輯是二元的『非真即偽』，不存在第三種選項，傳統的集合也是只有包含該元素或不包含該元素，沒有第三種關係。但在現實生活中存在著許多灰色地帶，我們常常無法一口咬定該元素屬於或是不屬於該集合，例如在語意上，身高 170 公分以上算是高個子，但身高 169 公分也不能說不算高個子，類似這種無法清楚劃分，只有程度上差別的現實狀況，傳統的集合便無法定義，解決這類的『灰色地帶』的問題便需要依靠模糊集合理論（fuzzy set theory）。

在模糊集合理論中，捨棄了原本傳統集合定出界限的做法，改為以從屬度（membership degree）來做區別，元素的從屬度越高就越接

近該集合的概念，例如對於『高個子』的集合中，身高 169 公分的從屬度會比 180 公分的從屬度低。計算該元素對該集合從屬度的方法是用從屬函數（membership function），例如若以下圖的從屬函數可算出身高 169 公分對高個子集合的從屬度是 0.6 ，而身高 180 公分的從屬度是 1 。

有了模糊集合理論後，就可以進行模糊推論的應用。傳統的邏輯推論是由『若條件符合，則某項事實成立』構成，應用在模糊理論上，則隨著條件符合的程度不同，事實成立的程度也相異。例如以『若為天氣寒冷，則開暖氣。若天氣不寒冷，則不開暖氣』的邏輯為例，以傳統集合理論來區分，攝氏 10 度以下算是天氣寒冷，暖氣也只有開或不開的選擇時，氣溫 11 度時的推論結果是不符合天氣寒冷的條件，所以不能開暖氣。這類的邏輯推論是與現在狀況相違背的，若套用模糊邏輯來進行推論，假設根據『天氣寒冷』此集合的從屬函數，11 度的從屬度是 0.9 ，則對於前項規則就有 0.9 的歸屬度，後項規則則有 0.1 的歸屬度，在經過推論後，我們可以得到『開 0.9 程度的暖氣』這個比較合理的結論。

當模糊推論應用在實際問題上時，系統主要包含了模糊集合的歸屬函數、模糊規則庫（fuzzy rule base）及模糊推論器（fuzzy

inference engine）等三大部份，處理資料的流程是： 1）模糊化（fuzzification）：以各個模糊集合的歸屬函數，計算輸入值的歸屬度。 2）模糊推論（fuzzy inference）：讀出模糊規則庫中的規則，根據模糊化後各模糊集合的歸屬度,進行模糊推論後得到結論。3）解模糊化：結合各規則推論所得的結論，以及各規則的歸屬度，將模糊集合結論換算成數值後輸出。

有許多模糊系統可供選擇，其中兩個代表性的分別是 Mamdani 和 TSK 系統：

1）Mamdani：此系統的模糊規則如下：

```
If x1 is A and x2 is B, then y is C
```

上述的 x1 和 x2 是輸入資料的屬性，y 是輸出，A、B、C 分別是三個模糊集合。

Mamdani 系統的優點是可以直接套用原本的邏輯法則，就像前述暖氣的例子，只要定義了『天氣寒冷』和『暖氣開』這兩個模糊集合的歸屬函數，就可以馬上進行模糊推論，對於人類來說比較直覺，也比較容易將已有的專家知識應用其中。

2）TSK: 此系統的模糊規則如下：

```
If x1 is A and x2 is B, then y = c0 + c1*x1 + c2 * x2
```

大部份參數與前式同義，除了 c0、c1、c2 代表三個常數外。

TSK 系統將原本的結論部份改為一個多項式，增加規則彈性的同時可以減少規則數量，節省解模糊化所需的時間成本。但相對於 Mamdani 而言，TSK 的多項式參數不容易以直覺設定，因此必須多花些功夫才能將既有的專家知識轉化為規則。有些領域，例如控制系統，卻可利用原有方法計算出來的數據套入參數中，反而較 Mamdani 適用。

模糊系統雖然有許多優點，但是需建立在正確的模糊集合與其從屬函數上，即便初始是經由專家所設定，隨著系統的進行也可能發現有更好的組合，如果能自行學習調整，就可以改進原本的模糊系統進而找到最佳解。目前普遍將類神經網路（neural network）的概念加入其中，結合而成模糊類神經系統（neuro-fuzzy system）。

類神經網路是仿照人體的神經系統，以結點間的連結『記憶』所受到的刺激。在實作中，連結的記憶便以權重（weight）的方式來儲存，隨著訓練資料的輸入和輸出，權重便加以變化修正系統的輸出值。以一般常用的前授型網路（feed-forward network）而言，將訓練資料輸入後，計算出輸出值與應有輸出值的差異，以倒傳遞(back-propagation）法則回推至各層與各個節點間的連結，用最陡坡降法（steepest descent method）修正各連結的權重，重複修改至穩

定為止。

類神經網路具有許多優點，包括準確度高、平行計算、軟硬體都容易實作等特性，卻也有相當大的缺點，其中之一便是類神經網路只能達到局部最佳化（local optimization）。常用的前授倒傳遞網路就深受影響，受限於初始設定的節點數、隱藏層數、及權重等，有可能無法達到全面最佳化（global optimization）而把錯誤降至最小。雪上加霜的是，初始值難以做有意義的設定，只能不斷嘗試或是隨機產生而已。

模糊系統是容易加入專家知識，放入初始值而難以調整，類神經網路可以輕易調整網路減低錯誤卻苦於尋找好的初始設定，於是模糊類神經系統就自然而然產生了。常見的模糊類神經系統是以類神經網路實作模糊系統而成，用各層的節點和各連結的權重作為模糊推論的功能，輸入層的節點中包含歸屬函數作為模糊化之用，輸出層則用做解模糊化。隨著訓練資料的輸入，以類神經網路的方式修改模糊推論規則以及模糊集合的歸屬函數。同時解決了類神經網路的初始化問題和模糊系統的訓練問題，可說是一舉兩得，因此也廣泛地應用於解決各種問題上。

