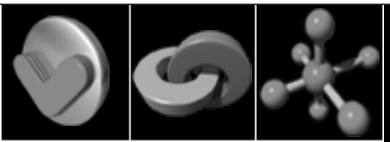
An Adaptive State Partition Approach to Reinforcement Learning

中正大學電機工程學系
黃國勝　博士



大綱

- 研究動機
- 背景介紹
- **Skill Presentation - Decision Tree**
- **Learning Skill by Demonstration**
- **Skill Synthesis**
- 實驗設計與討論
- 結論與未來展望

研究動機

- 設計機器人行為構成的困難：
  - 環境多變
  - 需要縝密的考量
  - 需要機器人學的知識
  - 由實測結果來調整修正

Hard Computing v.s. Soft Computing

研究動機

- 如何學習”技巧” (Skill)
  - 技巧該如何表示
    - 有技巧單 (primitive-based)
    - 無技巧單元(non-primitive-based)

無技巧單元

- 局部 - **Local Function Approximation**
  - 小腦模型(CMAC)
  - 徑向基底網路(RBFN)
  - 整合符號表示與RBFN 網路
- 整體 – **Global Function Approximation**
  - 多層類神經網路
  - 模糊邏輯
  - 隱藏式馬可夫模型

有技巧單元

- 動作基本單元(motion primitives)
- 感測與運動單元(perceptual-motor primitives)
- 運動基板(motor schemas)
- 運動程式(motor programs)
- 行為基礎系統(behavior-based systems)

人類示範機器學習法 (learning by demonstration)

- 主要是根據人類的動作來學習獲得機器人的技巧
  - **Assembly plans from observation(APO)**
  - **programming by demonstration(PBD)**
  - **programming by watching(PBW)**
  - **Learning by demonstration(LBD)**

工作(Task)表示法

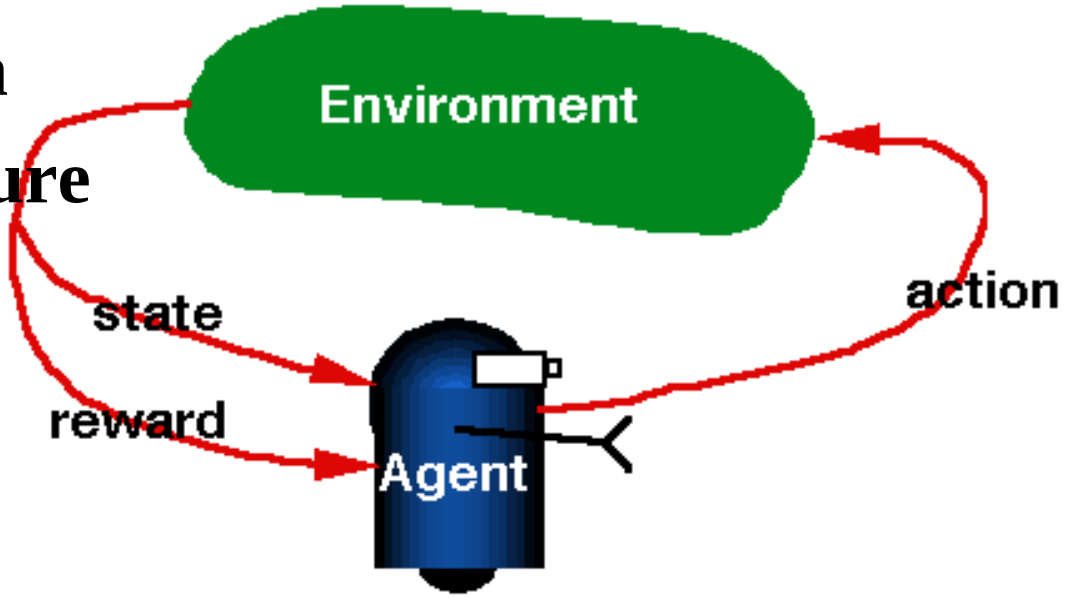
- 近似語言表示法(language-like expressions)
- 符號表示法如：
  - 圖形(graphs)
  - 階層式結構(hierarchical structures
  - 決策樹(decision trees)

技巧學習演算法

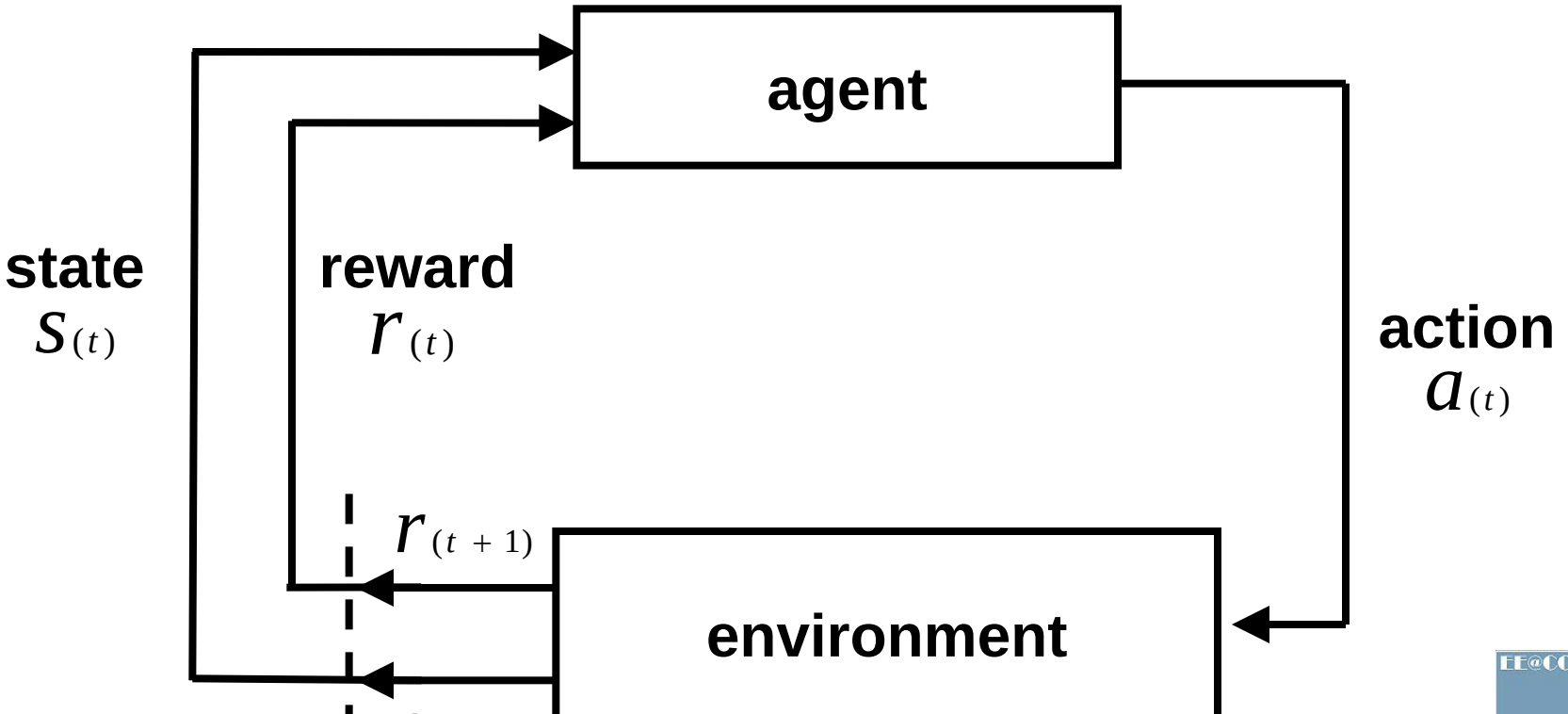
- 當技巧被表示後，技巧學習演算法必須與表示法一起配合，來使技巧對環境有適應性
  - 監督式
  - 非監督式
  - 加強式的學習方法。

專業背景介紹

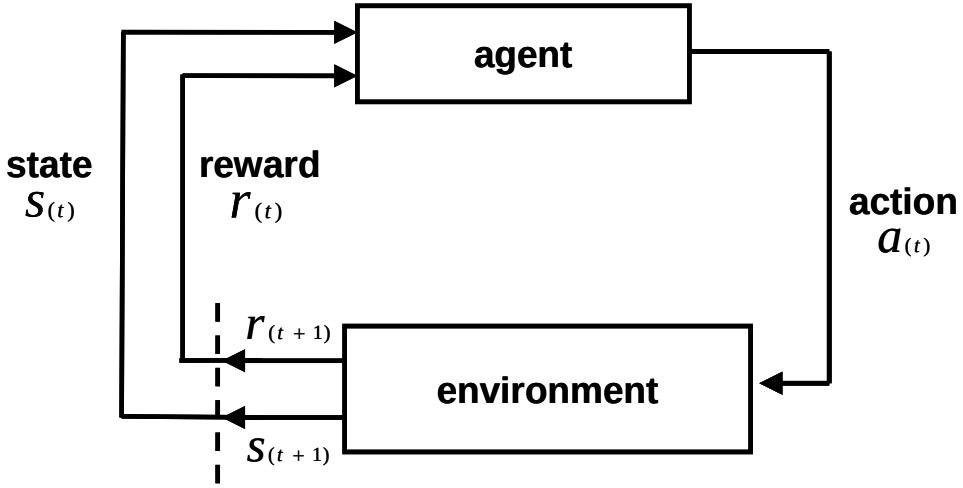
- **Reinforcement Learning**
- **Q-Learning**
- **Decision Tree Induction**
- **Subsumption Architecture**



背景介紹: Reinforcement Learning

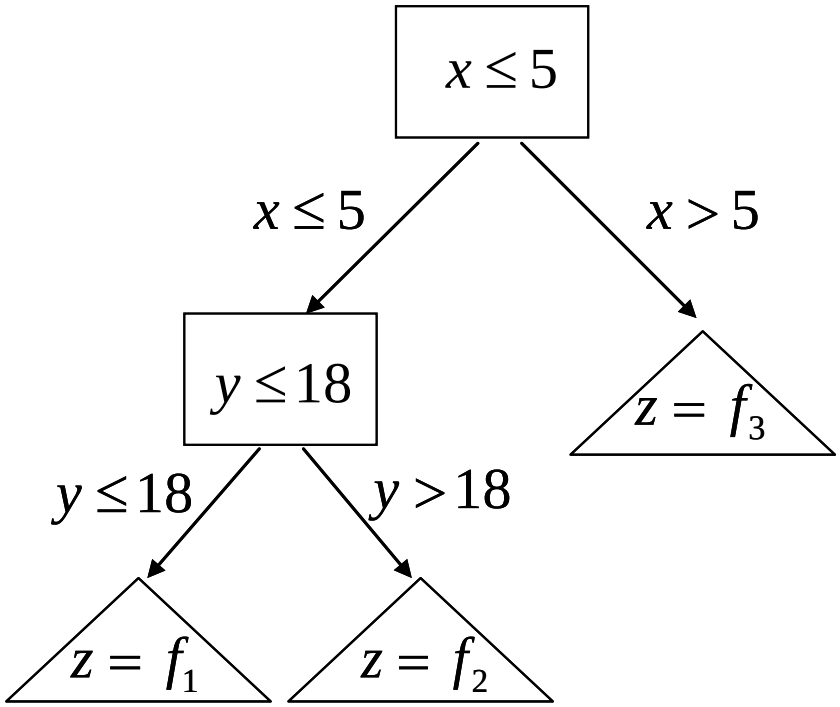


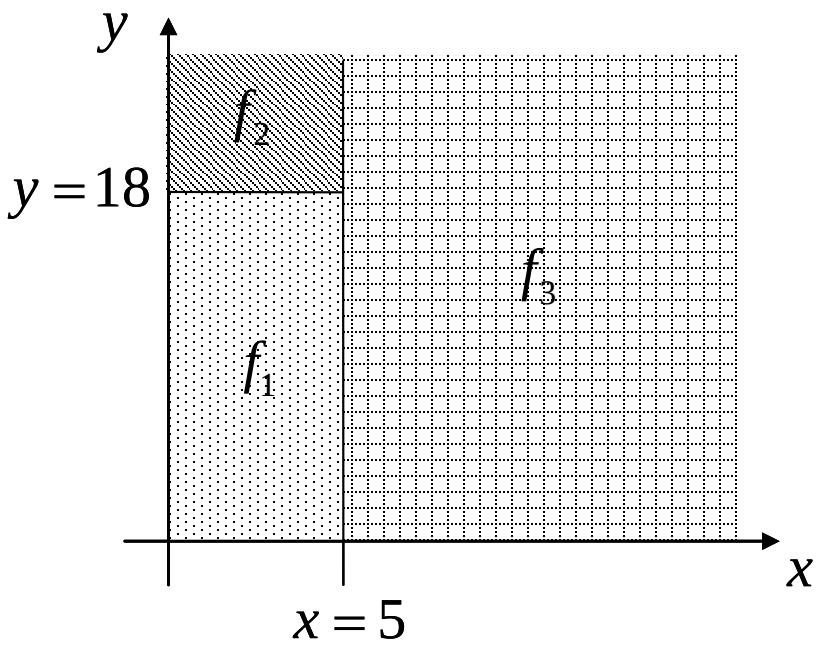
背景介紹: Q-Learning



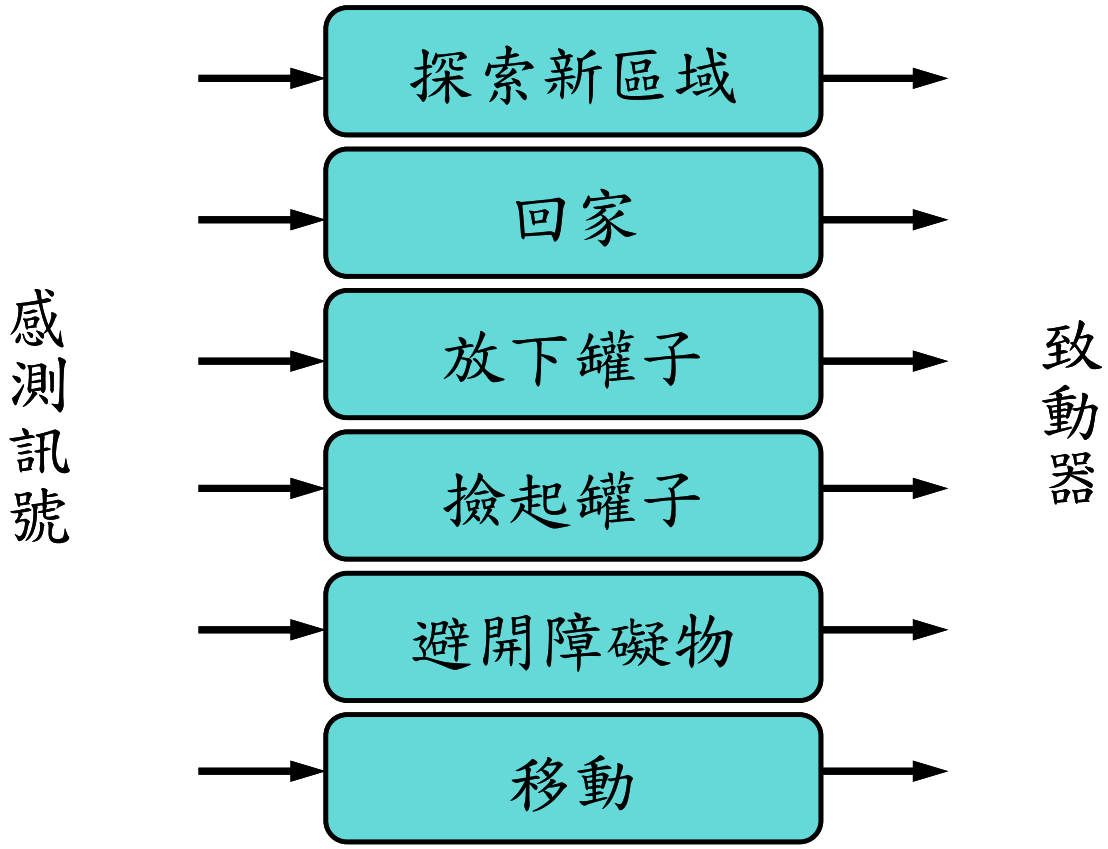
$$Q(s,a) \leftarrow Q(s,a) + \alpha * \left(r + \gamma * \max_{a'} Q(s',a') - Q(s,a) \right)$$

