

機器人基礎動作模仿學習之探討與實作

張人祐
國立中興大學資工所研究生

S9256028@mail.cs.nchu.edu.tw

林煜寰
國立中興大學
資工系學生

雷啟元
國立中興大學
資工系學生

洪國寶
國立中興大學資工系教授

gbhomg@cs.nchu.edu.tw

摘要

為了找出取代人力的方法，全世界已經越來越注重機器人這門機電整合的高技術域，目前已經出現許多能夠完成動作指令的機器人，但在指令下達的同時，伴隨著只是既有的經驗法則、固定的行為模式而已，當目標機器人必須更改目標工作時，帶來的往往是高成本的程式重寫、晶片重燒甚至到整個機體重新設定。如果能讓機器人藉由人工智慧的學習方式去瞭解新動作，而不是重新改寫程式的方法，再維護機器人時將會省下很多成本。本文提出一個新的動作模仿學習系統，此篇探討的架構主要建構在後端應用上，不需要探討到底層的動力學方程式、機電原理，討論方向將會著重在動作分析與自我學習的開發。

關鍵詞： 機器人，倒傳遞類神經，動作模仿，自我學習。

Abstract

In order to find a way to substitute manpower, many research labs have focused on the high-tech robot domain for many years. Nowadays, there are many robots with the ability to complete the motion by giving them an order that accompany with existed experiential rules and regular behavioral patterns. But in this way, when a robot has to change its function, there always comes the inefficiency incurred by reprogramming, chip redesigning and even resetting the robot. If the robot can understand the new movement by learning through A.I., instead of reprogramming, it will save lots of costs. In this paper, we discuss and implement the issues of a new motion imitating system. It focuses on the concept of motion analyzing and self-learning without involving the theories of kinetics and electrics.

Keywords: robot, self-learning, neural network.

1. 前言

行為模仿[1][7]的目的可以有效的降低維護機器人的成本、開發系統的高額費用。在一些高危險或者環境變數複雜的狀態下，行為學習的能力可以讓機器人具有自主應變的能力，甚至演變為機器人保護自我的能力，而不需額外花費大量人力去遙控操作。

如何讓個體去記取觀察模仿對象所做而個體從來未曾做過或不會作的某特定行為，在人工智慧上是一大議題。由於機器人本身就是設計要與外在環境的互動，於是產生了利用外在環境輔助定位並且定位物體本身就是機器人動作的想法，例如[3]讓示範者將鐘擺由靜止狀態擺動到一定的程度，再由一機器手臂模仿示範者的行為，將靜止的鐘擺擺動到一定程度。主要的實作方法是，將示範者的鐘擺軌跡紀錄下來，運動路徑以及鐘擺改變的角度將會決定擺動的方式。機器手臂藉由模擬與示範者相同的起始位置、擺動方式以及擺動到相同的角度來完成模仿的動作。

為了更加強機器人的自主判定能力，辨識形狀資訊[8]可以讓機器人學習特殊形狀的東西方便作於各部位的定位，[6]利用模仿學習的方式教導智慧型機器人分辨食物，並使用夾具將其夾起，在放置到適當的地點。所使用的方法就是讓機器人一次一次的辨識所要夾取物體的形狀資訊，並且不斷的從錯誤中學習。例如：要夾取三明治以及PIZZA，首先要先讓機器人知道三明治以及PIZZA的外型，三明治是三角形，PIZZA是圓形，取出圓形以及三角形的特徵給機器人去辨認。若是判斷錯誤則不斷的修正改進，已求在最後能夠達到確實的分辨食物的方法。

另外在單純運動學方面有利用軌跡控制與路徑規劃的原理測量出，機器人模仿的運動路徑，[4]智慧型機器人模仿學習人類揮動手的軌跡；模仿軌跡的方法，我們必須取出揮動手的起始位置以及結束的位置，以及整個運動的路徑，在路徑中取出樣本點，算出樣本點與樣本點之間的間格，角度等空間上的差異，機器人就可以藉由從起始位置，一一通過各個樣