2.3 應用強化式學習於控制系統

對於控制系統問題來說，強化式學習是個可以解決整體問題的架構，非常適合用來嘗試傳統控制系統難以達成或是效果不佳的控制需求，就 Q-learning 演算法來說，已被證明以表格方式實作可以有效搜尋行動策略並收斂至最佳解，也已經被用來解決許多控制的問題，但仍有一點問題，以表格而言只能處理離散的狀態和離散的動作，幾乎所有控制問題，環境狀態的參數都是連續的，雖然可以透過離散化(discretize) 的動作將連續切為離散,但所面臨的另外一個問題就是切得太粗輸入參數自然就不精確、切得太細狀態空間太大則不論是時間或空間成本均無法負荷，所以在有限的離散狀態和行動情況下，表格實作的 Q-learning 演算法尚可發揮效用，但除此之外的應用便得找到更好的強化式學習系統。

對於上述問題，改良的強化式學習系統所需具備的重要能力就是歸納 (generalization) ，如何能利用相似的狀態，推論出一個尚未記憶的狀態應有的動作，便需要對以往所得知的狀態做歸納整理後才能推論。應用類神經網路便是一個不錯的選擇，類神經網路使用權重記憶訓練值的方法本身便具有歸納的能力，若要取代原本的表格，只需輸入狀態和行動，便可輸出預期的價值，同時也將原本更新狀態動

作價值函數的方式套用至最陡斜坡法上,便可以處理連續狀態的問題。

連續狀態或許已經解決,但連續行動空間仍是個值得探討的問題,Q-learning 原本採用的ε-greedy演算法在連續空間更不可行,相較於狀態的查表，要找出最佳行動必須搜尋所有可能的行動，這使得細切動作空間比狀態空間增加數倍的時間複雜度，因此除了有限的動作以外，原本的 Q-learning 難以應用在大部份都是連續動作的控制問題上。

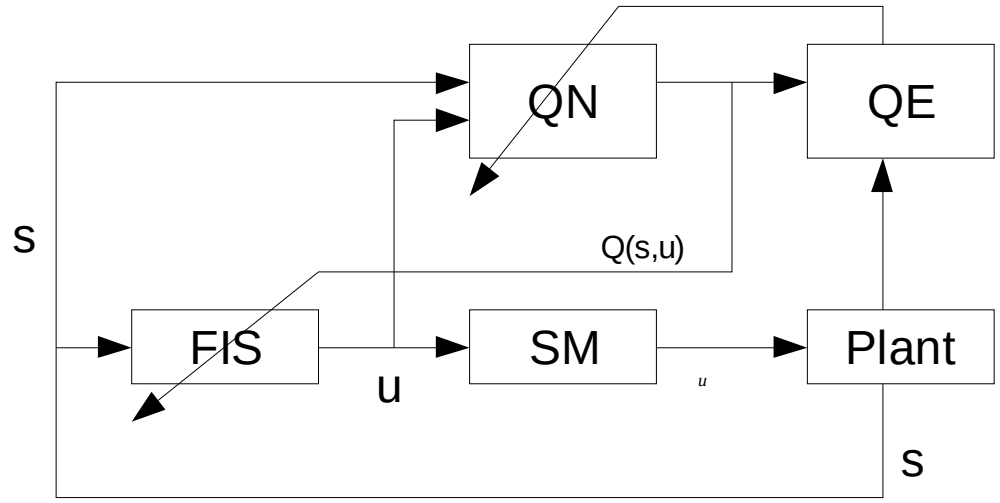
針對這個問題目前主要有兩種解決方案，一種是仍以離散動作為基礎，以模糊系統加權組合離散動作後輸出連續動作，另一種是以模糊類神經控制器直接輸出連續動作。對於前者來說，離散動作的選擇十分重要，正確或錯誤的離散動作對訓練結果影響甚大，也因此如果遇到無法事先獲知專家知識的問題，系統效能將大打折扣。後者雖然是直接輸出連續的動作，但有可能因為類神經網路的特性，構成價值函數的網路和找尋最佳動作的控制器網路都進入了局部最佳化，若任一掉入局部最佳化還有可能被另一部份修正，但若同時掉入的話很可能會進入穩定狀態而導致學習失敗。以上兩者都需要關鍵性的專家知識,但若在無法順利取得專家知識的環境下,很可能造成學習的失敗。

因此本篇論文循著以模糊類神經控制器實作 Q-learning 演算法的軌跡，提出能自動偵測初始值的系統，減低對專家知識的需求，以

期應用到更多的控制問題上。

第三章 研究方法

3.1 系統架構



系統架構圖如上圖，以下分別說明各元件功用。

3.1.1 FIS

Fuzzy Inference System, 模糊類神經網路控制器。負責輸入狀態後輸出能產生最大價值的動作。其功用可用方程式表達如下：

$$A = f(S) = \arg\max_{A} Q(S,A)$$

對於控制系統來說， TSK 是較 Mamdani 好的系統選擇，因此使用 TSK 作為推理系統：

If S_1 is G_1 *and* S_2 is G_2 *then* y is $c_0+c_1S_1+c_2S_2$

上述G_1和G_2是兩個模糊集合，其從屬函數是屬於高斯函數：

$$G(S_1) = \exp\left(-\frac{\|S_1 - C\|^2}{2\sigma^2} \right)$$

上述C是該從屬函數的中心點，σ是標準差。

使用高斯函數雖然相較於三角梯形等比較簡單的從屬函數計算量較大，但高斯函數具有連續可微的性質，若使用其他函數需得用不同的啟發式（heuristic）學習法,高斯函數因其可微性質可直接套用最陡坡降法修正。

