國立中興大學資訊科學與工程學系 （附件一）

資訊專題競賽報告

**智慧風扇**

AI FAN

專題題目說明、價值與貢獻自評（限100字內）：

|  |
| --- |
| 市面上的電扇大多只能手動調整風速大小，但若於不方便調整風速大小的環境下(如睡著)會相當不方便。因此我們藉由qlearning學習使用者可能的風速需求，設計一個自動化改變風速的智慧風扇。 |

專題隊員：

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | E-mail | 負責項目說明 | 專題內貢獻度(%) |
| 劉之岳 | jeff87905@gmail.com | 程式設計、概念構想 | 50% |
| 劉易鑫 | hahappast@gmail.com | 硬體組裝、實驗 | 50% |
|  |  |  |  |

【說明】上述表格之專題內貢獻度累計需等於100％。

指導教授簡述及簡評：

|  |
| --- |
|  |
| 指導教授簽名： |

中華民國 年 月 日

目錄

1. 摘要（含關鍵詞）
2. 專題研究動機與目的
3. 專題重要貢獻
4. 團隊合作方式
5. 設計原理、研究方法與步驟
6. 系統實現與實驗
7. 效能評估與成果
8. 結論
9. 參考文獻
10. 摘要

是不是常常爬起棉被開關電風扇呢? 明明剛開電風扇時相當舒服，但吹了一陣子後一陣涼意襲來，又不得已把棉被闔上。無奈的是，闔上棉被後卻又覺得悶熱煩躁不安，難以入眠。

每年5、6月是台灣的梅雨季，溫度變化大，一下子艷陽高照、悶熱異常，一下子又烏雲密布、寒風刺骨。每到這個時節，流行性感冒最容易趁虛而入。而在這種季節交替的環境下，人們往往在入夜時打開電風扇吹著強風，尤其是兒童，貪婪著一時的舒爽，卻沒想到在凌晨時，一場雷陣雨，帶走地表上的高溫，因而受了風寒。

人體生理機制與體溫的調節需要時間來因應環境溫度的變化，因此如果外部環境溫度變化過大，身體溫度調節機制反應不及或反應過度時，即可能對身體造成傷害或致命。

為了解決上述的情況，我們使用Q-learning設計了一個智慧型風扇，在熱時自動吹強，在冷時自動吹弱，解決睡眠上的困擾。

關鍵詞:Q-learning，智慧電風扇，個人化，藍芽

1. 專題研究動機與目的

每個家庭多多少少都會有老人、嬰兒的存在，老人們的行動不便、嬰兒對外界的感知能力尚未成熟…….等，對於短時間內動盪的環境溫度，無法像一般身強體壯的人們一樣做出最快的反應。就算聘請看護，由於每個人對於舒適溫度，感知上存在著根本差異，無法藉由人力上解決社會上的問題。

而市面上的電風扇大多只能手動控制風速，雖有些有遙控器可遠距改變電風扇轉速，但於睡覺時仍有困擾與不便。

此時便產生初步的構想，若電風扇能自動根據你的心聲改變風速的話那該有多好啊! 因此我們變設計出一個可自動根據你的需求改變風速的電風扇，並可適應多個不同的環境，讓你在不同的季節、地點皆可獲得最舒適的感覺。

三、專題重要貢獻

我們專題的研究成果可用於社會的普羅大眾，睡著後半夜溫度驟降，風扇會感測到溫度，調整大小，減少感冒機會。而由此種構想，可以進一步發展出將每個人對於

舒適溫度，及電風扇吹風後的溫度變化存進資料庫中。可以使多個使用者在登入帳號後，能調用過去資料，在最短時間內，抵達舒適溫度。

而對於醫療的貢獻上，無論是在醫院的病房中，又或是在家不便行動患者。病患若無法下床或按遙控器調整風扇開關，可藉由此風扇自動感知周遭環境變數，避免病人近一步的惡化、減少醫療資源的浪費。

由電子機械類切入，若功能強大化後，可用Q-learning系統對機房的電量及溫度控制，達到一個最佳化的結果。

四、團隊合作方式

前期:

合作想出大架構，根據架構分成硬、軟體。兩人分別實驗不同元件並共同測試元件能不能符合程式的需求。

後期:

共同撰寫演算法模型及實驗方式，一人負責程式設計的架構及流程圖，另一人負責將概念架構實作化。

共同實驗風扇與環境溫度感測器間的互動關係、觀察、解決問題。

五、設計原理研究方法與步驟

大綱:

分為主、從端。主端搭載感測器並根據感測器資料進行qlearning，qlearning完畢後即完成qtable。根據qtable選出最適合的風速結果給從端。從端搭載風扇，接收主端傳送之風速，然後吹送，完成智慧風扇的功能。

