

# 國立交通大學

## 機械工程學系

### 碩士論文

ANFIS 應用於四足機器人跨越非連續河石地形路徑  
規畫之研究



Path Planning of Quadruped Robot Walking through Discontinuous  
River-stone Terrain Based on ANFIS

研究生：曾俊融

指導教授：鄭璧瑩 博士

中華民國九十八年七月

ANFIS 應用於四足機器人跨越非連續河石地形路徑規劃之研究

Path Planning of Quadruped Robot Walking through Discontinuous River-stone  
Terrain Based on ANFIS

研究生：曾俊融

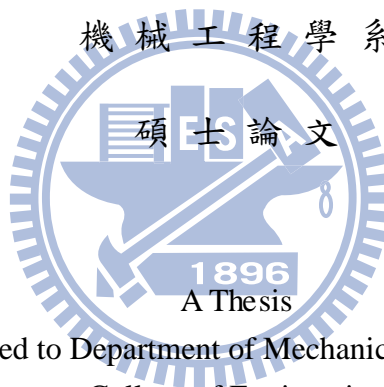
Student : Jyun - Rong Zeng

指導教授：鄭璧瑩 博士

Advisor : Dr. Pi-Ying Cheng

國立交通大學

機械工程學系



Submitted to Department of Mechanical Engineering  
College of Engineering

National Chiao Tung University  
in partial Fulfillment of the Requirements

for the Degree of

Master

In

Mechanical Engineering

July 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年七月

# ANFIS 應用於四足機器人跨越非連續河石地形路徑 規劃之研究

學生：曾俊融

指導教授：鄭璧瑩 博士

## 摘 要

本文研究目的為使用適應性網路模糊推論系統(ANFIS)演算法，應用於四足機器人跨越非平坦河石地形(River-stone Terrain)之路徑規劃，並搭配靜態 ZMP 穩定性分析，使機器人隨著不同地形做不同姿態調整，提高人工智慧步伐規劃的層次。

程式設計部分，首先針對本實驗室四足機器人(NC\_F4-09)架構中四組空間三自由度連桿手臂進行規劃研究，過程由簡入深，首先從平面三自由度連桿手臂開始做起，接著應用至空間三自由度連桿手臂，最後則是延伸至四足機器人做路徑規劃，處理平衡問題時使用 ZMP 理論，當行走時投影至地面的 ZMP 點，如果落在腳底支撐面上，行走過程中就能維持穩定。

本研究成果還包括以 MATLAB 為基礎所建立的步伐規劃自動產生的圖控式介面系統(GUI)，有助於簡化四足機器人跨越河石地形的步伐規劃與實際控制驗正。

實驗硬體方面主要採用壓克力板組成架構形狀，再加裝伺服馬達、7.2V 充電電池和馬達控制器(SSC-32)，經過測試調整後，完成四足機器人的實驗模組(NC\_F4-09)，四足機器人的步態規劃過程為透過 ANFIS 演算法，將

MATLAB 程式計算出的各軸運動數據，利用 RS232 傳輸線由電腦傳送至 SSC-32 中，讓四足機器人隨著不同使用者輸入的地形產生相對應的步伐控制數據，實際去跨越非連續或非平坦的河石地形。



關鍵字：模糊控制、倒傳遞類神經網路、零力矩點、四足機器人、適應性  
網路模糊推論系統

# Path Planning of Quadruped Robot Walking through Discontinuous River-stone Terrain Based on ANFIS

Student : Jyun - Rong Zeng

Advisor : Dr. Pi-Ying Cheng

## ABSTRACT

The study proposes an innovative method which combines Adaptive Neuro-Fuzzy Inference Systems(ANFIS) with Zero-Moment-Point(ZMP) algorithms and applies in path planning for quadruped robot walking through a discontinuous River-stone terrain. Therefore, the robot can change its walking figure for adapting to different terrain like humans.

Because the quadruped robot can be regarded as combinations of four 3-DOF manipulators and the ANFIS algorithm can also be regarded as combinations of Neural-Network and Fuzzy inference system, this thesis study is planned and proceeded from simple to difficult case. Firstly, we study the case of 3-DOF manipulator in a plane based on Fuzzy theorem. Second we compare the case of 3-DOF spatial manipulator based on both Neural-Network and ANFIS theorems. Finally this thesis proposes the ANFIS and ZMP methods to apply to the gait planning of 12-DOF quadruped robot.

This research deals with the stability control algorithms of quadruped robot by ZMP method, when the robot walks through discontinuous terrain. If its projection of ZMP on the ground is inside the triangle foot print area, the robot will keep stable on walking.

There are many experimental instruments were adoptive in the study, such as acrylic sheet, servo motors and SSC-32 servo controller. This gait planner transfers the angle datas made by ANFIS programs to the SSC-32 board by RS232 connected cable, and let the robot pass through the discontinuous River-stone successfully and stably.

Keywords: Fuzzy control, Error Back-Propagation, ZMP, Quadruped robot ANFIS.

## 誌 謝

在兩年的研究期間，最要感謝的是我的指導教授 鄭壁瑩博士給予我許多的指導。在學習過程中，無論是指導教授規劃的國科會計畫合作案、課程助教或是實驗室每週的會議裡，老師總是循序漸進的給予我指示事項，讓我學到的不僅僅是一個解決問題的方法，更是一種對處理問題的堅持及負責態度。此外 鄭老師對於未來工作的態度及許多經驗的分享上，都使我獲益良多，在平常生活中對我也是照顧有加，在這邊要對 鄭老師致上最深的謝意。

此外這兩年間非常感謝實驗室中的所有成員，學長健寶、皓光及俊宏，同學亮瑜、家宏、仲緣及志元，及學弟軒丞、詩漢、漢嘉、瑞昌及永修，此外還有在職專班的光耀、岳芳、勝龍，謝謝你們在生活與研究上都給我眾多的幫忙與協助，也讓我在這兩年間交到了許多的朋友。

另外，非常感謝論文口試委員所提出的建議與指教，讓本研究內容能夠更加充實、完善。

最後當然要感謝我的家人這兩年來在背後默默的支持我，讓我能夠無後顧之憂的完成這份學業，你們的辛苦我都知道，謝謝你們。

最後，要感謝的人實在太多了，謹於此，謝謝所有關懷我的人。

曾俊融謹致

# 目 錄

|  |     |
|--|-----|
| 摘 要 .....                                  | i   |
| ABSTRACT .....                             | iii |
| 誌 謝 .....                                  | iv  |
| 目 錄 .....                                  | v   |
| 表 目 錄 .....                                | vii |
| 圖 目 錄 .....                                | vii |
| 一、緒論 .....                                 | 1   |
| 1.1 研究背景與動機 .....                          | 1   |
| 1.2 文獻回顧 .....                             | 2   |
| 1.3 研究主題流程 .....                           | 7   |
| 1.4 論文架構 .....                             | 8   |
| 二、研究理論 .....                               | 9   |
| 2.1 基礎理論 .....                             | 9   |
| 2.1.1 零力矩點(ZMP)定義 .....                    | 9   |
| 2.1.2 機器人學座標系轉換 .....                      | 11  |
| 2.1.3 倒傳遞類神經網路(Back-Propagation ANN) ..... | 14  |
| 2.1.4 模糊控制(fuzzy control) .....            | 17  |
| 2.1.5 適應性類神經網路模糊推論系統 .....                 | 19  |
| 三、系統模擬軟體介紹 .....                           | 21  |

|                 |                              |    |
|-----------------|------------------------------|----|
| 3.1             | MATLAB 介面介紹 .....            | 21 |
| 3.2             | MATLAB – simulink .....      | 22 |
| 3.3             | MATLAB – 模糊工具箱 .....         | 23 |
| 3.4             | MATLAB – 倒傳遞類神經網路 .....      | 25 |
| 四、研究內容與方法 ..... |                              | 27 |
| 4.1             | 研究過程與架構 .....                | 27 |
| 4.2             | Fuzzy 應用至平面三連桿手臂 .....       | 28 |
| 4.3             | 倒傳遞神經網路應用至空間三連桿手臂 .....      | 34 |
| 4.4             | ANFIS 應用至空間三連桿手臂 .....       | 41 |
| 4.5             | 運算時間比較 .....                 | 49 |
| 4.6             | ANFIS 應用至 12-DOF 四足機器人 ..... | 51 |
| 4.7             | ZMP 平衡問題分析 .....             | 54 |
| 五、模擬與實驗結果 ..... |                              | 58 |
| 5.1             | 程式流程規劃 .....                 | 58 |
| 5.2             | 程式模擬結果 .....                 | 60 |
| 5.3             | 實驗硬體介紹 .....                 | 66 |
| 5.4             | 實驗結果 .....                   | 68 |
| 六、結論與未來展望 ..... |                              | 76 |
| 6.1             | 結論 .....                     | 76 |
| 6.2             | 未來展望 .....                   | 78 |



|           |    |
|-----------|----|
| 參考文獻..... | 79 |
|-----------|----|

## 表 目 錄

|   |    |
|---|----|
| 表 4.1 空間三連桿手臂 D-H 表.....                  | 35 |
| 表 4.2 不同轉移函數輸出角度誤差比較.....                 | 37 |
| 表 4.3 驗證樣本與BP輸出前 20 組比較.....              | 39 |
| 表 4.4 驗證樣本 與 ANFIS輸出 前 20 組比較.....        | 47 |
| 表 4.5 10 組比較樣本運算時間.....                   | 50 |
| 表 4.6 比較樣本、反向運動學和ANFIS 輸出結果比較.....        | 50 |
| 表 5.1 實際量測角度與座標值.....                     | 60 |
| 表 5.2 實際量測座標與D-H理論座標比較.....               | 61 |
| 表 5.3 Visual sequencer角度與ANFIS輸出角度比較..... | 62 |
| 表 5.4 伺服馬達規格.....                         | 67 |

## 圖 目 錄

|                               |   |
|-------------------------------|---|
| 圖 1.1 3-DOF手臂及其模糊控制架構圖.....   | 2 |
| 圖 1.2 兩足機器人與腳底力感測器圖.....      | 3 |
| 圖 1.3 求解ZMP值之流程圖.....         | 3 |
| 圖 1.4 反向運動學和反向動力學之類神經架構圖..... | 4 |
| 圖 1.5 足球機器人之模糊控制角色分配圖.....    | 5 |

|       |  |    |
|-------|--|----|
| 圖 1.6 | SCARA機械手臂控制方塊圖 .....                                   | 6  |
| 圖 1.7 | ANFIS應用於機械手臂控制方塊圖 .....                                | 6  |
| 圖 2.1 | 零力矩點位置圖 .....  | 9  |
| 圖 2.2 | 球面轉動之參考座標系與移動座標系關係圖.....                               | 11 |
| 圖 2.3 | 一般空間移動之參考座標系與移動座標系關係.....                              | 13 |
| 圖 2.4 | 倒傳遞類神經網路架構圖 .....                                      | 14 |
| 圖 2.5 | 各種轉移函數圖 .....  | 15 |
| 圖 2.6 | 模糊控制方塊圖 .....  | 18 |
| 圖 2.7 | ANFIS架構圖 .....   | 19 |
| 圖 3.1 | MATLAB介面視窗 .....                                       | 22 |
| 圖 3.2 | Simulink範例介紹圖 .....                                    | 23 |
| 圖 3.3 | fuzzy GUI介面 .....                                      | 23 |
| 圖 4.1 | 平面(a)和空間(b) 三連桿手臂示意圖 .....                             | 28 |
| 圖 4.2 | 四足機器人 3D圖(CIDM實驗室設計圖).....                             | 28 |
| 圖 4.3 | 三連桿手臂之Fuzzy架構圖 .....                                   | 29 |
| 圖 4.4 | 工作空間 .....   | 30 |
| 圖 4.5 | 輸入角度( $\theta_1$ 、 $\theta_2$ 、 $\theta_3$ )歸屬函數 ..... | 30 |
| 圖 4.6 | x座標歸屬函數.....   | 31 |
| 圖 4.7 | y座標歸屬函數.....   | 31 |

|        |  |    |
|--------|--|----|
| 圖 4.8  | Fuzzy輸出結果 .....  | 33 |
| 圖 4.9  | 空間三連桿尺寸與座標圖 .....  | 34 |
| 圖 4.10 | 3-DOF手臂之BP架構圖 .....  | 37 |
| 圖 4.11 | 轉移函數logsig .....   | 38 |
| 圖 4.12 | 20 組驗證樣本與BP角度比較圖 .....   | 40 |
| 圖 4.13 | ANFIS訓練架構圖 .....   | 42 |
| 圖 4.14 | 不同a、b、c、d、 $\sigma$ 值時的gbellmf與gaussm .....                              | 43 |
| 圖 4.15 | 學習後ANFIS1 輸入的歸屬函數 .....  | 45 |
| 圖 4.16 | 學習後ANFIS2 輸入的歸屬函數 .....  | 46 |
| 圖 4.17 | 學習後ANFIS3 輸入的歸屬函數 .....  | 46 |
| 圖 4.18 | 20 組驗證樣本與ANFIS角度比較圖 .....  | 48 |
| 圖 4.19 | ANFIS和反向運動學之時間與誤差比較流程 .....  | 49 |
| 圖 4.20 | 四足機器人座標系與步伐位置 .....  | 52 |
| 圖 4.21 | 踏步順序 .....   | 53 |
| 圖 4.22 | anfis1 求解馬達角度 $\theta_1$ 、 $\theta_4$ 、 $\theta_7$ 、 $\theta_{10}$ ..... | 53 |
| 圖 4.23 | anfis2 求解馬達角度 $\theta_2$ 、 $\theta_5$ 、 $\theta_8$ 、 $\theta_{11}$ ..... | 53 |
| 圖 4.24 | anfis3 求解馬達角度 $\theta_3$ 、 $\theta_6$ 、 $\theta_9$ 、 $\theta_{12}$ ..... | 54 |
| 圖 4.25 | ZMP穩定多邊形 .....   | 55 |
| 圖 4.26 | ZMP理論示意圖 .....   | 56 |

|        |                          |    |
|--------|--------------------------|----|
| 圖 4.27 | 四足機器人調整重心示意圖.....        | 57 |
| 圖 5.1  | 方法一 示意圖.....             | 58 |
| 圖 5.2  | 方法二 示意圖.....             | 59 |
| 圖 5.3  | 方法三 示意圖.....             | 59 |
| 圖 5.4  | 地形俯視圖.....               | 63 |
| 圖 5.5  | ANFIS GUI介面與實驗踏點圖.....   | 64 |
| 圖 5.6  | 機身與下一步踏點的相對座標.....       | 65 |
| 圖 5.7  | ANFIS計算馬達角度之程式流程圖.....   | 66 |
| 圖 5.8  | 伺服馬達.....                | 67 |
| 圖 5.9  | SSC-32 motor driver..... | 67 |
| 圖 5.10 | 腳底防滑套.....               | 67 |
| 圖 5.11 | 實驗硬體架構(CIDM實驗室設計).....   | 68 |
| 圖 5.12 | 實驗規劃.....                | 68 |
| 圖 5.13 | 平坦地形直線路徑.....            | 69 |
| 圖 5.14 | 平坦地形任意路徑.....            | 71 |
| 圖 5.15 | 非平坦地形直線路徑.....           | 73 |
| 圖 5.16 | 非平坦地形任意路徑.....           | 75 |

# 一、緒論

## 1.1 研究背景與動機

美國、日本與歐洲等許多國家投入機器人技術研究已數十年，早期主要應用在工業機器人，在生產中代替人類從事危險、精密或單調重覆性作業，近年來因先進國家高齡化社會的演變，加上機器人相關技術更加成熟，智慧型服務機器人趁勢而起，其未來產值將超越工業機器人。

在 1997 年 5 月年，IBM 研發出一台超級電腦「深藍」，在當時舉辦的西洋棋大賽以 1 分之差擊敗當時棋王 Kasparov，這件事情震驚全世界，可見超級電腦「深藍」具有「人工智慧」，具有可自行思考、學習及判斷的能力，故若將人工智慧演算法應用至機器人，便可使機器人代替人類做更多具有智慧的事情。

類神經網路及模糊理論的共同特性，兩者均是用來模擬人類的大腦學習模式，類神經網路所扮演的角色是模仿大腦中的神經元，即生理結構較低階的部份。相對的，模糊理論則是用來模擬人類大腦的思考與推理，即心理結構較高階的部份，若把類神經網路比喻成大腦的硬體，則模糊邏輯就好像軟體部份，ANFIS 結合兩者的優點，便能以數學函數描述人類的思考行為。

相較於傳統控制，類神經網路與模糊系統可在不需知道一個系統的數學模式下，便能成功的估測此一系統的能力。由於兩者均採用數值的方法

來解決問題，使得我們可以用工程軟體來處理，例如：matlab、visual c++。在面對多自由度的機械結構，要處理的計算公式相當冗長、複雜，而且幾乎都是非線性函數，故通常要簡化公式或做些假設，但簡化或假設後的結果與理論值誤差變更大，甚至是不符合實際狀況，因此簡化和假設是較不可行的，本文採用 ANFIS 演算法，以學習和推論的方式近似原本複雜的運動學公式，使模擬值更接近實際輸出值。

## 1.2 文獻回顧

Y. L. FU等人在[1]中藉由距離感測器測量末端手臂與障礙物的距離，並和各關節角度變化量一起視為模糊控制的輸入，經過模糊控制規則庫推論(IF-THEN)後，求得三個關節的扭力，使三連桿手臂在未知環境中成功達到立即(real-time)閃躲障礙物的功能，圖1.1為模糊控制的輸入和對應的輸出之架構圖。

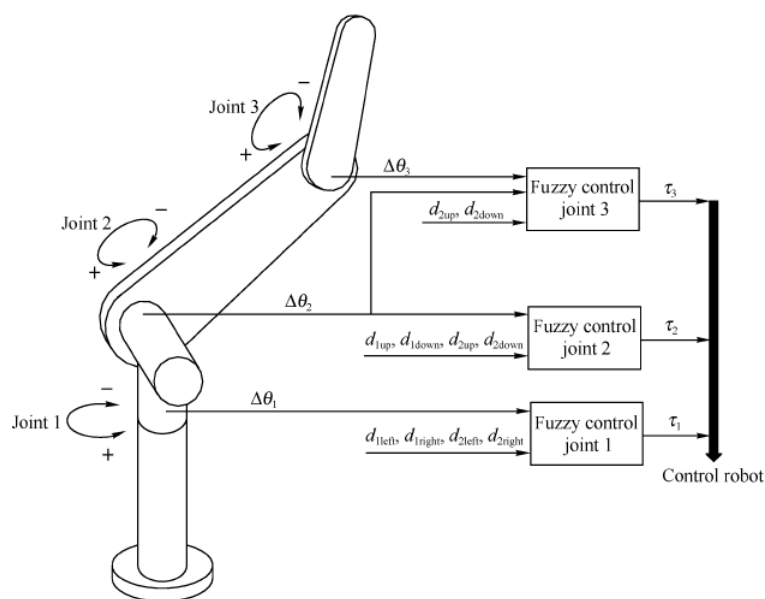


圖 1.1 3-DOF 手臂及其模糊控制架構圖

K. C. Choi 等人在[2]中，將十二個自由度的兩足機器人底部各安裝四個力感測器，藉由感測器量測值，透過類比數位轉換代入公式求解 ZMP 值，誤差值利用模糊理論補償至最佳，行走時 ZMP 值若落在腳底支撐面上，即可順利讓雙足機器人走過斜坡而不跌倒，參考圖 1.2。



圖 1.2 兩足機器人與腳底力感測器圖

C. L. Shih 等人在[3]中將兩足機器人腳底和腰部三點( $P_0$ 、 $P_6$ 、 $P_{12}$ )座標，代入正、反向運動學及牛頓-尤拉方程式求出各桿件的力 $F_i$ 和力矩 $M_i$ ，再藉由 ZMP 公式求得 x、y、z 方向的 ZMP 值，最後結合最佳化的方法規劃出最適合的路徑，圖 1.3 即為其流程圖。

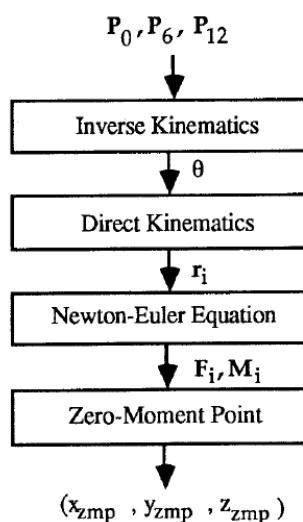


圖 1.3 求解 ZMP 值之流程圖

**Ken Onozato and Yutaka Maeda** 在[4]中將類神經網路架構應用至二連桿 **SCARA Robot**，經過不斷學習調整架構中參數使誤差慢慢收斂，學習結果成功取代逆向運動學與逆向動力學的公式，如圖1.4所示；因此，類神經理論可在不知控制的數學函數情況下，也能得到期望的輸出。

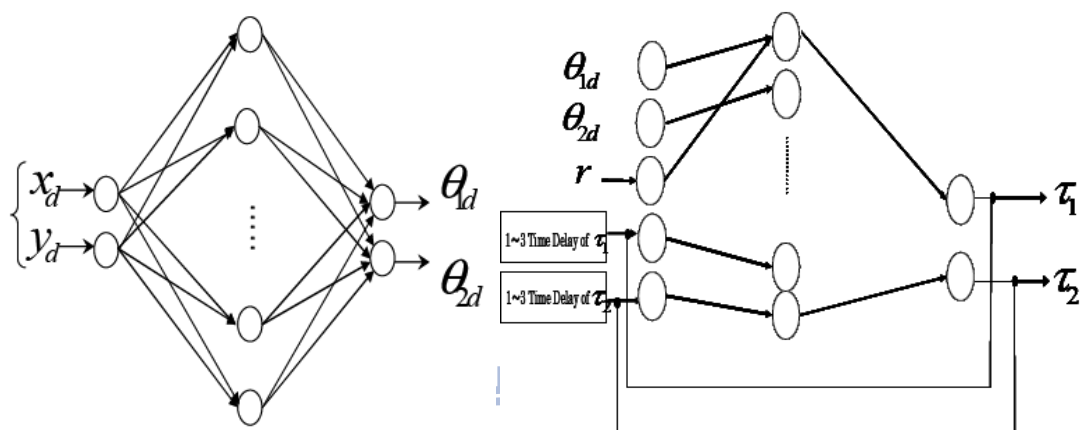


圖 1.4 反向運動學和反向動力學之類神經架構圖

**Bojun Ma** 等人在 [5] 中將五個自由度的操作手臂與三個自由度的移動平台結合，在處理多自由度的系統時，必須面對相當冗長的計算公式，**Bojun Ma** 等人針對具有冗餘(Redundant)自由度的移動操作臂提出一種反向運動學求解策略，和兩個假設條件，有效解決冗餘自由度分配的問題，也簡化了複雜的計算公式。

**Farzad Pourhohrat** 等人在 [6] 中將類神經網路應用於多自由度操作臂的反向運動學問題，但是求解結果會出現無窮多解，故作者加入限制條件使結果恰有一解，再利用 *lagrange multipliers* 把原本限制問題轉換為非限制問題，最後藉由學習演算法修正類神經架構裡的權重值，使 *energy function* 的值至最小。



**Prahlad Vadakkepat** 等人在 [7] 中將模糊控制應用至足球機器人，首先分配每個機器人的角色，Attacker、Midfielder、Defender、Goalie，而每個角色所執行的動作都不同，機器人可依照給予的角色去防守或攻擊，建立完模糊架構後，便能讓每個機器人精確地達到想要的目的，如圖1.5所示。

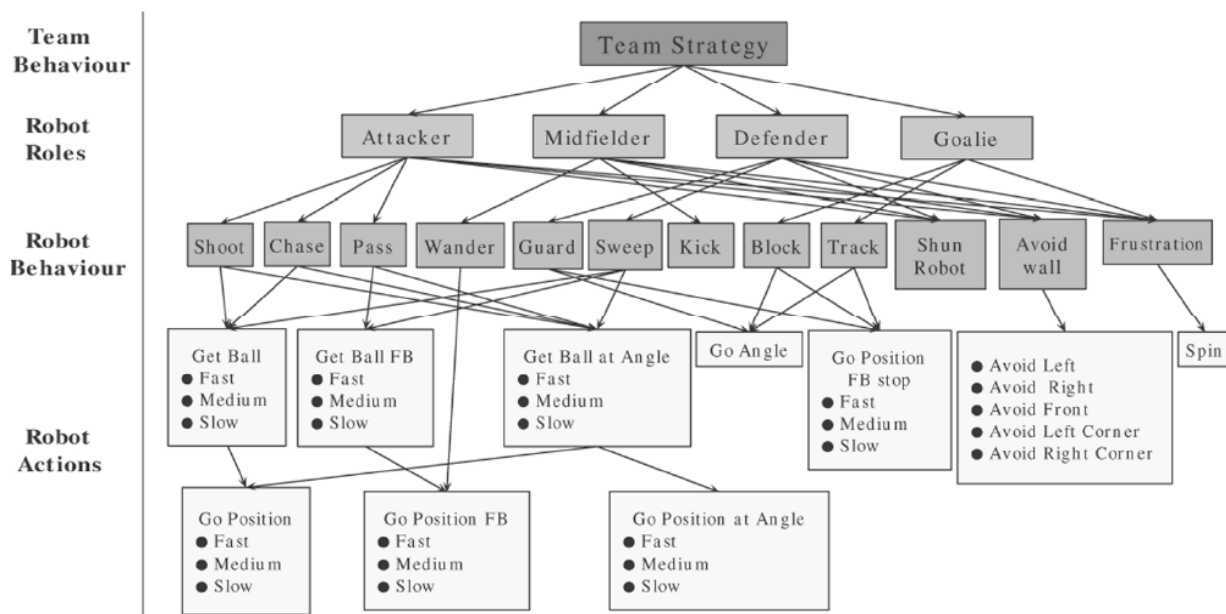


圖 1.5 足球機器人之模糊控制角色分配圖

**Prabu D** 等人在 [8] 中使用ANFIS控制器應用至三自由度 SCARA 機械手臂，對於不確定控制系統的數學函數，或是不確定機械手臂的結構時，只要我們有輸入和對應的輸出(Desired Trajectory)，便可以藉由ANFIS 架構經過學習而控制不確定的系統，ANFIS結合了模糊邏輯和類神經網路的優點，模糊邏輯可視為大腦的思考、推理生理部份，而類神經網路則為大腦神經元結構部份，論文中模擬部分使用 MATLAB來實現，圖1.6為其控制方塊圖。