此模糊類神經網路控制器共有五層，第一層是輸入層，負責接收狀態的屬性數值，並把數值複製送至各個模糊集合的從屬函數中，第一層和第二層間的連結其權重固定為 1 。第二層是模糊化層，計算各規則中各模糊集合的從屬值，從屬函數存於第二層的節點中。第三層是模糊推論層，計算各規則的從屬度，這裡使用相乘作為模糊交集運算。第四層是解模糊化層，將第一層的數值乘上各規則的結論部份，計算出結論多項式的值。第五層是輸出，將每條規則的結論值以規屬度加權平均，輸出即為控制命令。

3.1.2 QN

Q-learning Network 價值網路主要是用來預測狀態與動作的價值，即狀態動作價值函數，可以下式表示：$QN(S,A) = Q(S,A)$

價值網路是以類神經網路實作，同時為了某些控制問題中價值具有與時間相關的性質，使用動態的延時網路（time-delay neural network）來實作。在延時網路中，輸入包含了目前的狀態及過去幾個時間的狀態：$\mathrm{S}(t),\mathrm{S}(t+1\Delta),\mathrm{S}(t+2\Delta),\cdots,\mathrm{S}(t+n\Delta)$

上述的Δ是時間間隔，n是最大的過去狀態數目，再加上動作A就是該網路的輸入值。輸出值就是狀態動作的預測價值。

此網路使用三層的架構，分別是輸入層、隱藏層和輸出層。輸入層接收輸入值，節點個數與輸入值個數相同，若狀態有m個屬性數值，則共有$1+m(1+n)$個節點。隱藏層的節點中存有雙彎曲函數（sigmoid function）運算，如下式：

$$f(x)=\frac{1}{1+e^{-x}}$$

使該網路具有表達非線性函數的能力。輸出層只有一個節點，輸出計算出來的預期值，另因本應用主要著眼於迴歸（regression）而非分類（classification）問題，所以此層的節點只是單純輸出而不做任何處理。

3.1.3 SM

Stochastic Modifier 隨機修改器是將原本最佳化的輸出動作，修改以進行探索動作空間的工作。不同於離散動作狀況，這裡使用常

態分佈機率（normal distribution）找尋其他的可能性，以原本所輸出的最佳化動作為中心，以常態分佈機率隨機挑選兩旁的數值，以下式表達：

$$\hat{u} = u + x, x \square N\left(0, \sigma^2\right)$$

上式中u為原本挑選的最佳化動作，x是以常態分佈機率挑選出來的隨機數值，此常態機率N的中心點為零，標準差為σ。在訓練初期時使用比較大的標準差，增加輸出值變化的機率，而在訓練後期時減少標準差，減低輸出值的變化，穩定系統的表現。

3.1.4 QE：

Q-learning Evaluation 以環境傳回的資訊修正原有的價值網路，這裡的環境是指 Plant 受控體。

3.2 訓練方法

3.2.1 訓練價值網路

價值網路的修正便是要減少所預測的價值與實際上應有的價值間差異，可用下式表達：

$$\delta = \mathrm{R} + \gamma \max_{\mathrm{A'}} \mathrm{Q}(\mathrm{S'},\mathrm{A'}) - \mathrm{Q}(\mathrm{S},\mathrm{A})$$

上式中$\mathrm{R} + \gamma \max_{\mathrm{A'}} \mathrm{Q}(\mathrm{S'},\mathrm{A'})$是實際上應有的價值，$\mathrm{Q}(\mathrm{S},\mathrm{A})$是所預測

的價值，δ便是該差異。我們希望能逐漸減小差異：

$$\mathrm{Q}(\mathrm{S},\mathrm{A}) \leftarrow \mathrm{Q}(\mathrm{S},\mathrm{A}) + \alpha\delta$$

上式中α是學習率。而在類神經網路的學習上，使用平均平方差（mean square error）會有比較高的準確度，因此這樣訂立錯誤函數：

$$E = \frac{1}{2}\delta^2$$

對於每個在價值網路中的權重w，使用以下的更新規則：

$$w_{new} = w_{old} - \eta\frac{\partial E}{\partial w}$$

上式中η是學習率，取代了α的功用。而該錯誤的偏微分也可寫成：

$$\frac{\partial E}{\partial w} = \delta\frac{\partial \mathrm{Q}(\mathrm{s},u)}{\partial w}$$

3.2.2 控制器的訓練

控制器的訓練目標是在儘量輸出具有最大價值的動作，依據價值網路所預測的價值，調整輸出以增加輸出值的價值。如下式：

$$\xi_{new} = \xi_{old} + \beta\frac{\partial \mathrm{Q}(\mathrm{s},u)}{\partial \xi}$$

上式之ξ是控制器的參數，包括高斯歸屬函數的中心點與標準差、每項規則的結論多項式參數等，β是學習率。上式可看出參數是往增加評價預測值的方向改變，不過控制器並非直接產生評價預測值而是透過評價網路，亦即控制器將動作輸入至評價網路後才能影響評價預測值，所以是以下式來做更新：

$$\frac{\partial Q(s,u)}{\partial \xi} = \frac{\partial Q(s,u)}{\partial u}\frac{\partial u}{\partial \xi}$$

值得注意的是，在評價網路的訓練中，所用的$\max_{A'} Q(S',A')$資訊便是從控制器得到最佳化的動作，而控制器訓練中的$\frac{\partial Q(s,u)}{\partial u}\frac{\partial u}{\partial \xi}$也必須使用評價網路的權重值，這樣互相影響的結果，若是同時有不良的局部最佳化現象，有可能會趨於穩定而導致訓練失敗。

3.2.3 規則數量的調整

規則數量是影響模糊系統的因素，如果太少很可能會無法涵蓋輸入數值空間，造成不夠符合訓練（under-fitting）的狀況，反之如果太多不但系統資源時間成本無法負荷，也可能發生過適訓練(over-fitting）的狀況。理想上是找到適當的規則數量，增加模糊系統的效率。