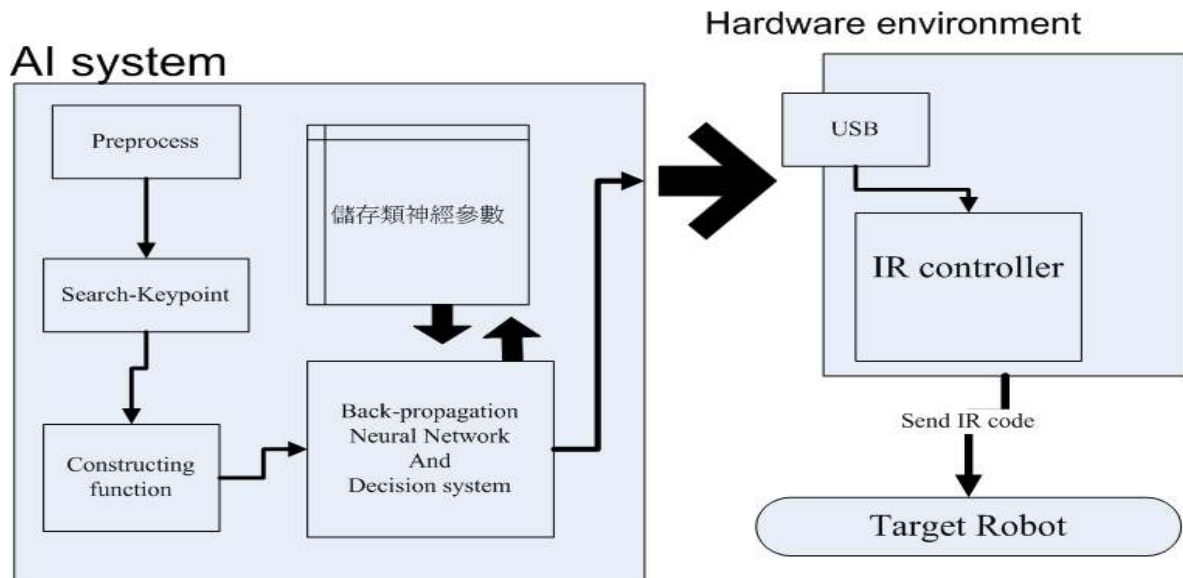


Fig.1. 系統架構圖

本點，最後到達終點的方式，這樣就可以做出相同軌跡的動作。運動軌跡的運算可以有效的提供準確行徑方法。

由於此類機器人在更改機器人動作後，運動學方程式和動力學模型都必須重新透過使用者修正。為了讓機器人更偏向人類思考方向，於是產生了標記運動部位的方法[2]，如何取得模仿目標身上的關鍵點。先在模仿目標上做特殊的標記，此特殊的標記包含兩種顏色（紅色和黃色），分別在目標的四肢上，左手，右手和左腳上貼上紅色作為標記，在身體中間和右腳貼上黃色作為標記，共五個位置。利用機器人頭上的攝影機去擷取 320*240 像素的影像。將 RGB 色彩模型轉換成 YUV 色彩模型。偵測目標物身上的五個標記，並個別計算其重心點。根據人體身上的標記的相對位置，辨識出五個關鍵點（身體，左手，右手，左腳和右腳）。利用身體的關鍵點，進而去求得其他的兩個額外的關鍵點（左肩和右肩）。在定位出四肢以後，可透過相對運動位置的關聯法則去加強動作判定的支持度。

特殊的標記可加速辨識，但在複雜環境或者外在變因多的情況下會嚴重影響系統準確度。於是我們產生了直接辨識各運動部位的方法，[5]讓智慧型機器人模仿人類四肢的動作；首先必須要明確的給定四肢的樣本，在人類的四肢以及身體的某些部分上面取出樣本點，辨識人類動作的方式就是以各個樣本點的相對位置來認定，例如我們有右手的樣本點以及右肩膀的樣本點，當機器人辨識出來，右手的樣本點在右肩的上方，那就可以做出舉手的

動作，如果右手的樣本點在右肩的旁邊，那就是將右手水平的張開。其他四肢部分依此類推。如此就可以模仿人類四肢的動作了。四肢的個別辨識將隨著被模仿目標的四肢特徵所更改，為了能夠讓系統更加的可攜性，這裡我們將提出一套新的模仿概念，不需計算動力學問題，也不用特殊標記點方法辨認出運動部位，而是嘗試使用類似人類思維方式[10]在影像上簡化成動作架構圖解決動作模擬的判定，在系統實驗上會先以機器人模仿機器人為，將在以下討論之。