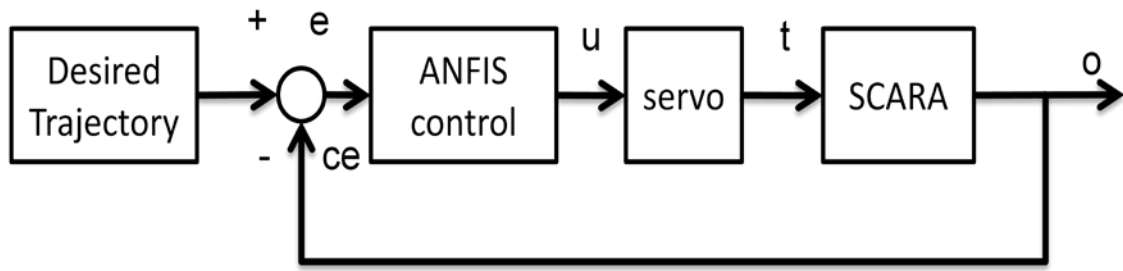


圖 1.6 SCARA 機械手臂控制方塊圖

L. X. Wei 等人在 [9] 中將 ANFIS 應用在求解反向運動學，只要有手臂末端位置座標，便能求解出馬達轉動角度，文章中結論提到模擬結果顯示此方法有更快速的學習效率，精準度也提高，且具有更好的及時處理能力，圖 1.7 為控制方塊圖。

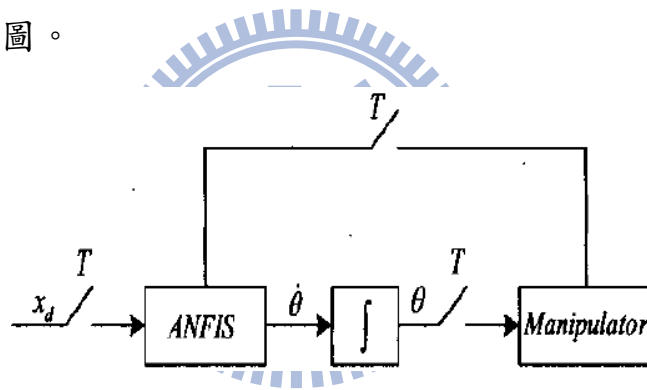


圖 1.7 ANFIS 應用於機械手臂控制方塊圖

Amar KHOUKHI 等人在 [10] 中將多層模糊-類神經架構應用於三連桿操作臂，ANFIS 架構主要分為五層，第一層模糊化、第二層模糊推論、第三層正規化、第四層解模糊化、第五層做加總，其輸入為馬達角度和角速度，輸出為馬達扭力。

### 1.3 研究主題流程

關於行走式機器人路徑規劃的研究目標，依據本實驗室訂定的研究主題與功能，研發步驟大致規劃為三個階段：

第一階段：發展一套路徑規劃方法應用於平坦的河石地形。

第二階段：加深地形的難度為非平坦的河石地形，以達到爬坡的需求。

第三階段：一舉將傾斜的角度拉大，甚至達到攀岩的效果。

本研究主旨在發展一套實用的 ANFIS 路徑規劃方法，能夠應用在行走式四足機器人上，並使用 MATLAB 來模擬其方法的可行性及穩定度，最後再用實驗加以驗證，以下規劃大致的研究流程為：

1. 以第一階段的路徑規劃法為基礎進行修改，使其能夠應用在非平坦河石地形
2. 建立四足機器人座標系與運動學分析
3. 建立自適應性類神經網路模糊推論系統架構
4. 搭配 ZMP 理論解決平衡問題
5. 撰寫 MATLAB 路徑規劃程式與 GUI 圖形化介面程式
6. 模擬與實驗結果討論

## 1.4 論文架構

### 第一章『緒論』

說明本論文研究之動機與背景、文獻回顧、研究流程與論文架構的介紹。

### 第二章『研究理論』

闡述有關機器人路徑規劃之理論介紹，如 ZMP、座標轉換、模糊理論、倒傳遞類神經網路及 ANFIS 的介紹。

### 第三章『系統模擬軟體介紹』

對 MATLAB、Simulink、模糊工具箱和倒傳遞類神經網路的視窗和一些簡單指令介紹。

### 第四章『研究內容與方法』

說明將倒傳遞類神經網路、模糊控制和 ANFIS 分別應用至三連桿手臂和四足機器人的詳細流程。

### 第五章『模擬與實驗結果』

說明將 MATLAB 模擬結果數據，匯入馬達控制器，觀察四足機器人行走非平坦河石地形時，ZMP 點是否落在支撐腳所形成的三角形內，實際的跨越動作，再以四足機器人實體的展示來驗證所提出的方法。

### 第六章：『結論與未來方向』

闡述本研究現階段研究成果，及下一階段論文研究方向。



## 二、研究理論

### 2.1 基礎理論

本章節簡單介紹了機器人相關的應用原理，包含了 ZMP 的定義、座標系轉換、類神經網路、模糊邏輯與 ANFIS 的介紹。

#### 2.1.1 零力矩點(ZMP)定義

零力矩點(Zero Moment Point)的概念最初在 1969 年被 Timcenko 及 Vukobratovic 提出，在 1984 年，第一次被應用在人形機器人上，從那時開始，ZMP 就成為動態平衡步行的指標。ZMP 的定義為地面上一點的慣性力和重力之淨力矩(net moment)無水平軸的分量  $M_x$  和  $M_y$  則稱此點為零力矩點，當 ZMP 落在人形機器人腳底所形成的支撐面內，則機器人可維持平衡不摔倒。而地面對機器人反作用力的合力所作用的點，則稱為壓力中心 (Center of Pressure)，當機器人步行維持平衡時，壓力中心和零力矩點是重疊的，如圖 2.1 所示

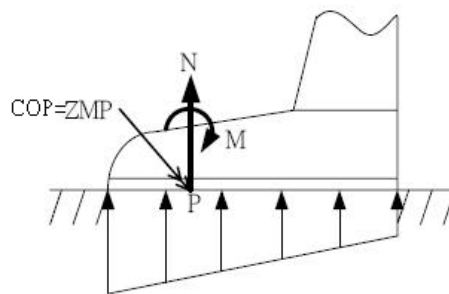


圖 2.1 零力矩點位置圖

ZMP 的計算公式如下：

$$\begin{aligned}
 x_{ZMP} &= \frac{\sum_{i=0}^n m_i (\ddot{z}_i + g) x_i - \sum_{i=0}^n m_i \ddot{x}(z_i - z_{ZMP}) - \sum_{i=0}^n I_{iy} \ddot{\theta}_{iy} + F_{ext,x} z_f}{\sum_{i=0}^n m_i (\ddot{z}_i + g)} \\
 y_{ZMP} &= \frac{\sum_{i=0}^n m_i (\ddot{z}_i + g) y_i - \sum_{i=0}^n m_i \ddot{y}(z_i - z_{ZMP}) - \sum_{i=0}^n I_{ix} \ddot{\theta}_{ix} + F_{ext,y} z_f}{\sum_{i=0}^n m_i (\ddot{z}_i + g)}
 \end{aligned} \tag{2-1}$$

其中

$m_i$ ：第  $i$  桿的質量

$(x_{ZMP}, y_{ZMP}, z_{ZMP})$ ：ZMP 的座標

$(x_i, y_i, z_i)$ ：第  $i$  桿件質心座標

$g$ ：重力加速度

$F_{ext,x}, F_{ext,y}$ ：外力干擾的分力

$I_{ix}, I_{iy}$ ：第  $i$  桿件慣性力的  $x$  軸  $y$  軸分量

$\ddot{\theta}_{ix}, \ddot{\theta}_{iy}$ ：第  $i$  桿件的  $x$  軸和  $y$  軸的角加速度

$z_f$ ：外力力臂

為了簡化複雜的動態平衡問題，忽略掉各桿件的慣性力及外力干擾，假設

所有質量集中在質心，則 ZMP 公式可簡化為

$$\begin{aligned}
 x_{ZMP} &= x_G - \frac{\ddot{x}_G (z_G - z_{ZMP})}{\ddot{z}_G + g} \\
 y_{ZMP} &= y_G - \frac{\ddot{y}_G (z_G - z_{ZMP})}{\ddot{z}_G + g}
 \end{aligned} \tag{2-2}$$

上式中  $(x_G, y_G, z_G)$  為機器人質心位置的座標， $(\ddot{x}_G, \ddot{y}_G, \ddot{z}_G)$  為質心加速度的分量。

通常機器人的垂直加速度非常小可以忽略，且當機器人只步行在平坦的地

面時  $Z_{ZMP}$  為 0，則 ZMP 公式可簡化為下式

$$x_{ZMP} = x_G - \frac{\ddot{x}_G z_G}{g} \quad (2-3)$$

$$y_{ZMP} = y_G - \frac{\ddot{y}_G z_G}{g}$$

當機器人移動緩慢，質心加速度  $\ddot{x}_G, \ddot{y}_G$  也都為 0，則零力矩點和質心座標，

在 xy 平面有相同的投影點

$$\begin{aligned} x_{ZMP} &= x_G \\ y_{ZMP} &= y_G \end{aligned} \quad (2-4)$$

### 2.1.2 機器人學座標系轉換

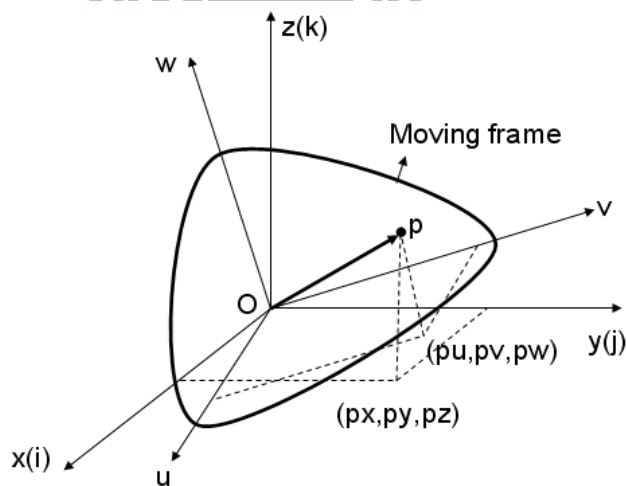


圖 2.2 球面轉動之參考座標系與移動座標系關係圖

圖 2.2 為一參考座標系和移動座標系之間的位置相關圖，令  $i, j, k$  為參考座標

系 A 的三軸之單位向量，而  $u, v, w$  為移動座標系 B 的三軸之單位向量，當以

參考座標系 A 來描述  $u, v, w$  這三個向量時，可以得到以下結果：

$$\begin{aligned}\vec{u} &= u_x \vec{i} + u_y \vec{j} + u_z \vec{k} \\ \vec{v} &= v_x \vec{i} + v_y \vec{j} + v_z \vec{k} \\ \vec{w} &= w_x \vec{i} + w_y \vec{j} + w_z \vec{k}\end{aligned}\quad (2-5)$$

該點 P 的位置向量由 A 來表示

$$\vec{p} = p_x \vec{i} + p_y \vec{j} + p_z \vec{k} \quad (2-6)$$

或是以移動座標系 B 表示

$$\vec{p} = p_u \vec{u} + p_v \vec{v} + p_w \vec{w} \quad (2-7)$$

將(2-5)式帶入(2-7)式可以得到以參考座標系 A 所表示的 P 點的位置向量

$$\begin{aligned}\vec{p} &= (p_u u_x + p_v v_x + p_w w_x) \vec{i} + (p_u u_y + p_v v_y + p_w w_y) \vec{j} \\ &\quad + (p_u u_z + p_v v_z + p_w w_z) \vec{k}\end{aligned}\quad (2-8)$$

比較(2-6)式與(2-8)式可得

$$\begin{aligned}p_x &= p_u u_x + p_v v_x + p_w w_x \\ p_y &= p_u u_y + p_v v_y + p_w w_y \\ p_z &= p_u u_z + p_v v_z + p_w w_z\end{aligned}\quad (2-9)$$

將(2-9)式以矩陣表示

$$\vec{p} = {}^A R_B \vec{p} \quad \text{其中} \quad {}^A R_B = \begin{pmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{pmatrix} \quad (2-10)$$

因為為正交矩陣，所以  ${}^B R_A = {}^A R_B^{-1} = {}^A R_B^T$ ，反求  $\vec{p}$  對  $\vec{p}$  間的關係，可得

$$\vec{p} = {}^A R_B^T \vec{p} \quad (2-11)$$

下面考慮一般性空間移動的座標轉換關係



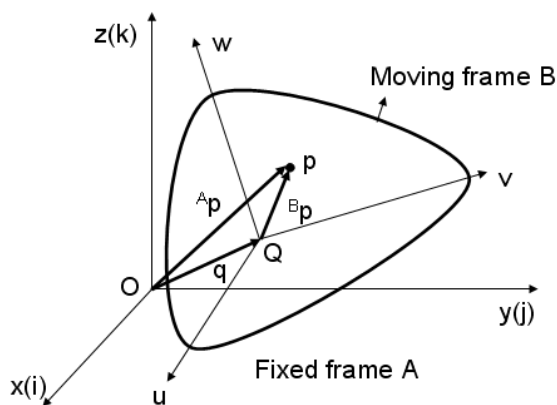


圖 2.3 一般空間移動之參考座標系與移動座標系關係

根據圖 2.3，發現移動座標系除了轉動外還包含了移動  $q$ ，從圖中的向量關係圖可以得到兩座標系之間的關係

$${}^A \vec{p} = {}^A R_B {}^B \vec{p} + {}^A \vec{q} \quad (2-12)$$

將上式整理成均質轉換矩陣 (Homogeneous Transformation Matrix) 可得

$${}^A \vec{p} = {}^A T_B {}^B \vec{p} \quad (2-13)$$

其中

$${}^A T_B = \begin{pmatrix} {}^A R_B & \vdots & {}^A \vec{q} \\ \dots\dots\dots & \vdots & \dots \\ 000 & \vdots & 1 \end{pmatrix} \quad (2-14)$$

要反求  ${}^A \vec{p}$  對  ${}^B \vec{p}$  間的關係時，因為  ${}^A T_B$  並非正交矩陣 ( ${}^A T_B^{-1} \neq {}^A T_B^T$ )，將(2-12)

式等號兩邊同乘  ${}^A R_B^{-1}$ ，又  ${}^A R_B^{-1} = {}^A R_B^T$ ，可得

$${}^B \vec{p} = {}^A R_B^T {}^A \vec{p} - {}^A R_B^T {}^A \vec{q} = {}^A T_B^{-1} {}^A \vec{p}$$

其中

$${}^A T_B^{-1} = {}^B T_A = \begin{pmatrix} {}^A R_B^T & \vdots & -{}^A R_B^T {}^A \vec{q} \\ \dots\dots\dots & \vdots & \dots\dots\dots \\ 000 & \vdots & 1 \end{pmatrix} \quad (2-15)$$

有了上述的關係式，即可輕易的求得兩座標系之間的轉換關係。

### 2.1.3 倒傳遞類神經網路(Back-Propagation ANN)

倒傳遞神經網路為多層神經網路，由三個部份組成，輸入層、隱藏層及輸出層，隱藏層可為單層或多層視系統難易而定，圖 2.4 為一單層隱藏層的網路架構；1987 年 Lippmann 提出，單層隱藏層的類神經網路足以應付線性或非線性函數，單一隱藏層神經元個數可為  $A(B+1)$ ， $A$  為輸出層神經元數目， $B$  為輸入層神經元個數；實際應用時通常只要較少神經元個數即可達到良好的收斂結果。

誤差倒傳遞類神經網路利用最陡坡降法(*gradient steepest descent method*)的觀念，將推論輸出值與目標輸出值的差以「倒傳遞」的方式修正網路中的權重和偏權值，直到誤差函數達到最小，也就是推論輸出值趨近目標輸出值，架構即訓練完成。

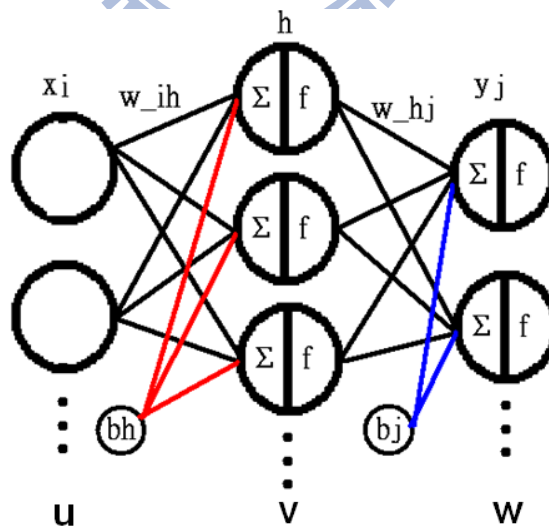


圖 2.4 倒傳遞類神經網路架構圖

$X_i$ : 架構輸入,  $i=1\sim u$        $h$ : 隱藏層神經元,  $h=1\sim v$        $Y_j$ : 架構輸出,  $j=1\sim w$

$W_{ih}$ : 輸入層與隱藏層之間的權重       $b_h$ : 隱藏層的偏權值

$W_{hj}$ ：隱藏層與輸出層之間的權重

$b_j$ ：輸出層的偏權值

$\Sigma$ ：summation

$f$ ：轉移函數(線性或非線性)

常見的轉移函數如圖 2.5 所示(圖片來源：MATLAB 工程軟體的「Help」資料庫)：

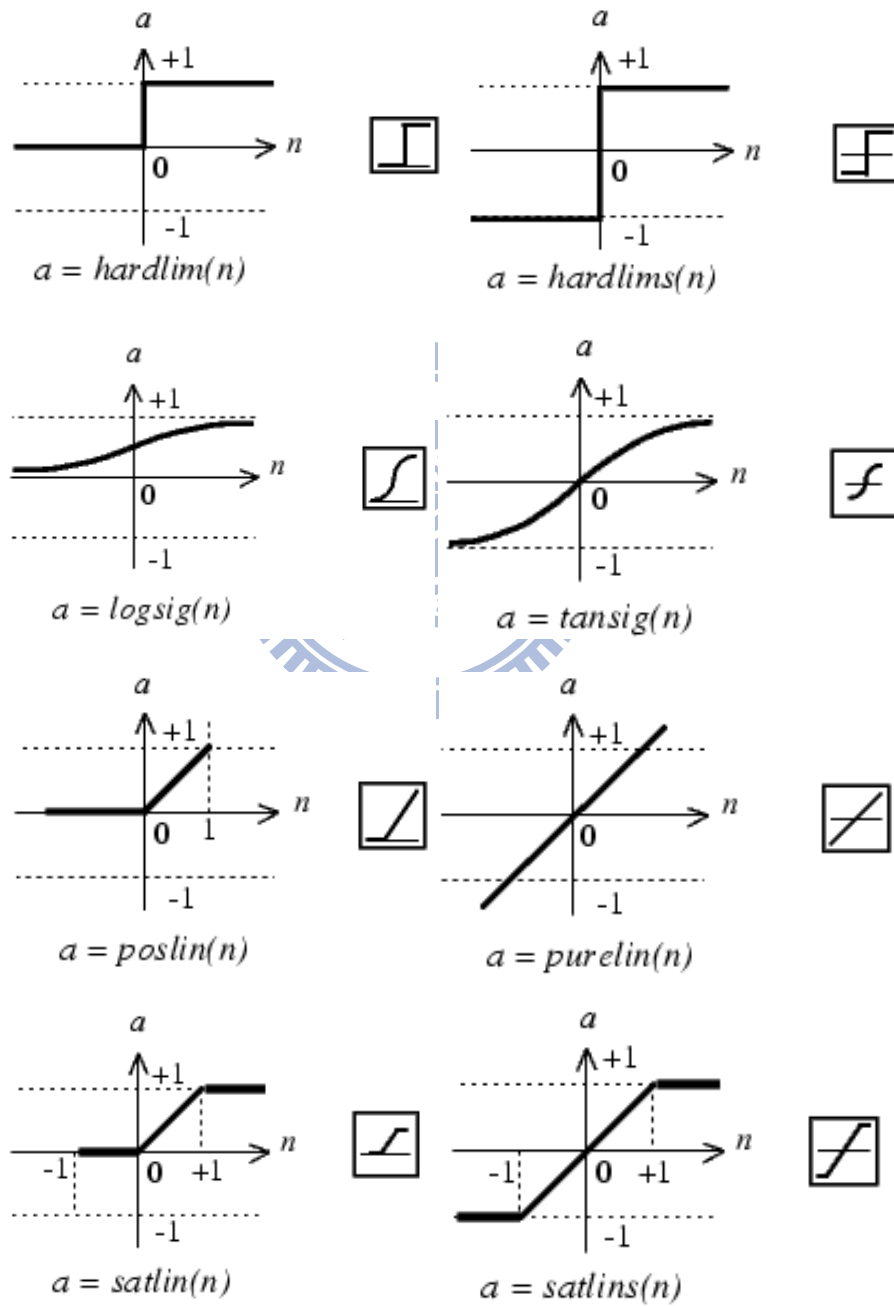


圖 2.5 各種轉移函數圖

學習過程中參數修正的數學函數如下：

$$\text{Net}_h = f(\sum_{h=1}^v \sum_{i=1}^u (X_i * W_{ih} - b_h)) \quad (2-16)$$

將輸入乘上對應的權重  $W_{ih}$  作加總後，扣掉對應的偏權值  $b_h$ ，再帶入非線性函數  $f$  得  $\text{Net}_h$ 。

$$Y_{jn} = f(\sum_{j=1}^w \sum_{h=1}^v (\text{Net}_h * W_{hj} - b_j)) \quad (2-17)$$

將  $\text{Net}_h$  乘上對應的權重  $W_{hj}$  作加總後，扣掉對應的偏權值  $b_j$ ，再帶入非線性函數  $f$  即為真實輸出  $Y_{jn}$ ，接著是修正的誤差函數。

$Y_j$ ：參考輸出       $Y_{jn}$ ：真實輸出       $\eta$ ：學習常數 0~1 之間

輸出層誤差函數：

$$D_j = \sum_{j=1}^w Y_{jn} * (1 - Y_{jn}) * (Y_j - Y_{jn}) \quad (2-18)$$

隱藏層誤差函數：

$$D_h = \sum_{h=1}^v (\text{Net}_h * (1 - \text{Net}_h) * \sum_{j=1}^w W_{hj} * D_j) \quad (2-19)$$

$$\Delta W_{hj} = \eta * D_j * \text{Net}_h \quad (2-20)$$

$$\Delta b_j = -\eta * D_j \quad (2-21)$$

$$\Delta W_{ih} = \eta * D_h * X_i \quad (2-22)$$

$$\Delta b_h = -\eta * D_h \quad (2-23)$$

修正後的權重與偏權值：

$$\begin{aligned} W_{hj}'' &= W_{hj} + \Delta W_{hj} & W_{ih}'' &= W_{ih} + \Delta W_{ih} \\ b_j'' &= b_j + \Delta b_j & b_h'' &= b_h + \Delta b_h \end{aligned} \quad (2-24)$$

均方誤差：

$$E = \sum_{j=1}^w 0.5 * (Y_j - Y_{jn})^2 \quad (2-25)$$

直到誤差函數 E 收斂至極小，也就是推論輸出值趨近目標輸出值，架構即訓練完成。

#### 2.1.4 模糊控制(fuzzy control)

Lotfi.A.Zadeh(札德) 在 1965 年提出一種定量表達工具，以數學的方式表達某些無法明確定義的模糊概念，用來處理人類口語表達特有的模糊性現象有很好的成果。

電腦優於人腦的地方在於計算能力，對於無法以有效的計算法解決的問題，如思考、推理、影像識別等，表現就不如人腦。因此模糊理論就是針對人腦對於模糊的訊息或不完全的資料，其不需經過精密繁雜的計算過程，仍能做出正確判斷。