找尋適當規則數量的想法是，一開始先以最少數量的規則建構模糊系統，當狀態資料輸入時即檢查是否有任一規則對其的歸屬度高於預先訂定的門檻值，若沒有則表示現有的模糊規則都無法適用於此新資料，因此應當新增規則。

新增規則的方法是，以該新進資料作為高斯歸屬函數的中心點，標準差則以適當的常數設定，結論多項式的參數是將現有規則的多項式參數平均而得。

3.3 初始化

為了解決先前文獻中，需要過多專家知識的問題，有必要設計一套自動找尋初始值的演算法。該演算法會有兩個特點，不能花費太多的時間資源成本、及不需要太優良的結果，以前者而言，如果成本花費過鉅，則有喧賓奪主之虞，不能與強化式學習搭配。至於初始值本為訓練改進用，不必太計較效果的好壞，但是方向不能差太多，例如本來就需靠強化式學習發掘馬達轉速與冰箱溫度的關聯性，但事先得知是轉速增強會使溫度降低或是相反等資訊，也具有相當的重要性。

在此我們提出應用已有的傳統控制技術來找尋初始值。基本的線性控制演算法 PID (Proportional Integral Differential) 是根據現有狀態與目標的誤差、誤差的加總及與上次誤差的差異等三項資訊，線性組合後得到控制值。PID 的三項線性參數分別是K_P、K_I、K_D等，可以經由系統分析得到這三項參數進行控制，也有自動偵測調整的方法。

根據 Ziegler-Nicholas 方法所設計的自動參數偵測法，是送出一固定的控制訊號，觀察受控體的變化情形，待變化穩定後，紀錄三項資料，一是最後穩定後受控體的變化X，二是在變化過程中最激烈的程度，也就是變化曲線中最大的斜率R，三是變化甚小的時間，其計算方法是，在變化曲線斜率最大該點做切線與時間軸相交，所得時

間與開始時間的差D。有了這三項資訊便可以求出 PID 控制器的參數，如下式：

$$K_P = \frac{X}{DR}$$

這項理論的根據是自然中很多受控體都具有一階非時變系統的特徵，因此是可以做簡單的控制系統。對於更高階的系統無法有可靠的表現，因此也有許多的改進設計，不過在此我們只需當作初始值產生用，不強求效果。

我們的模糊系統與 PID 控制器的共通處是，都有線性組合的部份，在受控體狀態以誤差及誤差速率表達下，模糊系統中結論多項式的參數與 PID 控制器的參數有相同的功能,也因此在模糊系統的結論多項式中：

$$u = c_0 + c_1 e + c_2 \dot{e}$$

e是誤差而$\dot{e}$是誤差速率的情形下，c_1就與K_p有等價的涵義。因此我們以下列規則初始化模糊系統：

$$if\ \ e \text{ is } \mathrm{G}_1 \ and\ \ e \text{ is } \mathrm{G}_2 \ \ then \text{ u is } \mathrm{K_p}\dot{e}$$

一開始模糊系統便僅只一條規則，$\mathrm{G_1G_2}$的高斯歸屬函數也設定中心點為零而標準差為一定值。

3.4 系統流程

本系統學習整體流程如下所示：

步驟一	初始化評價網路和模糊類神經控制器的參數。
步驟二	獲得受控體狀態S。
步驟三	使用控制器找出適合狀態S的動作u。
步驟四	使用隨機修改器修改動作u而成動作$\hat{u}$以實現隨機搜尋的需求。將$\hat{u}$送入受控體中，得到獎賞r與新的狀態ns。
步驟五	使用受控體找出最適合狀態ns的動作nu。
步驟六	以公式調整評價網路
步驟七	以公式調整模糊類神經控制器
步驟八	檢查模糊控制器是否有新增規則的必要，若有，則按照前述公式新增規則
步驟九	以ns取代S，並檢查是否已經到達學習目標，若否則跳回步驟二繼續進行

第四章 實驗與結果討論

我們以兩個實驗去測試我們方法的可行性，分別是較簡單的線性系統與較複雜的非線性系統，以下便分別討論之。

4.1 線性系統

我們先以較簡單的線性系統來做測試，同時探討設計一個強化式控制系統應該注意哪些細節。

該線性系統可以用下列狀態空間式表達：

$$\dot{X}_1 = X_2$$
$$\dot{X}_2 = -X_2 - 200X_1 + u$$
$$Y = X_1$$

其中 X_1、X_2 都是系統狀態變數，u 是輸入動作，Y 是系統輸出。兩個系統狀態變數均初始為零。學習的目標是要將控制系統輸出和目標間的差異儘量縮小。首先需要設計獎賞函數：

$$r(t) = \begin{cases} -1 & ,if \left| y(t) - R(t) \right| > \varphi \\ -\dfrac{1}{\varphi} & ,otherwise \end{cases}$$