背景介紹: Decision Tree Induction





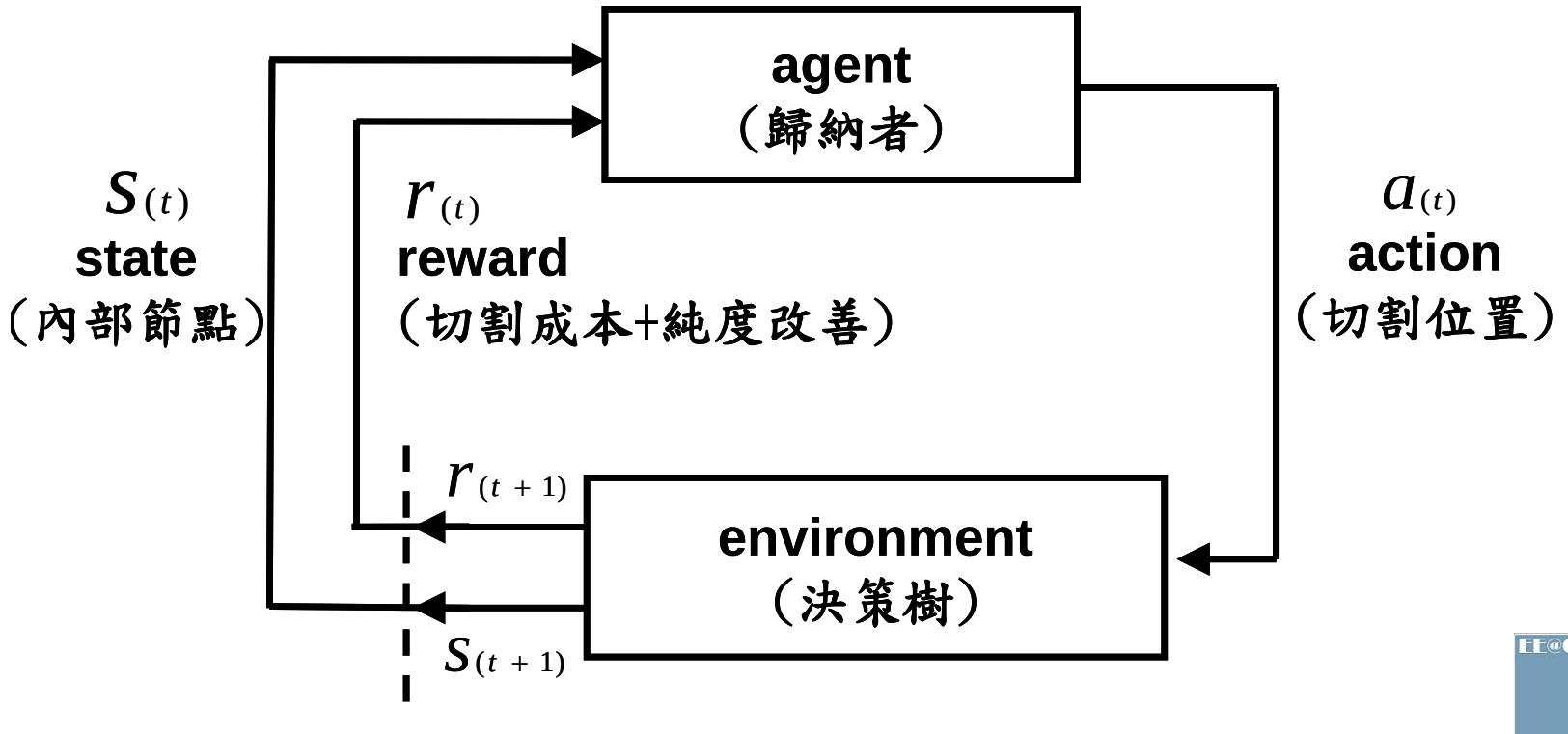
背景介紹: Subsumption Architecture



RL-based Decision Tree Algorithm

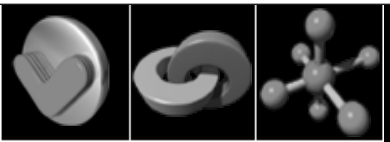
- **Reinforcement Learning-based Decision Tree Algorithm**
  - 示範技巧機器學習
- **Decision Tree-based Q-Learning**
  - 技巧合成

RL-based Decision Tree Algorithm



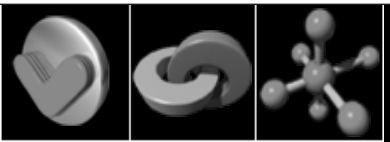

EE@CCU
EIS
Lab.
EMBEDDED INTELLIGENT SYSTEM






EE@CCU
EIS
Lab.
EMBEDDED INTELLIGENT SYSTEM



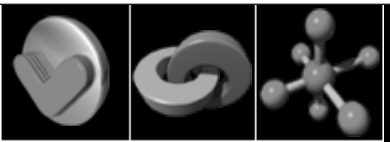


RL-based Decision Tree Algorithm

- 切割終止條件：
  1. 內部節點所包含的區域太小
  2. 內部節點所包含的資料量太少
  3. 切割過後，會導致某一邊沒有資料
  4. 內部節點的純度夠純


EE@CCU
EIS
Lab.
EMBEDDED INTELLIGENT SYSTEM





Hardware Architecture

- **Communication interface: CANbus**
- **Hardware platform: dsPIC33F**
- **Terminal controller: embedded system**

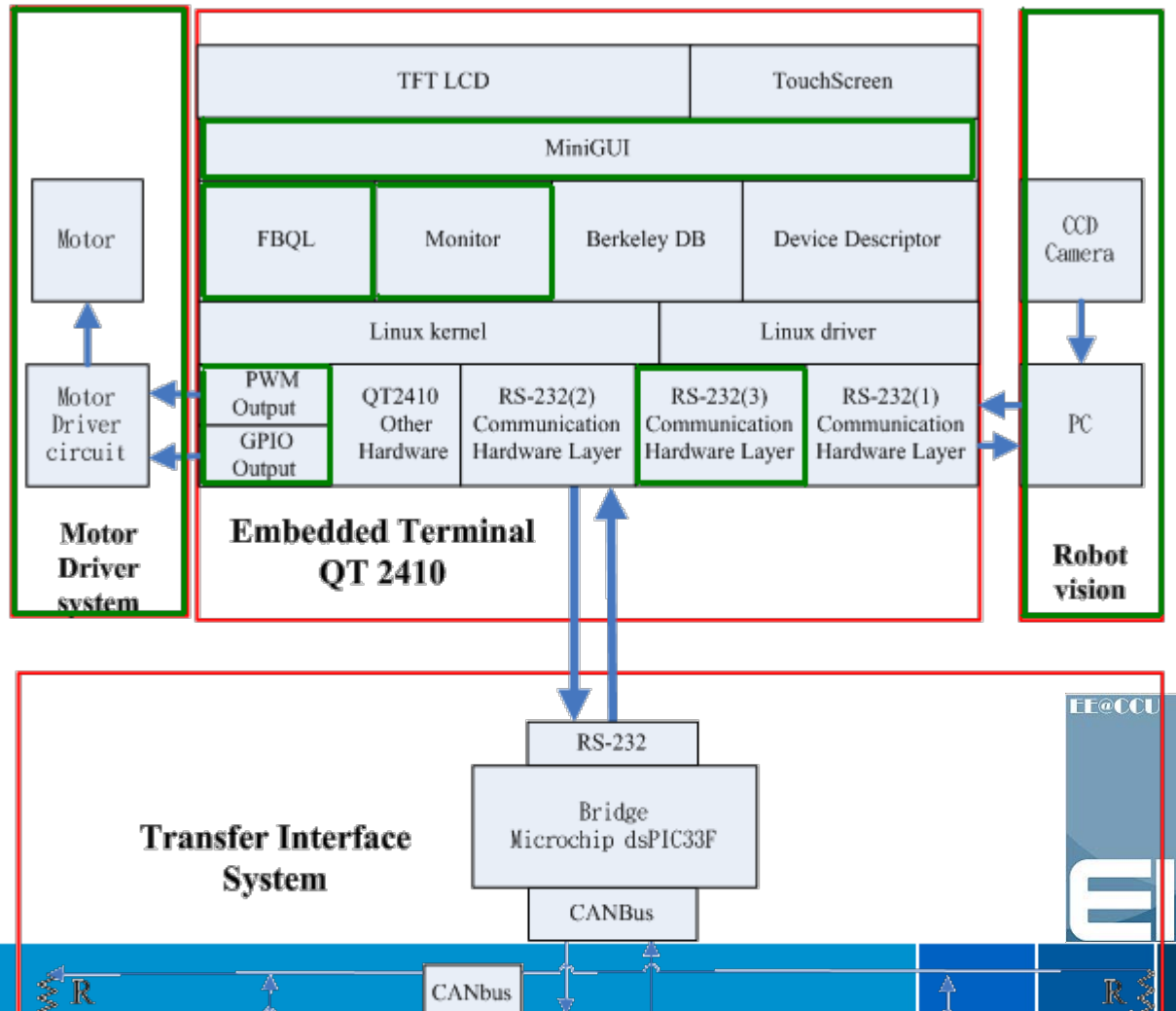
System Structure

Top: **Fusion**

Center:

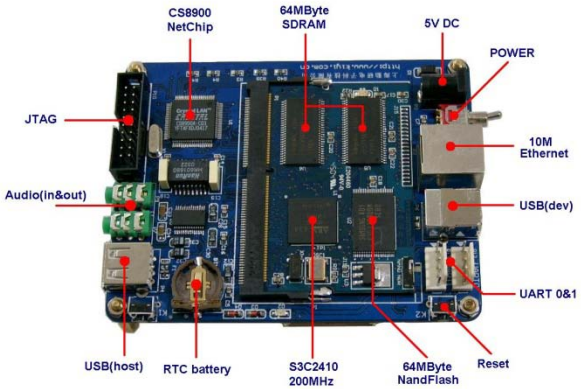
Communicate

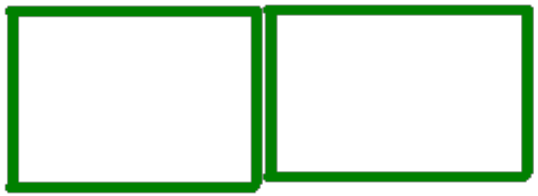
Bottom: **Module**



Complex behavior – Root Device

QT2410(new version) develop board Structure





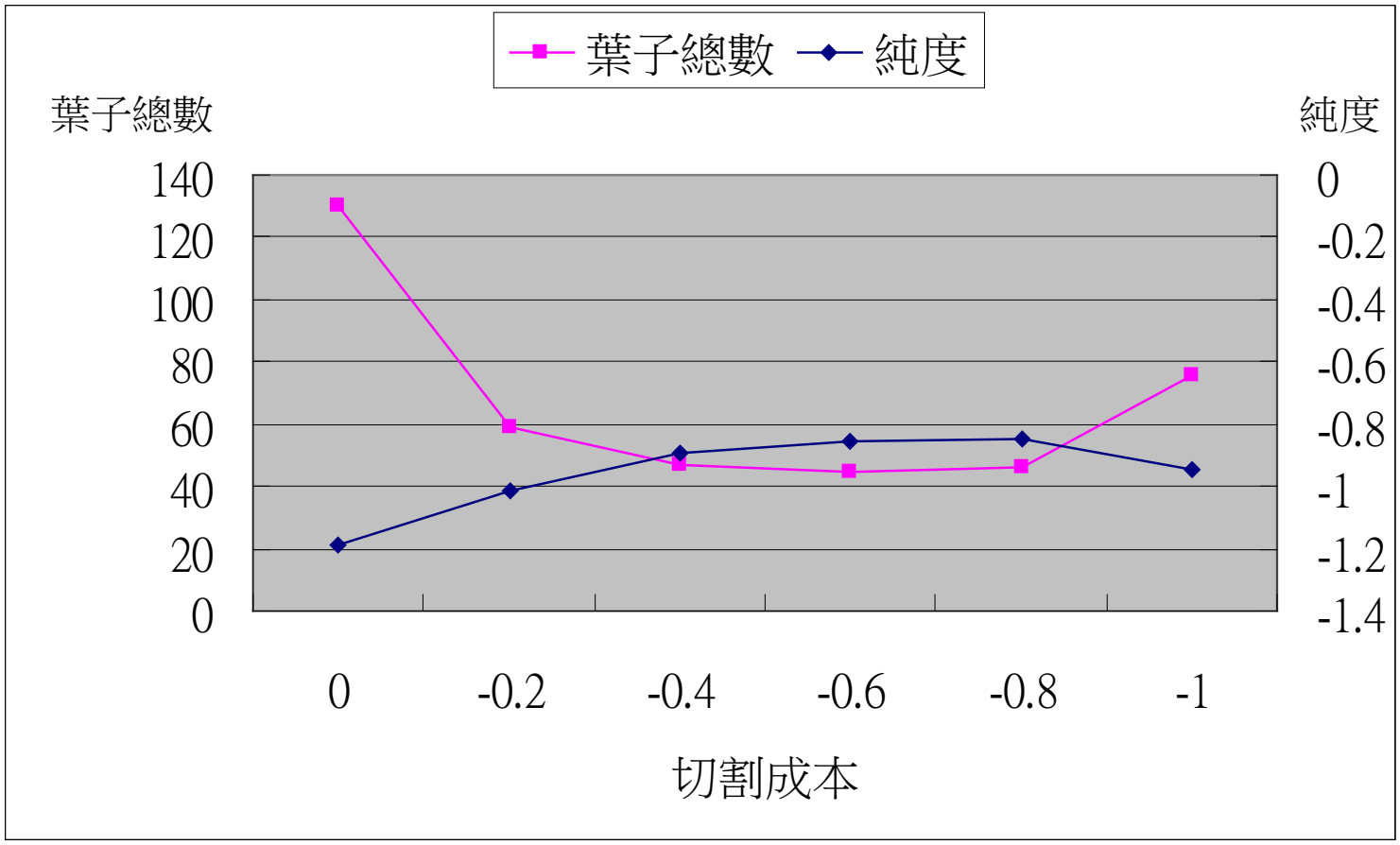
示範機器學習

- 追球行為模仿

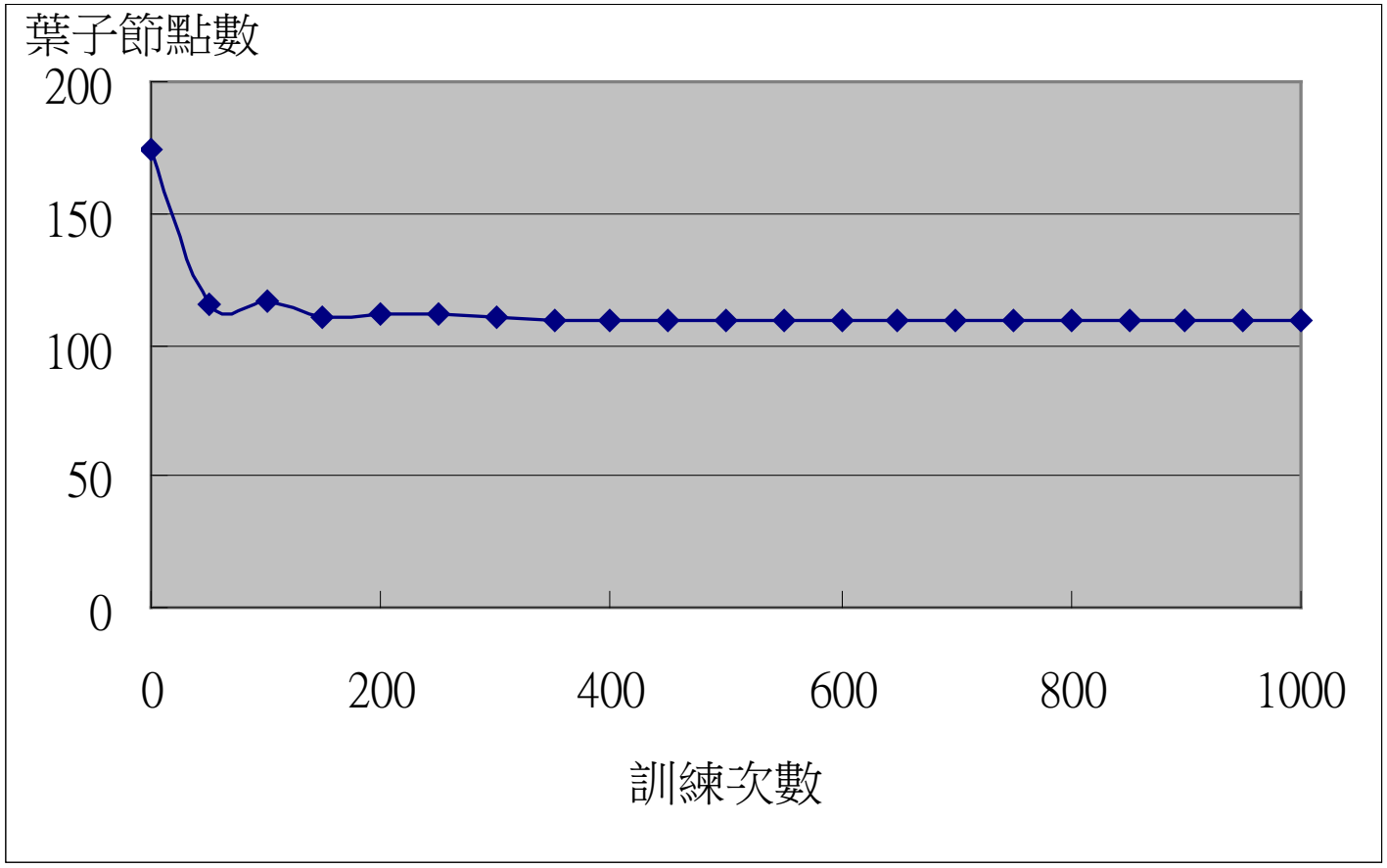
RL-based Decision Tree Algorithm

RL-based Decision Tree Algorithm

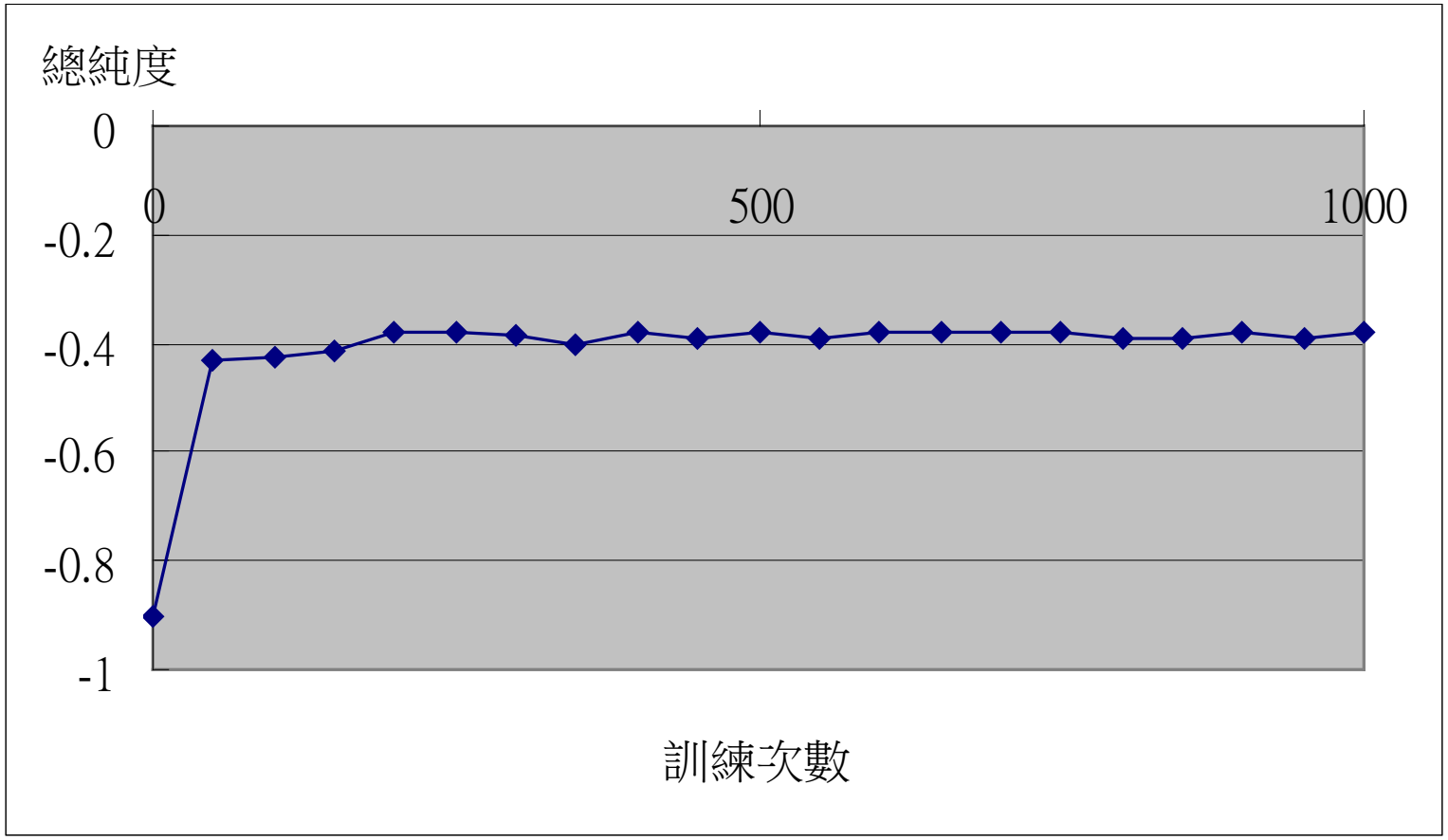
RL-based Decision Tree Algorithm



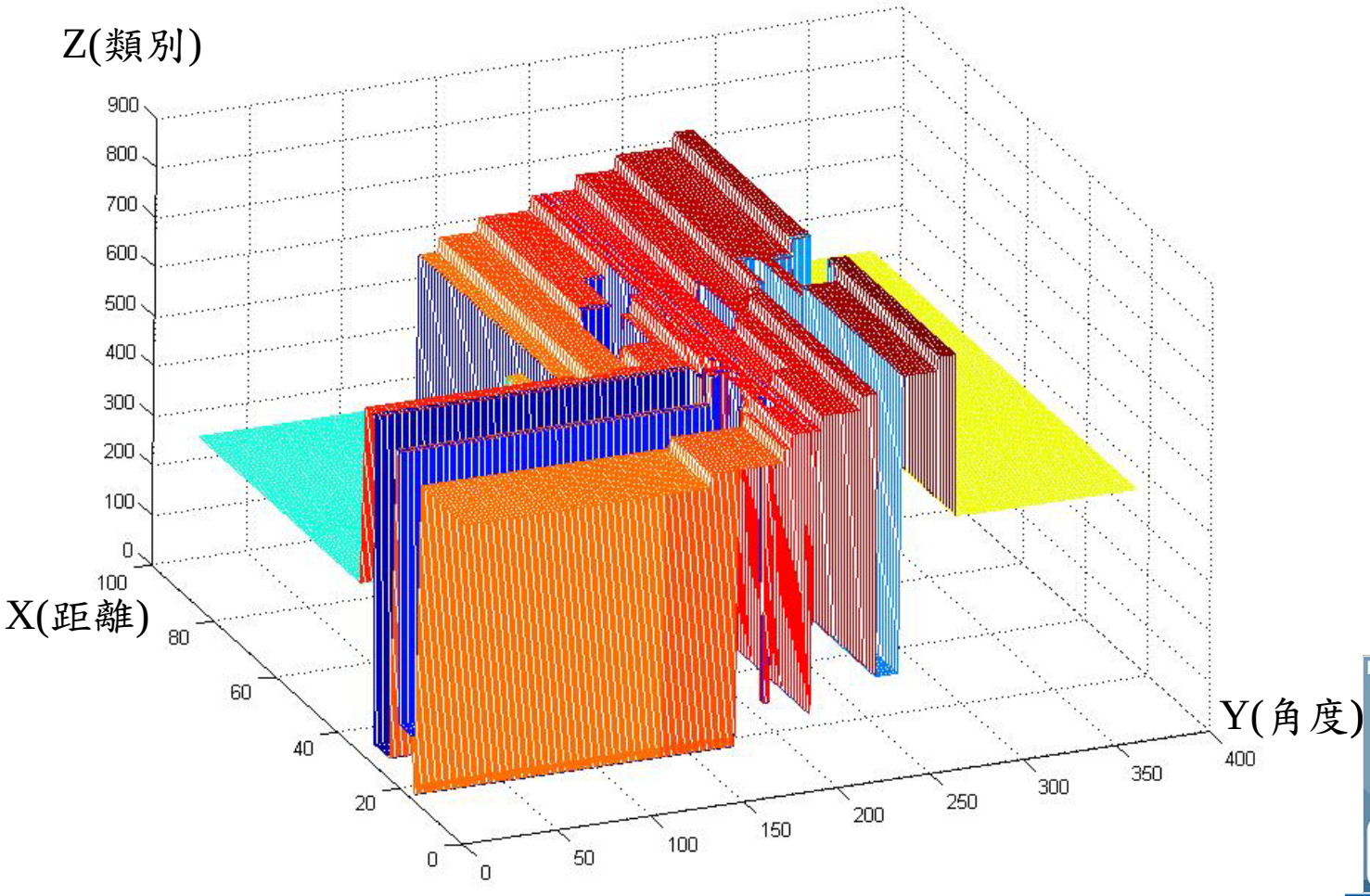
RL-based Decision Tree Algorithm



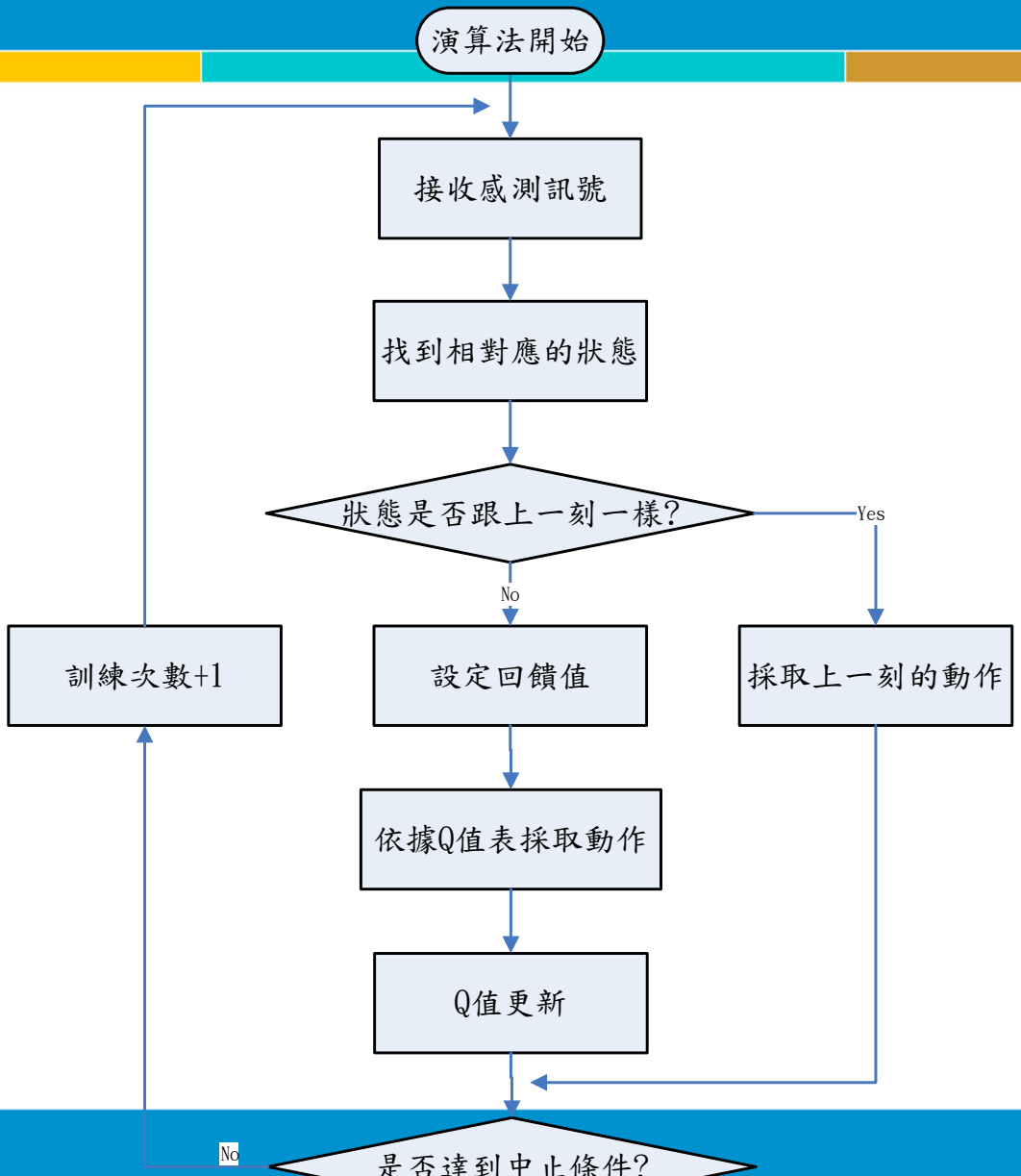
RL-based Decision Tree Algorithm