不論古典控制法則或是現代的控制理論，在理論方面已有相當的程度，實際應用上也有一定的成效，這些都只侷限在受控系統必須相當的明確或是可以用嚴謹的數學計算公式來表達時，才能使傳統控制理論有效的運作，真實世界裡，工業界上所存在的是複雜、多變、模糊的系統，受控場難以用明確的數學模式表達，導致傳統、現代控制無法有效的應用；相對的，模糊控制可在不知受控系統數學函式下，即可以達到良好的控制效果，突破傳統控制的限制，因此近年來被廣泛的應用。

模糊控制主要有四大部分，分別為模糊化、模糊推論、規則庫、解模糊化，如圖 2.6 所示。

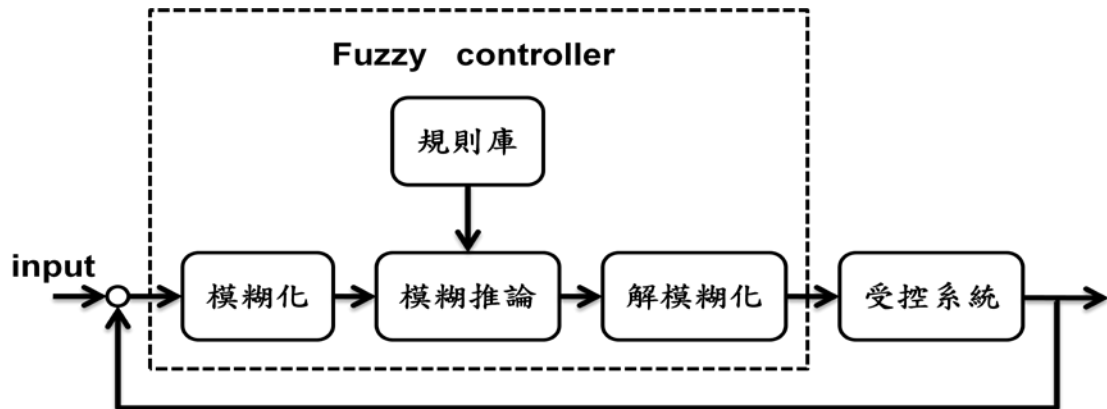


圖 2.6 模糊控制方塊圖

步驟一：模糊化 (fuzzifier)

將外界輸入對應到所屬語意變數的歸屬函數，經過模糊化函數(fuzzifier function)將一明確輸入轉化為一語意值(linguistic value)。

步驟二：規則庫 (data base)

由專家經驗和知識或控制系統的特性所訂定的規則，一般規則庫多以 IF-THEN 的方式來表示。

步驟三：模糊推論 (fuzzy inference)

此步驟為模糊控制的核心，在規則庫的法則下進行推論的合成運算，將每條規則推論的結果取聯集(union)。

步驟四：解模糊化 (defuzzifier)

將模糊運算所得的模糊輸出轉換成實際的物理量，解模糊化的方法很多，最常用的是重心法。

## 2.1.5 適應性類神經網路模糊推論系統

1993 年由 Jang 所提出，此架構為 Sugeno 模糊模型的延伸，此網路架構稱為適應性網路模糊推論系統，Sugeno 模糊模型由 ANFIS 架構的好處是，可以利用類神經網路學習演算法來設定 Sugeno 模糊模型的參數，使 Sugeno 模糊模型具有自我學習的能力，自行調整最佳的模糊規則。

圖 2.7 說明一具有兩個輸入  $x$ 、 $y$ ，一個輸出  $f$  的 ANFIS 架構

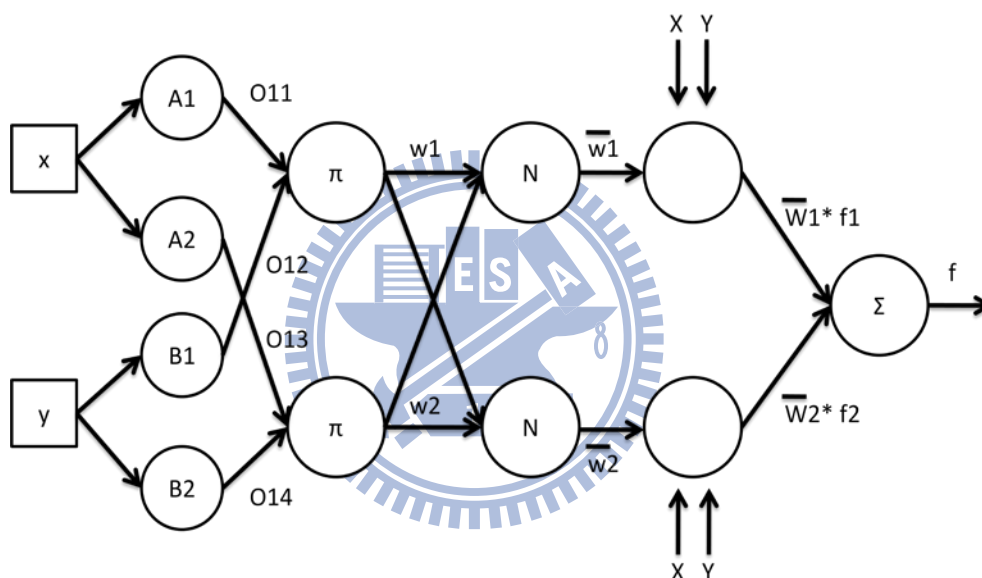


圖 2.7 ANFIS 架構圖

第一層：

$A1$ 、 $A2$  為變數  $x$  的歸屬函數，而  $B1$ 、 $B2$  為變數  $y$  的歸屬函數，第一層的工作為計算變數  $x$ 、 $y$  分別映射到  $Ai$ 、 $Bi$  的值。

$$O_{1i} = u_{A_i}(X) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \quad i = 1, 2 \quad (2-26)$$

$$O_{1i} = u_{B_i}(Y) = \frac{1}{1 + \left| \frac{y - c_i}{a_i} \right|^{2b_i}} \quad i = 3, 4 \quad (2-27)$$

第二層：

此層為模糊規則的運算，選取第一層觸發高度最低者或乘積

$$O_{2i} = W_i = U_{A_i}(x) * U_{B_i}(y)$$

$$or = \min(U_{A_i}(x), U_{B_i}(y)) \quad i=1,2 \quad (2-28)$$

第三層：正規化運算

$$O_{3i} = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad i=1,2 \quad (2-29)$$

第四層：每個模糊推論的運算

$$O_{4i} = \bar{w}_i * f_i = \bar{w}_i * (p_i x + q_i y + r_i), \quad i = 1,2 \quad (2-30)$$

第五層：計算前一層中神經元輸出值的總和，作為最後網路的輸出值

$$f = \sum_i \bar{w}_i * f_i = \bar{w}_1 f_1 + \bar{w}_2 f_2 \quad (2-31)$$

學習過程如下：

ANFIS 參數學習採用「混合式學習」，學習過程如下：

1. 首先先固定前提部的歸屬函數參數(a,b,c)，假設前提部的參數為已知。
2. 將所有輸入資料輸入網路，以最小平方估測法(Least-Square Estimator)估測出最佳的推論部參數(p,q,r)，得到近似的輸出值。
3. 計算網路輸出值與目標輸出值的誤差。
4. 利用最陡坡降法，修正前提部的參數。
5. 回到步驟 1 繼續學習，直到輸出誤差足夠小為止。



## 三、系統模擬軟體介紹

### 3.1 MATLAB 介面介紹

MATLAB 是由 Math Works 公司於 1984 年推出的數學軟體，最早發展理念是提供一套非常完善的矩陣運算，隨著市場需求，如今 MATLAB 已成為各種系統模擬、線性代數、數位訊號處理、科學目視(Scientific Visualization)的重要語言；開啟 MATLAB 會顯示圖 3.1 所示視窗，主要部份有 Workspace、命令歷史列(Command History)、命令視窗(Command Window)、路徑、工具列...等等，以下為一些簡單介面介紹。

Workspace：顯示所有定義變數和檔案名稱。

命令歷史列：顯示所有執行過指令。

命令視窗：所有指令都輸入在此視窗裡，按 enter 鍵執行。

路徑：顯示 MATLAB 呼叫函式的路徑，預設值為 work 資料夾。

工具列：一些工具箱快捷鍵，例如：M-file、Simulink、help...等等。

以下簡單說明幾點 MATLAB 優、缺點：

優點：

- 1.簡單易學，容易上手，相當多書籍可以參考。
- 2.網路上許多範例程式、教學檔。
- 3.提供許多矩陣運算指令。
- 4.針對不同領域開發許多應用程式，稱為工具箱(Toolbox)。

缺點：

1. MATLAB 編譯的執行檔必須在有安裝 MATLAB 的電腦才可順利執行。

2. 迴圈運算速度比 C 語言慢。

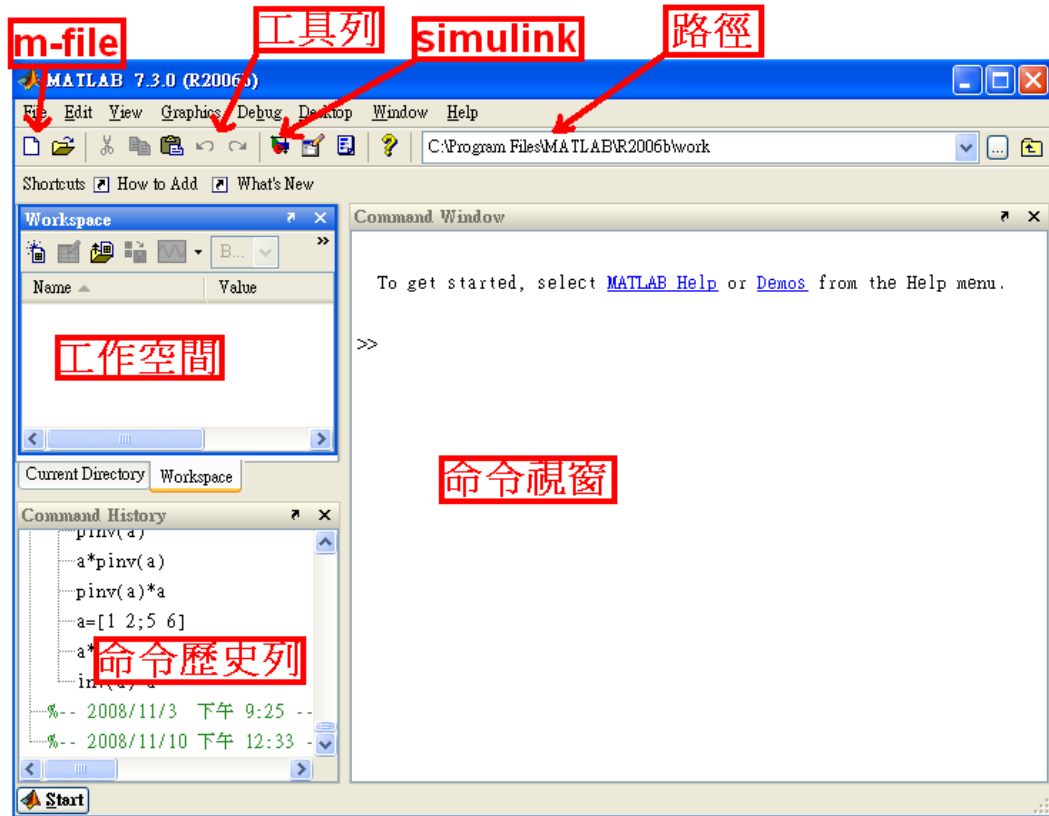


圖 3.1 MATLAB 介面視窗

### 3.2 MATLAB – simulink

Simulink 提供一個動態系統建構模擬和分析的環境，整合了許多 MATLAB 的工具箱，讓不同工具箱可在此圖形化介面上作連結，在這個環境中，只要透過簡單的圖形方塊，利用滑鼠拖曳，即可以建構出控制系統，如此一來使程式更一目了然，縮短了程式除錯的時間，有別於以往打指令撰寫程式的方式。

圖 3.2 為一簡單範例，將正弦和餘弦波用模擬示波器(scope)顯示。

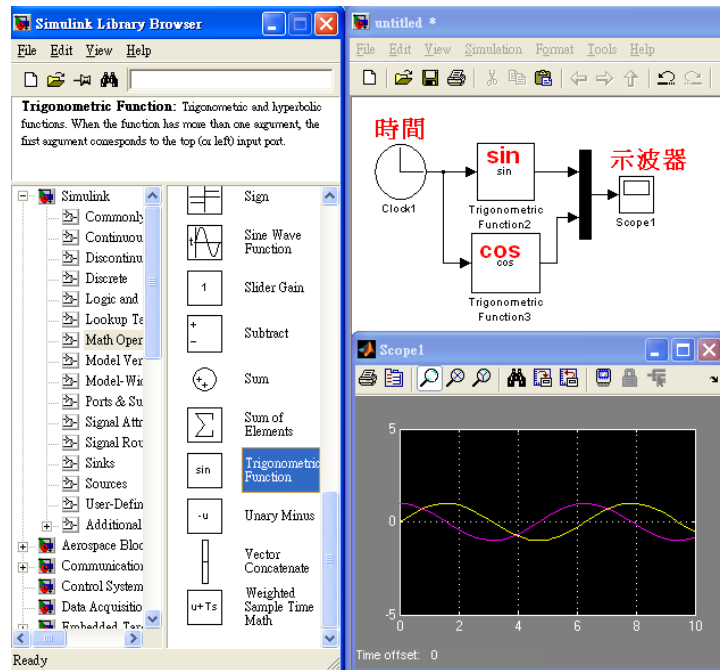


圖 3.2 Simulink 範例介紹圖

### 3.3 MATLAB – 模糊工具箱

在 MATLAB 命令視窗下輸入 fuzzy 即會顯示圖 3.3 圖形化介面視窗，

此為模糊控制主要架構(輸入、輸出、模糊推論、推論方法...等等)。

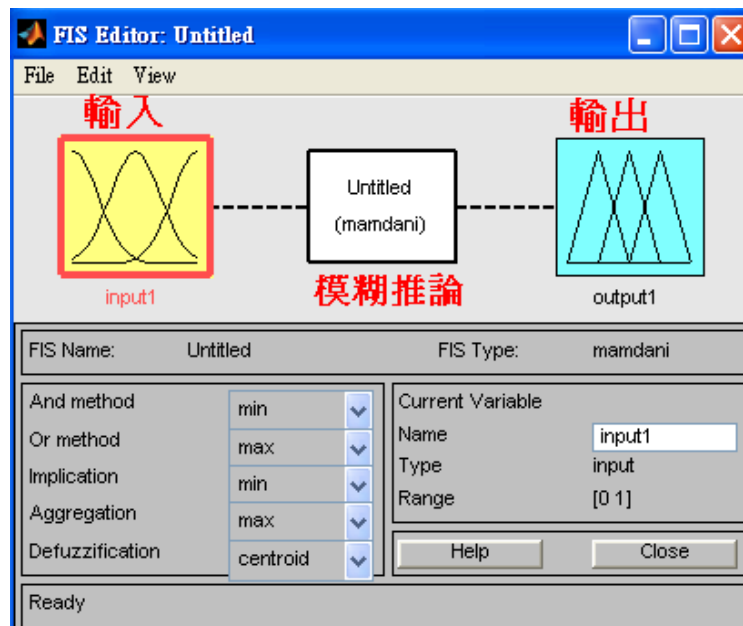
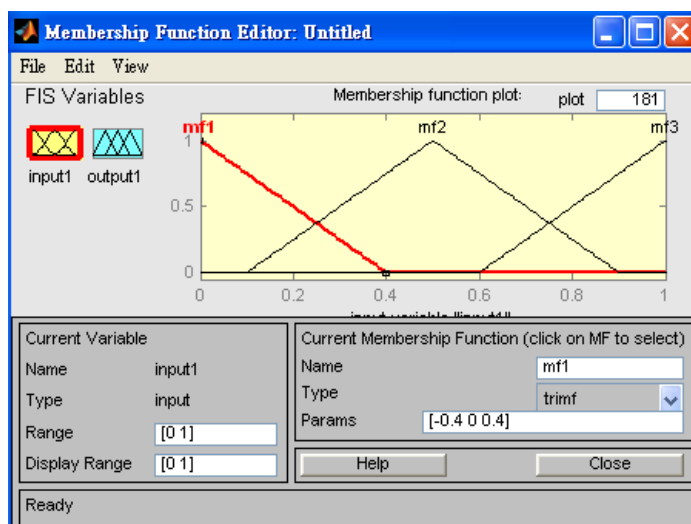
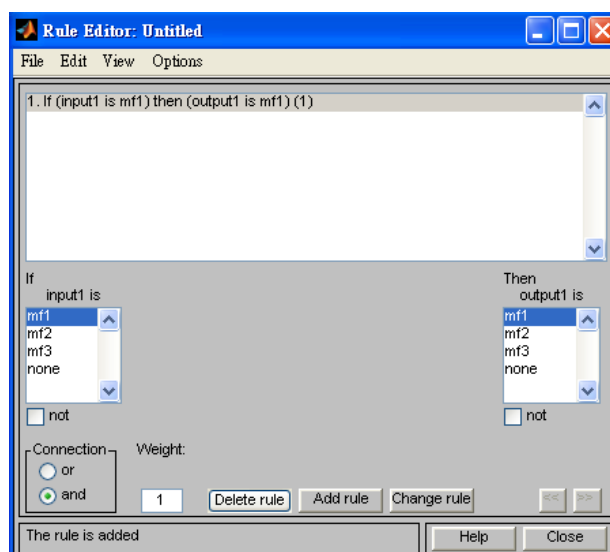


圖 3.3 fuzzy GUI 介面

滑鼠連點輸入圖案兩下或是鍵盤輸入(Ctrl+2)即可開啟建構輸入和輸出歸屬函數的視窗，歸屬函數有許多種類可選擇，例如：三角形、鐘形、高斯曲線、梯形...等等，如下圖所示，不同的歸屬函數有不同定義方式，定義完成後按 close 按鈕，視窗即關閉且自動儲存定義參數。



接著是規則庫的建立，點選 Edit 裡的 Rules 或鍵盤輸入(Ctrl+3)即出現編輯規則庫的圖形化視窗，如下圖所示，藉由專家經驗建立合適的規則以 IF-THEN 的方式進行推論，例如：IF input is... THEN output is...，在達到相同需求下，規則數越少越好，所有設定完成後可在 View-->Rules 觀看結果。



### 3.4 MATLAB – 倒傳遞類神經網路

倒傳遞類神經網路屬於前饋型 (feedforward) 類神經網路，故使用 MATLAB 來實現倒傳遞類神經網路的指令為 `newff`，`ff` 即代表 feedforward。

**newff** 指令說明：

`newff(PR, [S1 S2...SNi], {TF1 TF2...TFNi}, BTF, BLF, PF)`

PR：R x 2 矩陣，R 為輸入個數，其值為各輸入的最小和最大值

Si：第 i 層的神經元個數

TFi：第 i 層神經元的轉移函數 (預設值為 'tansig')

BTF：倒傳遞類神經網路訓練函數(預設值為'trainlm')

BLF：倒傳遞類神經網路權重與偏權值的學習函數(預設值為'learngdm')

PF：誤差函數 (預設值為='mse')

轉移函數 (TFi)有許多不同種類，例如： `tansig`、`logsig`、`purelin`。

學習函數(BTF) 有許多不同種類，例如：`trainlm`、`trainbfg`、`trainrp`、

`traingd`。

注意：`trainlm` 是系統預設值，因為它非常快速，但很耗系統記憶體，因此若出現 `out-of-memory error` 請嘗試其他函數。

以下為其他訓練函數：

`Slow trainlm`：藉由設定 `net.trainParam.mem_reduc` 的值為 2 或更大，可減少系統記憶體使用量。

`trainbfg`：學習速度較慢，但是記憶體使用效率優於 `trainlm`。

`trainrp`：學習速度較慢，但是記憶體使用效率優於 `trainbfg`。

修正權重與偏權值的函數有 `learngd` 和 `learngdm`。

輸出誤差函數種類有 `mse` 和 `msereg`。

由以上的指令說明(資料來源：MATLAB 工程軟體的「Help」資料庫)得知只要有訓練樣本，確認並建立好架構、所有學習參數與函數設定，便可立即學習，但未必每次學習結果都會收斂，若結果發散，此時要利用試誤法和經驗公式不斷修改架構參數，例如：隱藏層層數、神經元個數、轉移函數種類...等等。



## 四、研究內容與方法

本文研究最終目的為將 ANFIS 應用至四足機器人路徑規劃，ANFIS 融合了模糊推論系統與類神經網路學習功能，學習結果應用至四足機器人跨越非平坦河石地形，比傳統機器人相比更具有智慧。

本實驗室設計的四足機器人[11](圖 4.2)的腳為三個自由度可視為空間三連桿，研究過程有簡入深首先將模糊控制應用至平面三連桿圖 4.1(a)，接著比較 BP 與 ANFIS 應用至空間三連桿圖 4.1(b)，最後本研究重點是發展 ANFIS 程式為基礎的步伐規劃系統應用至四足機器人的運動控制[11]，然而當探討機器人行走時還必須考慮平衡問題，本文搭配 ZMP 理論，讓四足機器人行走時機身重心投影至地面的點落在腳底支撐面內，即可避免傾倒。

### 4.1 研究過程與架構

本文主要使用三種理論，分別是模糊理論、倒傳遞類神經網路和適應性網路模糊推理系統，藉由觀察輸出誤差，比較三種理論的優缺點，研究應用由簡入深，過程如下：

1. 模糊控制應用至平面三連桿手臂
2. 倒傳遞類神經網路應用至空間三連桿手臂
3. ANFIS 應用至空間三連桿手臂
4. ANFIS 應用至 12-DOF 四足機器人
5. 跨越非連續非平坦河石地形之實驗

以下為此三種架構的示意圖：

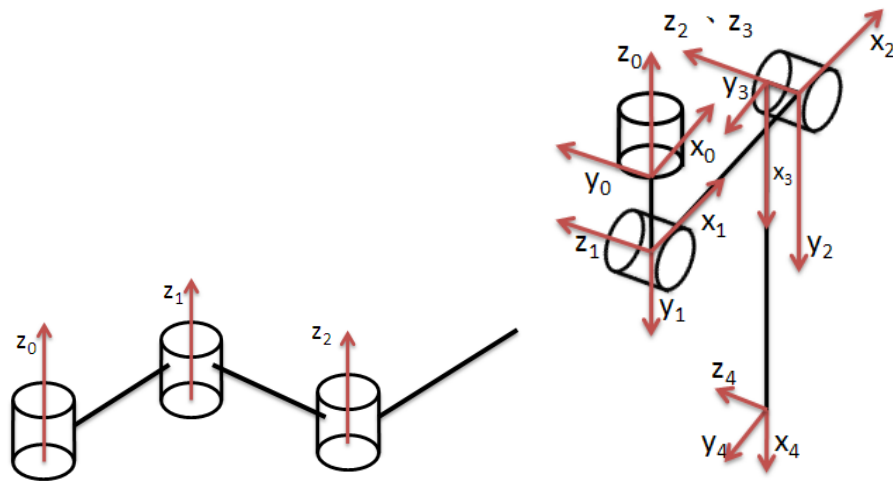


圖 4.1 平面(a)和空間(b) 三連桿手臂示意圖

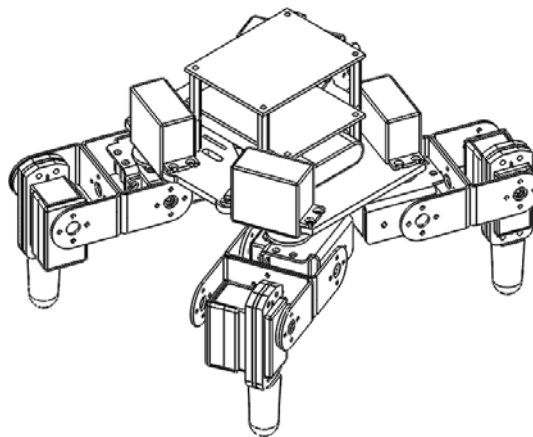


圖 4.2 四足機器人 3D 圖(CIDM 實驗室設計圖)

## 4.2 Fuzzy 應用至平面三連桿手臂

本節使用模糊推論系統替代平面三連桿手臂正向運動學函數，比較模糊推論輸出與運動學函數誤差大小，不斷以試誤法修改 Fuzzy 架構參數使誤差收斂至更小，則表示此 Fuzzy 架構可近似於正向運動學函數，換句話說，只要系統輸入和輸出為已知，便可靠經驗建立 Fuzzy 架構進而取代線性或是非線性系統轉移函數。



1. 首先確認架構輸入與輸出，本節利用 Fuzzy 近似平面三連桿手臂正向運動學，輸入為三個馬達角度( $\theta_1$ 、 $\theta_2$ 、 $\theta_3$ )、輸出為(x、y 座標)，歸屬函數選用三角形(trimf)，參考圖 4.3。