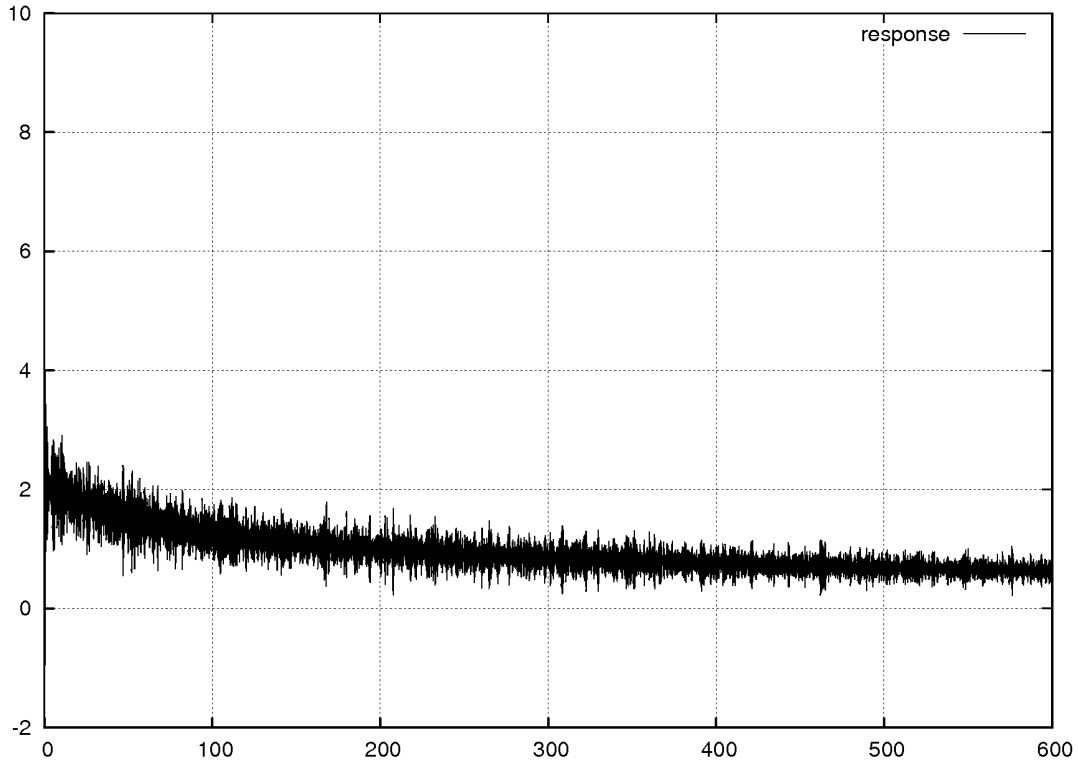
$R(t)$ 是目標值而 φ 是定義區間範圍用的常數，意即在常數區間範圍內獎賞是漸次放大的，超過此一範圍便縮小至 -1 。

開始進行實驗時要先初始化模糊系統，一開始只有一條模糊規則：

If e_1 is G_1 and e_2 is G_2 then u is $c_0 + c_1 e_1 + c_2 e_2$

G_1、G_2是兩個使用高斯歸屬函數的模糊集合，其函數的中心點為零而標準差設唯一。$e_1 = y(t) - R(t)$是代表系統輸出與目標間的差值，e_2則是e_1的微分項，代表差值改變的速率。

一開始先將c_0、c_2設為零，使用自動探測 PID 參數法找到並代入$c_1 = K_p = 776$，用其作為規則的結論部份。評價網路部份，將最大的過去變數時間設為 2 ，加上現在共有 3 個環境狀態傳入，每個環境狀態都有兩個屬性，外加動作值，總共 7 個輸入值，所以輸入層有 7 個節點，分別是$e_1(t), e_1(t+1\Delta), e_1(t+2\Delta), e_2(t), e_2(t+1\Delta), e_2(t+2\Delta), u(t)$，隱藏層設定 21 個節點，各節點間的權重隨機產生。實驗進行過程如下圖所示，隨著時間的進行，系統輸出與控制目標間的差異逐漸降至於零。



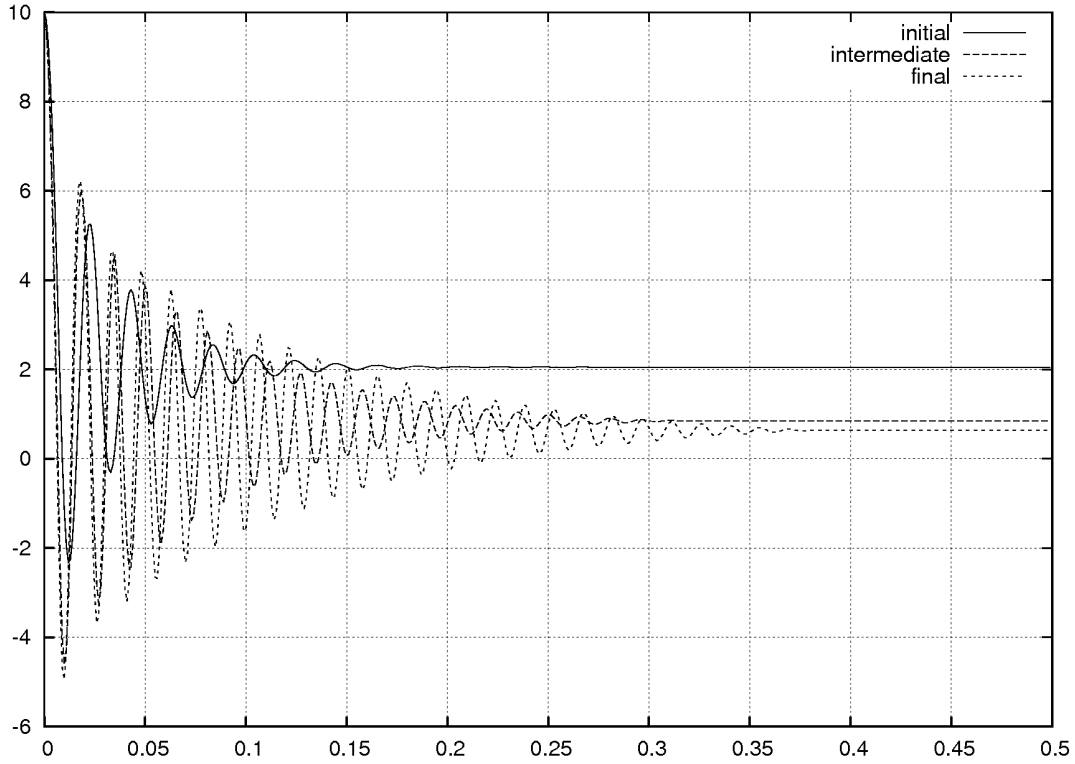
在訓練結束後由原本的一條規則增加成了四條，分別列出於下：

If e_1 is G_{11} and e_2 is G_{12} ， then $u = L_1$
If e_1 is G_{21} and e_2 is G_{22} ， then $u = L_2$
If e_1 is G_{31} and e_2 is G_{32} ， then $u = L_3$
If e_1 is G_{41} and e_2 is G_{42} ， then $u = L_4$