步驟1

將DHT11感測器裝置於Arduino主端

原理: 使用感測器監測環境變數。

研究方法: 1.使用體感溫度量測法替代傳統溫度感測貼片

2.利用dht11、風速計感測器測量環境變數轉化為體感溫度

步驟2

於Arduino主端機器學習

原理: Q-learning

研究方法:根據Q-learning 演算法:Qnew[st,at]=(1-α)\* Q [st,at]+ α\*(rt+**Υ**\*maxQ(st+1,a))

步驟3

Arduino 主端與Arduino從端連線

原理:藍芽模組HC06主端根據qtable傳送轉動作(即弱、中、強)給藍芽模組HC06從端執行風扇轉速。

架構圖如下圖所示

Arduino從端

Arduino主端

藍芽互傳

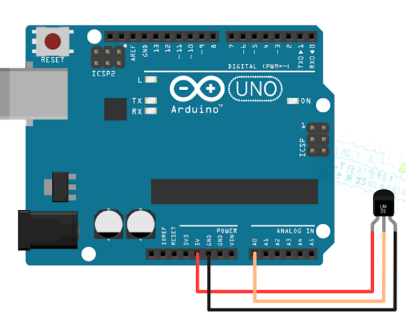
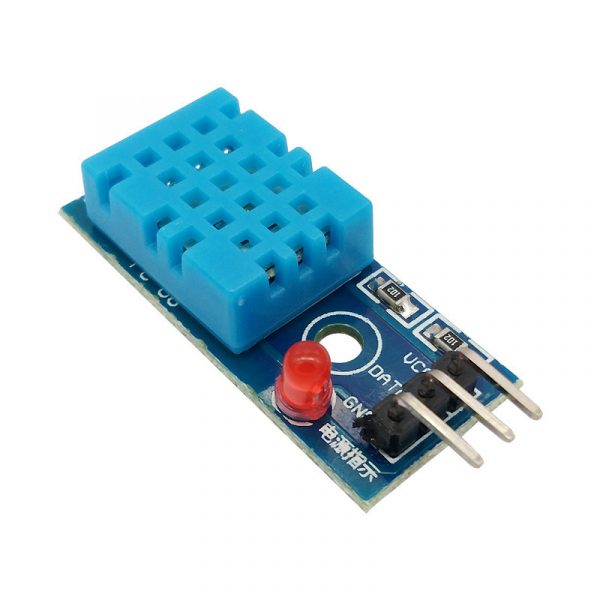
Q learning

溫度感測器

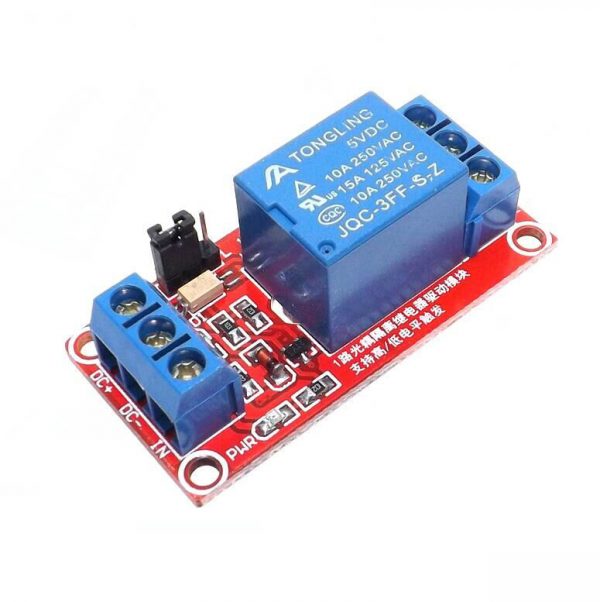
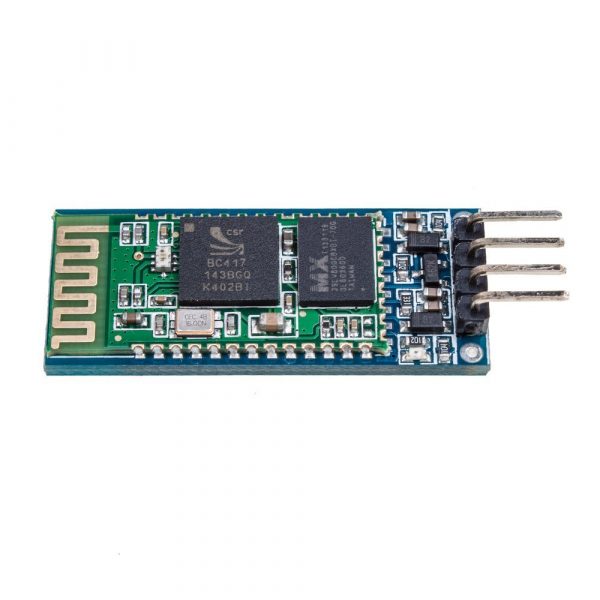
風扇

六、系統實現與實驗

實驗材料:

Arduino UNO \*2 DHT11\*1  

Relay\*1 bluetooth\*2(一為主端，另一個為從端)

Cpu風扇\*1 風速計\*1

系統實現:

1.