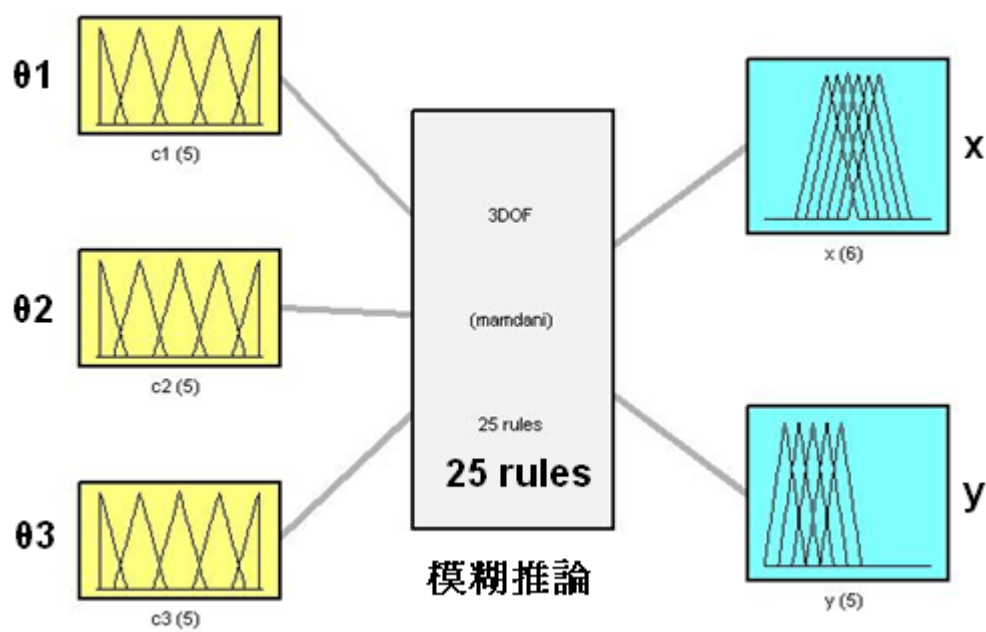


圖 4.3 三連桿手臂之 Fuzzy 架構圖

2. 建立輸出和輸入的歸屬函數，本文將 3 個輸入( $\theta_1$ 、 $\theta_2$ 、 $\theta_3$ )的歸屬函數設定皆使用 5 個三角形函數，角度範圍  $\theta_1$  為-90 度到-60 度之間， $\theta_2$  和  $\theta_3$  範圍相同為-90 度到+90 度之間；因為桿長( $L_1=L_2=10$ 、 $L_3=5$ )和角度隨機搭配關係，輸出  $x$ 、 $y$  座標的歸屬函數設定就大大不同，由工作空間可看出  $x$  範圍為-15 到+20、 $y$  範圍為-25 到+5 之間，參考圖 4.4、圖 4.5、圖 4.6、圖 4.7。

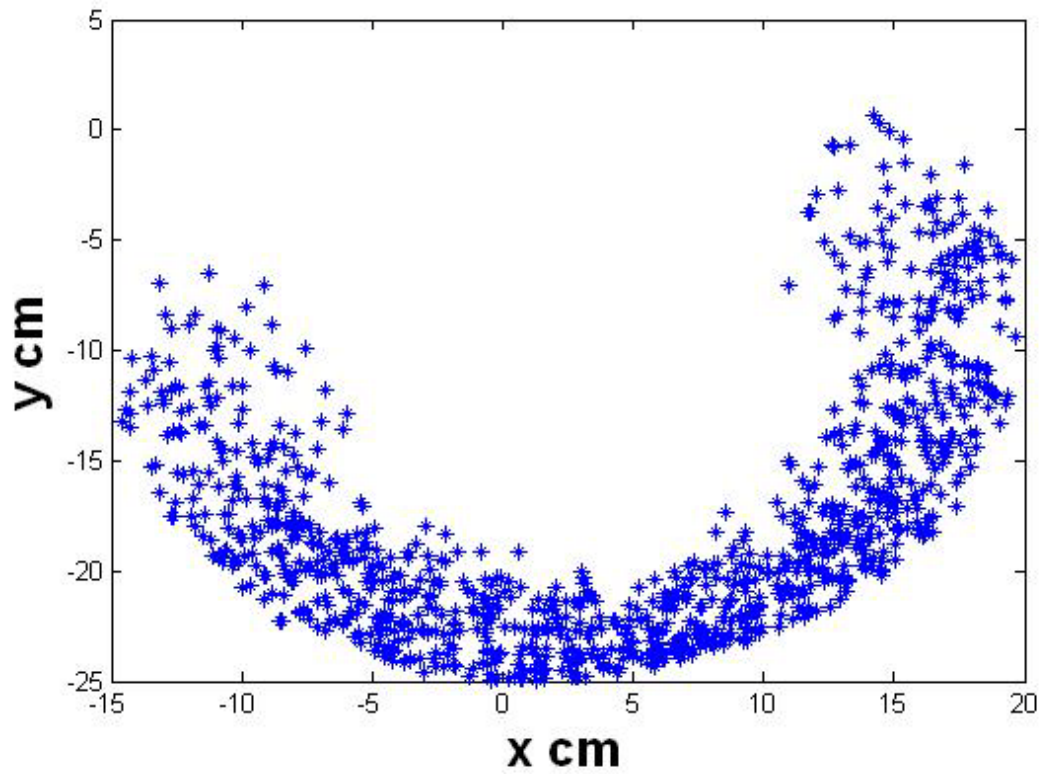


圖 4.4 ES 工作空間

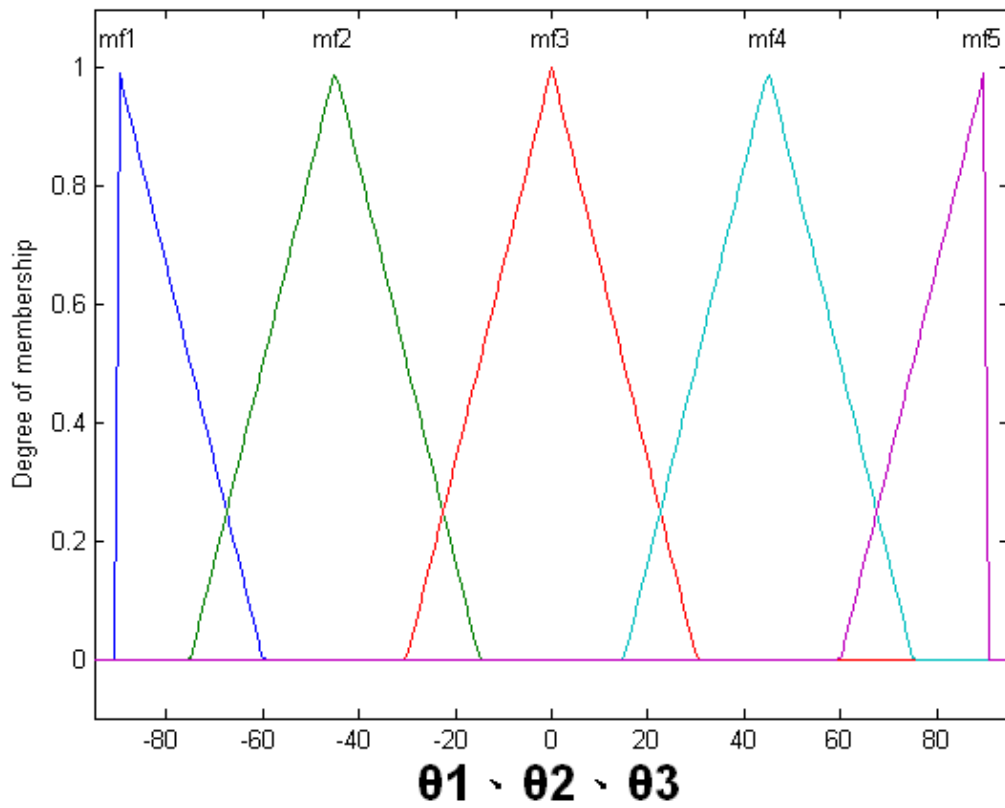


圖 4.5 輸入角度( $\theta_1$ 、 $\theta_2$ 、 $\theta_3$ )歸屬函數

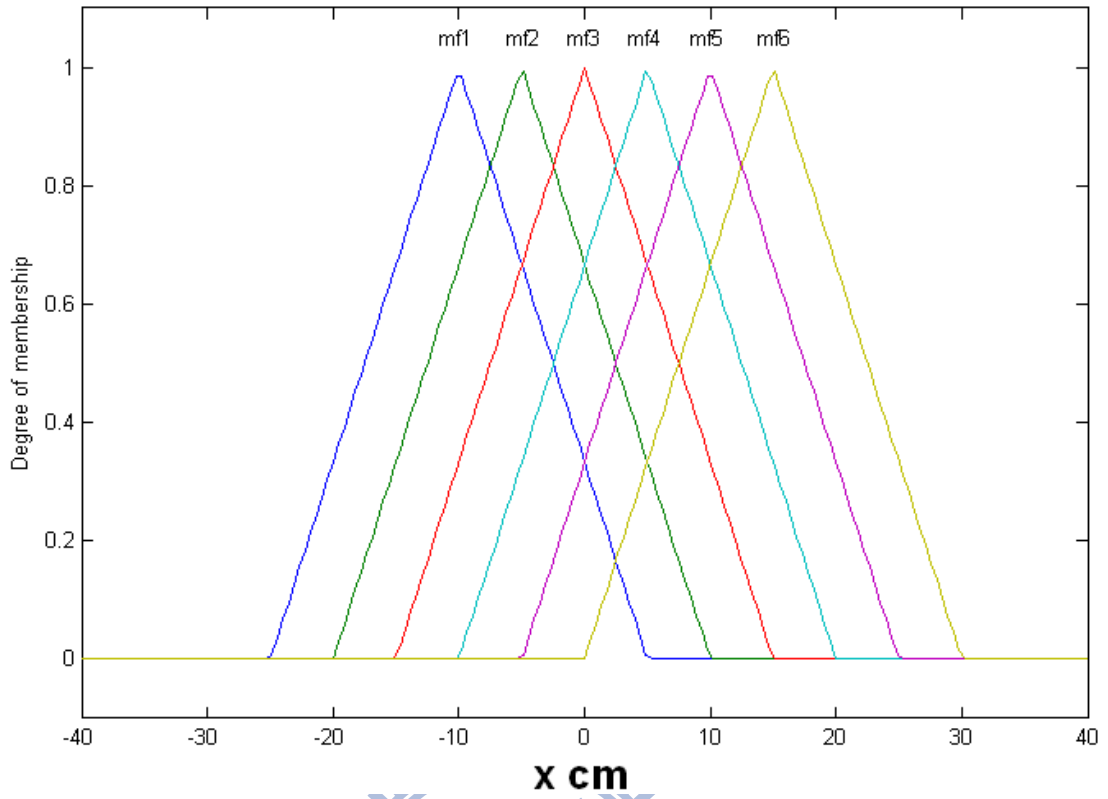


圖 4.6 x 座標歸屬函數

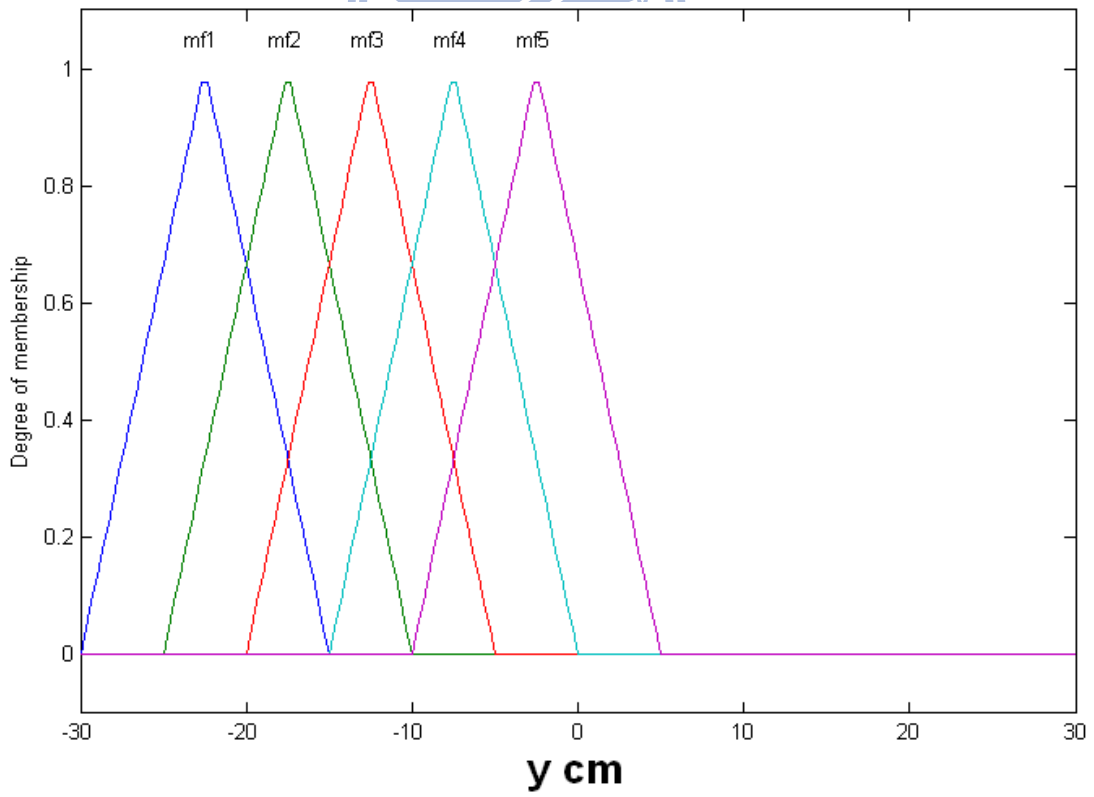


圖 4.7 y 座標歸屬函數

3. 將  $\theta_1$ 、 $\theta_2$ 、 $\theta_3$  歸屬函數做隨機組合搭配，產生不同  $x$ 、 $y$  座標，藉由觀

察工作空間圖建立模糊規則庫。

1. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf1) and ( $\theta_3$  is mf1) then ( $x$  is mf2)( $y$  is mf4)
2. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf1) and ( $\theta_3$  is mf2) then ( $x$  is mf2)( $y$  is mf3)
3. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf1) and ( $\theta_3$  is mf3) then ( $x$  is mf3)( $y$  is mf3)
4. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf1) and ( $\theta_3$  is mf4) then ( $x$  is mf3)( $y$  is mf2)
5. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf1) and ( $\theta_3$  is mf5) then ( $x$  is mf4)( $y$  is mf2)
6. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf2) and ( $\theta_3$  is mf1) then ( $x$  is mf4)( $y$  is mf3)
7. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf2) and ( $\theta_3$  is mf2) then ( $x$  is mf4)( $y$  is mf2)
8. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf2) and ( $\theta_3$  is mf3) then ( $x$  is mf3)( $y$  is mf2)
9. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf2) and ( $\theta_3$  is mf4) then ( $x$  is mf4)( $y$  is mf2)
10. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf2) and ( $\theta_3$  is mf5) then ( $x$  is mf4)( $y$  is mf3)
11. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf3) and ( $\theta_3$  is mf1) then ( $x$  is mf4)( $y$  is mf2)
12. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf3) and ( $\theta_3$  is mf2) then ( $x$  is mf4)( $y$  is mf2)
13. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf3) and ( $\theta_3$  is mf3) then ( $x$  is mf5)( $y$  is mf2)
14. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf3) and ( $\theta_3$  is mf4) then ( $x$  is mf5)( $y$  is mf4)
15. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf3) and ( $\theta_3$  is mf5) then ( $x$  is mf5)( $y$  is mf4)
16. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf4) and ( $\theta_3$  is mf1) then ( $x$  is mf5)( $y$  is mf4)
17. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf4) and ( $\theta_3$  is mf2) then ( $x$  is mf6)( $y$  is mf4)
18. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf4) and ( $\theta_3$  is mf3) then ( $x$  is mf6)( $y$  is mf5)
19. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf4) and ( $\theta_3$  is mf4) then ( $x$  is mf7)( $y$  is mf5)
20. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf4) and ( $\theta_3$  is mf5) then ( $x$  is mf7)( $y$  is mf5)
21. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf5) and ( $\theta_3$  is mf1) then ( $x$  is mf7)( $y$  is mf5)
22. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf5) and ( $\theta_3$  is mf2) then ( $x$  is mf7)( $y$  is mf4)
23. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf5) and ( $\theta_3$  is mf3) then ( $x$  is mf8)( $y$  is mf6)
24. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf5) and ( $\theta_3$  is mf4) then ( $x$  is mf8)( $y$  is mf6)
25. If ( $\theta_1$  is mf1) and ( $\theta_2$  is mf5) and ( $\theta_3$  is mf5) then ( $x$  is mf8)( $y$  is mf6)

4. 任意給定  $\theta_1$ 、 $\theta_2$ 、 $\theta_3$  觀察推論結果與理論值比較，若輸出結果與理論值

誤差很大，則必須回到步驟 2 改變歸屬函數參數並修改模糊規則庫直到

誤差足夠小時，此架構才可被應用。

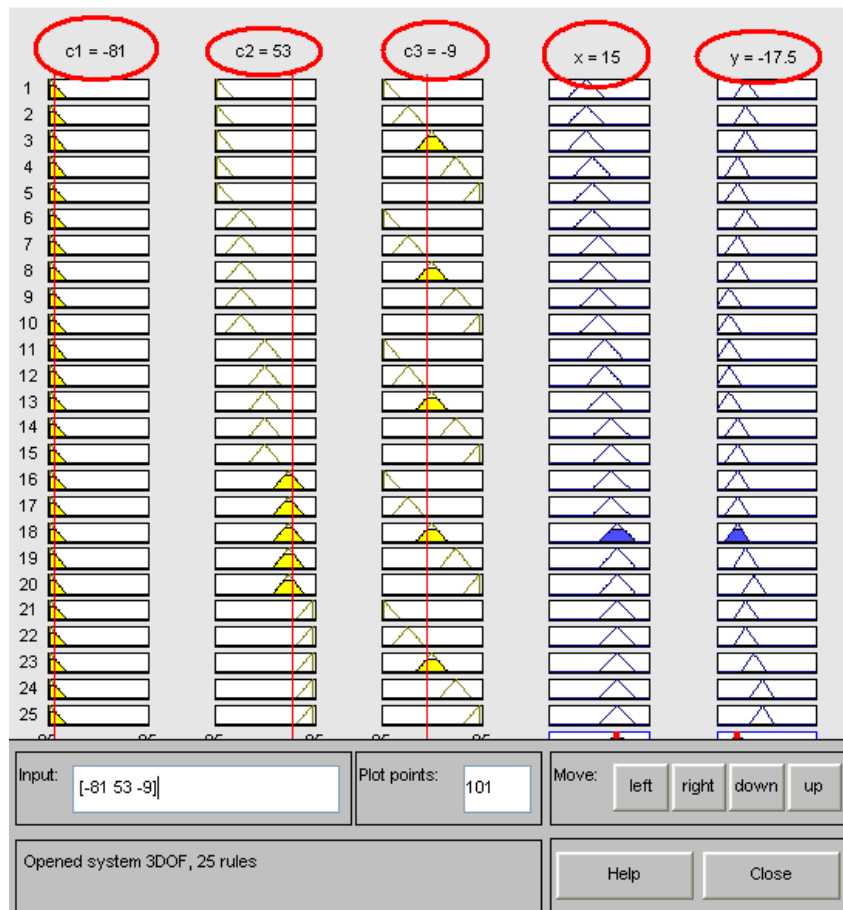


圖 4.8 Fuzzy 輸出結果

討論：

圖 4.8 說明當 Fuzzy 輸入為  $\theta_1=-81$ 、 $\theta_2=53$ 、 $\theta_3=-9$  時，推論結果  $x=15$ 、 $y=-17.5$  與理論值  $x=14.387$ 、 $y=-17.58$  比較得知，模糊控制近似線性或非線性函數誤差較大，因其中有許多參數設定，只能依賴試誤法和經驗不斷嘗試，而且模糊理論算是開路系統，沒有回授訊號可自行修正參數，所以參數調整必須仰賴手動，故通常模糊控制被應用在輸出誤差的修正，輸出誤差的修正是比較主觀、容易想像、推理的，而模糊邏輯正好是模擬大腦的推理系統，因此使用模糊控制「修正誤差」是比用來「近似線性或非線性函數」較容易的。

### 4.3 倒傳遞神經網路應用至空間三連桿手臂

本節將類神經網路應用至空間三連桿手臂，最大優點和模糊理論相同，兩者皆可在不知道受控場轉移函數下，只要有輸入和其對應的輸出，便可用來近似系統的轉移函數；另外類神經網路本身可自行調整參數，此方面優於模糊控制，類神經理論就類似人類大腦行為，當人類在學習一新事件時，大腦並不會去計算數學函式，而是藉由不斷學習最後便熟能生巧，倒傳遞類神經網路應用至空間三連桿手臂流程如下：

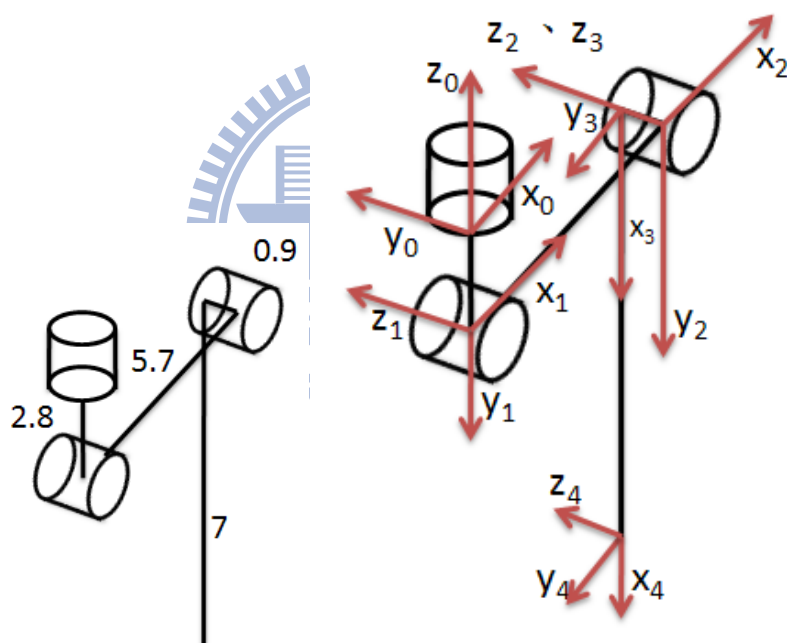


圖 4.9 空間三連桿尺寸與座標圖

1. 在空間三連桿手臂(圖 4.9)上配合馬達起始位置建立座標系，使用 D-H 理論建立 D-H 表和 D-H 矩陣，將 D-H 矩陣相乘後，第四行前三列即為正向運動學公式。

表 4.1 空間三連桿手臂 D-H 表

| Joint | $\Theta$   | A   | d    | A        |
|-------|------------|-----|------|----------|
| 0     | $\theta_1$ | 0   | -2.8 | $-\pi/2$ |
| 1     | $\theta_2$ | 5.7 | 0    | 0        |
| 2     | $\pi/2$    | 0   | 0.9  | 0        |
| 3     | $\theta_3$ | 7   | 0    | 0        |

以下分別為空間三連桿手臂 D-H 矩陣  ${}^0A_1$ 、 ${}^1A_2$ 、 ${}^2A_3$  及  ${}^3A_4$ ：

$${}^0A_1 = \begin{bmatrix} \cos \theta_1 & 0 & -\sin \theta_1 & 0 \\ \sin \theta_1 & 0 & \cos \theta_1 & 0 \\ 0 & -1 & 0 & -2.8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1A_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 5.7 * \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & 5.7 * \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2A_3 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3A_4 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 7 * \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & 7 * \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

將四個矩陣相乘  ${}^0A_1 * {}^1A_2 * {}^2A_3 * {}^3A_4$  可得末端位置 x、y、z 座標：

$$x = -7 * \cos(C1) * \sin(C2) * \cos(C3) - 7 * \cos(C1) * \cos(C2) * \sin(C3) - 9/10 * \sin(C1) + 57/10 * \cos(C1) * \cos(C2)$$

$$y = -7 * \sin(C1) * \sin(C2) * \cos(C3) - 7 * \sin(C1) * \cos(C2) * \sin(C3) + 9/10 * \cos(C1) + 57/10 * \sin(C1) * \cos(C2)$$

$$z = -7 * \cos(C2) * \cos(C3) + 7 * \sin(C2) * \sin(C3) - 14/5 - 57/10 * \sin(C2) \quad (4-1)$$

- 利用(4-1)式正向運動學公式計算出倒傳遞網路學習樣本，以隨機的方式產生 4000 組三個馬達的角度，代入正向運動學公式求得其對應的 4000 組 x、y、z 座標，其中 2000 組當作學習樣本，剩下 2000 組當作驗證樣

本，MATLAB 部分程式如下：

```
theda1=(-35:110) % 馬達一角度範圍  
theda2=(-45:45) % 馬達二角度範圍  
theda3=(-90:45) % 馬達三角度範圍  
for j=1:4000 % 迴圈  
c1(j)=theda1(round(rand*145)+1); %隨機選取-35~110之間的任意角度  
c2(j)=theda2(round(rand*90)+1); %隨機選取-45~45之間的任意角度  
c3(j)=theda3(round(rand*135)+1); %隨機選取-90~45之間的任意角度  
C1=c1(j)*pi/180; C2=c2(j)*pi/180; C3=c3(j)*pi/180; % 徑度與度 換算  
x(j)= -7*cos(C1)*sin(C2+C3)-0.9*sin(C1)+5.7*cos(C1)*cos(C2);  
y(j)= -7*sin(C1)*sin(C2+C3)+0.9*cos(C1)+5.7*sin(C1)*cos(C2);  
z(j)= -7*cos(C2+C3) -2.8-5.7*sin(C2);  
end
```