2. 系統架構

在傳統行為模擬上，會先把影像切割成手、頭、腳身體各重要部位，做好辨識，讓機器人了解手的定位在哪，腳的定位在哪來做出相對應動作。或者用各種環境感測器訂立出世界座標系下去分析，有點類似定位系統。在這篇的方法裡，我們嘗試著導向一個非傳統的思維，一個類似於仿生的概念，我們不需要特別偵測手或者腳部位的變化，而是讓機器人去看懂整體的動作架構，就像人類看同類的動作一樣。在實驗中將機器人分成五個動作分別為站立、左轉、右轉、後仰、彎腰等學習目標。

以下將介紹系統的流程與演算法設計（如圖一），並提出不同於以往的特徵點萃取法。我們在設計系統上，大致分了兩大區塊，一為我們的軟體決策系統與演算法設計，我們稱之為 AI system，分別為

- 1) Preprocess function: 將要處理的圖片做去背景、框出被模仿機器人的大概範圍。

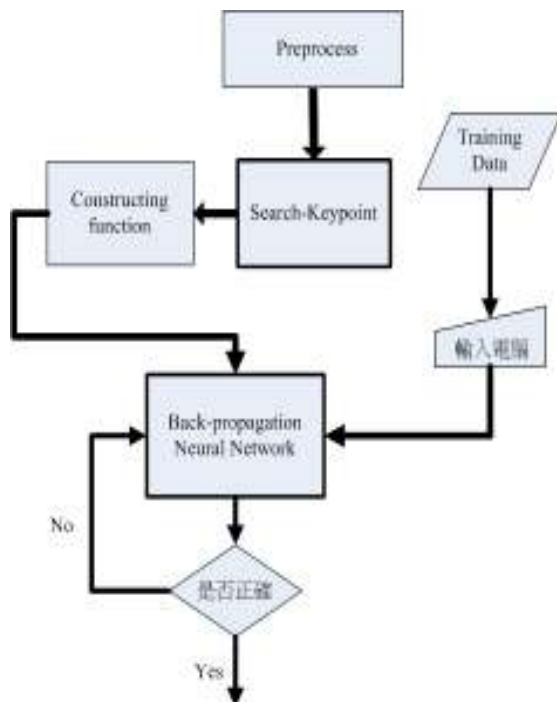


Fig. 2. 系統流程圖 (AI system)

2) Search key-point function: 找出動作機器人的特徵點。

3) Linking function: 找出一個新的動作簡化圖, 可以有效的傳達欲辨識物體的資訊。

4) 倒傳遞類神經系統: 由於倒傳遞具有隱藏層特性, 且有自我學習效果, 因此讓我們採用此種網路架構做為最終決策系統。

另一部份在於指令如何在硬體環境的傳遞和轉換並且成功命令機器人達成欲模仿的動作。我們利用了紅外線, USB 等傳輸設備來實作, 在稍後段落將會詳細說明所需的設備, 還有實驗的環境。

系統流程

在系統流程上 (Fig. 2), 我們將圖片依序導入不同的 function, Data 在 function 與 function 的傳送之間是採用轉換後一維的整數矩陣, 陣列數值只會有 -1 (代表雜質)、2 (萃取後特徵點)、1 (機器人特徵點)、3 (最後處理架構圖的特徵資訊), 如此可大量降低運算量和複雜度, 並藉由除存在電腦上各種被模仿動作的訓練資料, 不斷的訓練類神經的參數直到近似收斂為止, 在這邊將採用非批次的更新權重值方法, 可更精準的學習每次的錯誤更新。

Preprocess:

由於此篇探討的重點不在於影像處理的

技術, 所以我們在實作上先採用了單純背景下去作拍攝動作, 可加速我們獲得整體機器人的資訊。我們利用人工標定和程式上的分析, 在 RGB 的色域上找出六個值:

$$(R_{up}, R_{low}, G_{up}, G_{low}, B_{up}, B_{low})$$

$$RGB_{low} = \min(\text{GetRGBvalue})$$

$$RGB_{up} = \max(\text{GetRGBvalue})$$

我們找出概略的 RGB 範圍的判定經由以下的運算可獲得一張單純的二元圖陣列 (屬於機器人範圍內的點在展示圖上顯示成白色, 傳遞陣列中我們設成 1, 以下簡稱特徵值, 其餘在展示圖上顯示成為黑, 以下簡稱雜訊, 我們存成 -1 以方便日後做連線有無超出範圍的判定)。為了加速執行效率來達成 real-time 的目標, 所以在此步驟會產生的小範圍雜訊值 (不屬於機器人卻判定成白色, 屬於機器人被判定為黑色) 我們不會作任何處理, 由於這些雜訊往往會產生極小色塊或雜點, 經由實驗證明這些雜訊並不會影響最後判定結果。

Search key-point function:

將獲得的二元圖平均切成 n 等分, n 越大可讓圖片資訊更精細, 做出的架構模型更具細節, 但相對的伴隨著龐大的空間與運算資源的耗損, 在此實作中為了取捨效率與準確率的平衡, 將採用 $n=15$ 下去執行。因次將從上一步驟的二元矩陣平均切成 15×15 的區域, 每一區域內的中心如果為白色則作為該區域的特徵點代表, 如果呈黑色代表不在機器人範圍於是進行朝八個方向的運算 (分別為上、下、左、右、左上、左下、右上、右下算其距離平方) 找出最短距離且是屬於白色範圍的特徵點, 設成此區域的代表特徵值, 反之則判定此區塊無任何代表性特徵值。為了方便特徵的傳遞, 我們在這段程式中會產生變數 k' 存入陣列 K' , 將 K' 陣列傳入下一步驟。

Step 1:

將圖片切成 n 等分, 並找出重心點 k 。

Step 2:

If $k \in \text{White}$ $k' = k$ goto step 5

if $k \in \text{Black}$ goto step 3

Step 3:

$$(d_{up}, d_{up-right}, d_{right}, d_{down-right}, d_{down}, d_{down-left}, d_{left}, d_{up-left}) \in d_i$$

$$d_i = \text{distance}(x_{d_i} - x_k, y_{d_i} - y_k)$$

Step 4:

If $d_i \in \phi$ 放棄該區域

Else find $k' = \min(d_i)$

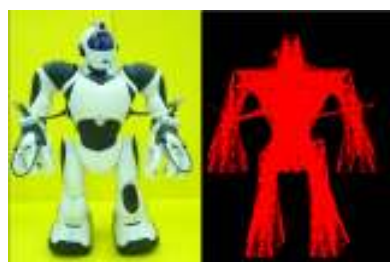
Step 5:

Save k' to $K' [i]$

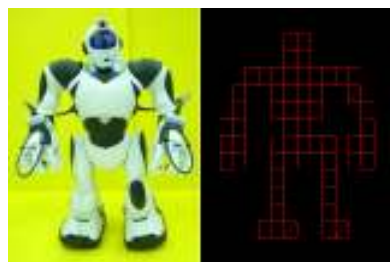
不斷重複步驟 2 到步驟 5，我們會得到陣列 K' ，裡面依序存了我們過濾過後的特徵點，導入下一 function。

Linking function:

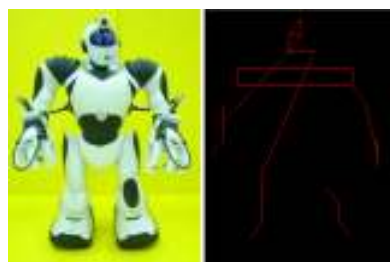
獲得此動作的代表特徵點後，必須要建構成簡單的動作示意圖，於是我們將設一陣列儲存剛剛的代表特徵值也就是上一步驟的 K' ，然後將每個特徵值對於其他特徵值做以下運算：過濾雜訊演算法（1. 算出兩點斜率，然後從兩點中偏左邊那點開始往右檢查 假設座標是(15, 50)、(25, 80)，我們就從 $X=15$ 跑到 25 做 $(X-15)*\text{斜率}+50$ 這樣就會跑過兩點連線的所有點了）完成這一步後我們會獲得一張簡單的動作示意圖，由於實作發現不同的連線規則



(a) 第一種架構圖



(b) 第二種架構圖



(c) 第三種架構圖

Fig. 3. (a, b, c 各種架構圖)