體感溫度演算法:

AT=1.07T+0.2e-0.65V-2.7

e=(RH/100)\*6.105\*exp(17.27T/(237.7+T))

其中AT為體感溫度（°C）、T為氣溫（°C）、e為水氣壓（hPa）、

V為風速（m/sec）、RH為相對濕度（%）

氣溫:dht11感測

水氣壓:dht11感測

風速:風速計感測，其中弱風4 (m/s),中風4.4(m/s),強風5(m/s)

2.

設定舒適初始溫度:藉由使用者按按鈕實現。其中含有冷鍵(風扇風速下降)、熱鍵(風扇風速上升)、即”確定鍵”、RESET鍵(可於環境不同使用)

3.

建立reward\_table: 不同溫度下操作不同的動作(弱、中、強)，給予相對應的獎勵，即越接近舒適溫度，其獎勵值越高。建立delta\_table 紀錄每個溫度操作不同動作後，會到達的溫度值。

4.

建立qtable的方法:

實作Q-learning 演算法:Qnew[st,at]=(1-α)\* Q[st,at]+ α\*(rt+**Υ**\*maxQ(st+1,a)):

其中Q為q table、 s為狀態、a為動作、r為獎勵值、α為學習率(0<=α<=1)**,Υ**為衰退值(0<=**Υ**<=1)。

在epsilon greedy=0.9情況下操作Q-learning

即有0.9的機率操作Q-learning 另外0.1的機率使用argument\_max的方法

Q-learning: qtable的更新方法為少量學習過去的舊資料以及預測大量預測未來的資料。

其中未來資料的預測包含3. reward\_table,而在maxQ(st+1,a)中下一步state使用隨機變數預測。

Arugement\_max: 在maxQ(st+1,a)中下一步state使用當前qtable最大值。

將qtable值寫入eeprom，方便斷電後的下次使用。

5.

建立智慧風扇:

透過4.完成的qtable 找出最短循環之風扇動作

使用者溫度初值(state) 其對應之qtable有3個action(分別為弱、中、強)將其中最大的qtable[state][action]值找出，並根據此動作藉由delta table delta[state][action]跳往下一個溫度，不斷循環。

實驗:

實驗1

實驗前:使用者根據感覺不停切換風扇轉速。

實驗過程:發現人體體感溫度變化須持續一段時間後趨於穩定而非瞬間值。

解決方案:

Step1:

使用者於初始設定溫度時按不同的按鈕(太冷鍵、太熱鍵、確定鍵)

Step2:

使用者需吹10秒相同轉速的風扇才可按確定鍵。

實驗2

實驗現象:

Step1:

實測使用者按確定鍵的前一步動作。

Step2:

觀察reward\_table

Step3:觀察實驗結果，結果如下:

當使用者按”確定鍵”前一步動作選項為”弱”時，整個reward\_table有值的部分偏向於

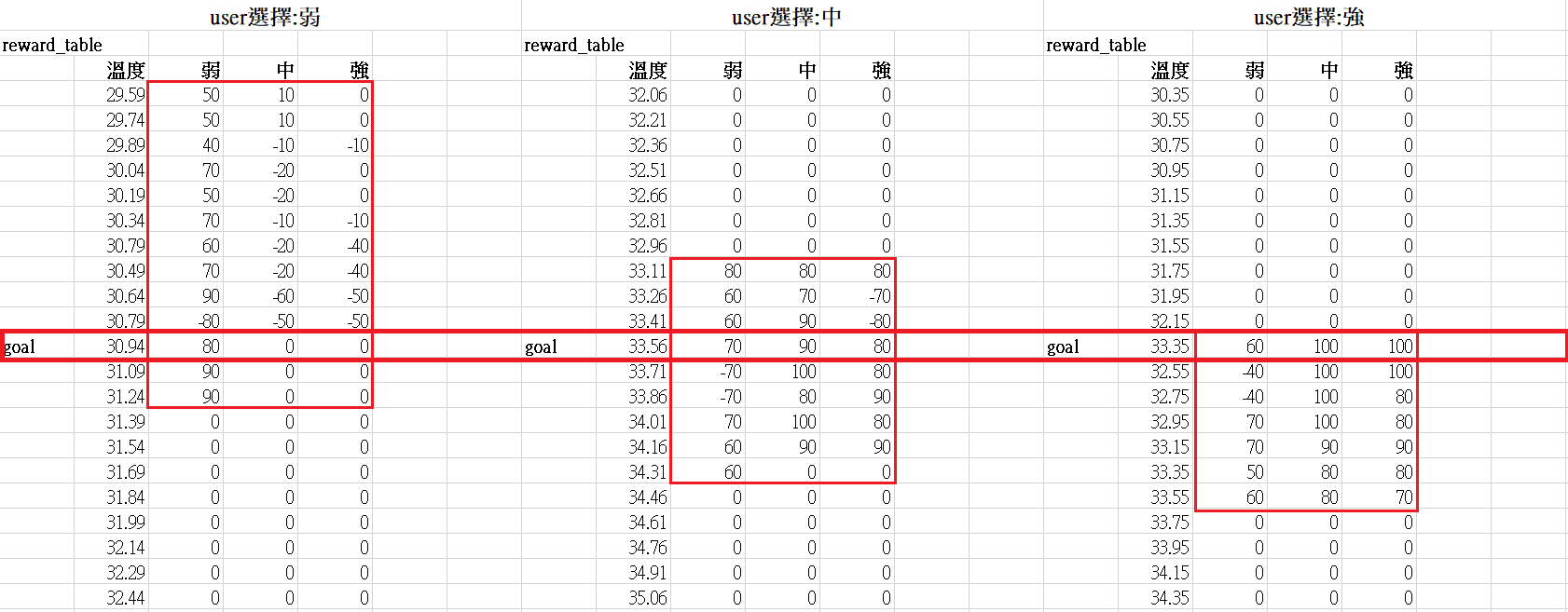
Goal溫度值的上方

當使用者按”確定鍵”前一步動作選項為”中”時，整個reward\_table有值的部分較為均勻

當使用者按”確定鍵”前一步動作選項為”強”時，整個reward\_table有值的部分偏向於

Goal溫度值的下方

實驗2圖表:



實驗解讀:

使用者若按”確定鍵”前一步為"弱"，則使用者在該環境下偏好習慣為弱風。

使用者若按”確定鍵”前一步為"強"，則使用者在該環境下偏好習慣為強風。

實驗3.

實驗前:

設定reward\_table初值，規則如下:

step1.

若使用者在目前狀態(當下溫度)下，操作完某一動作(電風扇轉速:弱、中、強)後， 而新的狀態接近目標狀態(即舒適溫度)，則該事件我們給予”正數分數”的獎勵。

Reward正數實施辦法:若目前狀態，操作該動作後，至目標狀態則給予滿分(100分)的獎勵，每距離目標狀態1格距離，則依次遞減10分。

Reward正數意義:在同一個狀態下，分數越高者，代表在該狀態下實施分數高者之動作

，為最靠近舒適狀態的選擇。

step2.

若使用者在目前狀態下，操作完某一動作，而新的狀態卻遠離目標狀態，則該事件我們給予”負數分數”的懲罰。

Reward負數分數實施辦法:將所有負數分數訂為-1。

實驗過程:

發現由實驗前step2所建立的負數獎勵模型，會導致qlearning階段時，若發生以下事件，則無法作出最好的動作選擇。

事件:

在某一狀態下，執行3個動作(弱、中、強)，皆會遠離目標狀態(舒適溫度)。

解決方法:

重新制定新的”負數分數”懲罰制度。

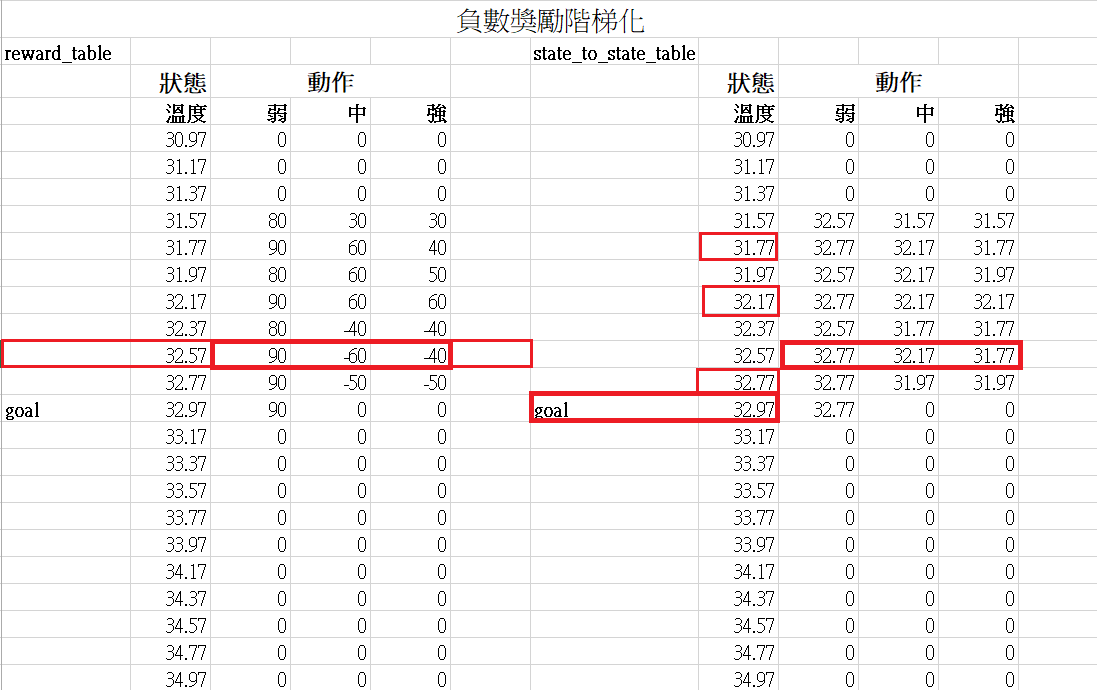
新的Reward負數分數實施辦法: 若目前狀態，操作該動作後，遠離目標狀態1格，

則給予-10分。依次遞減，最低-100分。

實驗3圖表:



表一(實驗前) 註: state\_to\_state table 中的值表示為狀態值，即在該狀態下操作某動作會到達的”下一個”狀態值



表二(實驗後)

實驗4.

實驗前:

設定reward\_table:

Step1:

設定reward\_table時一併更新state\_to\_state table

Step2:

每收到一個新的狀態(溫度)，則根據該狀態操作某動作(電風扇風速:弱、中、強)時，則

reward\_table、state\_to\_state table對應的表格會不停更新。

Step3:

將reward\_table的獎勵值定為實驗3之實驗結果。

實驗過程:

發現若使用Step1之方法會發生以下事件:

事件:

出現某一個狀態，在q\_table中對應3個動作皆無值。

事件意義:

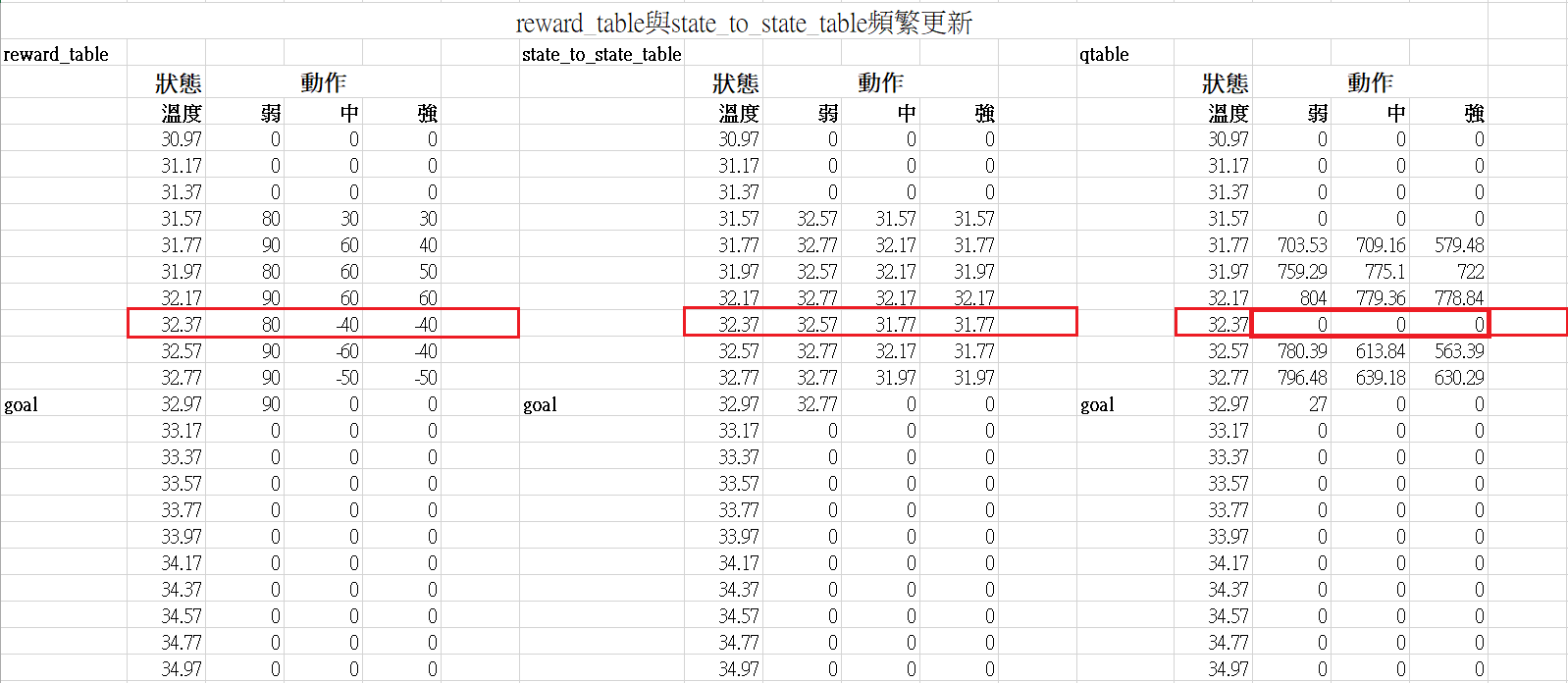
1.