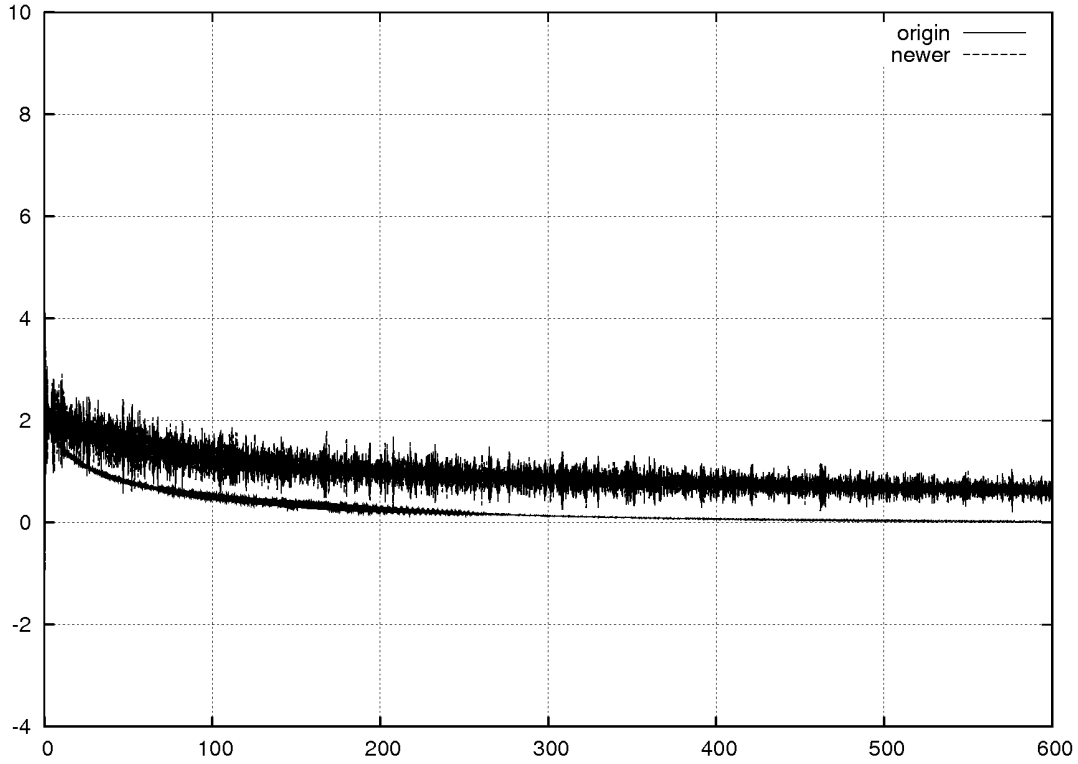
其中

$G_{11} = G(-0.023, 9.977)$ $G_{12} = G(-0.005, 10.000)$
$L_1 = 803.602 + 1964.544 \times e_1 + 30.092 \times e_2$
$G_{21} = G(5.000, 0.100)$ $G_{22} = G(0.000, 0.100)$
$L_2 = -0.104 + 775.475 \times e_1 + 0.000 \times e_2$
$G_{31} = G(5.000, 0.101)$ $G_{32} = G(-0.500, 0.101)$
$L_3 = -1.378 + 769.108 \times e_1 + 0.637 \times e_2$
$G_{41} = G(5.000, 0.104)$ $G_{42} = G(0.500, 0.101)$
$L_4 = -0.513 + 773.434 \times e_1 - 0.009 \times e_2$

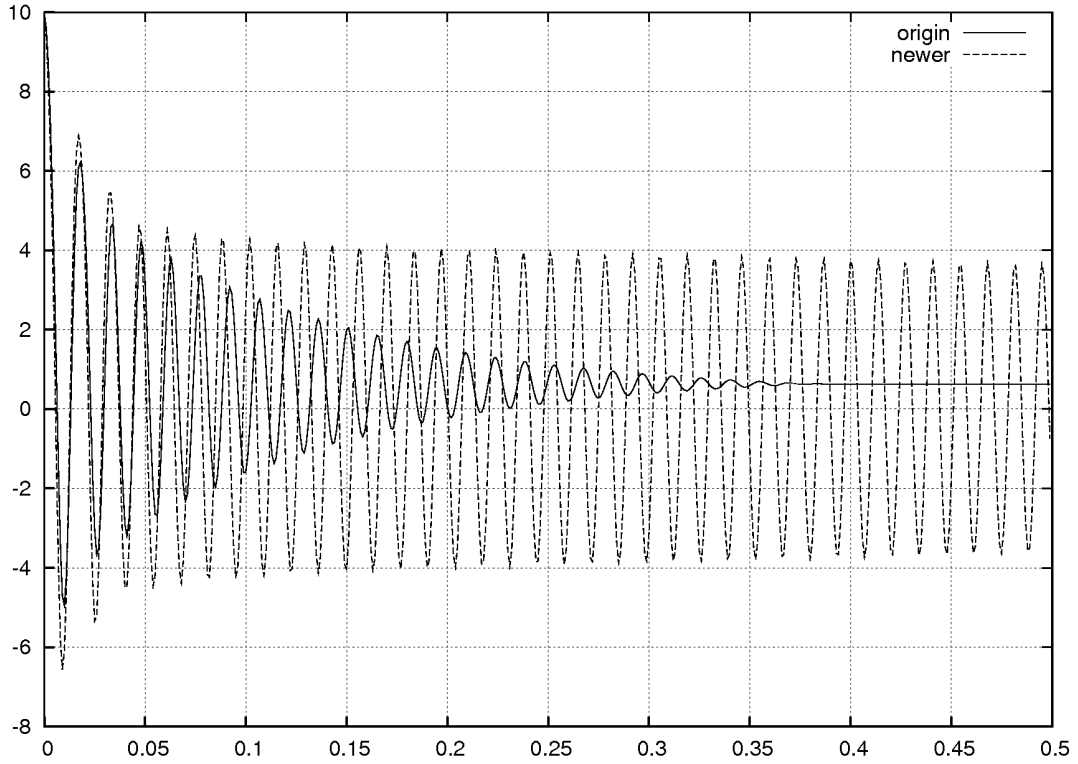
比較訓練過程中不同時段的模糊控制器效能，由下圖可以發現隨著訓練的演進，效能也不斷增加。