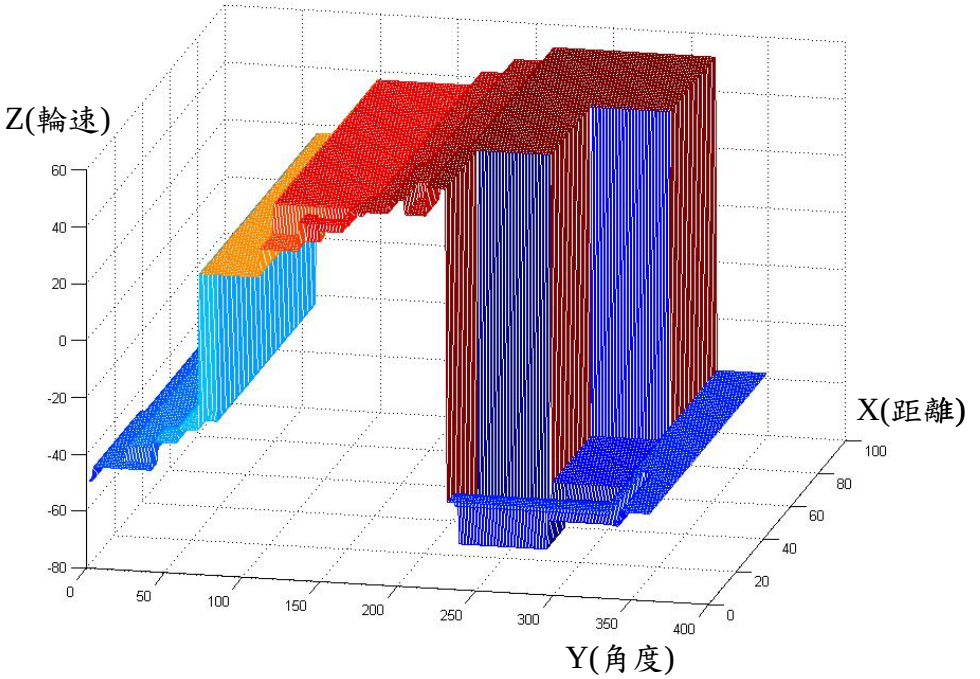
RL-based Decision Tree Algorithm

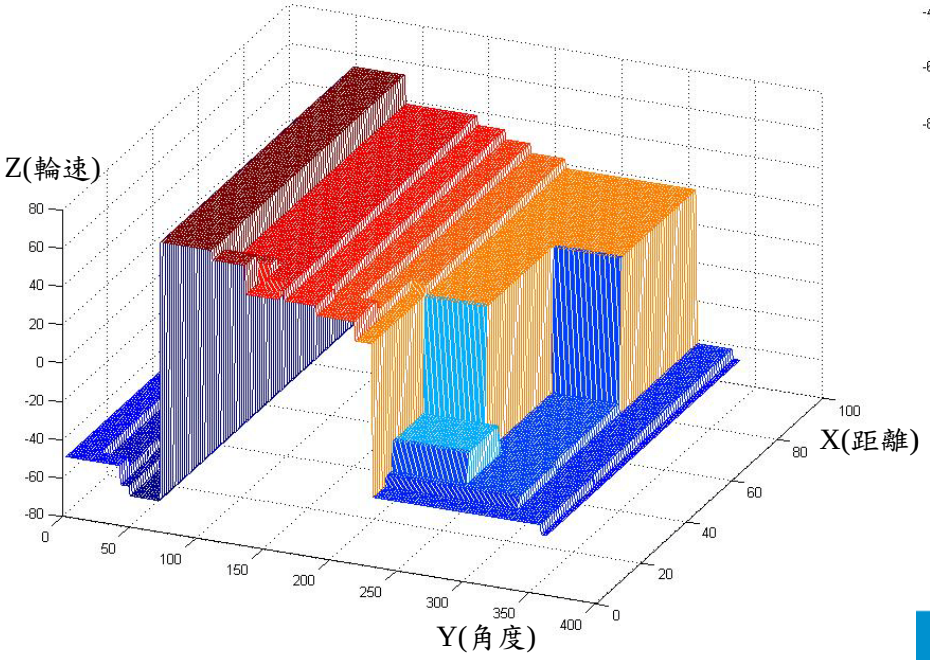


DT-based Q-Learning

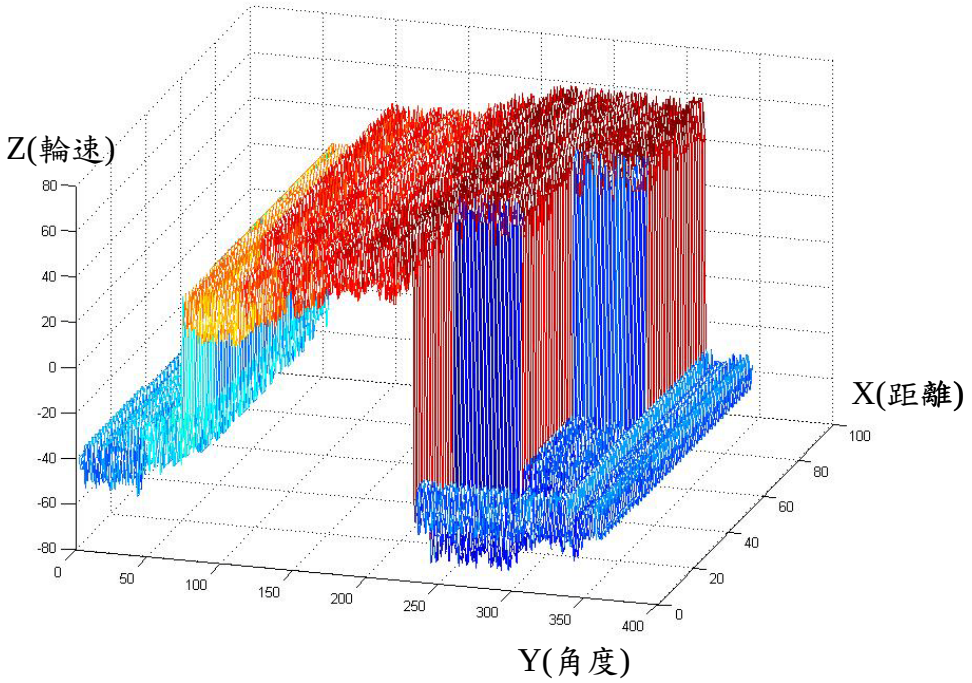


DT-based Q-Learning

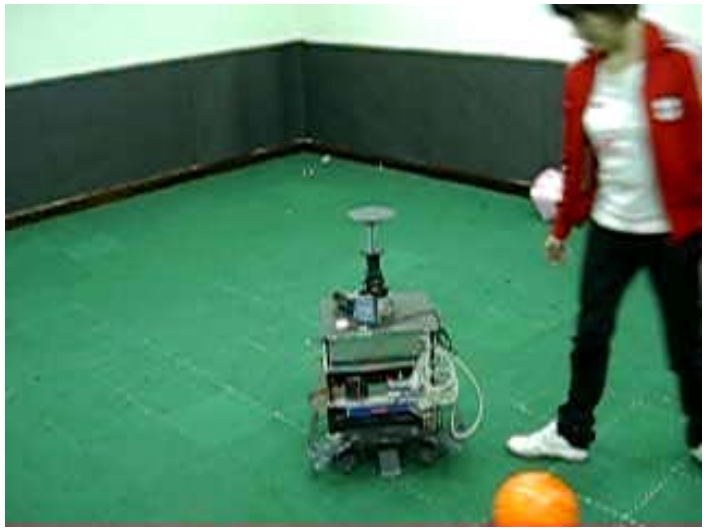




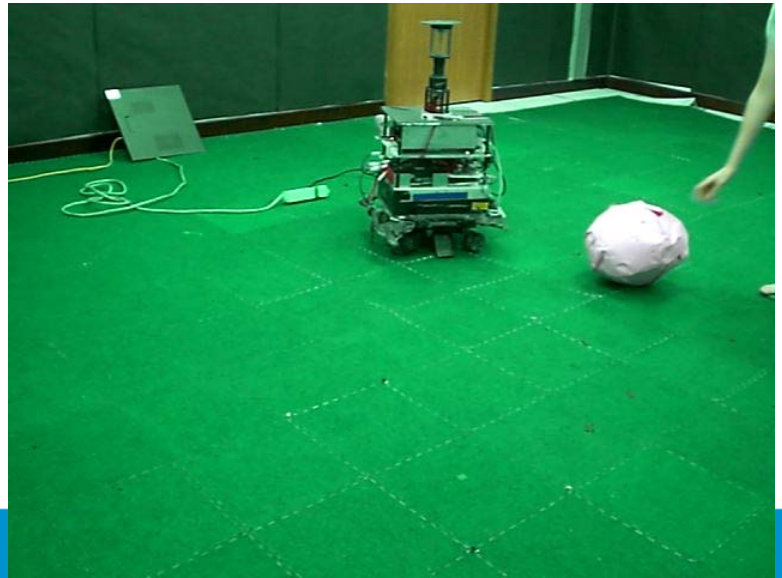
DT-based Q-Learning



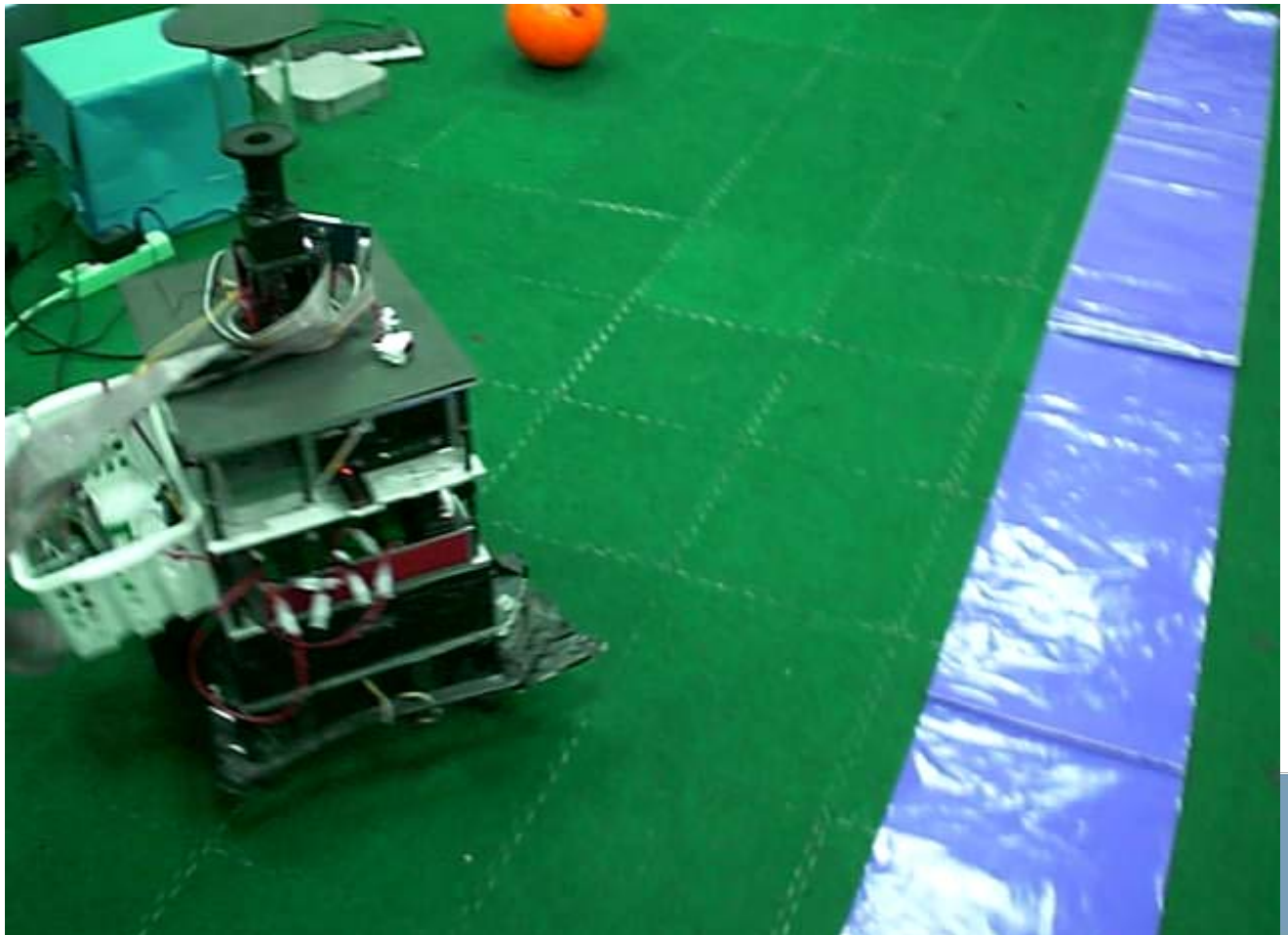
Goal seeking



Obstacle avoidance

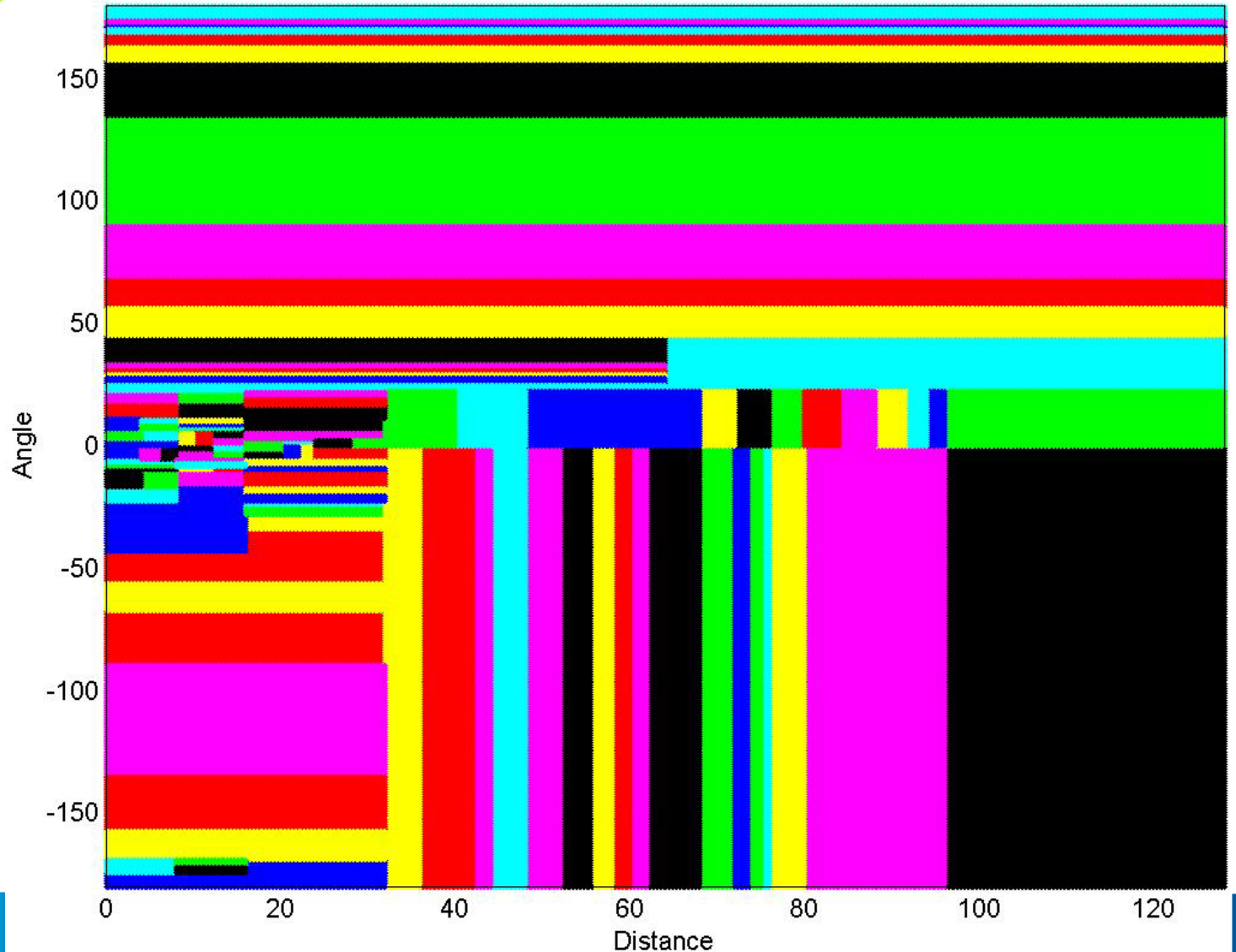


Line following



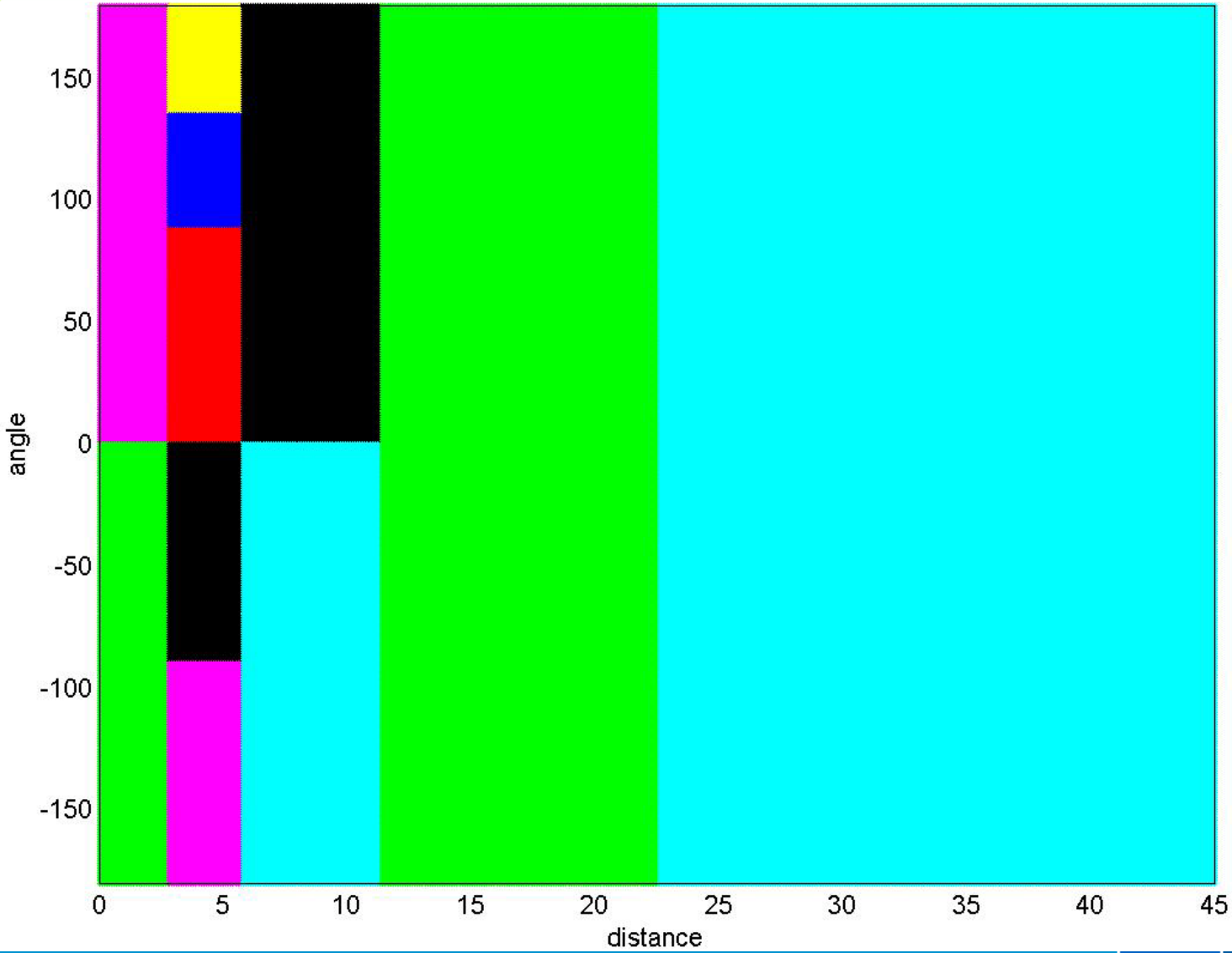
實驗設計與討論:尋標行為模仿

共122種類



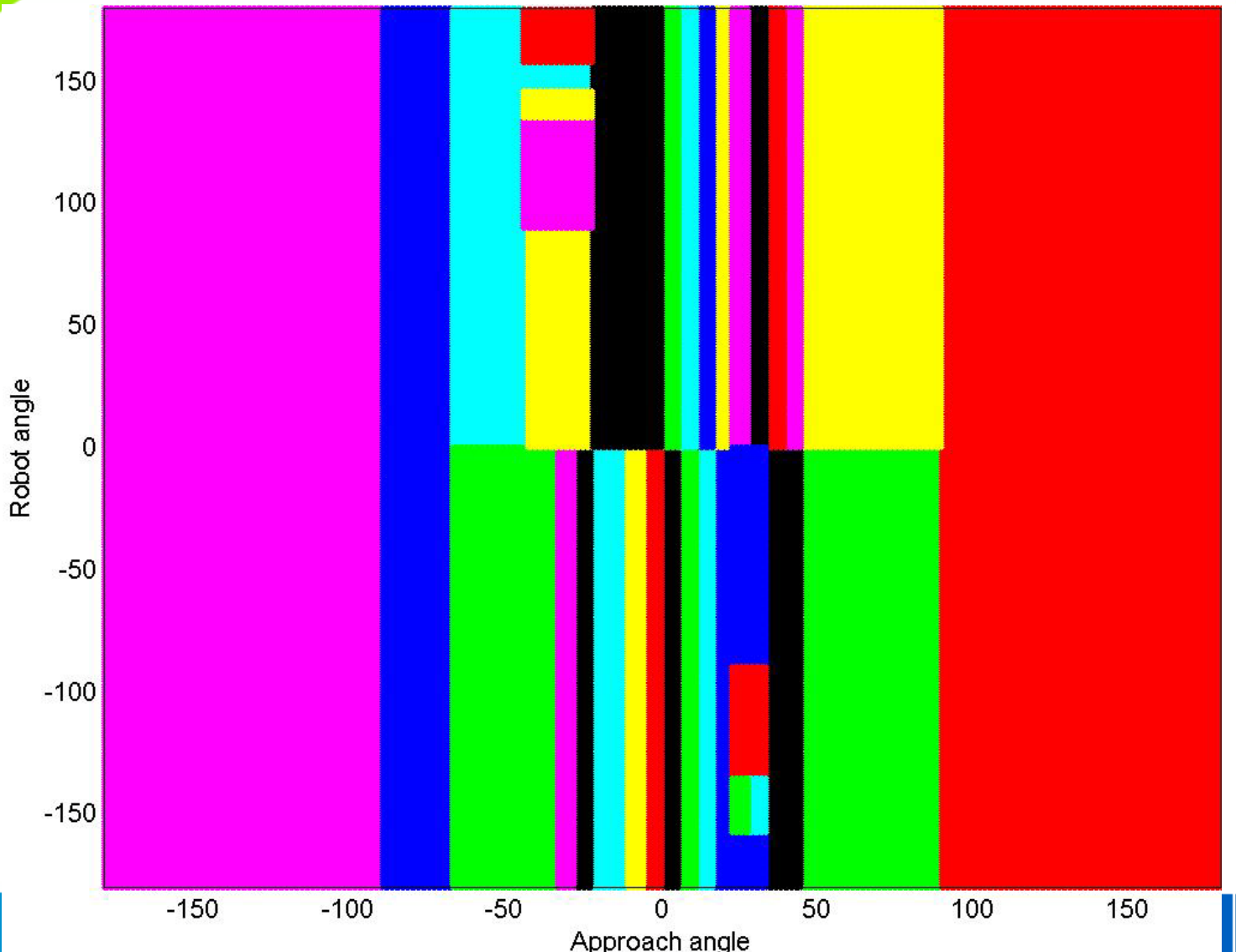
實驗設計與討論: 避障行為模仿

共11種類

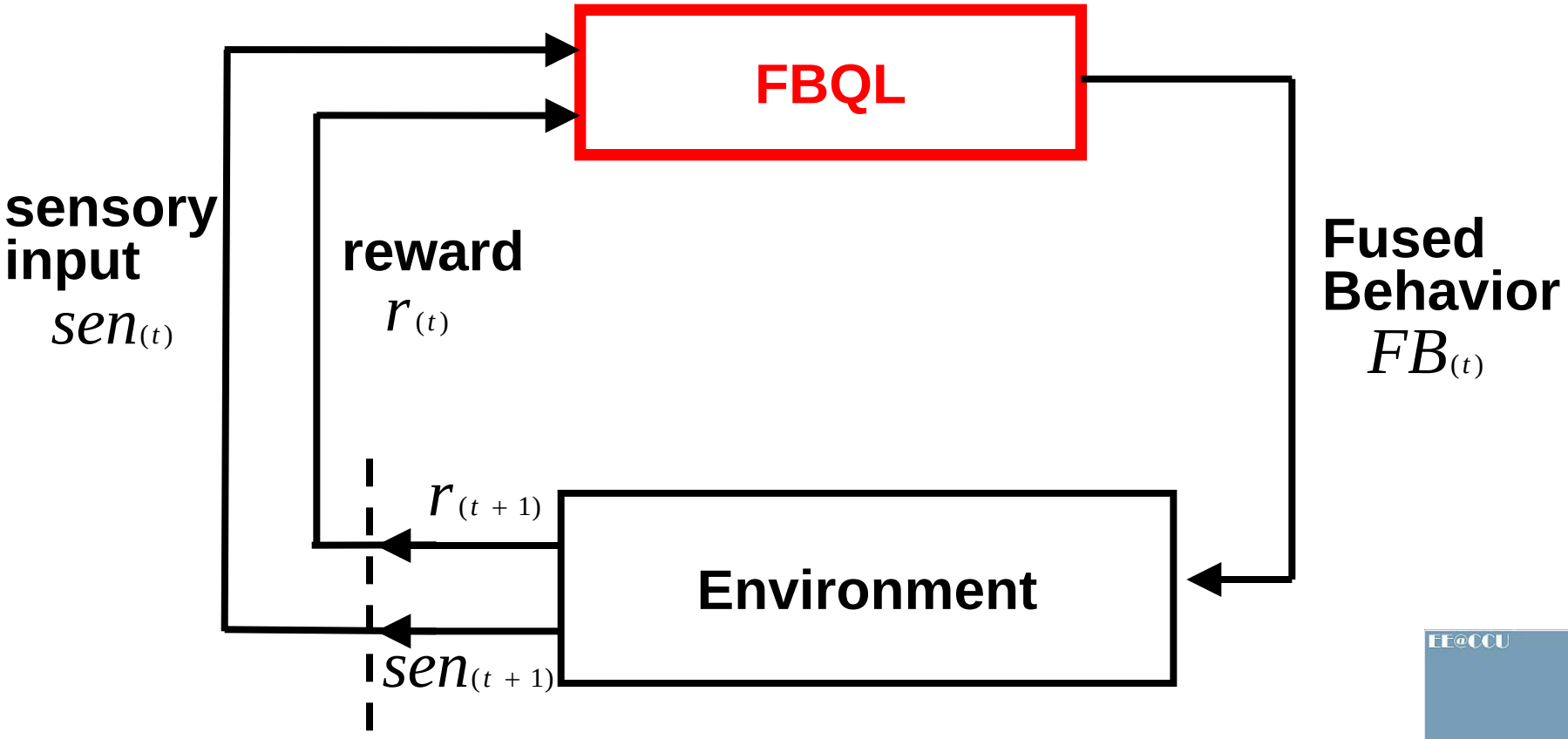