會產生不同的架構圖，且架構圖的資訊也會產

生巨大變化。於是我們在此提出三種連線方式，並概略講解三種連線方法差異度，第一種 (Fig. 3. a) 為不省略任何特徵點與線，接近暴力法把所有可能的點全部連出來，辨識度高但伴隨著相對運算量大且雜訊也多，為了加速系統演算效率，我們必須省略大部份無用的資訊。於是我們提出第二種方法搜尋所有的斜線 (Fig. 3. b)，省略畫上去以後出現曾經交叉過的線，產生了完整的格子架構圖，主體結構構圖明顯，動作構顯而易見，但在實作上會產生轉身系列動作無法分析的錯誤，也就是機器人左右轉動作分析不夠大無法有效辨識出目標物體動作，為了兼顧資料簡單化與辨識度的平衡，我們再次改進方法，成為目前第三種方法 (Fig. 3. c)。

第三種成果可以有效的節省大量資訊，並成功的勾勒出簡單明瞭的架構圖。我們提出概略的程式寫法：

Constructing Function 的參數為上一步驟所獲得 K' ，buffer 為系統暫存矩陣，主要儲存特徵點與座標資訊。

```
Constructing Function ( $K'$ ) {
  Store all  $K'$  in buffer[k][2] by sorting
  For i>=0 to  $K'-1$ 
    For j>=1 to  $k'-1$ 
      Slope=(buffer[i][0]-buffer[j][0])/(
        buffer[i][1]-buffer[j][1])
      If(Slope==0)
        Then linking(I,j)
      If(Slope<0)
        Then
          linking(buffer[j][1],buffer[i][1])
          h=(1-buffer[k][1])*de+
            buffer[k][0]
      If(Slope>0)
        Then linking(buffer[i][1]+1,h)
    }
}
```

另外在特徵點存取上，我們使用了以下方法用於最後存取的判定。將每列取兩點的方法把特徵點存入矩陣時多加一些判斷先把找到的特徵點放在暫存陣列裡，然後繼續找下一個，如果下一個與暫存陣列裡的特徵點在同一列，就找下一個，並把現在這個存到暫存陣列裡直到連續兩個特徵點不在同一列，就把暫存陣列裡的特徵點放進矩陣，並把下一個特徵點也放進去，因為下一個特徵點必然是那一列的第一個。在這步驟我們可成功獲得一張簡單且易辨識的架構圖（在系統傳遞中仍是以一微矩陣傳遞）。

決策系統：

由於我們需要具有學習機制、且適合演化的系統，因此我們採用了倒傳遞式類神經網路（Back-propagation neural network）當作決策系統結構[8][9][12]。倒傳遞類神經網路可說是目前類神經學習模式的代表，屬於監督式學習法則，特別適用於預測、診斷方面，現行生活中已有多項成熟運用。其基本原理就是利用最陡坡降法（the gradient steepest descent method）的觀念，將誤差函數達成最小化。在與傳統感知機網路比較上，主要多了隱藏層可以藉由網路表現出各單元間的交互影響，並改用平滑可微分的轉換函數，使的最陡坡降法導出修正網路的加權值公式。以下將說明倒傳遞演算法如何應用之公式：