值得一提的是，如果改進原本的訓練方式，將原本每輪只做一次修正的控制器，改成多次修正而至穩定，會加速訓練過程，也有較佳的收斂效果，如下圖所示：



但遺憾地也同時加速控制器的震盪增加幅度，下圖是修正前和修正後，訓練完成的控制器效能比較：

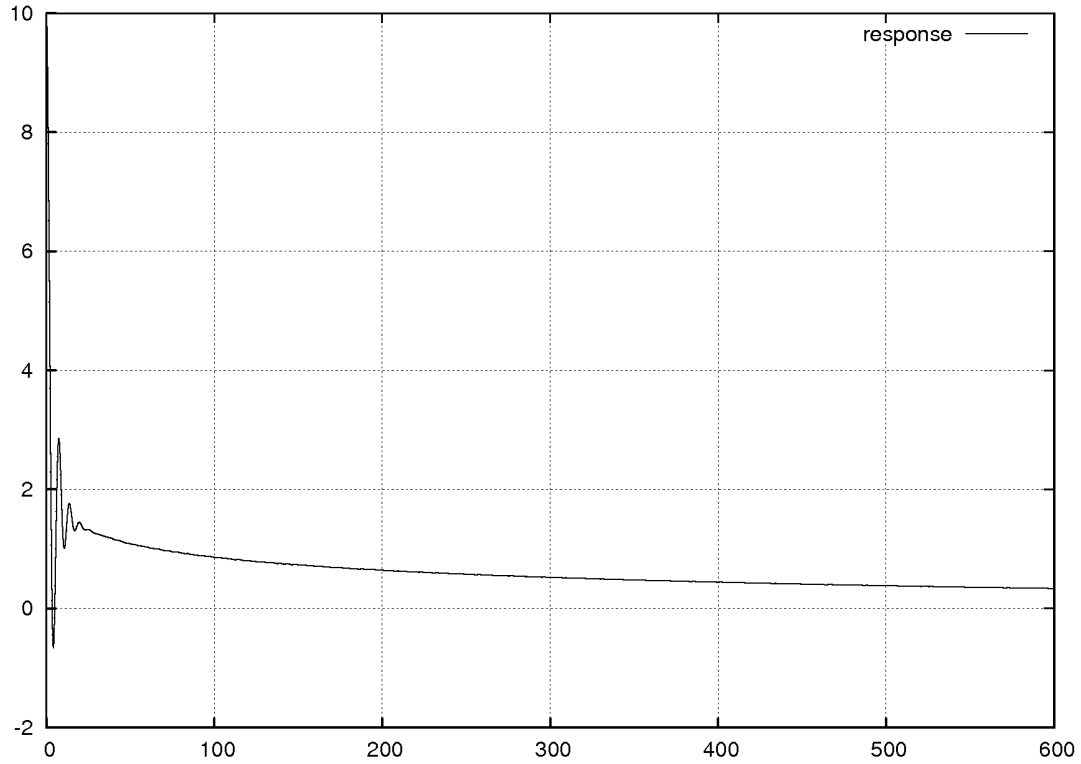


4.2 非線性系統

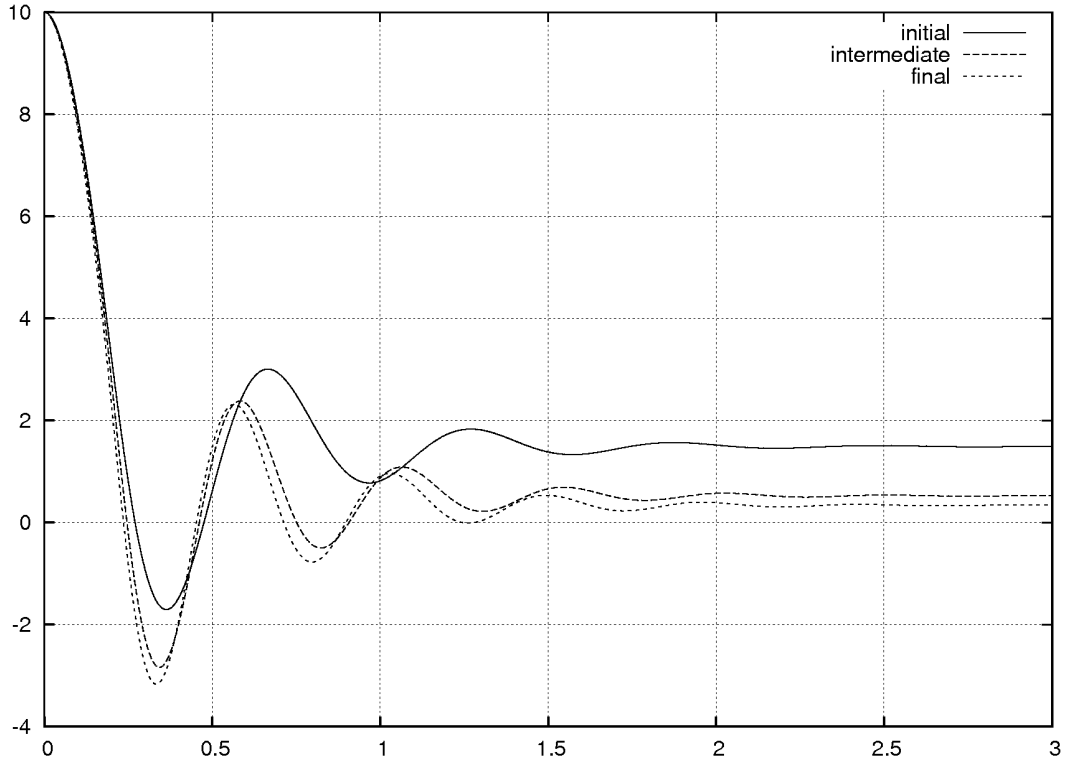
以下是以一個實際的非線性系統來測試我們的系統，此非線性系統可寫成以下的狀態空間式：