實驗設計與討論:沿線行為模仿

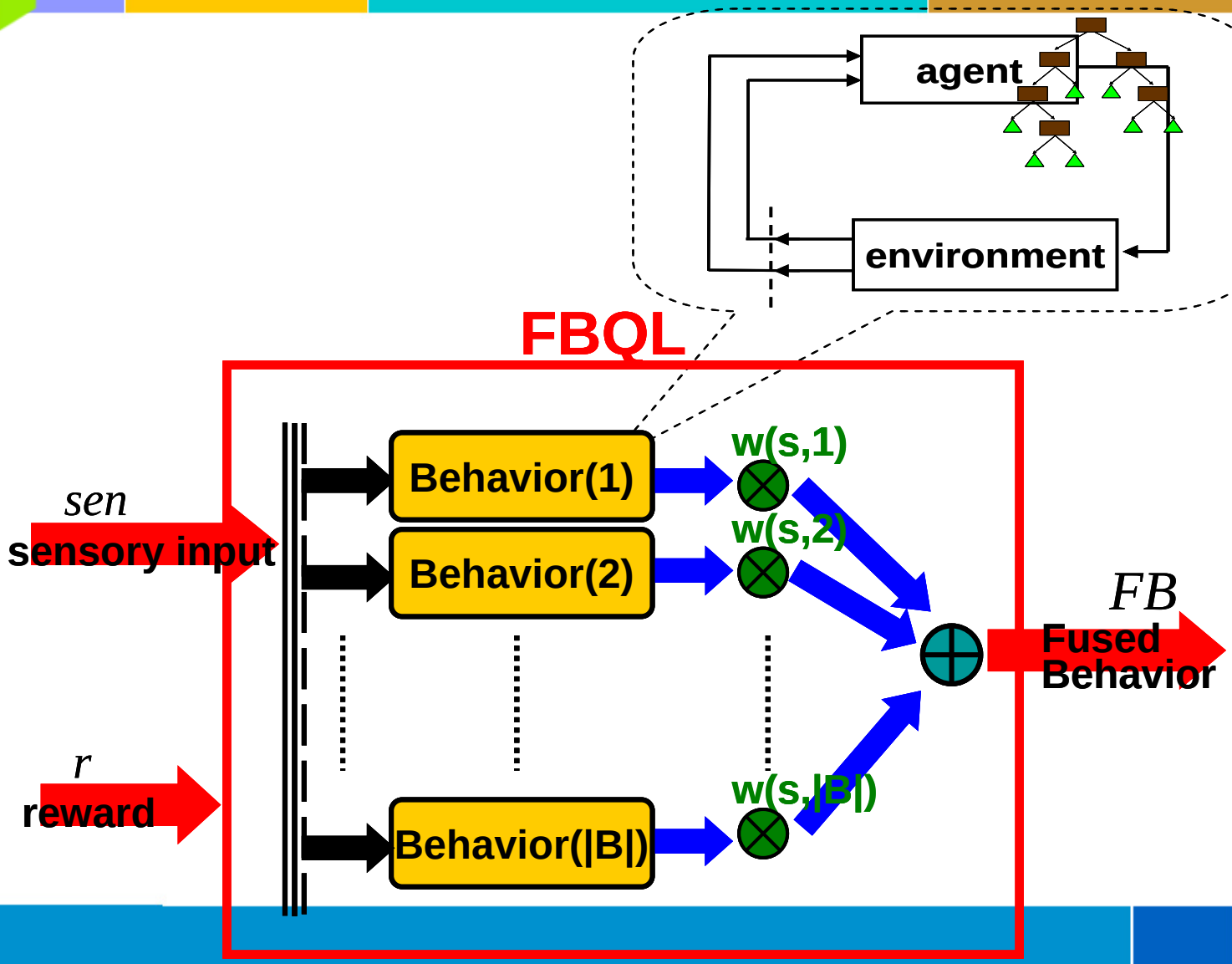
共37種類



技巧組合 – Skill Synthesis

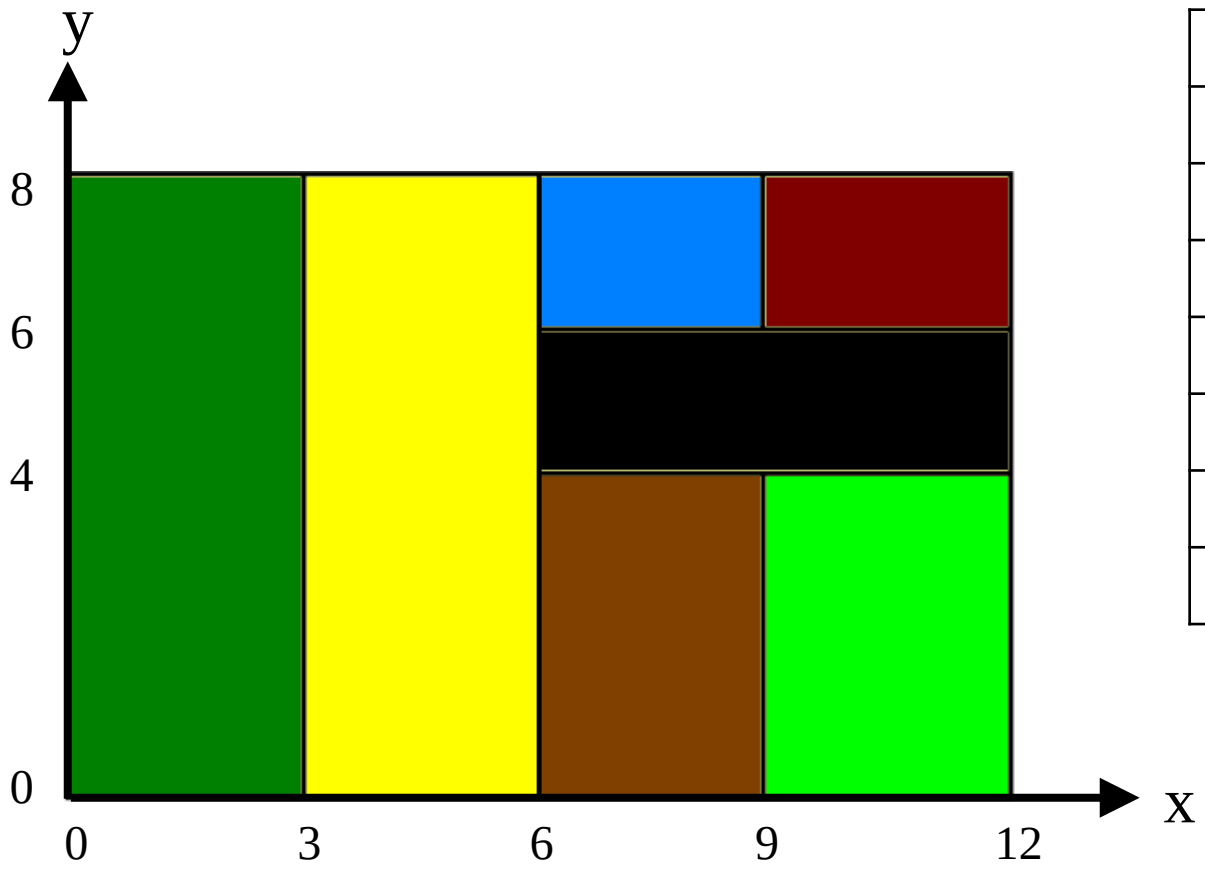


技巧組合學習演算法(FBQL)



技巧組合學習演算法(FBQL)

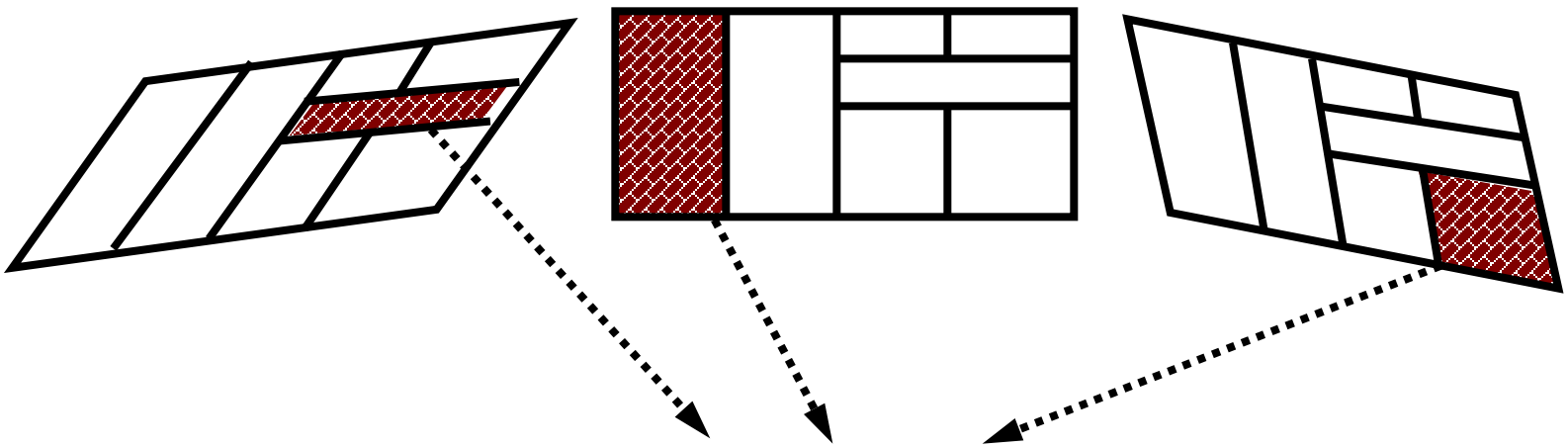
- 狀態表示法(RL-based Decision Tree)：



分類	表示範圍(x,y)
N1	(0,3,0,8)
N2	(3,6,0,8)
N3	(6,9,0,4)
N4	(9,12,0,4)
N5	(6,12,4,6)
N6	(6,9,6,8)
N7	(9,12,6,8)

技巧組合學習演算法(FBQL)

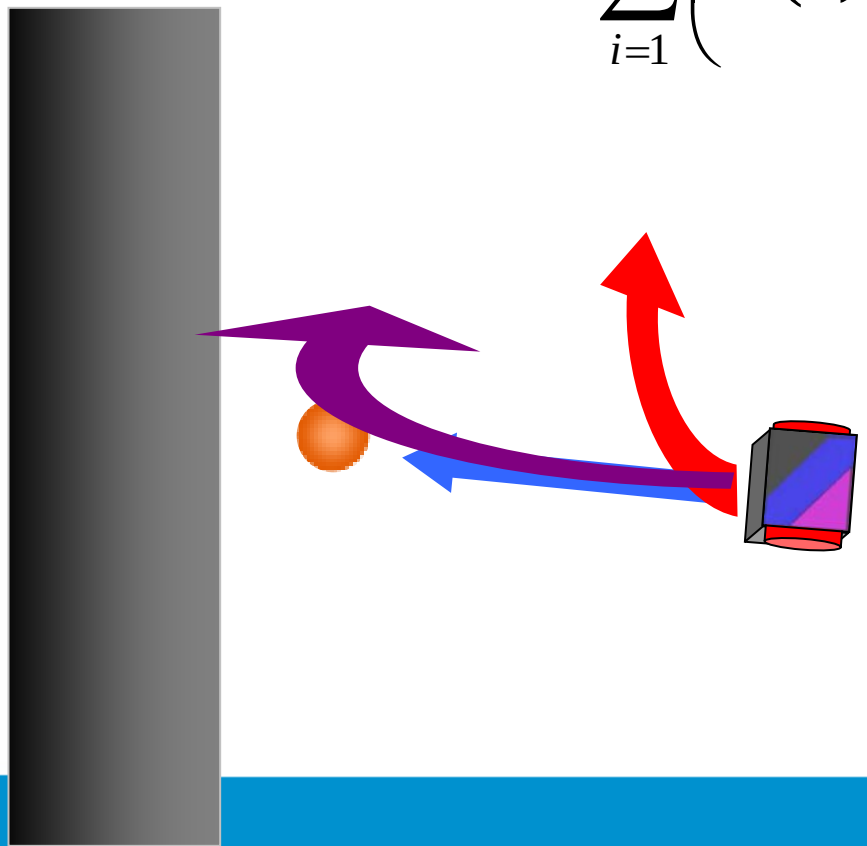
- 狀態表示法(FBQL)：



技巧組合學習演算法(FBQL)

- 融合輸出動作(Fused Behavior)：

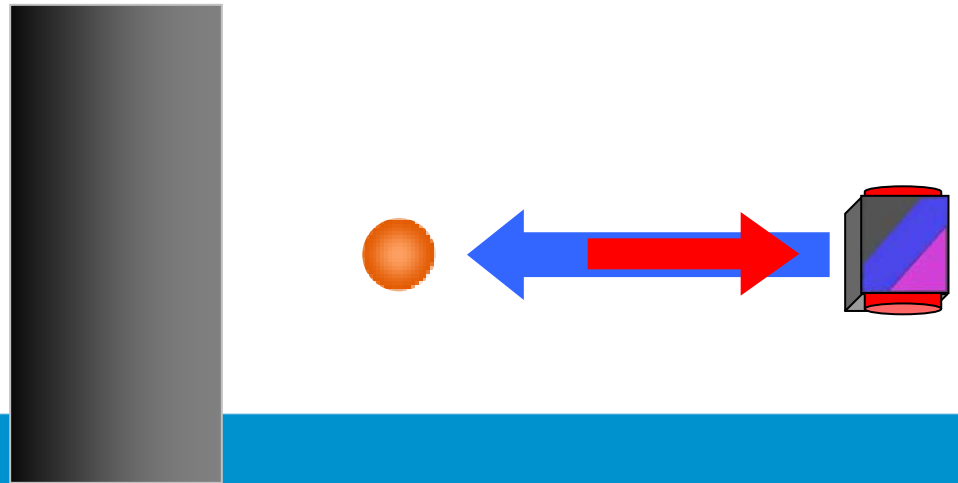
$$FB \leftarrow \sum_{i=1}^{|B|} \left(w(s,i) * B_i \right), \qquad \sum_{i=1}^{|B|} w(s,i) = 1$$