權重值： W_xhW_hy

偏權值： $\theta_y\theta_x$

輸入為： X 輸出為 Y

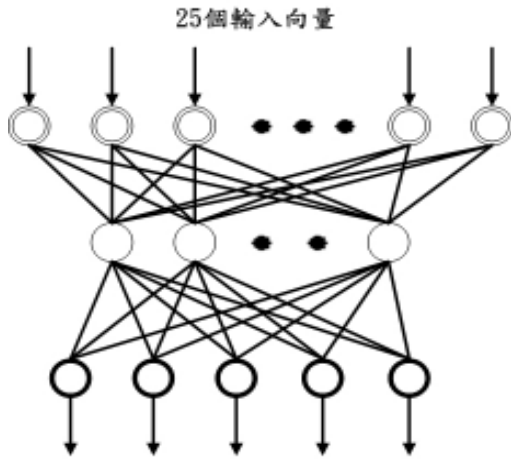


Fig. 4. 類神經示意圖

$$net_h = \sum_i W_xh_{ih} \bullet X_i - \theta_h_h \quad (1-1)$$

$$H_h = f(net_h) = \frac{1}{1 + \exp^{-net_h}} \quad (1-2)$$

由 (1-1)、(1-2) 式可得出隱藏層輸出 H ，再來我們需要隱藏層輸出 H 來求得 Y 。

$$net_j = \sum_h W_hy_{hj} \bullet H_h - \theta_y_j$$

$$Y_j = f(net_j) = \frac{1}{1 + \exp^{-net_j}} \quad (1-3)$$

在 (1-3) 式我們將會獲得輸出向量 Y ，也就是類神經結果，僅有這樣是無法加入學習功能

的，所以我們必須修正並更新神經元權重值。在下面我們將探討其學習能力方程式。由(1-4)先求得差距量 δ

$$\delta_j = Y_j(1 - Y_j)(T_j - Y_j)$$

$$\delta_h = H_h(1 - H_h) \sum_j W_hy_{hj} \delta_j \quad (1-4)$$

利用差距量我們必須求得權重值修正量 ΔW 和偏權值修正量 $\Delta \theta$ 利用(1-5)式可求得權重值修正量、偏權值修正量，利用這些修正量在(1-6)式中更新權中值、偏權值來達成類神經的學習策略。

利用以上的類神經特性，我們架構出以倒傳遞為主體的決策系統，在實作中(Fig. 4)，我們將圖切成 5*5 等份，每一等分的連線數目視為一個 input，所以在系統裡我們採用了 25

$$\Delta W_hy_{hj} = \eta \delta_j H_h$$

$$\Delta W_xh_{ih} = \eta \delta_h X_i$$

$$\Delta \theta_y_j = -\eta \delta_j$$

$$\Delta \theta_h_h = -\eta \delta_h \quad (1-5)$$

$$W_hy_{hj} = W_hy_{hj} + \Delta W_hy_{hj}$$

$$W_xh_{ih} = W_xh_{ih} + \Delta W_xh_{ih}$$

$$\theta_y_j = \theta_y_j + \Delta \theta_y_j$$

$$\theta_h_h = \theta_h_h + \Delta \theta_h_h \quad (1-6)$$

輸入向量、單層隱藏層、15 隱藏神經元、5 輸出向量。輸入向量代表每方格的圖中連線數目，而 5 輸出向量分別表示 5 個決定動作。最後輸出向量最高值的就是系統最後決定的動作。

3. 硬體環境

我們使用偉客多公司[13]出產的 CRO(Fig. 5)紅外線控制器，具備功能有分析並學習紅外線功能，可直接由 PC 端透過 C 語言發射紅外線指令，此遙控器經由 RS232 介面與我們主機作連接，預先分析好完整的遙控機器人的紅外線控制碼，可讓我們直接在 C 語言中操作。

在測試機體平台上，我們採用了由 Wow Wee[14] 玩具公司推出的史賓 V2 的娛樂兩足機器人(Fig. 5)，他的特色是運動機制簡單，



Fig. 5. (由左到右依序為紅外線控制發射器，史賓 V2 型機器人，一般畫面擷取裝置)



(a)執行畫面，訓練介面



(b)為彎腰範例，類神經成功判定

Fig. 6. (a)(b)分別為訓練執行畫面跟預測未知動作判定結果畫面

在關節靈活度不高的考量下，恰巧適合我們單純動作模仿的平台，在操作時不需要考慮到重心轉移與運動機制問題，也沒有一般兩足機器人因控制馬達力道回朔所產生的振動問題。藉由紅外線控制端可直接控制機器人動作。在軟體開發環境上我們採用 C++ 語言、MFC 平台，未來將導入 Directshow sdk 開發。

4. 實驗討論

在實驗中我們定義了五個要學習的動作，分別為機器人的左轉、右轉、站立、彎腰、後仰。由於隱藏神經元數量多，所以同樣資訊必須不斷重複訓練才会有好效果，在實驗中以每樣動作 25 張圖為樣本下去連續跑 300 次，經由類神經學習系統處理後，由各動作五個未知範例下去做測試，都能準確辨識，辨識率高達 92%(25 個未知動作範例判定，僅有兩個判定錯誤)，並且正確的送出指令，Fig. 6. b 顯示一個成功判定的畫面。但由於拍攝圖片的問題，我們不能處理角度太大的照片，這代表著與真正的人類視覺還有一段差距，也是未來極需挑戰的問題，另外在拍攝圖片的技術上，未來將導入簡單的去背景技術，讓這套系統可以

