

# 資工系實務專題研究計畫成果報告

## NXT 樂高機器人之全自動學習投籃

專題編號：099-CSIE-S022

執行期限：98 年 1 學期至 99 年 1 學期

指導教授：郭忠義 教授

專題計劃參與人員：錢銅岳、王欣陽

### 一、專題摘要

本專題以 LEGO MINDSTORMS NXT 建立硬體架構，使用 Eclipse 作為開發平台、用 Java 程式語言來撰寫，並套用 Q-Learning 學習法來實現機器人自動學習的最終目標。

#### 關鍵詞：

MINDSTORMS、可程式化、智慧型機器人、Q-Learning 學習法、機器學習、全自動化

### 二、緣由與目的

在這個科技日新月異的時代，為了讓系統達到更快的速度與更大的儲存容量，硬體與軟體每分每秒不斷的在更新，並套用了許多複雜的演算法來產生更高的效能，但這樣的模式所輸出的結果，不但系統本身無法解釋其動作，且在實際中經常會發生不可預期的錯誤。

而運用了機器學習的系統，在現實的應用場合中，比較能符合使用需求，且由於是採取經驗累積的方式，系統本身也能夠解釋其動作。

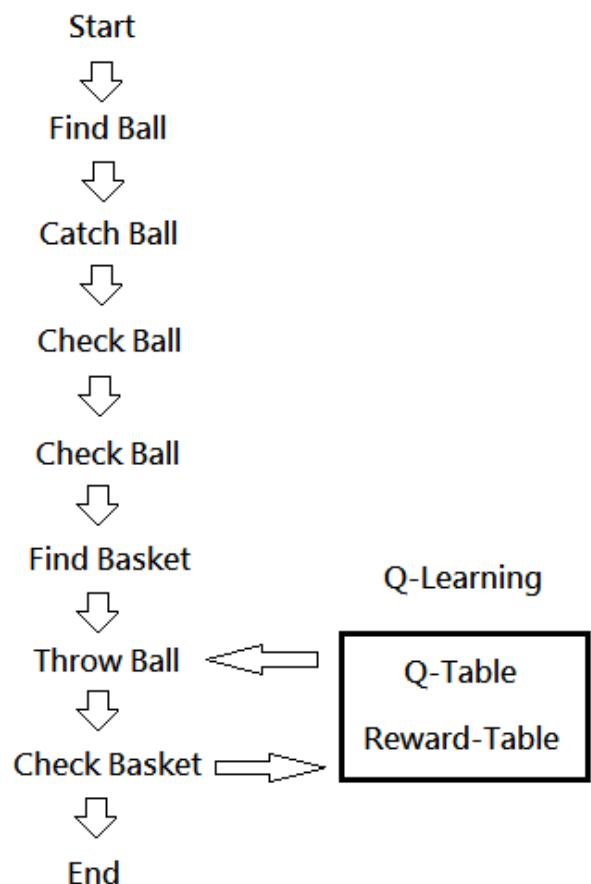
所以我們這次的專題以全自動化的方式來進行機器的學習，期望能夠達到讓程式與實際運作有一定程度的契合度。

### 三、系統架構與技術說明

本專題的系統可以分為「Find Ball」、「Catch Ball」、「Check Ball」、「Find Basket」、「Throw Ball」、「Check Basket」

這幾個項目。

其中「Throw Ball」與「Check Basket」會使用到 Q-Learning，以進行不同資料的嘗試，並在投球結束後給予 Feedback，得到最佳的投球力道。



〈圖一、系統流程圖〉

Q-Learning (增強式學習法) 可以細分為 Q-Table 與 Reward-Table 兩個不同的表格，藉由提供資訊與系統的 Feedback，改變這兩個表格的資料，讓內部的 Q 值更加的精確。