$$X_1 = X_2$$
$$X_2 = -9.8\sin(X_2) - 0.5X_1 + 0.5u$$
$$Y = X_1$$

系統的初始狀態兩個變數均為零，自動偵測而得的參數，訓練過程可見下圖：



同樣的也可以將誤差逐漸縮小趨近於零。在訓練完成時，總共產生了 11 條規則，下圖是比較訓練過程中控制的進步程度：



4.3 實驗討論

由以上兩個實驗可以了解到，偵測參數的方法能找到一份可以使用的參數，而強化式學習具備了將這些參數最佳化的能力，同時群聚方法能使規則具備足夠的數量。

整套系統可以在需要最少的專家知識下運作，能夠自動偵測參數並做最佳化，與以往不同的是，具有處理連續狀態及動作的能力。

第五章 結論與未來展望

5.1 結論

強化式學習具有監督式或非監督式學習所不及的優點，能夠以全面性的觀點來處理問題，兼具學習及決策的能力。然而在處理連續空間及連續動作的問題上，仍仰賴專家知識給予指導，在控制問題上，常常無法得到這些專家知識或不符成本。

本文提出了新的修正，改善既有的強化式學習演算法，賦予自動偵測所需知識的能力，增加了強化式學習在控制上的應用性。實驗也證實，系統確實可以自行找到一組尚可的初始值，避免系統掉入錯誤的局部最佳化中，而能找到比較符合需求的最佳化解，所付出的成本微乎其微。

5.2 未來展望

本系統所使用的自動參數找尋技巧是來自於已有的控制研究，這使得本系統的應用範圍也被迫局限於控制系統，將原有的參數尋找技巧泛用化是增加本系統應用度的必要工作。

此外強化式學習本身也有必須要解決的問題，獎賞函式的設定目前均只用很直覺的觀念設計，但獎賞函式影響甚鉅，甚至超過初始值

的影響，找到設計獎賞函式的準則也是強化式學習改進必須要做的事。

第六章 參考文獻

[1] K. J. Astrm and T. Hgglund, Automatic tuning of PID controllers. Research Triangle Park, NC : Instrument Society of America, 1988.

[2] C.-T.Lin and C.S.G.Lee, “Reinforcement structure/parameter learning for neural-networkbased fuzzy logic control systems,” IEEE Transaction Fuzzy System, vol. 2, no. 1, pp. 46－63, Feb. 1994.

[3] X. Dai, C.-K. Li, and A. B. Rad, “An approach to tune fuzzy controllers based on reinforcement learning for autonomous vehicle control,” IEEE Transactions on Intelligent Transportation Systems, vol. 6, no. 3, pp.285－293, Sept. 2005.

[4] M. J. Er and C. Deng, “Online tuning of fuzzy inference systems using dynamic fuzzy q-learning,” IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 34, no. 3, pp. 1478－1489, Jun.2004.

[5] K.-S. Hwang and H.-J. Chao, “Adaptive reinforcement learning system for linearization control,” IEEE Transaction on Industrial Electronics, vol. 47, no. 5, pp.1185－1188, Oct. 2000.

[6] L. Jouffe, “Fuzzy inference system learning by reinforcement methods,” IEEE Transactions on Systems,Man, and Cybernetics, Part C: Applications and Reviews,vol. 28, no. 3, pp. 338－355, Aug. 1998.

[7] C.-F. Juang, “Combination of online clustering and qvalue based ga for reinforcement fuzzy system design,” IEEE Transactions on Fuzzy Systems, vol. 13, no. 3, pp.289－302, Jun. 2005.

[8] C.-T. Lin, Neural fuzzy control systems with structure and parameter learning. Singapore ; River Edge, NJ : World Scientific, 1994.

[9] A. G. B. Richard S. Sutton and R.J.Williams, “Reinforcement learning is direct adaptive optimal control,” IEEE Control System Magzine, vol. 12, no. 2, pp. 19－22, Apr. 1992.

[10] P. S. Shimon Whiteson, Matthew E. Taylor, “Empirical studies in action selection with reinforcement learning,” Adaptive Behavior, vol. 15, no. 1, 2007.

[11] R. S. Sutton, “Learning to predict by the methods of temporal differences,” Machine Learning, vol. 3, no. 1,pp. 9－44, 1988.

[12] R. S. Sutton and A. G. Barto, Reinforcement learning : an introduction. Cambridge, Mass.: MIT Press, 1998.
[13] K.-S. H. S.-W. Tan and M.-C. Tsai, “Reinforcement learning to adaptive control of nonlinear systems,” IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 33, no. 3, pp. 514–521, Jun.2003.
[14] L.-X. Wang, “Stable adaptive fuzzy control of nonlinear systems,” IEEE Transactions on Fuzzy Systems, vol. 1, no. 2, pp. 146–155, May 1993.
[15] L.-X. Wang,, A course in fuzzy systems and control. Upper Saddle River, N.J. : Prentice Hall PTR, 1997.