在不只是單純背景色下運作。在運算效率的檢測上，訓練好的類神經系統在 PC 端運算是極快的，在程式最佳下有可能達成一個真正 real-time[11]學習系統。而動作分析方面，目前只能達成模仿目標的基本簡單動作，動作的變化複雜性將會是實驗裡的一大變數，在複雜環境中取得效率與準確度的平衡，會是能更深探討的議題。

5. 結論

人工智慧至今已發展好幾十年，但與機器人領域間的整合最常使用就是模糊理論與類神經網路，機器人可以因為準確的數學方程式而有效率的達到我們的目標，但卻往往少了邏輯、推理、思考這些特性，我們認為在未來的發展上，機器人更具擬人化將會是一大議題，跳脫以往數學與物理的觀點，這領域應該更適合導入類學和生物學的觀點。

而學習與模仿是任何生物智慧最重要的也是必要的行為，我們從機器人的仿生可以獲得並改良出更好的人工智慧演算法。在學習演算法中應該可以嘗試著建構更複雜的神經網路系統，來達到讓學習不只是參數的更動，而是達到類似記憶、推理的特性。在實驗環境上，必須挑戰更複雜或者更真實的環境，系統必須具備可攜性的穩定度挑戰才能真正應用在人類世界中。

參考文獻

- [1] 游天龍，”彩色影像辨識技術應用於機器人運動控之研究“，*全國碩博士論文網*，2005. 06
- [2] 周立柏，”具模仿能力之智慧型雙足機器人之設計與實現“，*全國碩博士論文網*，2007. 06
- [3] Atkeson, C.G. and Schaal, S., ”Robot Learning From Demonstration” *Proceedings of the Fourteenth International Conference (ICML '97)*, Edited by Douglas H. Fisher, Jr. pp. 12-20, Morgan Kaufmann, San Francisco, CA, 1997.
- [4] Demiris, J. and Hayes, G., ”Imitative Learning Mechanisms in Robots and Humans“，*Proceedings of the 5th European Workshop on Learning Robots*, July 1996.

- [5] Demiris, J. and Mataric, M. J., "Perceptuo-Motor Primitives in Imitation", *Workshop on Agents in Interaction-Acquiring Competence Through Imitation, 2nd International Conference on Autonomous Agents (Agents '98)*, Minneapolis/St. Paul, May 10-13, 1998.
- [6] Williams, T. G. and Rowland, J. J. and Lee, M. H. and Neal, M. J. "Teaching by Example in Food Assembly by Robot", *Proceedings of the 2000 IEEE International Conference on Robotics & Automation*.
- [7] Maja J. Mataric, "Getting Humanoids to Move and Imitate" *IEEE, Intelligent systems and their application*, Vol. 15, NO. 4, 2000, pp 18-24
- [8] Xiaoyuan He, Ryo Kojima, and Osamu Hasegawa, "Developmental Word Grounding Through a Growing Neural Network With a Humanoid Robot" *IEEE Trans, Syst., Man, Cybern.*, Vol 37, NO. 2, 2007, pp 451-462
- [9] F. Shen and O. Hasegawa, "An incremental network for on-line unsupervised classification and topology learning," *Neural Netw.*, vol. 19, no. 1, pp. 90-106, Jan. 2006.
- [10] D. Roy, K. Hsiao, and N. Mavridis, "Mental imagery for a conversational robot," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 3, pp. 1374-1383, Jun. 2004.
- [11] S. Schaal, C. G. Atkeson, and S. Vijayakumar, "Real-Time Robot Learning with Locally Weighted Statistical Learning," *Proc. IEEE Int'l Conf. Robotics and Automation, IEEE Press*, Piscataway, N.J., 2000, pp. 288-293.
- [12] J. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, no. 1, pp. 71-99, 1993.
- [13] 偉克多微電腦工作室,
<http://vic.seeder.net/>
- [14] WowWee 娛樂機器人公司,
<http://www.wowwee.com/>