表示某一個狀態曾經可以經由另一狀態到達，可是因為reward\_table與state\_to\_state table不停在更新的，而qtable的判斷依據又是根據reward\_table 與state\_to\_state table。

2

所以當設定完reward\_table與state\_to\_state table時，會出現某個狀態曾經被聯繫過，但因為不停的更新，所以在最後一次更新時剛好與所有曾經聯繫過的狀態的斷訊，導致qtable無法由reward\_table和state\_to\_state table得知該”失聯”的狀態應該賦予何種值。

實驗4圖表:



從實驗四表格可以發現狀態(32.37)度無法由其他狀態藉由不同的動作抵達，而其周圍的狀態皆有值，單獨跳過該狀態所造成的奇特現象。

解決方案:

reward\_table與state\_to\_state table中每個值在設定時，只會更新一次。

實驗5.

實驗前:

根據實驗四的結論將reward\_table與state\_to\_state table中每個值在設定時，皆只會更新一次。

實驗過程:

發現若使用實驗四之結果會發生以下事件:

事件:

若某個狀態下3個動作皆為負值，則有可能某些狀態永遠無法被抵達。

舉例:

假設狀態(29度)為舒適溫度

Step1:

若狀態(28.5度)下操作動作(弱、中、強)而reward\_table給予狀態(28.5)度的3個動作皆為負值，則代表該狀態經由所有的動作皆會到達28.5度以下。

Step2:

假設該狀態(28.5度)經過動作弱到達新狀態(28.4度)，則由於先前的判斷制度，若新狀態(28.4度)經過3個動作無法大於28.5度，則會導致操作qlearning時，只要一進入狀態(28.5度)或是狀態(28.4度)會永遠卡死，不停的循環。

解決方案:

將設定reward\_table與state\_to\_state table的機制更改

新的方法:某一個狀態(溫度)下，若操作所有動作(弱、中、強)後，reward\_table所有皆為負值時，可以接受更新值並一併更改state\_to\_state table。

**七效能評估與成果**

效能評估:

1.

實驗測試: **Υ**(耗減率)=0.2, **Υ**=0.9

原理:

由公式中:Qnew[st,at]=(1-α)\* Q[st,at]+ α\*(rt+**Υ**\*maxQ(st+1,a)).

特別比較公式中**Υ**與reward的迭代可發現Q(s1)=r2+**Υ**\*Q(s2)=r2+**Υ**[r3+Υ\*Q(s3)]=…

整理:Q(s1)=r2+**Υ**\*r3+(**Υ**)^2 \*r4+(**Υ**)^3\*r5

數學意義:當**Υ**數值越大時，智慧型代理人便更加重視未來獲得的長期獎勵，

**Υ** 數值越小時，智慧代理人便更加短視近利，只在乎目前可獲得的獎勵。

實驗過程:

Step1:

固定相同的reward\_table與state\_to\_state table

Step2:

觀察gamma=0.2與gamma=0.9時qtable內的值。

實驗1圖表:

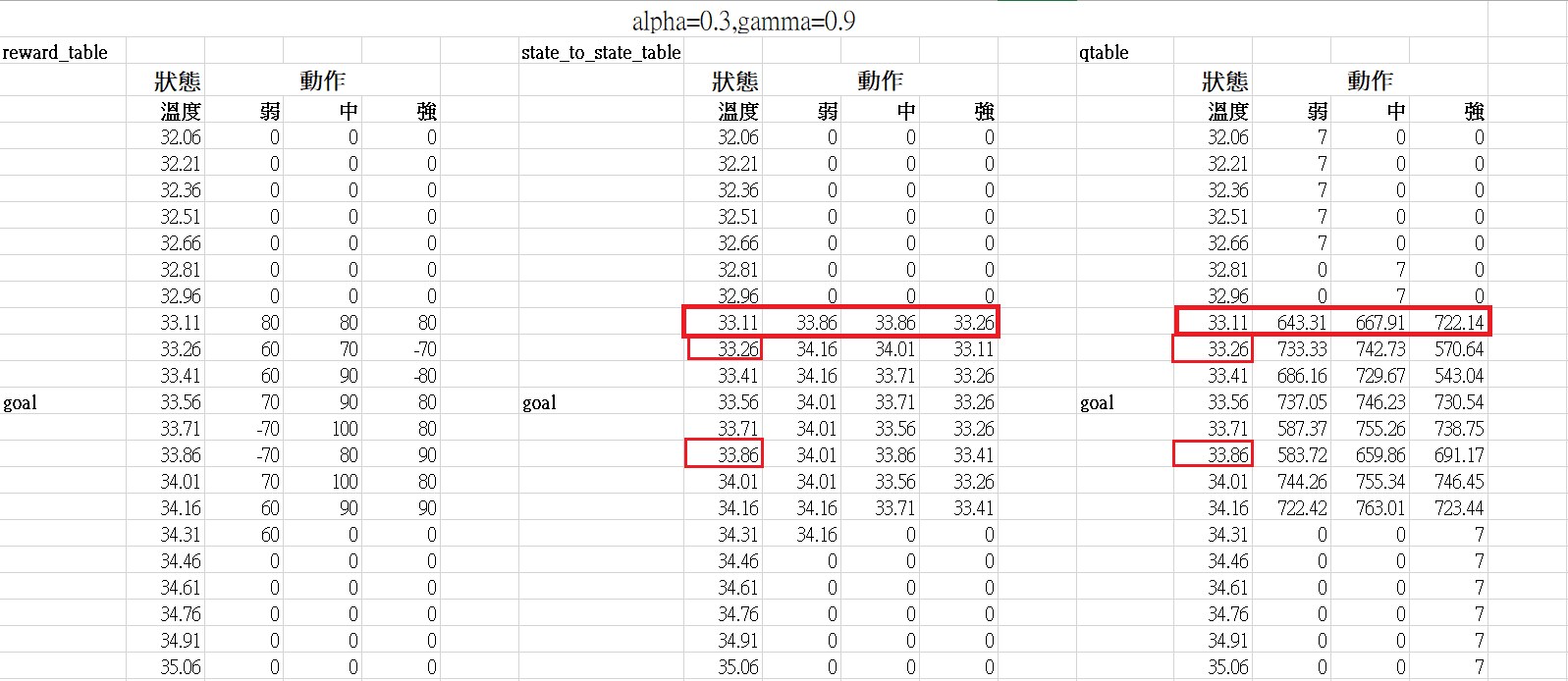


表1

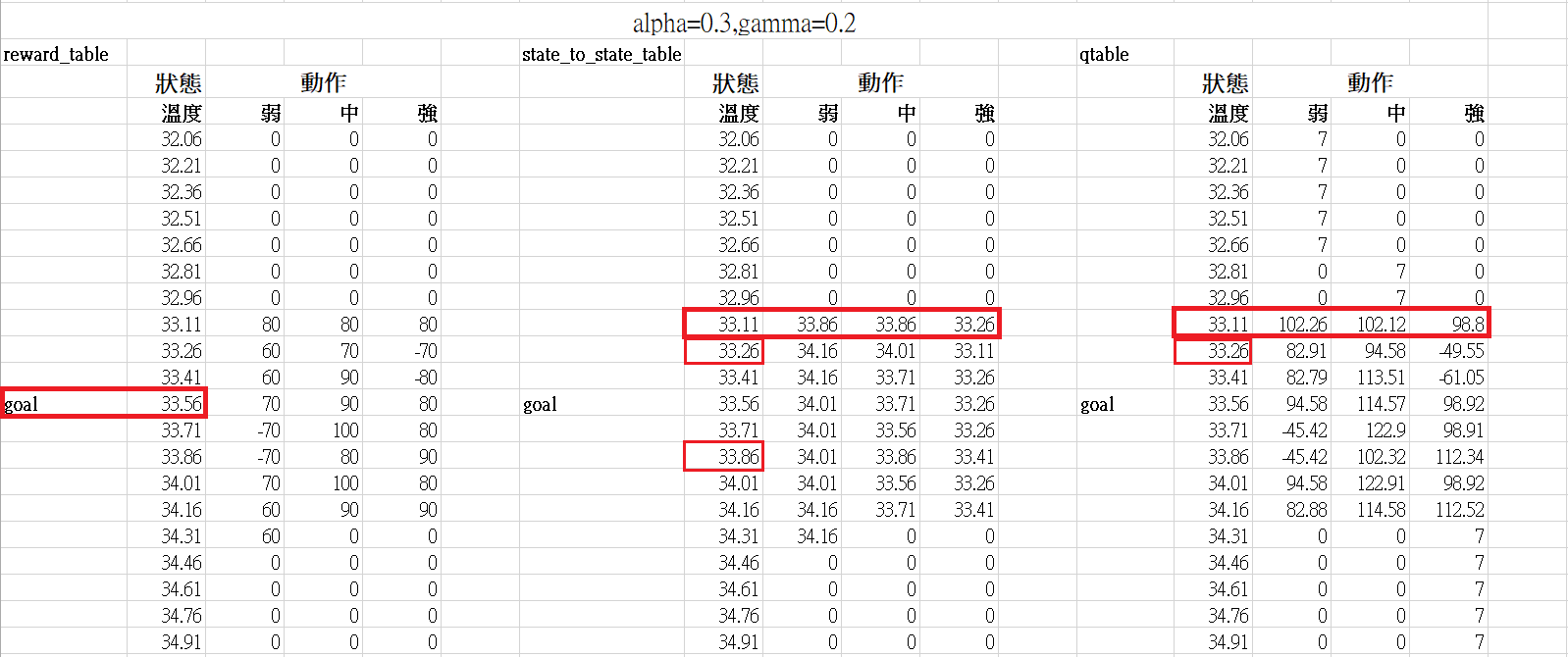


表2

實驗結果:

比較表1與表2:

於狀態(33.11度)，**Υ**=0.2 在智慧風扇階段下選擇為弱; **Υ**=0.9 在智慧風扇階段下選擇為強

比較過程:

Step1:

選定相同溫度比較，表一與表二的選擇為狀態(33.11度)

Step2:

表一在狀態(33.11度)中，選擇qtable分數最高的值，所以表一的第一個選擇為”強”，

觀察state\_to\_state table中狀態(33.11度)選擇動作(“強”)，會到達新的狀態(33.26度)。

Step3:

持續迭代Step2直到到達目標狀態為止。

比較結果:

狀態(33.11度)的第一個動作選擇”強”，會擁有最短路徑，因此效能評比: **Υ**=0.9>**Υ**=0.2

數字分析:

根據上圖qtable，同一狀態的3個動作中若出現相近的數字，表示自該狀態到達目標狀態的步數有高機率相同。

實驗總結:

由於在複雜的state\_to\_state table中無法輕易看出由任意狀態該選擇何種動作，才能以最短路徑到達目標狀態。

因此設計qlearning 模型將qtable根據reward\_table的值與state\_to\_state table的值不斷更新。

最後，在智慧風扇的階段，則可以根據qtable的值判斷出最短路徑到達目標狀態，並且不斷循環。

八、結論