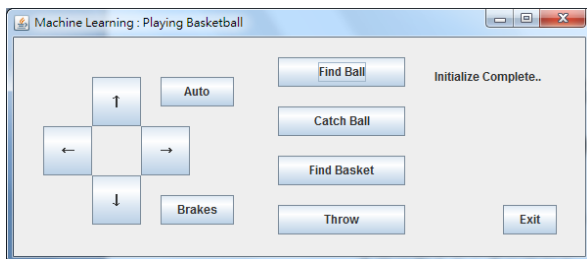
$$Q_n(s, a) \leftarrow Q_{n-1}(s, a) + \alpha_n \times [r - Q_{n-1}(s', a')]$$

$$\begin{cases} Q_{n-1}(s, a) : \text{old value} \\ \alpha_n : \text{learning rate} \\ r : \text{reward} \end{cases}$$

〈圖二、Q-Learning 學習法〉

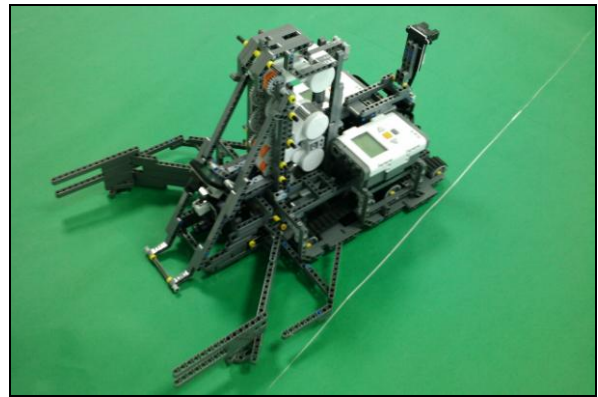
#### 四、研究結果

使用者可以藉由 GUI 介面〈圖三〉對機器人進行操控，進行分段式的測試或是直接點擊 Auto 進行全自動的機器學習。



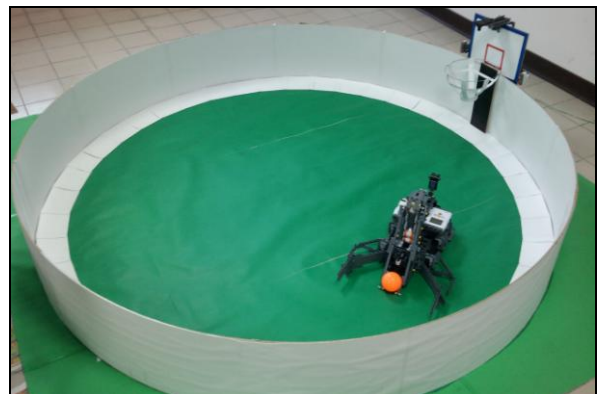
〈圖三、控制介面〉

1. 方向鍵  
手動控制機器人的移動及方向的改變。
2. 停止鍵  
手動控制機器人立即停止目前的動作。
3. Find Ball  
讓機器人自動尋找場地內的球，利用 NXTCam 偵測畫面中球是否存在，並做微調使機器人對準球之正中央。
4. Catch Ball  
利用球的大小變化來偵測是否已接近球，若行進過程中出現了偏差，程式會自動微調至正確方向，使機器人抓取到已完成鎖定的球。
5. Find Basket  
讓機器人自動尋找場地內的籃框。
6. Throw Ball  
讓機器人根據與目標的距離，向 Q-Learning 取得適合的力道來進行嘗試，並在投籃後，從裁判端取得結果，回覆投籃結果給 Q-Learning 來進行 Q 值的修改，達到學習的目的。



〈圖四、機器人主體〉

機器人長 48 公分、寬 27 公分、高 33 公分，可支援行走、旋轉、找球、夾球、找籃框、丟球等功能。



〈圖五、場地〉

場地直徑為 2 公尺、內圈設有防撞機制，避免機器人手臂卡住，且設有籃框與裁判機器人。

#### 五、結論與未來發展

由於硬體本身的產生的誤差，讓準確度無法達到百分之百的命中，但在實驗中也證明了透過 Q-Learning 學習法是能夠提升資料正確性的，也在實驗中學到了，如果要讓 Q-Learning 能夠更精準的給予資訊，則必須提供更多的 Feedback。

#### 六、參考文獻

- [1] LeJOS NXT API  
<http://lejos.sourceforge.net/>
- [2] ROBOTC.net forums  
<http://www.robotc.net/forums/>
- [3] LeJOS.org  
<http://sourceforge.net/forums>