技巧組合學習演算法(FBQL)

- 回饋值(reward)：

	Dense reward	Sparse reward
設計難易度	困難	容易
學習速度	快	慢
學習系統複雜度	低	高
局部極小值問題	可能發生	不會發生



技巧組合學習演算法(FBQL)

- 評估值Q值更新：

$$Q(s,i) \leftarrow Q(s,i) + \alpha * \left(\underbrace{w(s,i) * r}_{\text{立即回饋值}} + \underbrace{\gamma * \sum_{j=1}^{|B|} \left(w(s',j) * Q(s',j) \right)}_{\text{累積回饋值}} - Q(s,i) \right)$$

技巧組合學習演算法(FBQL)

- 權重值W值更新：

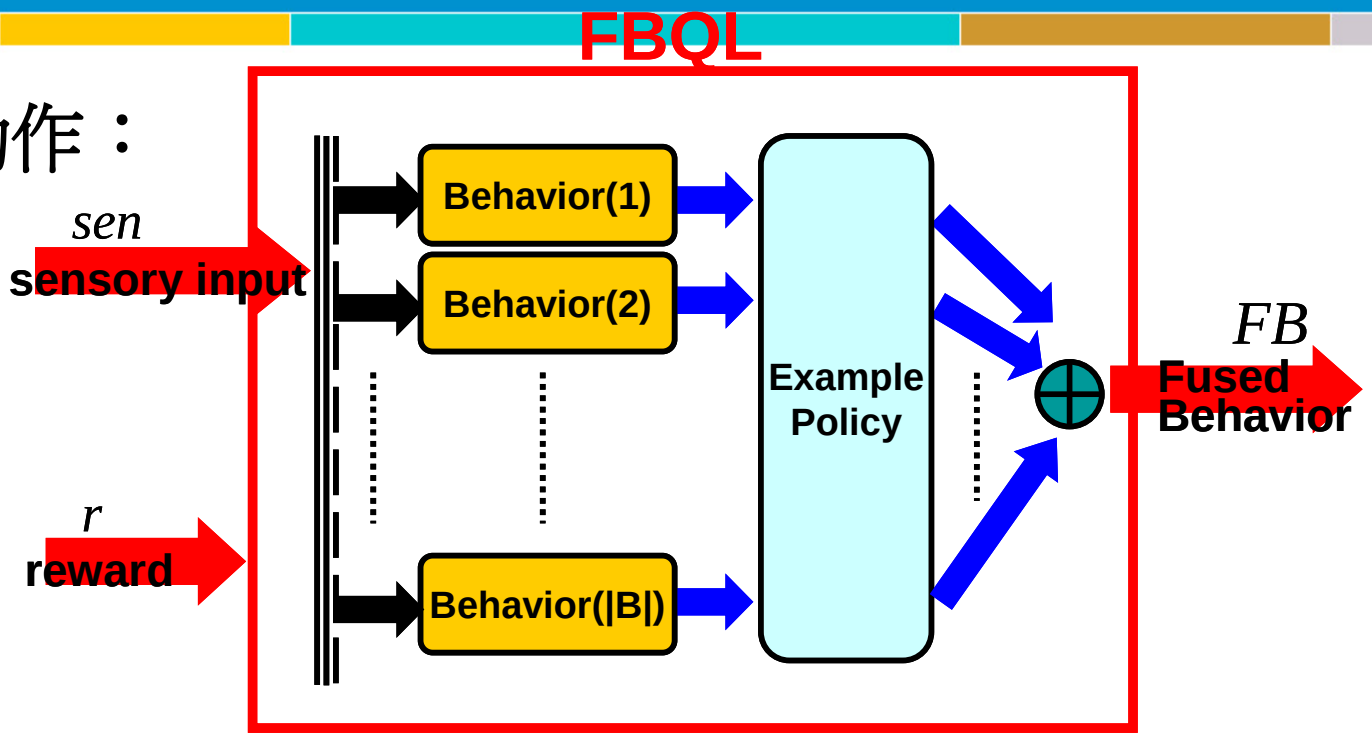
$$w(s,i) \leftarrow w(s,i) + \begin{cases} (1 - w(s,i)) * \delta * \left(Q(s,i) + f \right) & , if \quad i = \arg\max_i Q(s,i) \\ (0 - w(s,i)) * \delta * \left(Q(s,i) + f \right) & , otherwise \end{cases}$$