3. 本節使用倒傳遞類神經網路近似反向運動學，所以架構與前一節模糊理論剛好顛倒，輸入為手臂末端位置  $x$ 、 $y$ 、 $z$  座標，輸出為三個馬達的角度( $\theta_1$ 、 $\theta_2$ 、 $\theta_3$ )，如圖 4.10 所示。

4. 利用經驗公式或試誤法給定隱藏層層數及隱藏層神經元個數，通常學習的系統越複雜則隱藏層層數和神經元個數要越多，值得注意的是隱藏層層數以兩層為上限，神經元個數的經驗公式有許多，常見為以下兩種：

$$N = \sqrt{n+m} + a \quad \text{or} \quad N = m * (n+1) \quad (4-2)$$

$\sqrt{\quad}$ ：開平方根       $N$ ：神經元個數

$m$ ：輸出神經元個數       $n$ ：輸入神經元個數       $a$ ：1~10 的正整數



表 4.2 中 N 值表示神經元個數，E1、E2、E3 分別表示  $\theta_1$ 、 $\theta_2$ 、 $\theta_3$  類神經網路輸出與驗證樣本的誤差，比較的轉移函數有兩種分別為 logsig 及 tansig，透過試誤法不斷嘗試與多組模擬比較結果使用兩層隱藏層，每層各 12 個神經元並選定 logsig(圖 4.11)當作轉移函數。

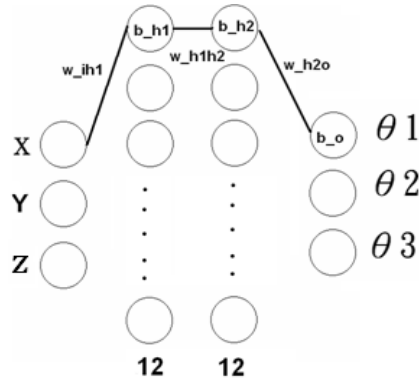


圖 4.10 3-DOF 手臂之 BP 架構圖

表 4.2 不同轉移函數輸出角度誤差比較

| N  | Logsig  |        |         | Tansig  |          |         |
|----|---------|--------|---------|---------|----------|---------|
|    | E1      | E2     | E3      | E1      | E2       | E3      |
| 1  | 16.0114 | 45.473 | 33.4699 | 97.666  | 50.7485  | 67.219  |
| 2  | 36.9713 | 44.527 | 33.9157 | 11.7125 | 16.8468  | 22.0226 |
| 3  | 11.6539 | 5.1924 | 7.0284  | 10.956  | 5.3643   | 5.4632  |
| 4  | 8.9392  | 5.7753 | 67.6195 | 72.2367 | 43.519   | 67.219  |
| 5  | 5.3585  | 4.3185 | 67.3805 | 11.3072 | 43.8325  | 5.4694  |
| 6  | 3.4233  | 2.6725 | 3.4286  | 12.4811 | 5.153    | 5.347   |
| 7  | 3.8288  | 1.6013 | 1.7113  | 4.9562  | 2.3491   | 2.6928  |
| 8  | 3.2629  | 2.1801 | 1.9481  | 3.8831  | 2.1348   | 2.5519  |
| 9  | 3.5693  | 1.5855 | 1.8438  | 72.742  | 133.8967 | 67.219  |
| 10 | 3.7753  | 1.5788 | 1.9434  | 4.6043  | 2.7514   | 56.8442 |
| 11 | 3.5238  | 1.6077 | 1.7975  | 62.8318 | 36.9166  | 52.9101 |
| 12 | 3.3757  | 1.2955 | 0.9887  | 3.0896  | 1.2874   | 1.2617  |
| 13 | 3.1594  | 1.3214 | 1.7157  | 3.5768  | 1.4094   | 1.3309  |

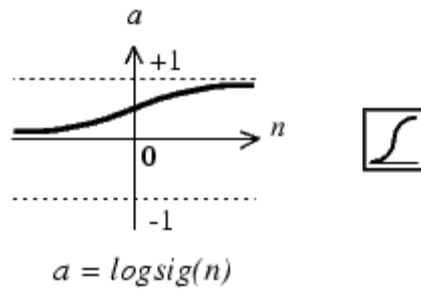


圖 4.11 轉移函數 logsig

5. 以隨機方式設定權重值(weight)和偏權值(bias)，設定學習次數 300 次

6. 將輸入和輸出正規化後代入架構開始學習，式(4-2)為正規化公式。

$$\begin{aligned} & (x - \min(x)) / (\max(x) - \min(x)) \\ & (y - \min(y)) / (\max(y) - \min(y)) \\ & (z - \min(z)) / (\max(z) - \min(z)) \end{aligned} \quad (4-3)$$

7. 輸出誤差以「倒傳遞」的方式修正權重(weight)和偏權值(bias)

權重、偏權值修正公式：

$Y_j$ ：參考輸出       $Y_{jn}$ ：實際輸出       $\eta$ ：學習速率

$$D_j = Y_{jn} * (1 - Y_{jn}) * (Y_j - Y_{jn}) \quad \text{輸出層的誤差函數} \quad (4.4)$$

$$D_h = \text{Net}_h * (1 - \text{Net}_h) * \sum_j W_{hj} * D_j \quad \text{隱藏層的誤差函數} \quad (4.5)$$

$$\Delta W_{hj} = \eta * D_j * \text{Net}_h \quad \Delta b_j = -\eta * D_j \quad \text{隱藏層與輸出層間神經元權重和偏權值修正} \quad (4.6)$$

$$\Delta W_{ih} = \eta * D_h * x_i \quad \Delta b_h = -\eta * D_h \quad \text{隱藏層與輸入層間神經元權重和偏權值修正} \quad (4.7)$$

8. 直到輸出的平均誤差收斂時，此架構便可拿來應用。

$$E = \sum_j | (Y_j - Y_{jn}) | / 2000 \quad Y_j : \text{參考輸出} \quad Y_{jn} : \text{真實輸出} \quad (4.8)$$

模擬結果：

2000 組驗證樣本馬達角度( $\theta_1$ 、 $\theta_2$ 、 $\theta_3$ )的輸出平均誤差(E1、E2、E3)分別為 3.4712 度、1.6145 度和 1.7871 度，礙於論文版面無法將 2000 組數據都陳列出來，表 4.3 僅列出驗證樣本與倒傳遞類神經網路前 20 組輸出比較。

表 4.3 驗證樣本與 BP 輸出前 20 組比較

| $\theta_1$ | BP 輸出 $\theta_1$ | $\theta_2$ | BP 輸出 $\theta_2$ | $\theta_3$ | BP 輸出 $\theta_3$ |
|------------|------------------|------------|------------------|------------|------------------|
| 29         | 27.5539          | 10.        | 9.4553           | 17         | 18.4993          |
| 24         | 23.6071          | 39         | 41.1387          | 34         | 35.9568          |
| 16         | 19.0681          | 28         | 24.3575          | -89        | -82.5413         |
| 53.        | 52.9903          | -21        | -21.1805         | -63        | -63.3282         |
| 100        | 102.6274         | -3         | -3.9693          | -33        | -32.1822         |
| 62         | 60.2894          | 30         | 28.1211          | -87        | -81.7025         |
| 38         | 38.2252          | 19         | 18.7706          | -32        | -32.2693         |
| 64         | 64.0132          | -18        | -18.3010         | -17        | -19.3471         |
| 90         | 90.6248          | 32         | 35.5384          | -10        | -11.2443         |
| 59         | 60.0246          | 29         | 29.0799          | -1         | -1.0361          |
| 42         | 41.4340          | 20         | 20.1251          | -48        | -46.6613         |
| 67         | 68.1003          | 4          | 3.4228           | -30        | -29.8762         |
| 104        | 91.4752          | 2          | 9.2159           | 29         | 29.6188          |
| 2          | -0.7299          | 34         | 32.7259          | 10         | 9.9776           |
| -6         | -7.7882          | -18        | -17.9085         | -1         | -1.9236          |
| 108        | 107.9987         | 7          | 8.6719           | -33        | -31.1010         |
| -2         | 1.1437           | 7          | 7.0958           | 13         | 14.0967          |
| 20         | 17.4165          | 25         | 23.2298          | 2          | 2.6888           |
| -26        | -26.1298         | 9          | 9.7733           | -83        | -79.6302         |
| 53         | 53.0356          | -21        | -20.7982         | -63        | -63.0234         |

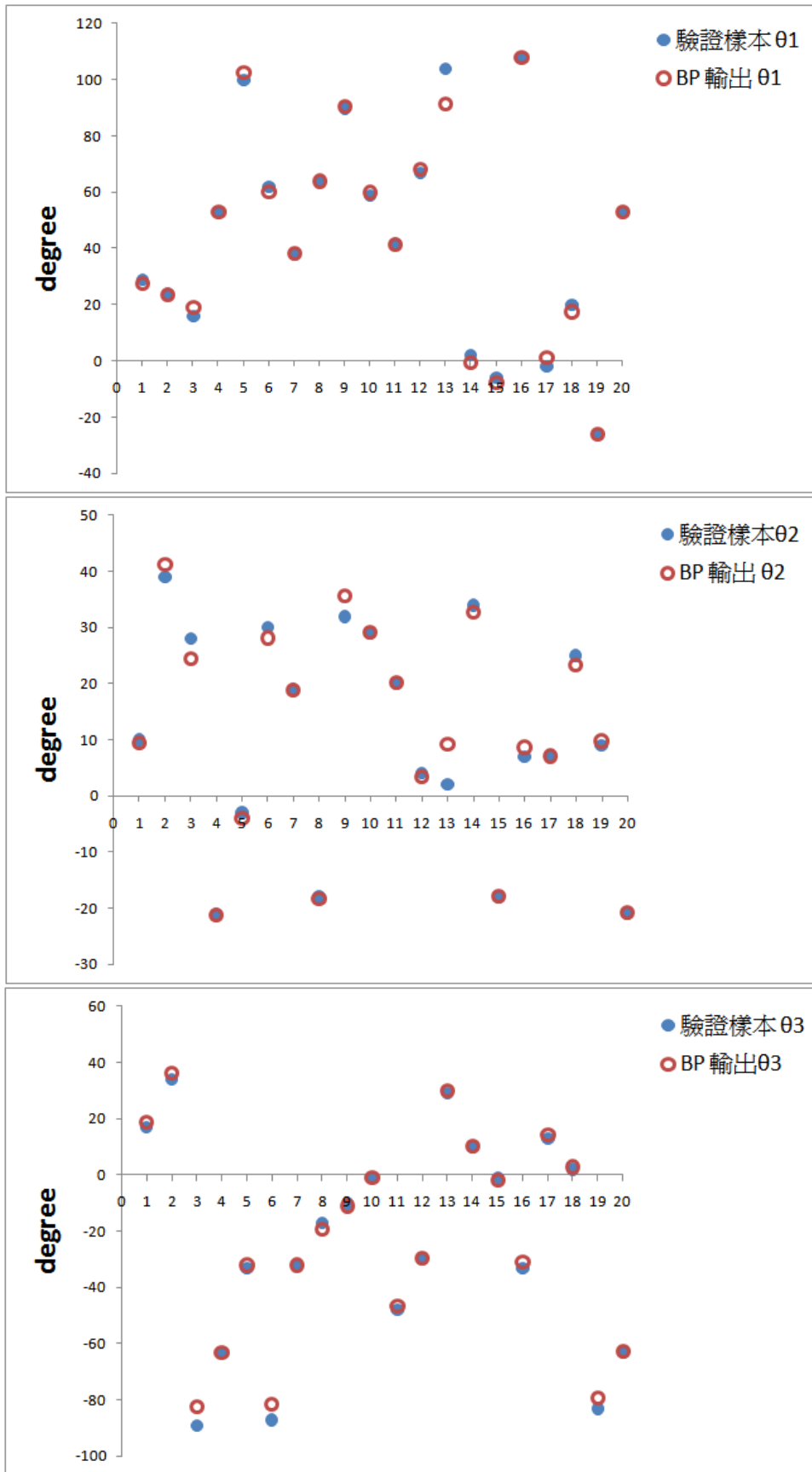


圖 4.12 20 組驗證樣本與 BP 角度比較圖

討論：

模擬結果證實倒傳遞神經網路可近似非線性反向運動學函數，使我們在不知道受控場數學函式下，依然可以獲得理想輸出，這就是與傳統控制最大不同點；但是對於倒傳遞神經網路架構中參數，例如：神經元個數、初始權重值、初始偏權值、轉移函數選擇...等等，只能藉由一些經驗公式和試誤法，不斷修改嘗試而得最佳解，不過類神經網路在學習中，會將輸出誤差以倒傳遞方式一層一層往回修正所有函數參數，這是他最大的優點，此行為就像控制系統的回授訊號；另外參數和架構相同情況下並不是每次學習都會收斂，推斷是因為權重和偏權值是以隨機方式設定所導致的。

#### 4.4 ANFIS 應用至空間三連桿手臂

ANFIS 以傳統模糊理論為基礎、類神經網路為架構所發展的一套理論。和傳統模糊理論相同之處在於，ANFIS 運用專家的經驗，根據 If...then...推論方式產生一組輸入與輸出之間的關係法則，而不限於一般的數學理論基礎。而優於傳統模糊理論之處則是利用類神經網路架構，每一個節點都能藉由學習調整參數，達到適應性的功能。

ANFIS 是由許多的節點與特定函數所構成，改變節點和函數值以接近學習樣本，經過多次的學習之後，此節點與函數將得到最佳值。令原始學習樣本與學習後的輸出值之差為目標函數，以梯降法(Gradient Descent Method)和連鎖律(Chain Rule)為調整的依據，目的在於降低其目標函數之均方誤差，

其中歸屬函數和規則庫都可在學習中得到標準法則。

步驟如下：

1. 首先建立四足機器人右前腳圖 4.9 座標系，使用 D-H 理論求得 D-H 表和 D-H 矩陣，此步驟與上一節倒傳遞類神經網路步驟 1 相同，目的是比較不同理論應用到空間三連桿時，經由誤差觀察兩種理論的優缺點
2. 利用正向運動學公式以隨機的方式產生 4000 組三個馬達的角度，代入正向運動學公式求得其對應的 4000 組  $x$ 、 $y$ 、 $z$  座標，其中 2000 組當作學習樣本，剩下 2000 組當作驗證樣本，此步驟也與上一節倒傳遞類神經網路步驟 2 相同。
3. 建立 ANFIS 架構，本文利用 3 個架構(圖 4.13)替代反向運動學函數，因為從反向運動學角度而言，求解  $\theta_1$  僅需用到  $x$ 、 $y$  座標與  $z$  座標無關，故 anfis1 的輸入只有兩個( $x$ 、 $y$ )，如此一來簡化了 anfis1 架構，大量減少了 anfis1 的程式學習時間，而  $\theta_2$ 、 $\theta_3$  與  $x$ 、 $y$ 、 $z$  相依，故 anfis2 和 anfis3 的輸入有三個( $x$ 、 $y$ 、 $z$ )。

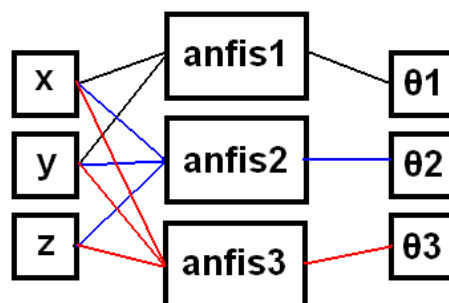


圖 4.13 ANFIS 訓練架構圖

使用 MATLAB 的「genfis1」和「anfis」兩個指令，「genfis1」以網格分割(grid partiton)方式產生 ANFIS 前提部與推論部的初始參數值，「anfis」會呼叫「genfis1」函數產生的初始參數來進行學習，本節以隨機方式產生 4000 組樣本，其中 2000 組學習樣本、2000 組驗證樣本、學習次數

$n=300$ 、轉移函數為 gbellmf、gaussmf，其函數標準式為：
$$\frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}}$$

$e^{-\frac{(x-d)^2}{2\sigma^2}}$ ，圖 4.14 為不同  $a$ 、 $b$ 、 $c$ 、 $d$ 、 $\sigma$  值時的圖形，可得知鐘形函數  $a$  參數影響寬度、 $b$  參數影響兩側斜率、 $c$  參數影響  $x$  方向偏移量；

而對於高斯函數  $d$  參數影響  $x$  方向偏移量、 $\sigma$  參數影響兩側斜率。

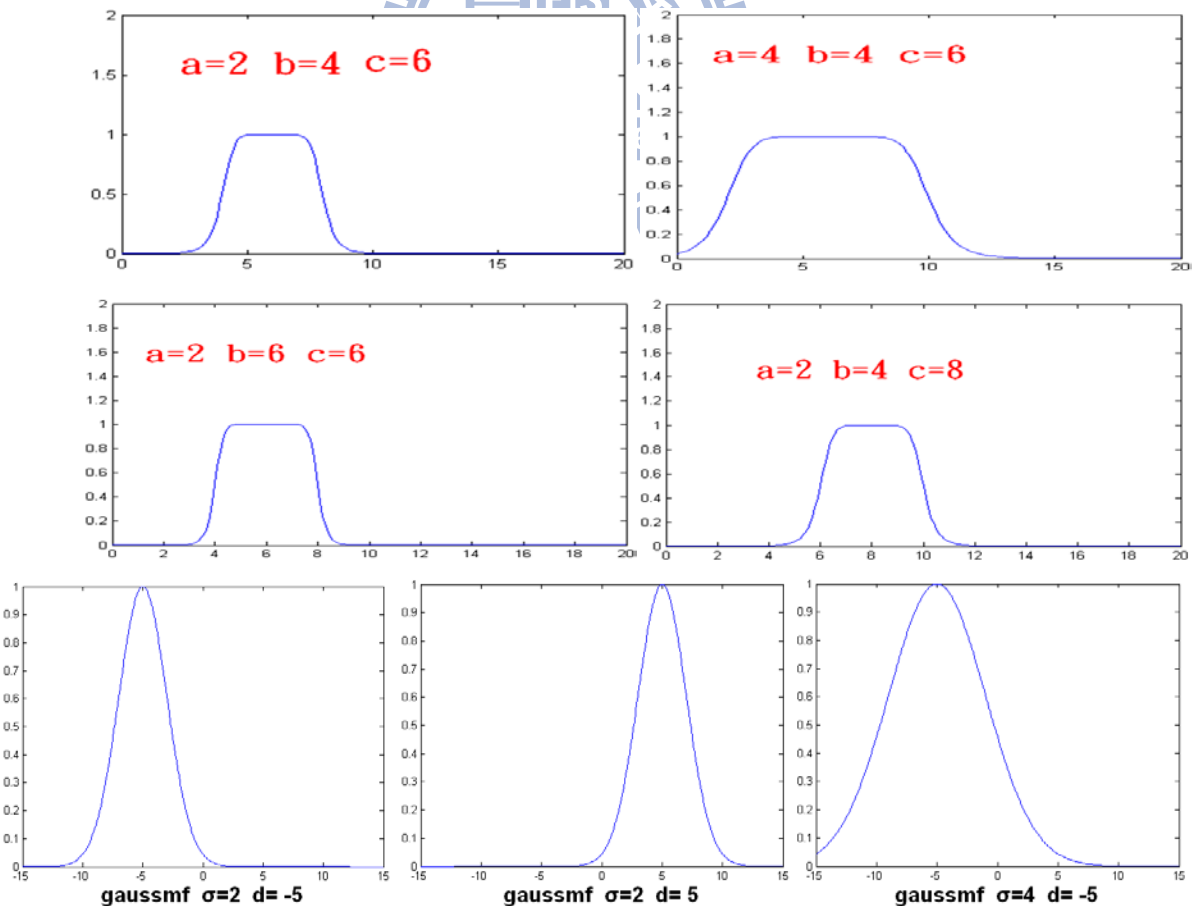


圖 4.14 不同  $a$ 、 $b$ 、 $c$ 、 $d$ 、 $\sigma$  值時的 gbellmf 與 gaussm

MATLAB 部分程式如下：

```
n=300 % 學習次數

% anfis1
a=genfis1([x1' y1' c1'],[8 8],
char('gaussmf','gaussmf'));
a=anfis([x1' y1' c1'],a,n);
% anfis2
b=genfis1([x1' y1' z1' c2'],[6 6 6],
char('gbellmf','gbellmf','gbellmf'));
b=anfis([x1' y1' z1' c2'],b,n);
% anfis3
c=genfis1([x1' y1' z1' c3'],[6 6 6],
char('gbellmf','gbellmf','gbellmf'));
c=anfis([x1' y1' z1' c3'],c,n);

代入 2000 組驗證樣本，比較 ANFIS 輸出與理論值

% anfis1
result1=evalfis([check_x1' check_y1'],anfis1);
compare1=[ check_c1' result1]
% anfis2
result2=evalfis([check_x1' check_y1' check_z1'],anfis2);
compare2=[ check_c2' result2];
% anfis3
result3=evalfis([check_x1' check_y1' check_z1'],anfis3);
compare3=[ check_c3' result3];
```

4. 若輸出誤差不理想，必須回到步驟 3，利用試誤法不斷嘗試修改參數(學習樣本數、轉移函數種類、歸屬函數個數、學習次數...等等)，直到輸出誤差收斂至可接受時，此架構便能應用至實驗上。

本文使用誤差公式：

$$\text{Error} = \sum | \text{ANFIS輸出} - \text{D-H理論值} | / 2000;$$



```

% anfis1
E1=abs([check_c1']-[result1]);
E1=(sum(E1)/(d/2))
% anfis2
E2=abs([check_c2']-[result2]);
E2=(sum(E2)/(d/2))
% anfis3
E3=abs([check_c3']-[result3]);
E3=(sum(E3)/(d/2))