自20世紀中後期，嵌入式系統就已漸漸踏入人們的生活，小至觸控式電燈、電子錶，大至汽車、冷氣其背後都有嵌入式系統的影子。時至今日，人工智慧、大數據，蔚為主流的年代，2016年電腦圍棋軟體AlphaGo 挑戰韓國棋王李世石藉由人工智慧預判棋路。因此，利用演算法預測未來可能發生的事件並非不可能的任務。

本研究將嵌入式系統結合人工智慧，使產品具有智慧效能管理。在研究過程中發現人類舒適溫度其對應結果應為舒適溫度，包含許多不同的環境變數。而Q-learning模型可以有效率的實現自我學習並分析數據，藉由表格將體感溫度最佳路徑循環具象化。

九、參考文獻

 Francisco S. Melo, ["Convergence of Q-learning: a simple proof"](http://users.isr.ist.utl.pt/~mtjspaan/readingGroup/ProofQlearning.pdf)Robert G. Steadman , 1984, J Clim Appl Meteorol 23:1674–1687.

Hasselt, Hado van. Reinforcement Learning in Continuous State and Action Spaces. Wiering, Marco; Otterlo, Martijn van. [Reinforcement Learning: State-of-the-Art](https://books.google.com/books?id=YPjNuvrJR0MC). Springer Science & Business Media. 5 March 2012: 207–251.