- 權重值W值正規化：

$$w(s,i) \leftarrow \frac{w(s,i)}{\sum_{j=1}^{|B|} w(s,j)} \qquad for\ all \quad i \in B$$

技巧組合學習演算法(FBQL)

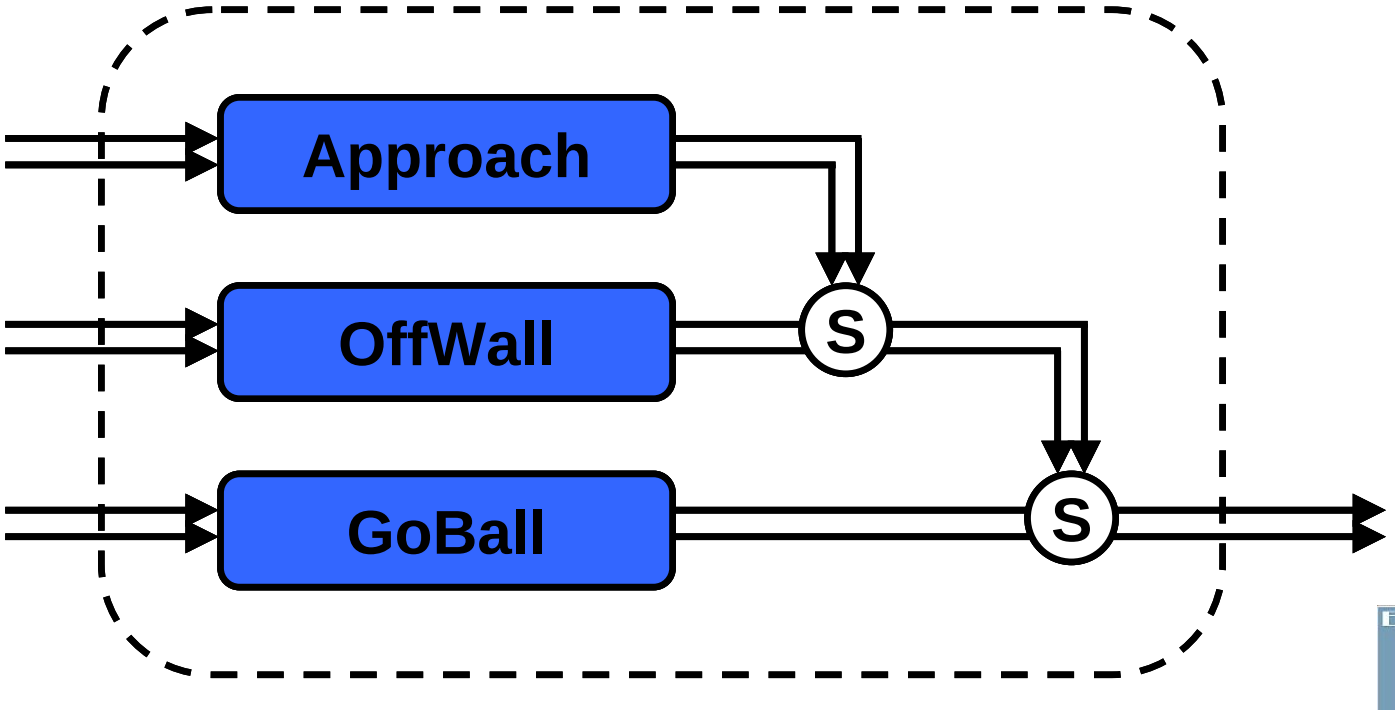
- 示範動作：



1. 為了加快FBQL的學習速度
2. 引領機器人去接觸到特定的回饋值
3. 節省隨機搜尋所空耗的時間
4. 不需要特殊的操控

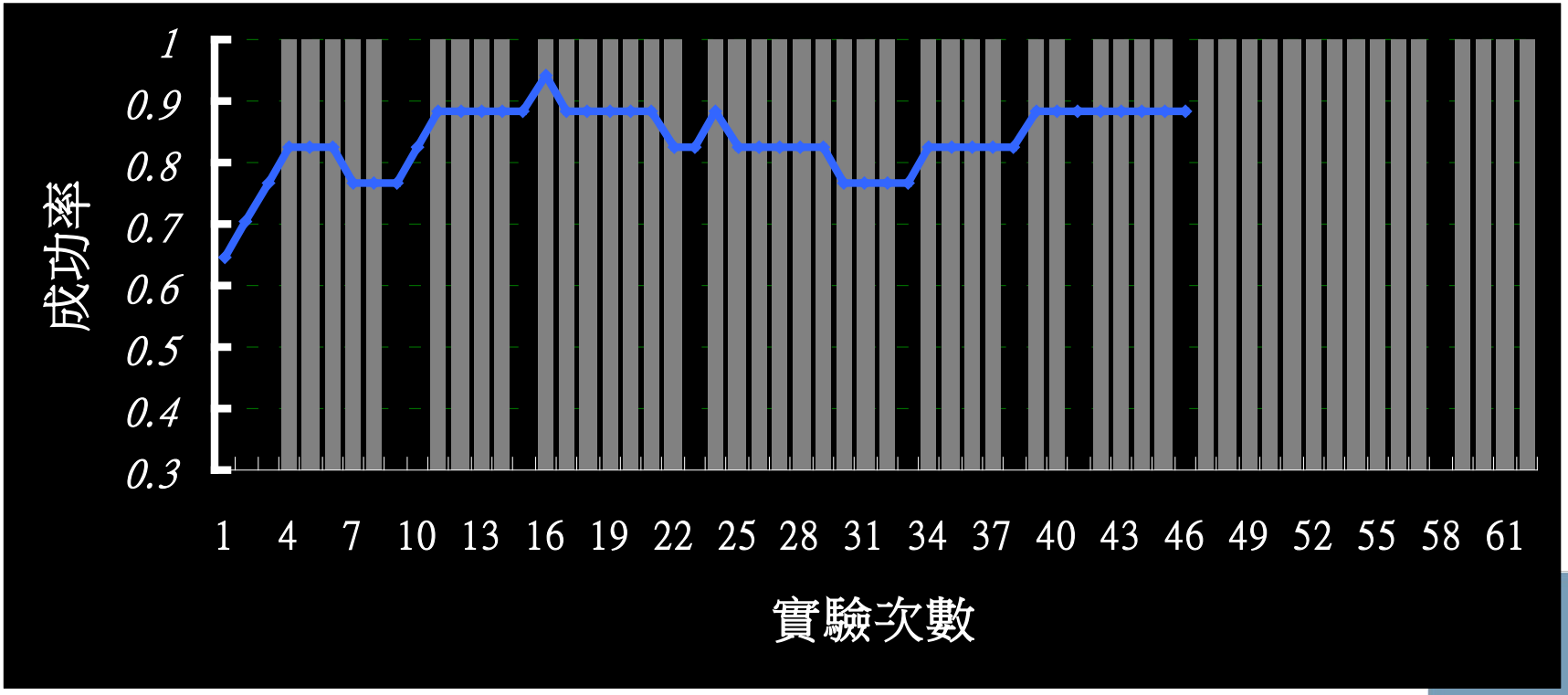
實驗設計與討論: Subsumption

- 以Subsumption Architecture的融合方式：



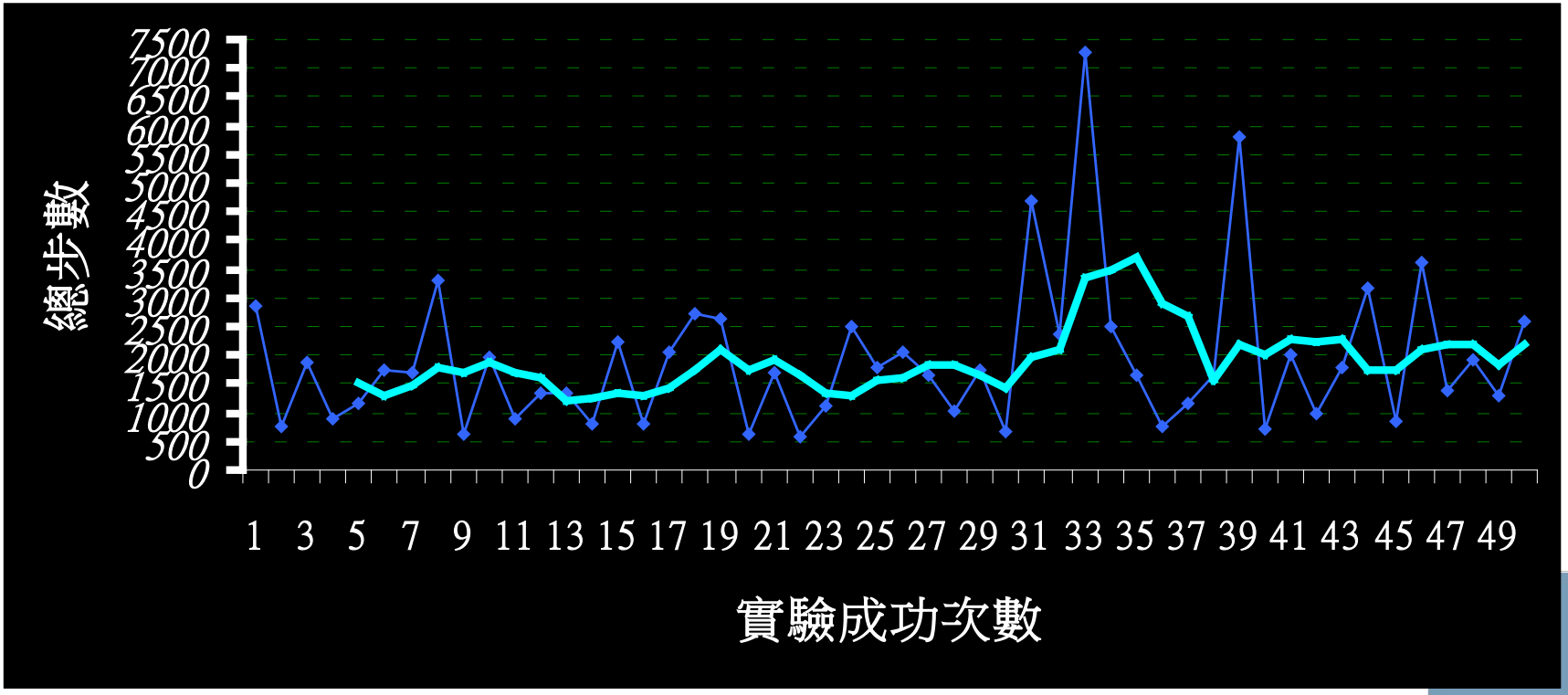
實驗設計與討論: Subsumption

- 成功率：

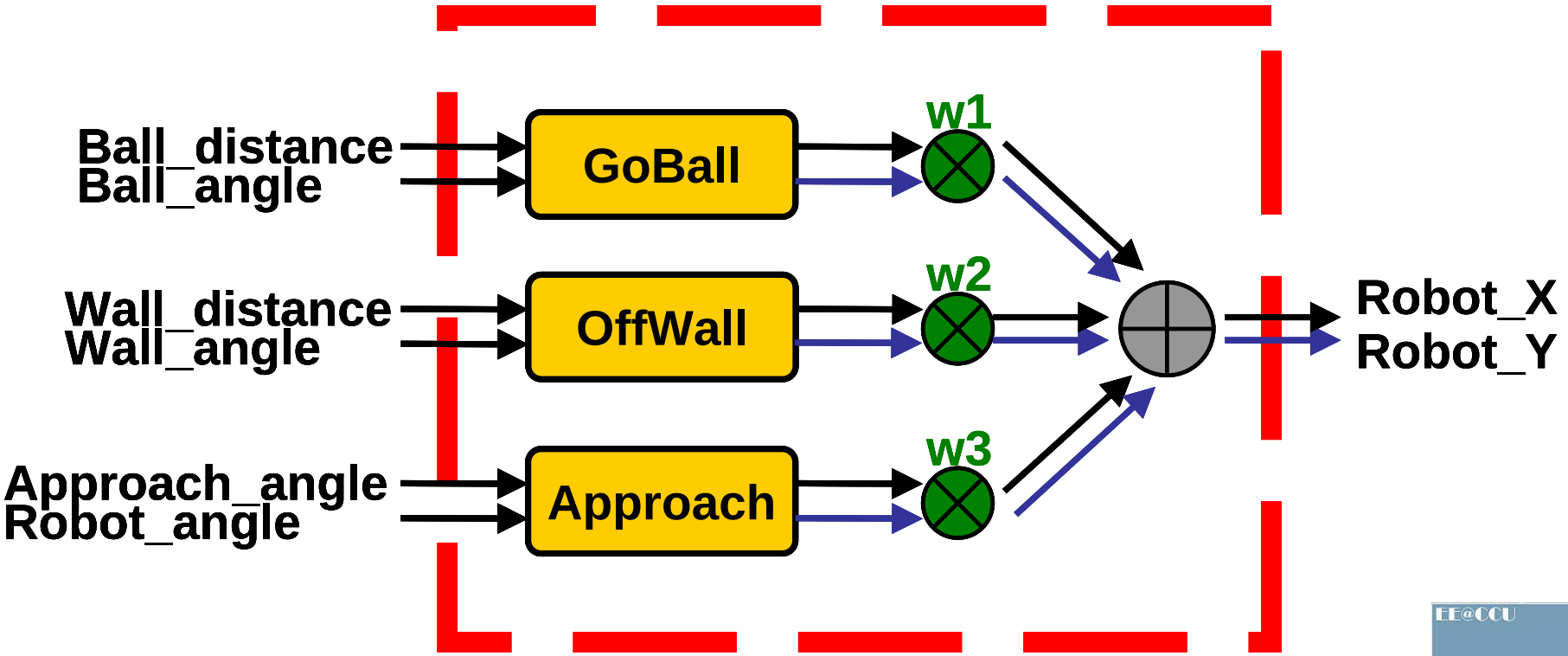


實驗設計與討論: Subsumption

- 成功總步數：

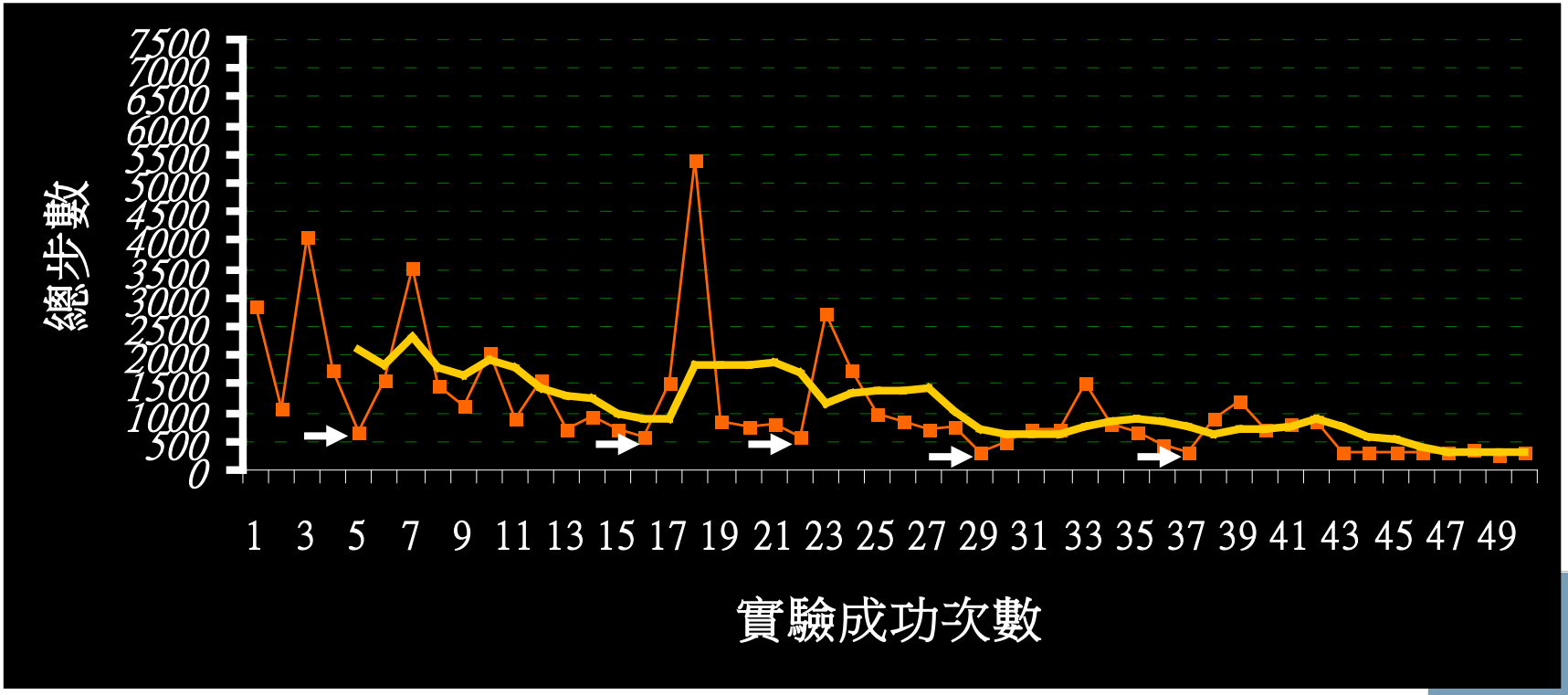