```

2000 組驗證樣本的輸出平均誤差(E1、E2、E3)分別為 2.85 度、1.682

度、0.9404 度，圖 4.15、圖 4.16、圖 4.17 為學習完成後 anfis1、anfis2、

anfis3 輸入的歸屬函數。

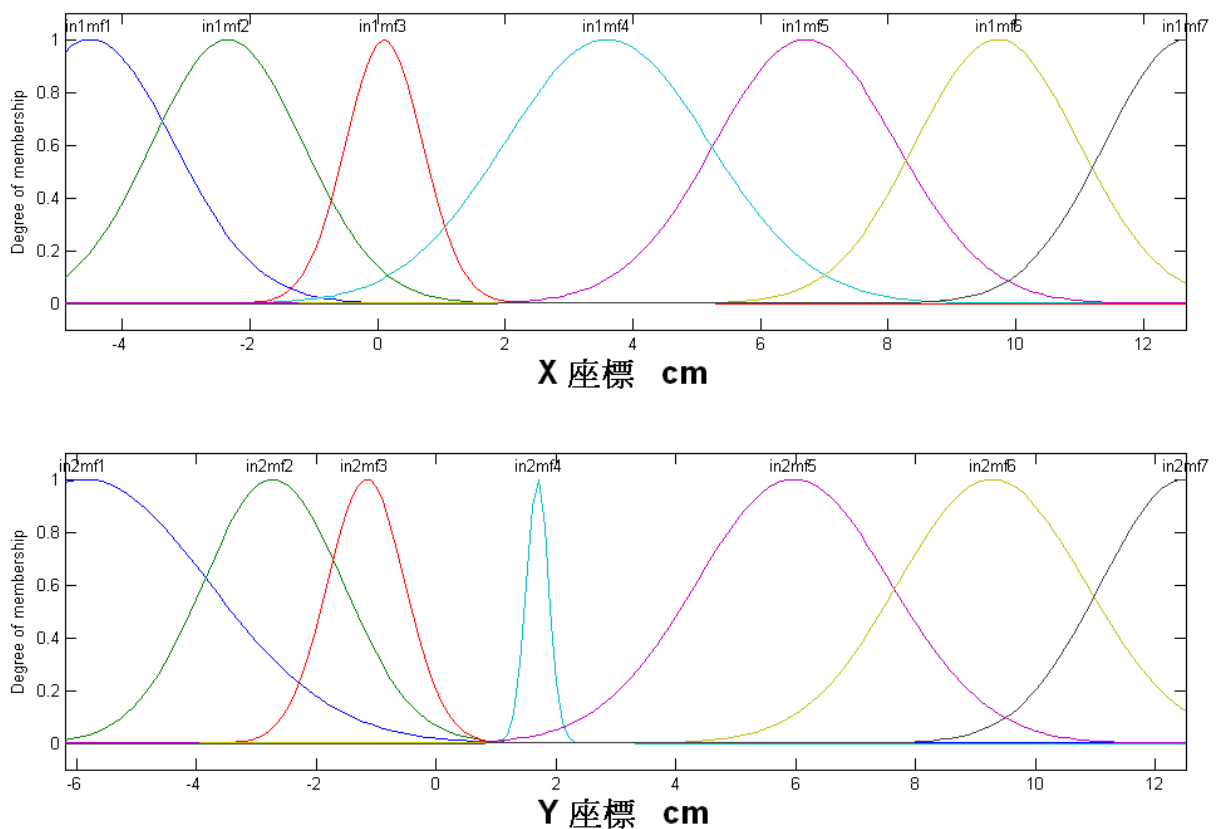


圖 4.15 學習後 ANFIS1 輸入的歸屬函數

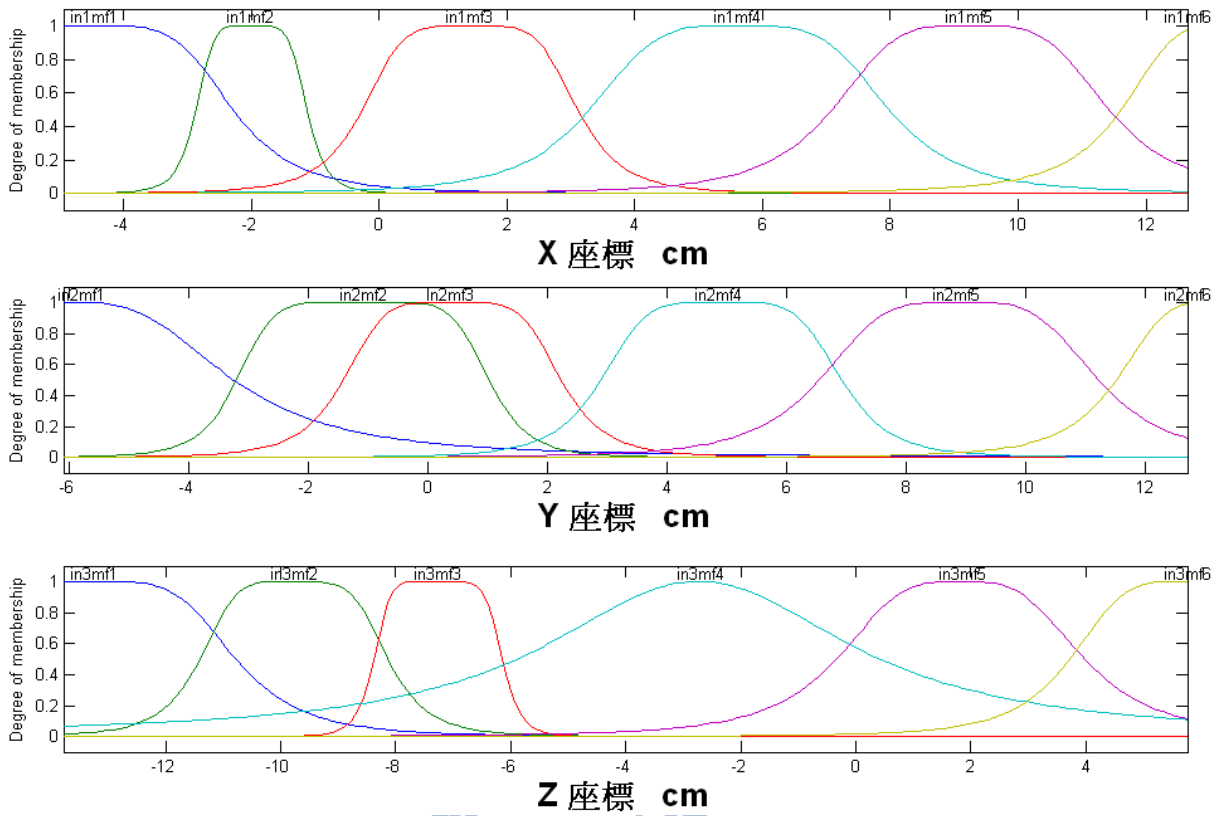


圖 4.16 學習後 ANFIS2 輸入的歸屬函數

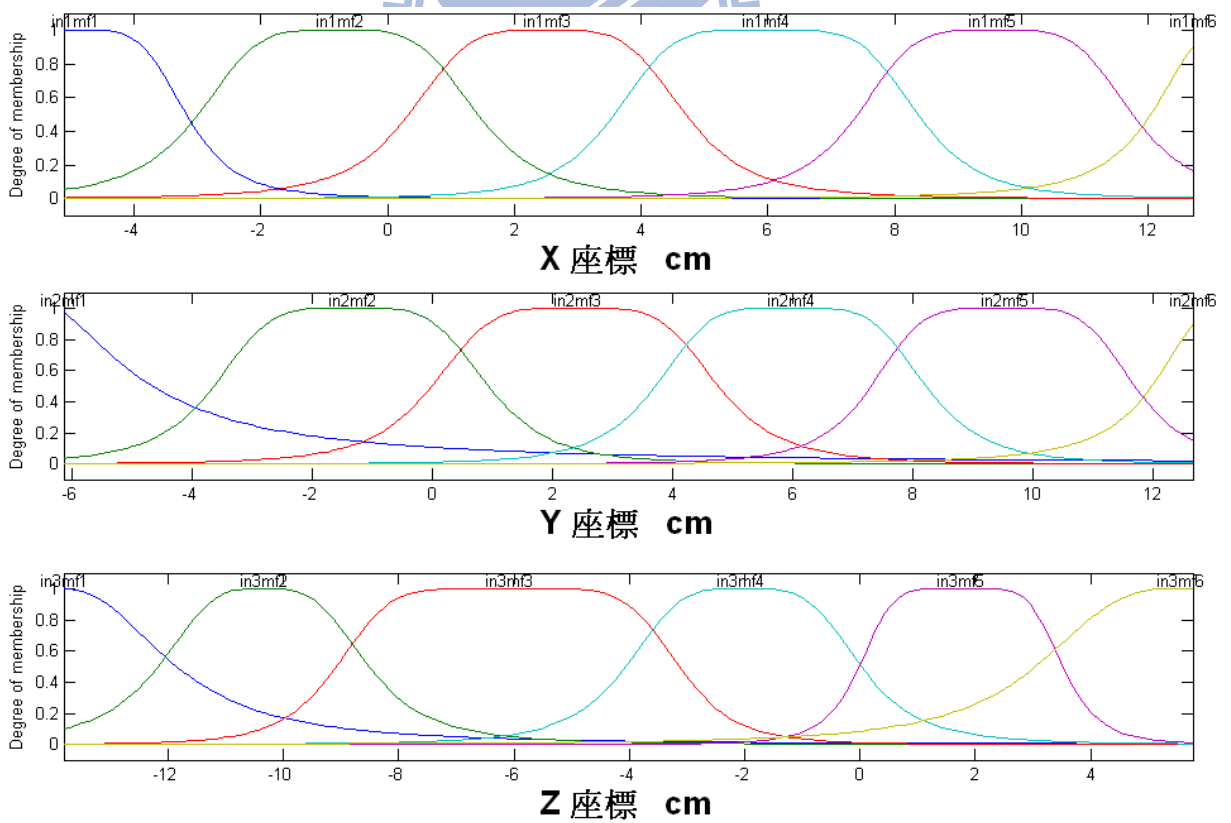
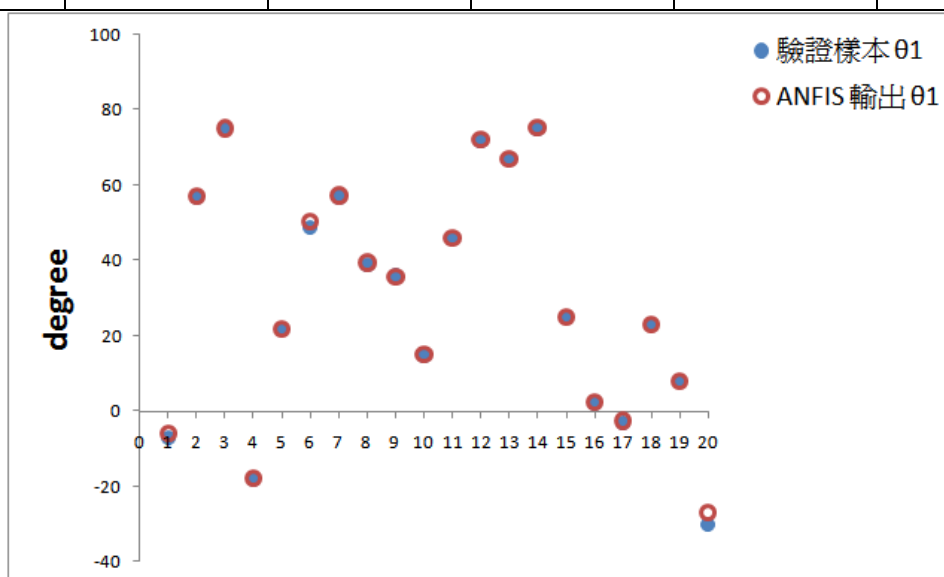


圖 4.17 學習後 ANFIS3 輸入的歸屬函數

驗證樣本總共有2000組，表4.4為僅列出前20組驗證樣本與ANFIS輸出比較

表 4.4 驗證樣本 與 ANFIS 輸出 前 20 組比較

| 驗證 01 | ANFIS 01 | 訓練 02 | ANFIS 02 | 訓練 03 | ANFIS 03 |
|-------|----------|-------|----------|-------|----------|
| -7    | -6.0289  | -20   | -19.8036 | 35    | 34.9989  |
| 57    | 56.9626  | -29   | -28.9231 | -36   | -37.2762 |
| 75    | 74.9925  | -9    | -8.7645  | -64   | -63.9375 |
| -18   | -17.9160 | -42   | -41.8775 | 27    | 27.0663  |
| 22    | 21.7804  | 29    | 29.2369  | -44   | -43.4977 |
| 49    | 50.1324  | 43    | 42.5853  | -27   | -26.9310 |
| 57    | 57.1793  | -12   | -12.2386 | -32   | -31.8188 |
| 39    | 39.3247  | 44    | 42.6811  | -65   | -64.4153 |
| 36    | 35.4740  | 42    | 44.7640  | 22    | 21.6420  |
| 15    | 14.9877  | 21    | 19.8582  | -82   | -81.6900 |
| 46    | 45.9444  | -27   | -26.9043 | 26    | 25.8109  |
| 72    | 72.0721  | 0     | 0.5773   | -83   | -80.1369 |
| 67    | 66.9801  | 27    | 26.5440  | -62   | -62.3571 |
| 75    | 75.1106  | 36    | 36.0343  | -33   | -33.2756 |
| 25    | 24.8369  | -14   | -13.2536 | 0     | 0.2171   |
| 2     | 2.2728   | -31   | -31.2698 | -15   | -15.4296 |
| -3    | -2.5775  | 0     | -3.6238  | -89   | -83.9106 |
| 23    | 22.9491  | -9    | -8.5006  | -56   | -55.7687 |
| 8     | 7.8842   | -3    | -3.1042  | -84   | -84.1068 |
| -30   | -26.8598 | 43    | 43.4997  | -23   | -22.7756 |



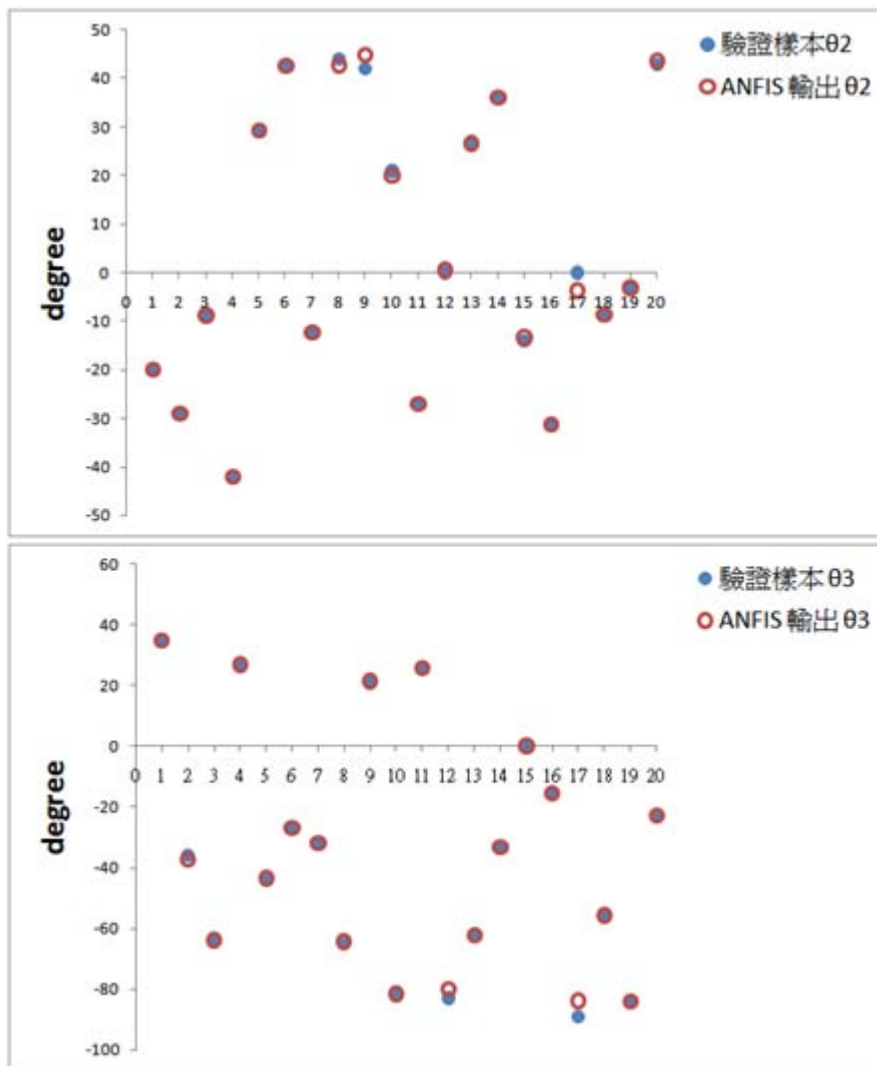


圖 4.18 20 組驗證樣本與 ANFIS 角度比較圖

討論：

4.3節與4.4節主要是將倒傳遞類神經網路與ANFIS分別應用至空間三連桿，因為兩者理論架構不同，所以無法將兩者理論設定在相同條件下比較誤差，只能在相同的2000組學習樣本、2000組驗證樣本和學習300次情況下，比較誤差；模擬時發現一個現象，BP在架構不改變時每次程式執行的結果有時會發散，差異很大；但是ANFIS架構不改變時，不論程式執行多少次，其輸出誤差皆差異不大，換句話說，ANFIS相對於BP是較穩定了。

## 4.5 運算時間比較

本節比較 ANFIS 與反向運動學應用於空間三連桿手臂的電腦程式

MATLAB 運算時間，比較流程如下：

1. 挑選 2000 組驗證樣本其中 10 組  $\theta_1$ 、 $\theta_2$ 、 $\theta_3$ ，代入正向運動學產生 10 組  $x$ 、 $y$ 、 $z$  座標視為「比較樣本」。
2. 使用 D-H 理論求解反向運動學函數，給定馬達角度範圍，讓聯立方程式恰有一組解。

$$({}^0A_1)^{-1} * {}^0A_4 = {}^1A_2 * {}^2A_3 * {}^3A_4 \quad (4-9)$$

比較(4-9)式左右兩矩陣的第四行 1~3 列，可得三個聯立方程式。

$$y * \cos(\theta_1) - x * \sin(\theta_1) = 0.9 \quad (4-10)$$

$$7 * \cos(\theta_2 + \theta_3) + 5.7 * \sin(\theta_2) = -2.8 - z_1 \quad (4-11)$$

$$-7 * \sin(\theta_2 + \theta_3) + 5.7 * \cos(\theta_2) = x_1 * \cos(\theta_1) + y_1 * \sin(\theta_1) \quad (4-12)$$

3. 將 10 組  $x$ 、 $y$ 、 $z$  座標分別帶入反向運動學公式和 ANFIS，使用 MATLAB 程式計時指令「tic」和「toc」，比較 10 組比較樣本 MATLAB 運算的時間和誤差，如圖 4.19 所示。

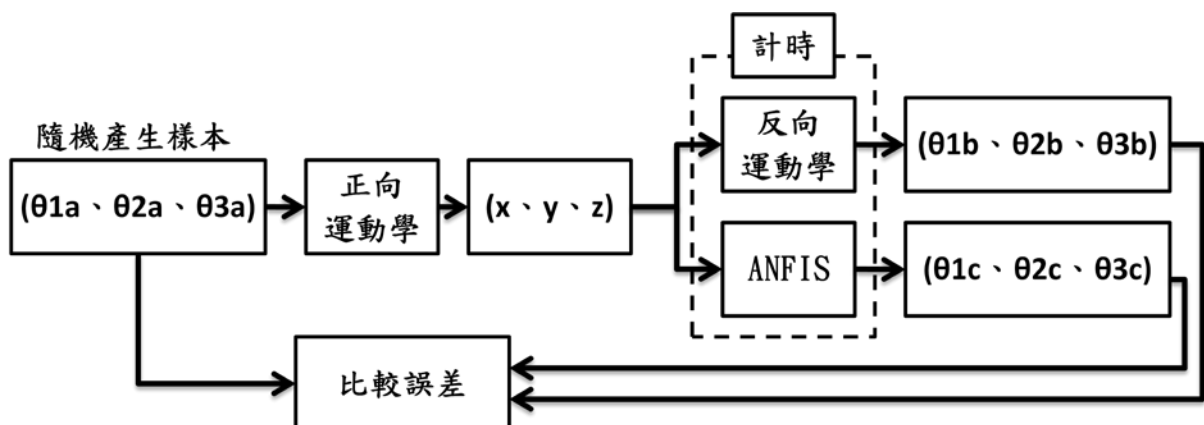


圖 4.19 ANFIS 和反向運動學之時間與誤差比較流程

模擬結果：

表 4.5 10 組比較樣本運算時間

| samples | Time(second)       |          |
|---------|--------------------|----------|
|         | Inverse kinematics | ANFIS    |
| 1       | 3.7289             | 0.9111   |
| 2       | 3.9458             | 0.5817   |
| 3       | 3.9171             | 0.5662   |
| 4       | 4.1316             | 0.5636   |
| 5       | 4.1467             | 0.5641   |
| 6       | 2.8720             | 0.5639   |
| 7       | 4.4723             | 0.5638   |
| 8       | 3.2113             | 0.5639   |
| 9       | 3.3154             | 0.5650   |
| 10      | 4.7061             | 0.5653   |
| 平均      | 3.8447 秒           | 0.6008 秒 |

表 4.6 比較樣本、反向運動學和 ANFIS 輸出結果比較

| 比較樣本       |            |            | 反向運動學      |            |            | ANFIS 輸出   |            |            |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| $\theta_1$ | $\theta_2$ | $\theta_3$ | $\theta_1$ | $\theta_2$ | $\theta_3$ | $\theta_1$ | $\theta_2$ | $\theta_3$ |
| -5         | 14         | -47        | -5.0000    | 14.0000    | -47.0000   | -4.8614    | 14.0809    | -45.6855   |
| 21         | 18         | -46        | 21.0000    | 18.0000    | -46.0000   | 20.6160    | 17.6328    | -46.1630   |
| 2          | 14         | 2          | 2.0000     | 14.0000    | 2.0000     | 2.3379     | 14.1692    | 1.9676     |
| -2         | -7         | -25        | -2.0000    | -7.0000    | -25.0000   | -1.9719    | -6.4805    | -25.2122   |
| 16         | 19         | -34        | 16.0000    | 19.0000    | -34.0000   | 16.2744    | 18.4095    | -33.6657   |
| 35         | -1         | -70        | 35.0000    | -1.0000    | -70.0000   | 35.0584    | -0.3217    | -69.7816   |
| 28         | 43         | -1         | 28.0000    | 43.0000    | -1.0000    | 64.7729    | 38.4214    | -1.1391    |
| 28         | -1         | -77        | 28.0000    | -1.0000    | -77.0000   | 28.0628    | 1.5179     | -79.0892   |
| 25         | -12        | -75        | 25.0000    | -12.0000   | -75.0000   | 24.9991    | -9.6569    | -78.7341   |
| 16         | 29         | -16        | 16.0000    | 29.0000    | -16.0000   | 16.1728    | 29.4933    | -15.9399   |

從表 4.5 可看出 ANFIS 比反向運動學節省了 6.4 倍的運算時間，而且從表

4.6 中可看出雖然 ANFIS 輸出角度( $\theta_1$ 、 $\theta_2$ 、 $\theta_3$ )會有些許誤差，但是對於本文跨越河石地形的研究是可接受的範圍，當四足機器人行走河石地形過程中腳底皆是踏在河石區域內，此部分可由後面實驗證明。

#### 4.6 ANFIS 應用至 12-DOF 四足機器人

完成空間三連桿(右前腳)模擬後，便可延伸至四足機器人其他三足，四足機器人具有12個自由度，運動學公式相當繁雜和冗長，更別說是反向運動學公式推導，因此要利用ANFIS理論取代反向運動學函數，因為本實驗室設計的四足機器人架構為對稱架構，利用D-H理論，建立完座標系後便可發現以下兩點：

1. 右前腳與左後腳踏在相同數字位置時(圖4.20)，馬達1與馬達7角度相同、馬達2與馬達8角度相同、馬達3與馬達9角度相同。
2. 右前腳與左前腳踏在相同數字位置時(圖4.20)，馬達1與馬達4角度差負號、馬達2與馬達5角度差負號、馬達3與馬達6角度差負號。

簡而言之，當右前腳踏在方格A1時，若將7、8、9馬達角度設定成和1、2、3馬達角度一樣，會使左後腳也踏在方格C1位置；若將1、2、3馬達角度加上負號後丟給左前腳4、5、6馬達，則左前腳也會踏在方格B1位置，以此類推右後腳也可順利踏在方格D1位置，因此四足機器人前進時馬達角度，可由右前腳去推算其他三腳，其他數字位置以此類推。

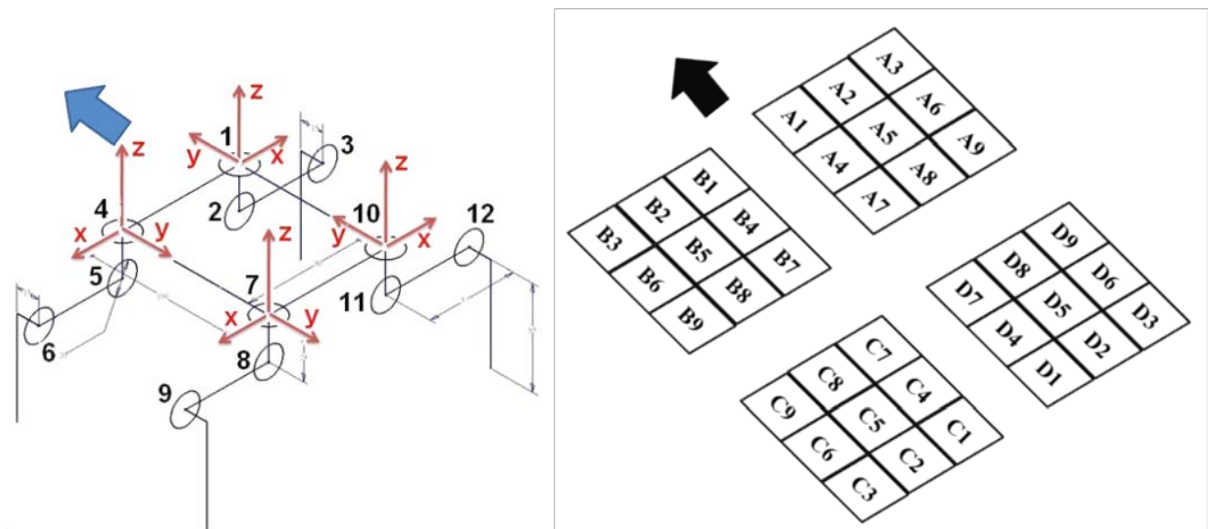


圖 4.20 四足機器人座標系與步伐位置

如此一來可簡化12個自由度導致運動學公式太過複雜的問題，將12-DOF四足機器人架構簡化成4個3-DOF空間三連桿，因此可延續4.4節3-DOF空間三連桿模擬結果應用至本節四足機器人，步驟如下：

1. 確定四足踏步順序(圖4.21)，本文順序為右前腳→左後腳→左前腳→右後腳，此種舉腳順序稱為波浪步態，這不但是自然界中四足動物最常使用的步態，也是目前各種四足機器人研究中，直走時的最佳步態順序 [16],[17],[18]。
2. 規劃踏步前機身重心調整規則，重心投影至地面的點必須落在腳底三足所形成的三角形裡面，才能使四足機器人行走時不傾倒。
3. 量測四足下一步位置座標，代入右前腳的anfis1、anfis2、anfis3架構求得四足機器人12個馬達的轉動角度。



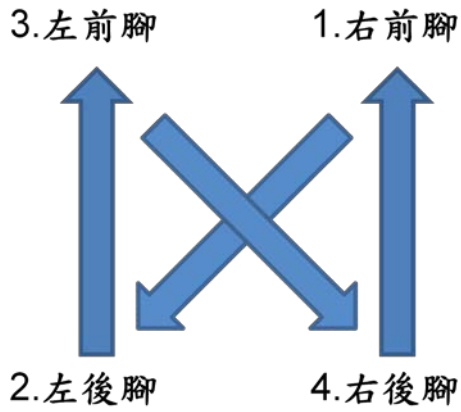


圖 4.21 踏步順序

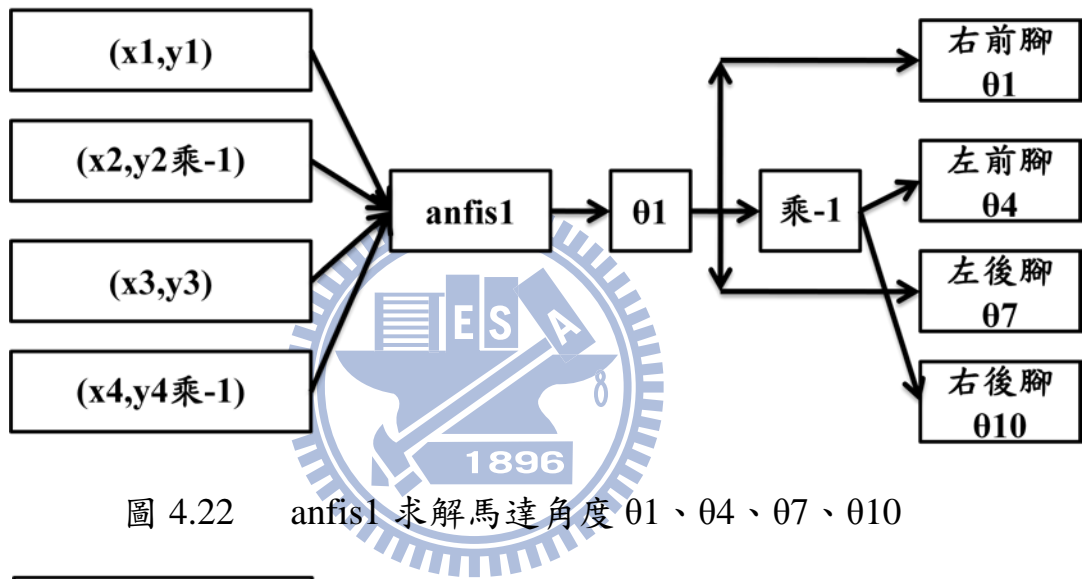


圖 4.22 anfis1 求解馬達角度  $\theta_1$ 、 $\theta_4$ 、 $\theta_7$ 、 $\theta_{10}$

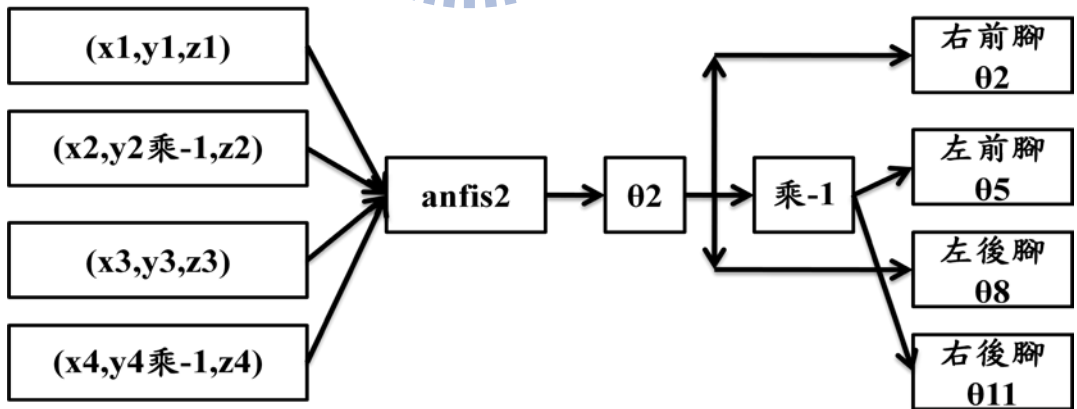


圖 4.23 anfis2 求解馬達角度  $\theta_2$ 、 $\theta_5$ 、 $\theta_8$ 、 $\theta_{11}$

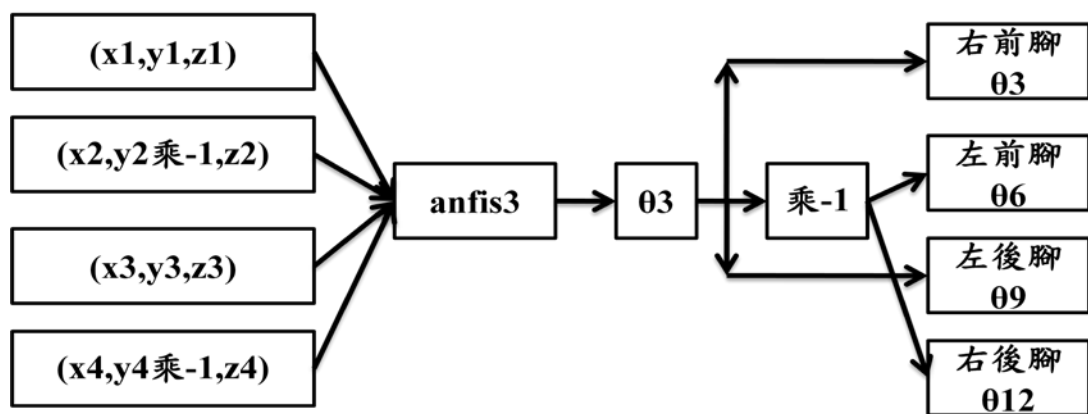


圖 4.24 anfis3 求解馬達角度  $\theta_3$ 、 $\theta_6$ 、 $\theta_9$ 、 $\theta_{12}$

#### 4.7 ZMP 平衡問題分析

四足機器人行走時最重要且最難的課題就是平衡問題，此問題分為兩部份，靜態和動態平衡，動態平衡主要是探討機器人行走時加速度對機身的影響，因為本文四足機器人，前進速度較緩慢，體型輕故加速度可省略不計；探討靜力平衡必需知道重心位置，當四足機器人抬起單腳時，重心投影至地面的投影點必須在支撐腳所構成的三角形內，才能使機身不傾倒，此外若要達到靜態穩定，當四足機器人前進時，同時只能抬起單腳，剩下支撐腳才能形成三角形支撐面，若同時移抬起兩腳，腳底支撐面積為一條線，是無法達成靜態平衡的，故本文四足機器人前進時步伐分成四個步驟順序為：右前腳→左後腳→左前腳→右後腳，此種舉腳順序稱為波浪步態，這不但是自然界中四足動物最常使用的步態，也是目前各種四足機器人研究中，直走時的最佳步態順序[16],[17],[18]。

我們將機器人系統分為五個部份，即機器人本體與4隻腳，它們的質量

分別為 $m_c$ 與 $m_i$ ， $i=1\sim 4$ 。且每個部份的質心位置則分為X與Y兩方向，X方向為 $X_{c.m.c}$ 與 $X_{c.m.i}$ ， $i=1\sim 4$ ；y方向為 $Y_{c.m.c}$ 與 $Y_{c.m.i}$ ， $i=1\sim 4$ 。

由以下兩式可求出整個機器人的質心座標在XY平面的投影：

$$X_{c.m} = \frac{\sum_{i=1}^4 m_i \times g \times X_{c.m.i} + m_c \times g \times X_{c.m.c}}{\sum_{i=1}^4 m_i \times g + m_c \times g} \quad (4.13)$$

$$Y_{c.m} = \frac{\sum_{i=1}^4 m_i \times g \times Y_{c.m.i} + m_c \times g \times Y_{c.m.c}}{\sum_{i=1}^4 m_i \times g + m_c \times g} \quad (4.14)$$

當然，質心座標在XY平面的投影就是靜態ZMP的位置，此外穩定度的定義為：由機器人之重心往地面作投影，其投影點，也就是ZMP與機器人腳在地面上的支撐點連線所構成之多邊形(又稱穩定多邊形)兩者間的最短距離(垂直距離)，稱為穩定度。為使機器人在整個運動的過程中保持穩定，不至於翻倒或倒下。穩定度必須大於或等於0。

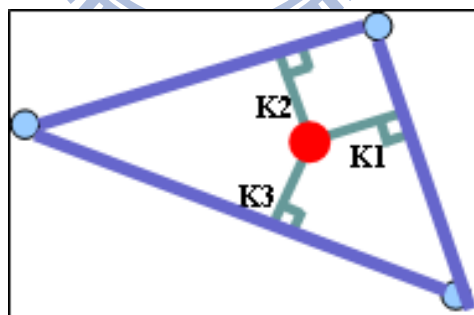


圖 4.25 ZMP 穩定多邊形

如圖 4.25 所示，當重心投影點與穩定多邊形的最短距離  $K_1$ 、 $K_2$ 、 $K_3$  皆大於 0，表示此四足機器人為穩定狀態，即使在機器人本體有傾斜的時候，由於假設機器人的四足接觸外力的方向皆與重力方向平行(自身重力,地平面的反作用力)，故使機器人翻倒的力矩只來自於機器人自身的重力，即為

參考座標系 Z 軸的方向，就算機器人本身有傾斜的情況，重心的重力還是直指向負 Z 軸，所以在靜態平衡的狀況下，可以只關注在 XY 平面上，重心投影點是否落在支撐腳的穩定多邊行內即可。

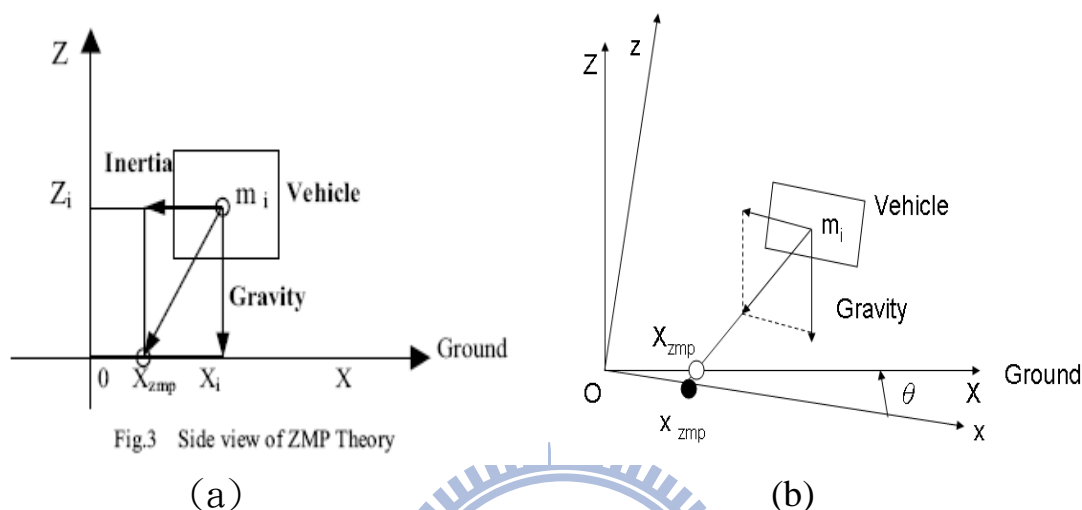


圖 4.26 ZMP 理論示意圖

當物體為靜止時，靜態平衡狀態是指重心會落在腳的支撐區域內，但如果物體是在移動的狀態，重力與慣性力的合力會指向腳之支撐區域內，則稱為動態平衡，這個與地面的交點稱做動態 ZMP(零矩點)，如圖 4.26(a)，若機器人慣性方向與重力方向有夾角則如圖 4.26(b)。在實際情況下，隨著機器人的移動而產生動量，重心位置還需要計算慣性力，此為動態穩定度所考量的問題。而靜態穩定度假設機器人的動量的考量小到可以忽略的情況下，通常應用在移動慢、體型輕、加速度不大的機器人上，本論文所採用的機器人因主要目的為跨越河石地形，故跨越時設定穩定性高、移動速度較慢，因此本論文中使用靜態 ZMP 來進行穩定性的分析。

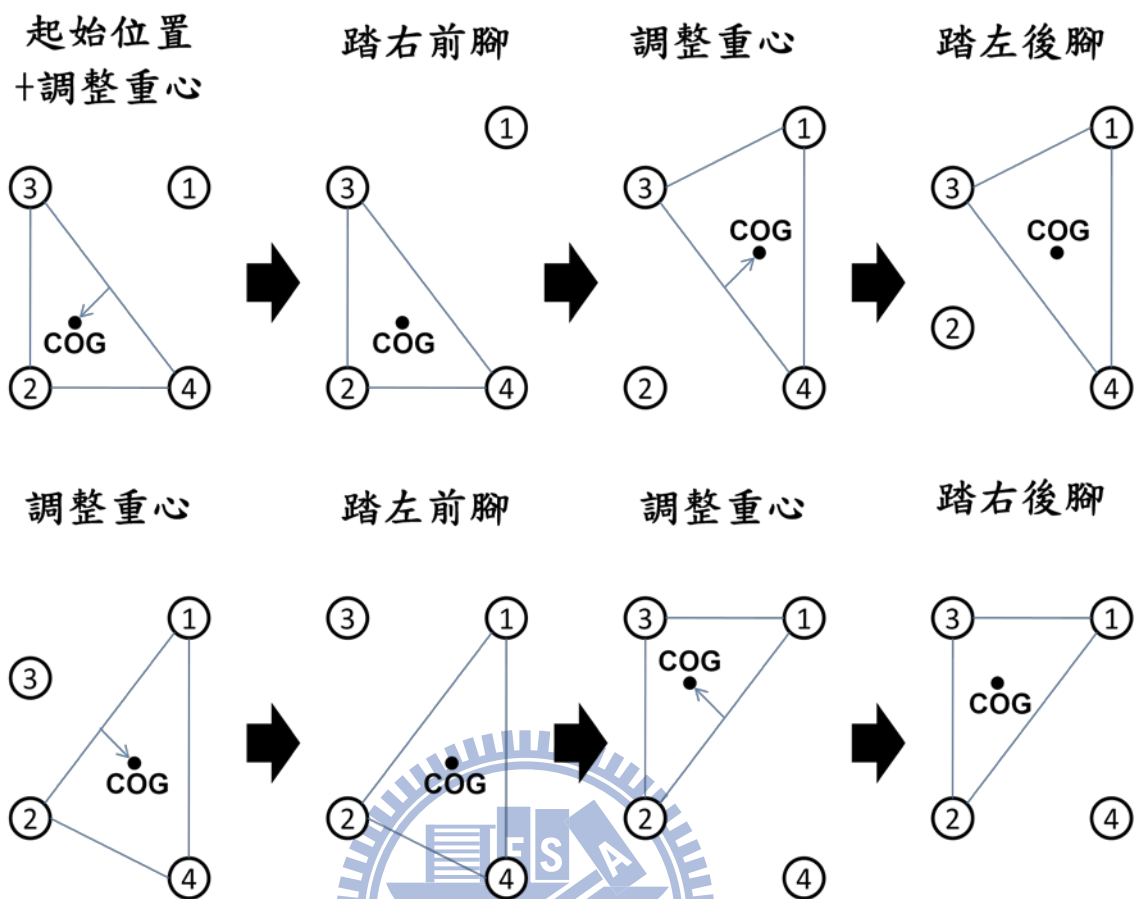


圖 4.27 四足機器人調整重心示意圖

圖4.27說明當四足機器人行走時本文偏移機身重心的順序，簡而言之，當要抬起單足時，為了符合靜態ZMP平衡，其重心投影至地面的點必須落在三角形支撐面內，因此12個馬達必須配合調整角度使機身重心偏移至安全區域，此外當移動機身時必須盡量避免腳底打滑現象，才不會造成誤差越來越大。

## 五、模擬與實驗結果

本文實驗目的為使用自適應類神經網路模糊推論系統理論計算四足機器人每一個步態的 12 個馬達角度，搭配 ZMP 理論調整重心，讓四足機器人行走時的每一個步態重心位置，都落在支撐腳所形成的三角形內，使四足機器人順利跨越非平坦河石地形而不傾倒。

### 5.1 程式流程規劃

本實驗程式流程規劃可分為三種方式進行：

1. 使用 PC 編輯 Visual sequencer 程式，透過 RS232 傳輸線將程式燒錄至 SSC-32 馬達控制器中，四足機器人便會照著程式設計的步伐流程跨越非平坦河石地形，此方法由實驗室另一個同學負責，參考圖 5.1。



圖 5.1 方法一 示意圖

2. 將 Visual sequencer 匯出.csv 檔案裡的角度，代入正向運動學 D-H 矩陣求得 $(x_i, y_i, z_i)$ 座標，再將座標代入以學習完成的 ANFIS 架構，求解相對應的角度，觀察比較.csv 檔案裡的角度和 ANFIS 輸出角度，若兩者角度誤差極小，便可說明 ANFIS 可以用來驗證方法一的結果，參考圖 5.2。

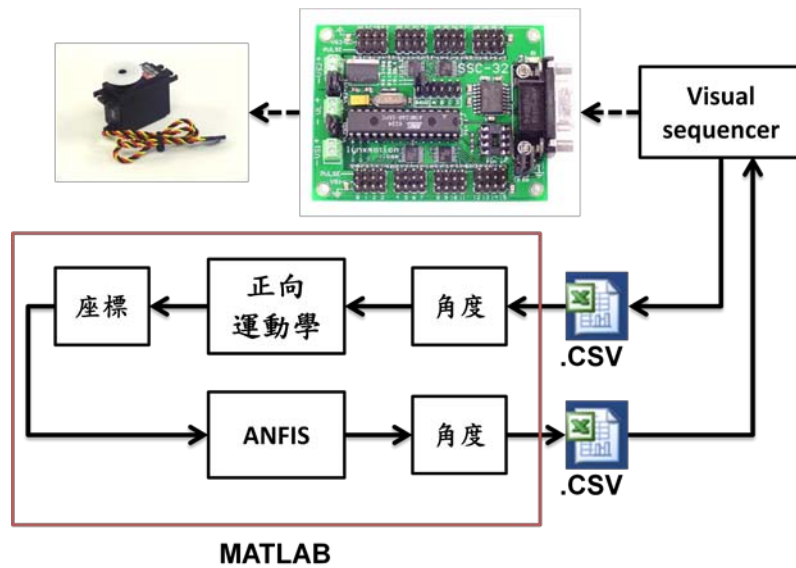


圖 5.2 方法二 示意圖

3. 使用者透過 GUI 介面點選隨機地形，透過地面絕對座標計算四足機器人每一步態機身與踏點的相對座標，將座標值輸入右前腳的 anfis1、anfis2、anfis3 求得 12 個馬達的角度( $\theta_i$ )，將每個步態的所有馬達角度存成矩陣，最後匯出成.csv 檔，讓 Visual sequencer 讀取.csv 檔中的角度，再透過 RS232 傳送至 SSC-32 馬達控制器中，量測實驗與模擬之間的誤差，若誤差大則必須修正模擬 ANFIS 架構參數，直到誤差很小為止，參考圖 5.3。

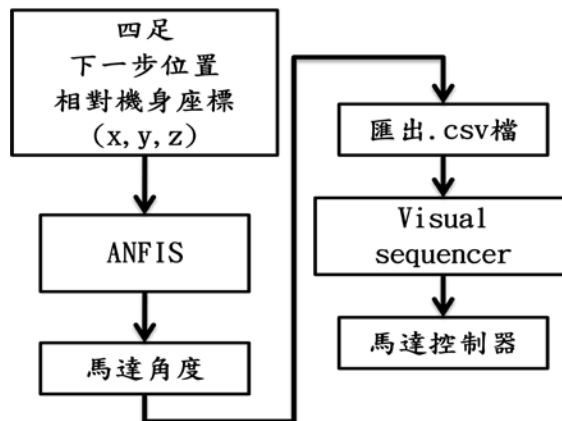


圖 5.3 方法三 示意圖

方法一由實驗試另一個同學負責，方法二主要是在驗證方法一的結果，本文重點研究部份為方法三，方法三為利用 MATLAB 撰寫一可套用至隨機地形的 ANFIS 通式程式，讓四足機器人路徑角度數據，隨著使用者輸入不同路徑而改變，替代傳統逆向運動學函數，有效減少電腦求解多自由度逆向運動學的冗長時間。

## 5.2 程式模擬結果

本節接續 5.1 節的方法二與方法三作詳細的說明並展示模擬結果，前者方法二僅是驗證方法一踩踏九宮格全部九個踏點位置的結果，而方法三為使用 ANFIS 理論取代複雜的反向運動學函數針對隨機踏點位置，並且搭配靜態 ZMP 來進行穩定性的分析，與傳統控制最大不同點為，不需知道控制系統數學函數情況下，就可藉由不斷學習達到理想的控制結果，首先是方法 2 流程如下：

1. 首先量測右前腳周圍九宮格踏點相對於機身的( $x_i$ 、 $y_i$ 、 $z_i$   $i=1\sim9$ )座標與當時馬達的角度。

表 5.1 實際量測角度與座標值

| 位置 | $\theta_1$ | $\theta_2$ | $\theta_3$ | X    | Y   | Z    |
|----|------------|------------|------------|------|-----|------|
| A1 | 92.26      | -1.22      | -15.14     | -1.2 | 7.7 | -9.1 |
| A2 | 63.4       | 0.12       | -20.03     | 2.8  | 7.7 | -9.1 |
| A3 | 43.12      | 14.04      | -56.17     | 6.8  | 7.7 | -9.1 |



|    |        |       |        |      |      |      |
|----|--------|-------|--------|------|------|------|
| A4 | 94.72  | -1    | 17.96  | -1.4 | 3.7  | -9.1 |
| A5 | 40.74  | -2.93 | 13.12  | 2.8  | 3.7  | -9.1 |
| A6 | 21.18  | -1.22 | -15.14 | 6.8  | 3.7  | -9.1 |
| A7 | 63.79  | 29.71 | 28.11  | -1.2 | 0.1  | -9.1 |
| A8 | -26.68 | 2.93  | 22.94  | 2.9  | -0.2 | -9.1 |
| A9 | -10.96 | -3.2  | -5.65  | 6.9  | -0.2 | -9.1 |

2. 將 Visual sequencer 匯出.csv 檔案裡的角度，使用 MATLAB 讀取後，代入正向運動學 D-H 矩陣求得座標，再跟實際量測的座標比較，由表 5.2 可看出，機身高度 z 值，實際明顯比理論降低約 2mm，此現象是因機身重量所造成，此為模擬與實際間的誤差，故模擬時必須修正此誤差量，才能更接近真實實驗。

表 5.2 實際量測座標與 D-H 理論座標比較

| 位置<br>(cm) | 實際<br>X | 理論<br>X | 實際<br>Y | 理論<br>y | 實際<br>z | 理論<br>Z |
|------------|---------|---------|---------|---------|---------|---------|
| A1         | -1.2    | -1.2018 | 7.7     | 7.6290  | -9.1    | -9.3952 |
| A2         | 2.8     | 2.8149  | 7.7     | 7.6311  | -9.1    | -9.3935 |
| A3         | 6.8     | 6.8486  | 7.7     | 7.6463  | -9.1    | -9.3742 |
| A4         | -1.4    | -1.1979 | 3.7     | 3.5707  | -9.1    | -9.3961 |
| A5         | 2.8     | 2.7875  | 3.7     | 3.5888  | -9.1    | -9.3982 |
| A6         | 6.8     | 6.8271  | 3.7     | 3.6105  | -9.1    | -9.3952 |
| A7         | -1.2    | -1.2376 | 0.1     | -0.4763 | -9.1    | -9.3530 |
| A8         | 2.9     | 2.7614  | -0.2    | -0.3804 | -9.1    | -9.3899 |
| A9         | 6.9     | 6.8157  | -0.2    | -0.4032 | -9.1    | -9.3985 |

3. 將實際量測座標，代入以學習完成的 ANFIS 架構，可求得輸出馬達角度，比較 ANFIS 輸出結果與 Visual sequencer 匯出的馬達角度的誤差大小(表 5.3)，從 27 個角度數據可計算出平均誤差為 0.5658 度，此結果說明 ANFIS 模擬的輸出結果接近方法一實驗結果，因此可藉由 ANFIS 驗證方法一。

表 5.3 Visual sequencer 角度與 ANFIS 輸出角度比較

| 位置 | 實際 01  | ANFIS 01 | 實際 02 | ANFIS 02 | 實際 03  | ANFIS 03 |
|----|--------|----------|-------|----------|--------|----------|
| A1 | 92.26  | 93.6636  | -1.22 | -0.0145  | -15.14 | -15.7254 |
| A2 | 63.4   | 63.5906  | 0.12  | 0.6073   | -20.03 | -20.2295 |
| A3 | 43.12  | 43.2094  | 14.04 | 14.6456  | -56.17 | -55.1151 |
| A4 | 94.72  | 90.8140  | -1    | -1.1456  | 17.96  | 18.3172  |
| A5 | 40.74  | 40.5636  | -2.93 | -2.8484  | 13.12  | 13.4329  |
| A6 | 21.18  | 21.1503  | -1.22 | -0.7079  | -15.14 | -15.4630 |
| A7 | 63.79  | 64.3373  | 29.71 | 28.7685  | 28.11  | 28.0695  |
| A8 | -26.68 | -28.0966 | 2.93  | 2.9149   | 22.94  | 22.7992  |
| A9 | -10.96 | -11.0458 | -3.2  | -3.0056  | -5.65  | -5.8786  |

4. 最後再將 ANFIS 輸出角度匯出.csv 格式檔案，讓 Visual sequencer 軟體讀取數據後去控制 12 個伺服馬達，使四足機器人完成指定的動作。