實驗設計與討論: FBQL



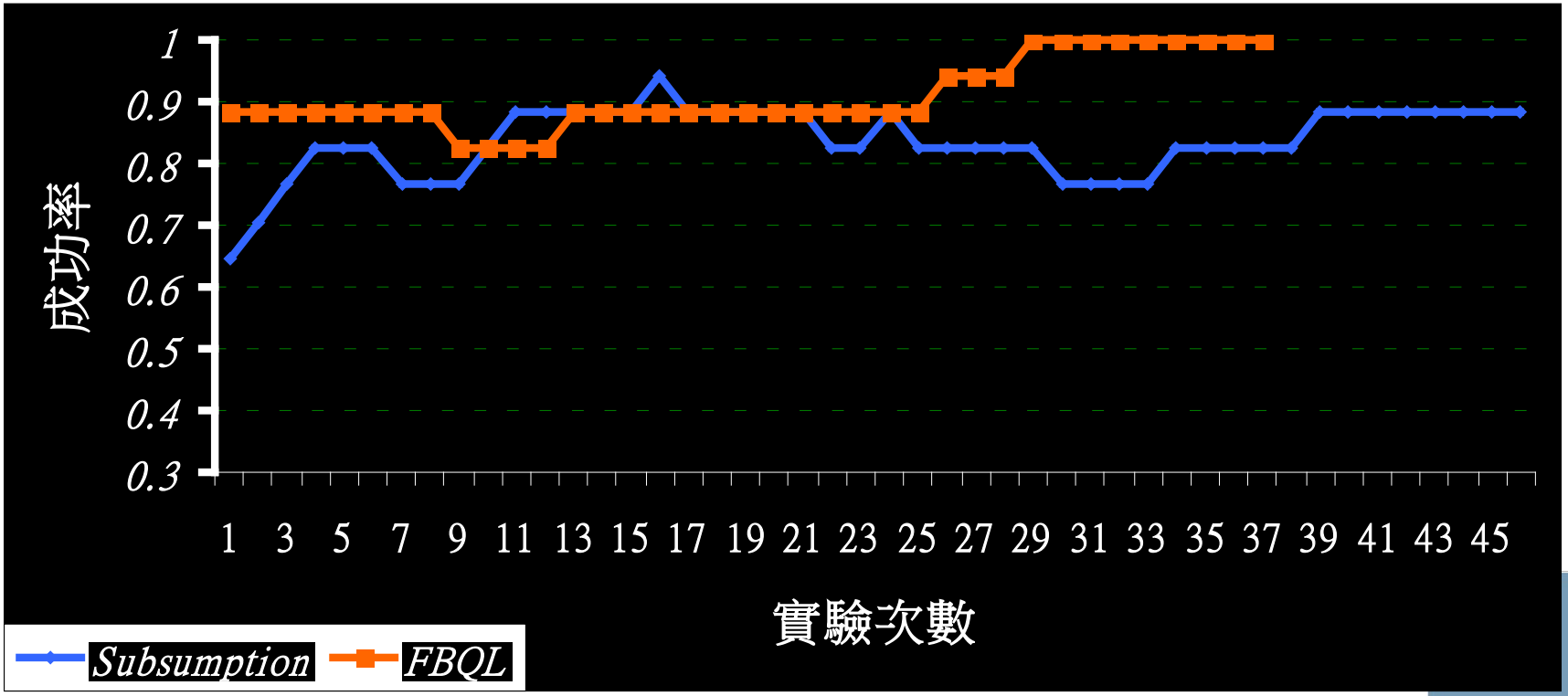
實驗設計與討論

- 成功總步數：



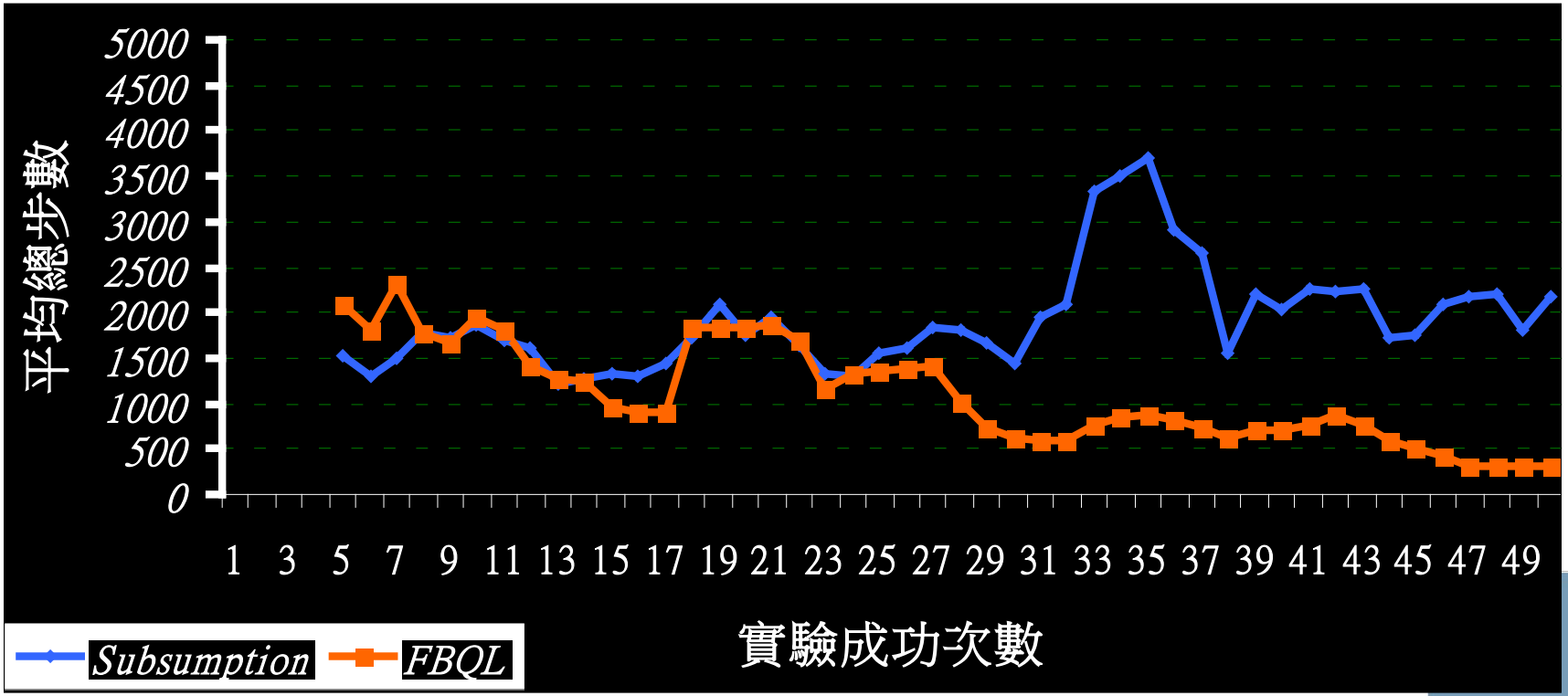
實驗設計與討論

- 成功率的比較：

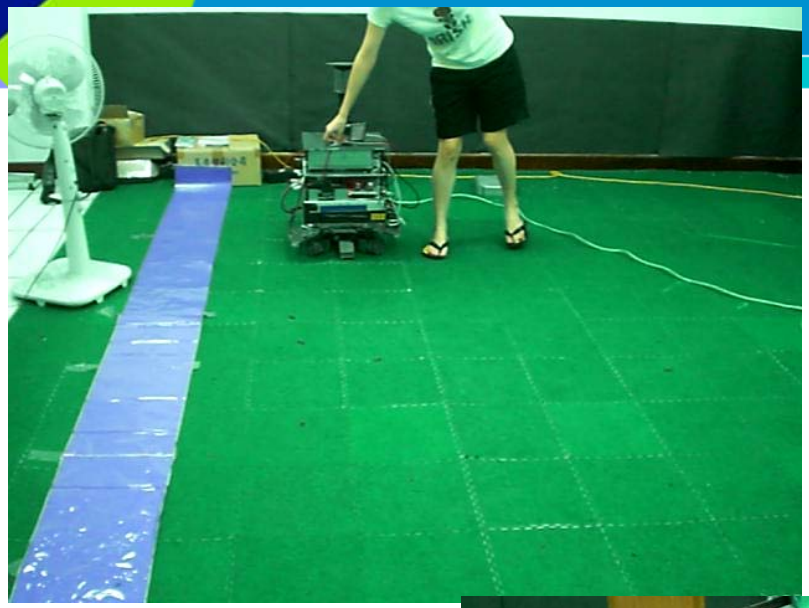


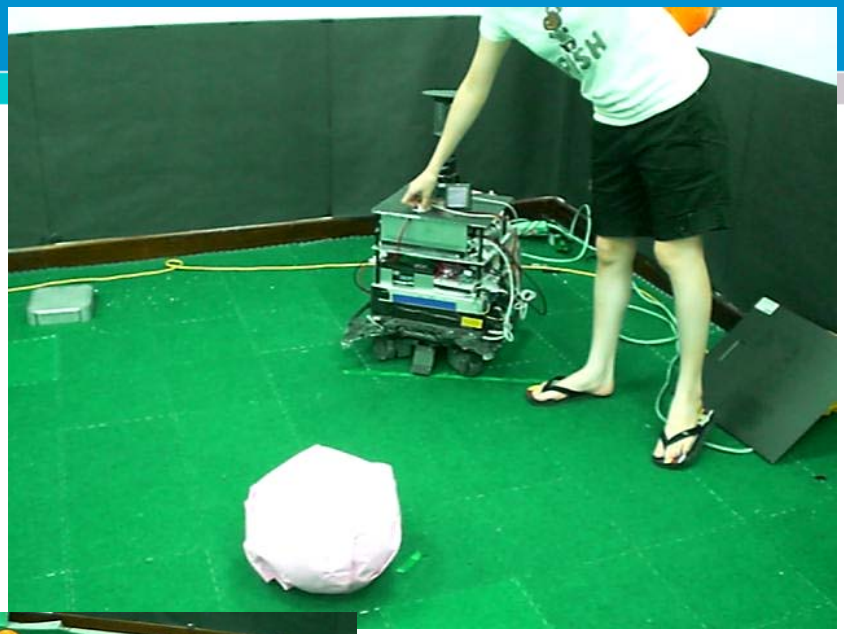
實驗設計與討論

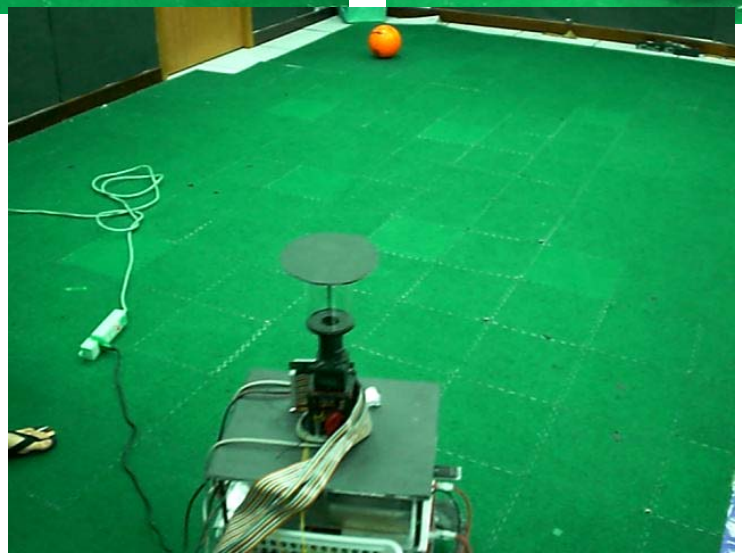
- 平均成功總步數的比較：



Fused three individual behaviors







實驗設計與討論

- **FBQL於撞牆率效果不佳的原因：**
  1. **FBQL重視長遠的目標**
  2. **Subsumption Architecture著重避牆行為**
  3. **FBQL希望找出更有效的移動方式**
  4. **FBQL的總步數很低(避牆率會被放大)**
  5. 先學完總步數之後，避牆率才會收斂