總而言之，方法二說明了利用 Visual sequencer 撰寫程式控制 12 個馬達也可使用 ANFIS 理論做到同樣結果，但是方法二是讀取方法一的角度做驗證，故只能說是驗證了方法一；但是本文研究最終目的是跨越隨機地形，

為了解決此問題必須撰寫一通式程式，使四足機器人前進步態可以隨著使用者輸入的地形不同而改變，因此提出方法三，流程如下。

1. 確定前進的方向和步伐大小，方向分別有右方、右前方、前方、左前方、左方、左後方、後方和右後方 8 個方向，右後方對腳底而言為機身最裡面的踏點，因為機身結構限制，若不是必要，盡量避開此踏點；前進步伐大小為 4 cm 和  $4\sqrt{2}$  cm 兩種，圖 5.4 為四足機器人前進路徑的地形俯視圖，A5、B5、C5、D5 分別為右前腳、左前腳、左後腳、右後腳預備動作時的踏點，而 A7、B7、C7、D7 為四足機器人機身底下最內側的踏點。

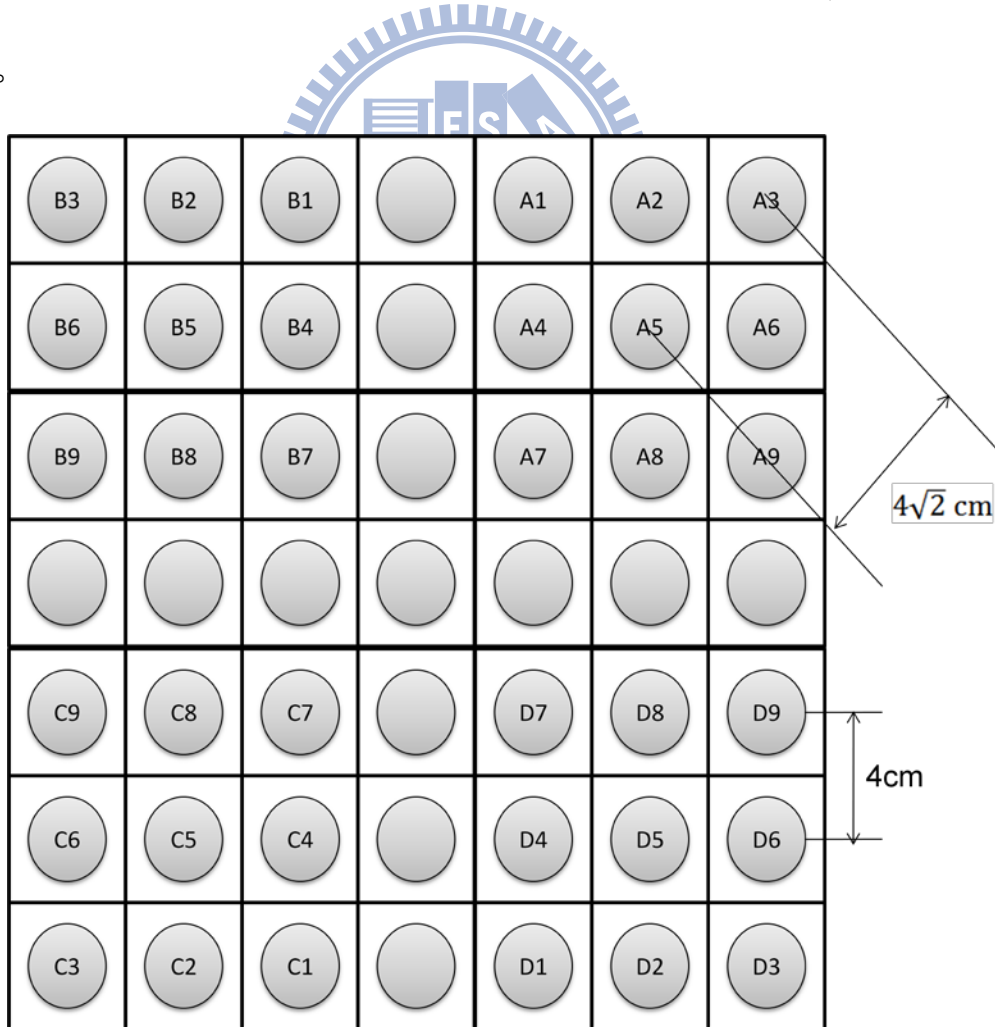


圖 5.4 地形俯視圖

2. 決定踏步順序，本文踏步順序為右前腳→左後腳→左前腳→右後腳，搭配靜態 ZMP 理論偏移重心位置，配合踏步順序，偏移重心順序為左後偏移→右前偏移→右後偏移→左前偏移，目的是將重心位置偏移至支撐腳所形成三角形的斜邊中點往內偏移一公分的位置，參考圖 4.27
3. 如圖 5.5 所示，建立圖形使用者介面(GUI)，繪製模擬地形，只要使用者使用滑鼠左鍵點選欲前進路徑，按下 OK 按鈕後可產生前進路徑之座標矩陣(xi、yi、zi)，給定座標後按下 calculate 按鈕，便可產生相對應的馬達角度數據，如此一來，不需再一一輸入踏點座標，讓操作更方便。

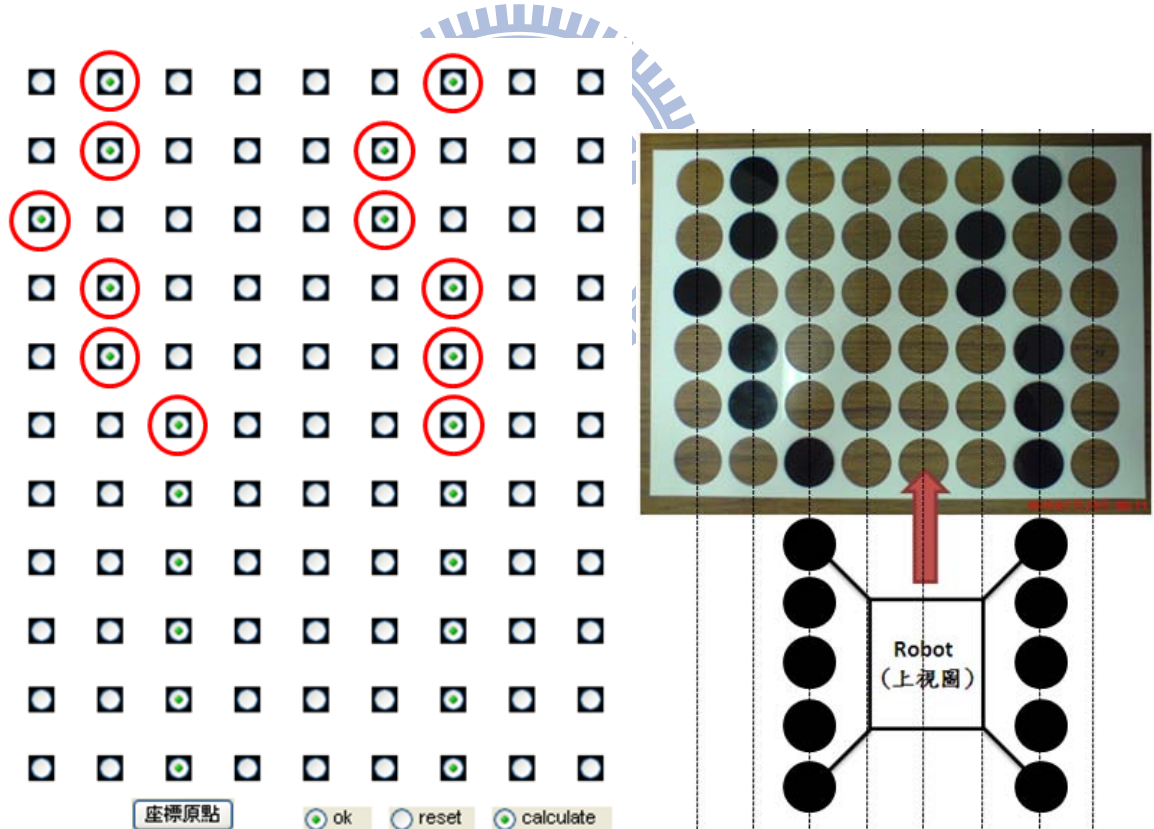


圖 5.5 ANFIS GUI 介面與實驗踏點圖

4. 在圖 5.4 地形 C5 的位置設立地面絕對座標(0, 0, 0)，藉由此固定座標可以推算所有踏點和機身上移動點的絕對位置座標，將踏點絕對座標與機

身上移動點絕對座標相減，求得四足下一步踏點相對機身的相對座標，參考圖 5.6。

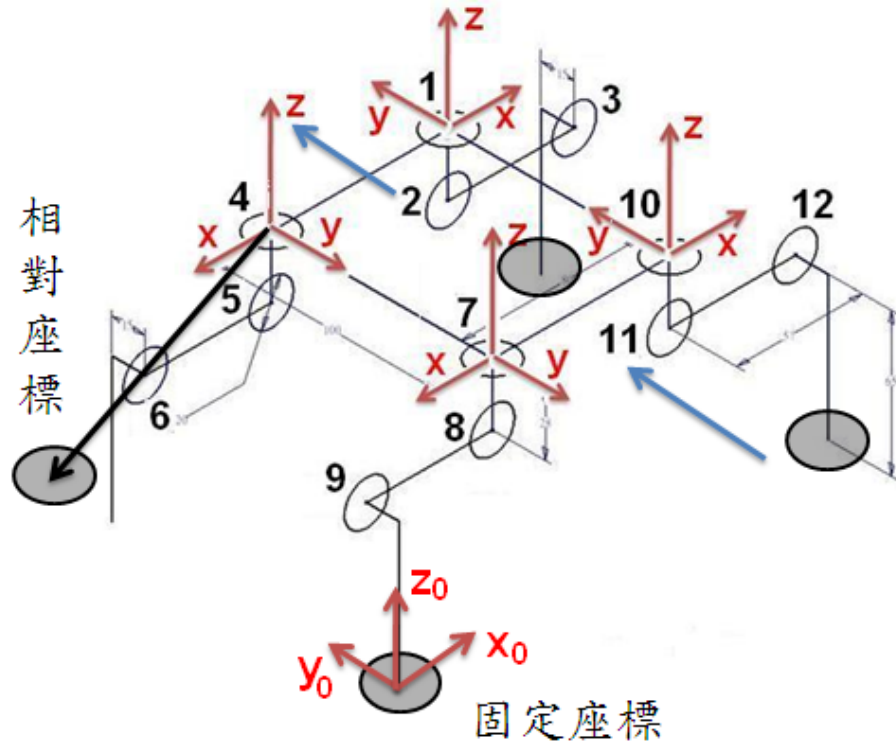


圖 5.6 機身與下一步踏點的相對座標

5. 將每個步態機身相對踏點的相對座標，加上實際與模擬之間的誤差修正，再代入學習完成的 ANFIS 求得每個步態所有馬達角度數據。
6. 將每個循環馬達角度利用 MATLAB 程式匯出.csv 檔，將.csv 檔匯入至 visual sequencer 透過 RS232 傳輸線傳送至馬達控制器 SSC-32 後，進一步控制四足機器人，觀察實驗結果是否如同模擬期望的隨機路徑前進，實驗結果於下一節以圖片表示，程式流程如圖 5.7 所示。

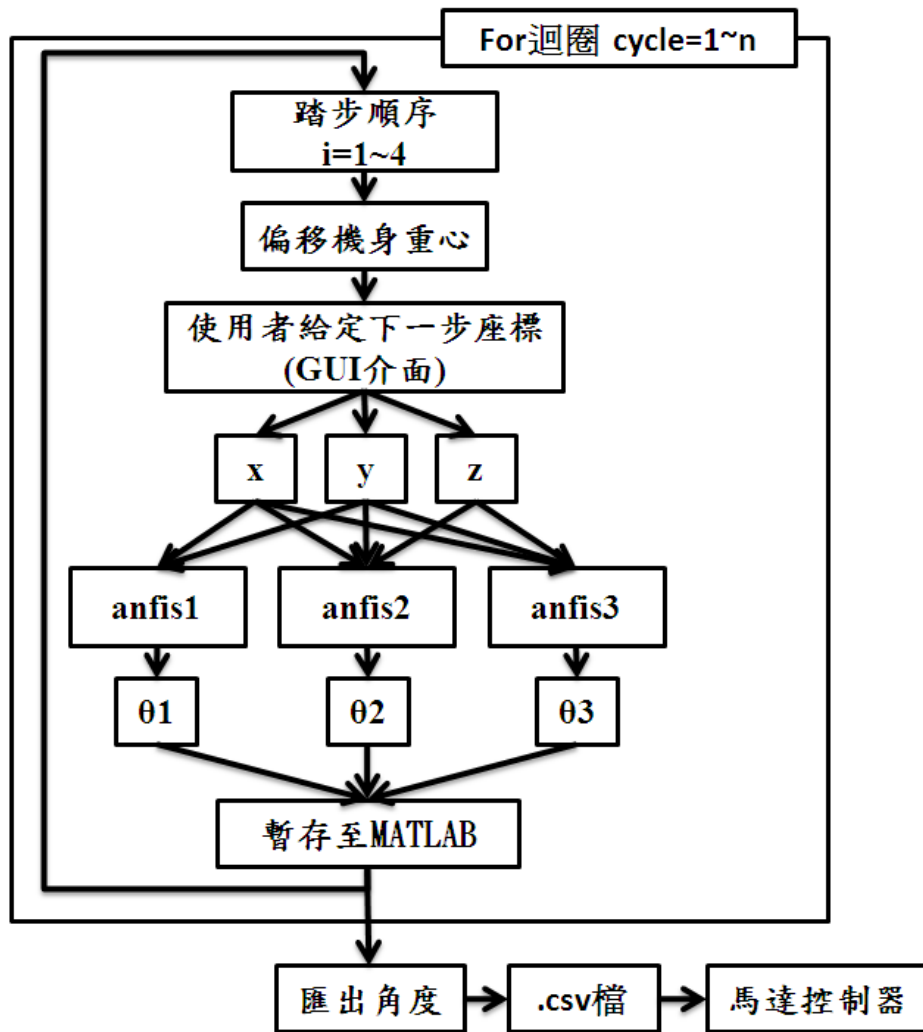


圖 5.7 ANFIS 計算馬達角度之程式流程圖

### 5.3 實驗硬體介紹

實驗硬體主要為 12 個伺服馬達(圖 5.8)、SSC-32 馬達驅動器(圖 5.9)、壓克力板、7.2V 充電電池，硬體架構建立過程如下：

1. 首先利用 3D 繪圖軟體 Solidworks 設計架構尺寸。
2. 利用實驗室雷射切割機，按照 CAD 設計尺寸對壓克力板加工。
3. 將 SSC-32 馬達控制器組裝到本體上，再將 12 個伺服馬達連接到控制器對應的 pin 腳上。

4. 將 7.2V 充電電池放置在 SSC-32 馬達驅動器下方。
5. 四足機器人腳底套上防滑套(圖 5.10)，增加機器人行走時與地面的摩擦力，減少腳底打滑現象和實驗誤差，組裝完成後如圖 5.11 所示。



圖 5.8 伺服馬達

表 5.4 伺服馬達規格

|      |                    |
|------|--------------------|
| 馬達型號 | HiTEC HS-475HB     |
| 角度範圍 | -90°~90°           |
| 電壓   | 4.8 - 6.0v dc      |
| 扭力   | 76 oz.-in          |
| 重量   | 1.52 oz            |
| 速度   | 0.18s / 60 degrees |

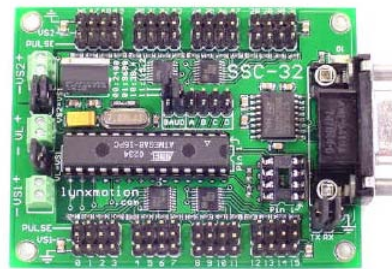


圖 5.9 SSC-32 motor driver



圖 5.10 腳底防滑套

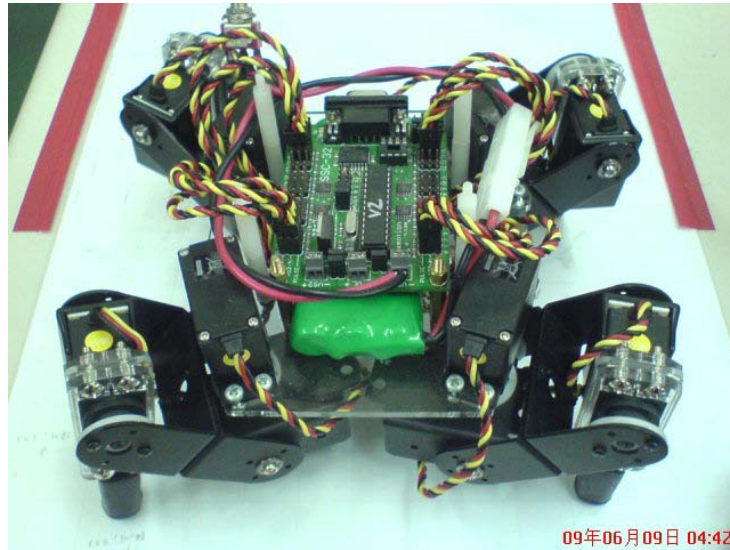


圖 5.11 實驗硬體架構(CIDM 實驗室設計)

#### 5.4 實驗結果

實驗有簡入深分為兩個部份，平坦地形與非平坦地形，每個部分再細分兩個階段，分別為直線和隨機方向前進。

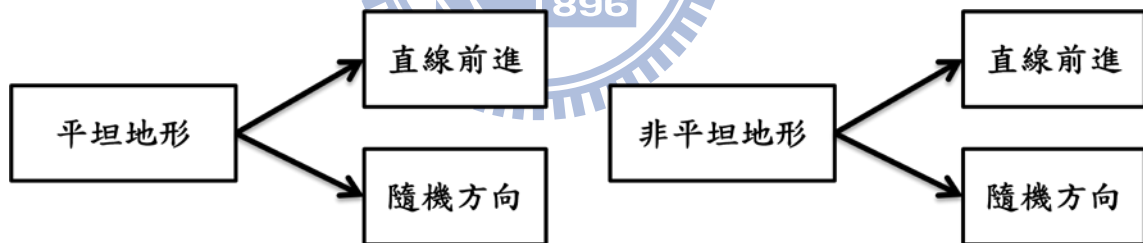


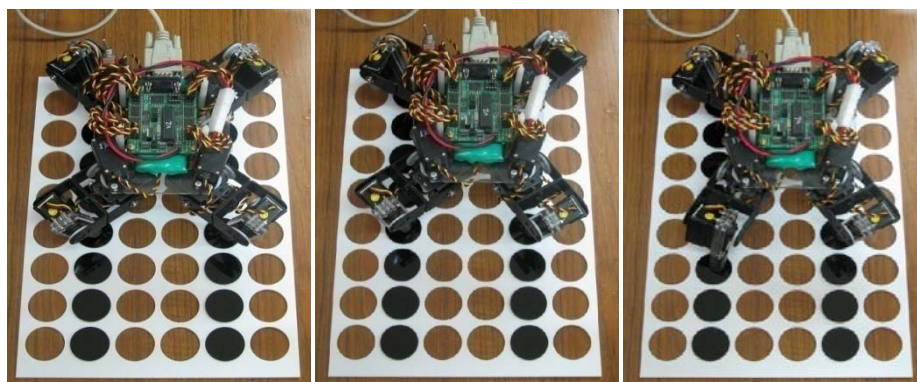
圖 5.12 實驗規劃

##### 1. 平坦地形→直線前進

不論任何地形其踏步過程皆為：偏移重心→踏右前腳→偏移重心→踏左後腳→偏移重心→踏左前腳→偏移重心→踏右後腳；圖 5.13 為平坦地形直線前進的俯視圖，下方為前進方向；其黑色壓克力圓板厚度 4mm 為可拆除式、可任意放入厚度 2mm 白色壓克力洞洞板上，可以很方便地改變



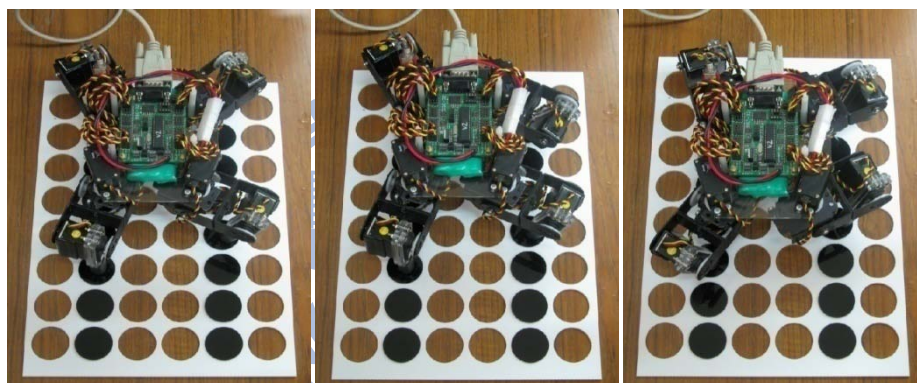
四足機器人前徑路徑，此地形機身相對黑色圓板高度為 9.5 cm、偏移重心距離為 1cm、抬腳高度為 2cm。



(1)預備動作

(2)偏重心

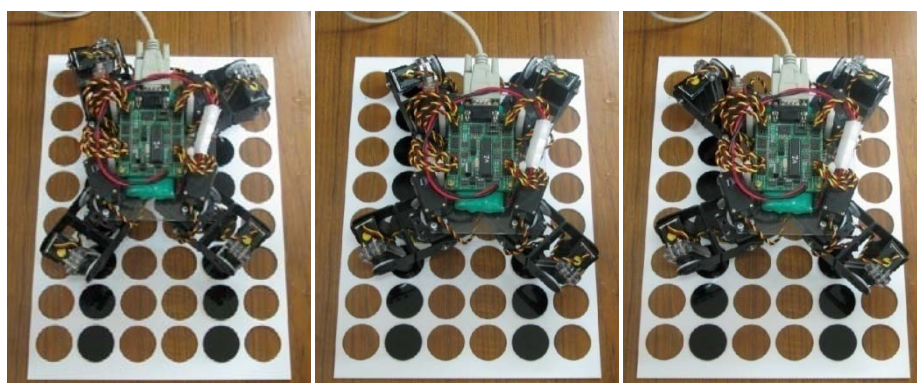
(3)踏右前腳



(4)偏重心

(5)踏左後腳

(6)偏重心



(7)踏左前腳

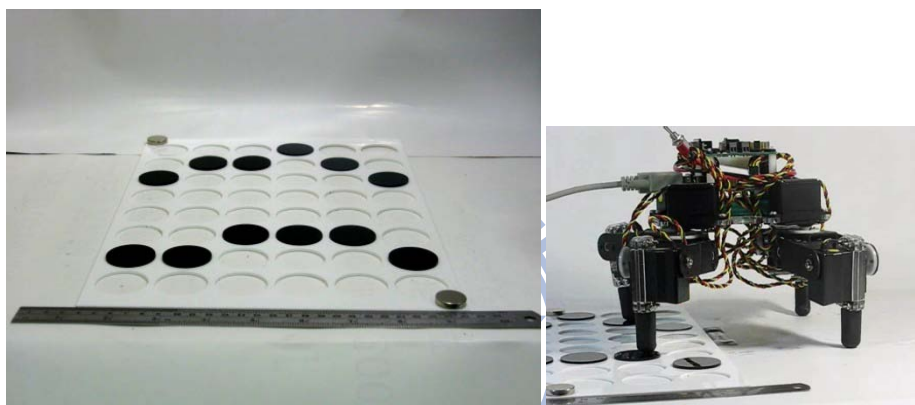
(8)偏重心

(9)右後腳

圖 5.13 平坦地形直線路徑

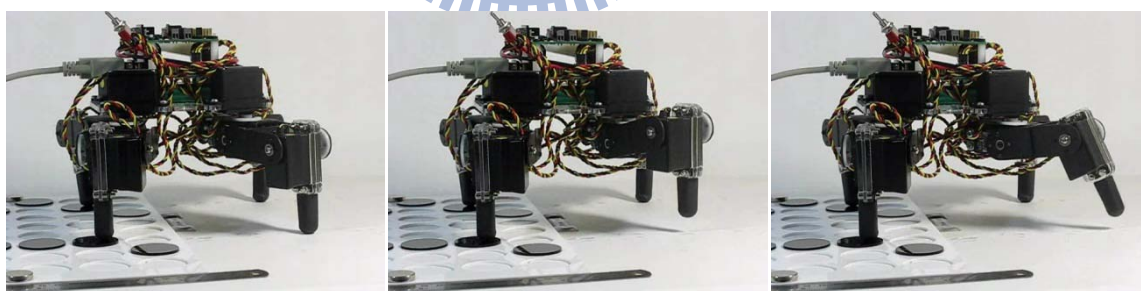
## 2. 平坦隨機地形

完成直線前進後，進一步要跨越隨機地形，此地形面對的不僅是直線還有斜前方向前進，故 ZMP 調整相較之下更為重要，礙於版面關係圖 5.14 僅列出其中一個週期的踏步過程圖片，所謂一個週期就是每隻腳往前一步的過程，此實驗機身相對黑色圓板高度為 9.5cm，偏移重心距離設定為 1cm，抬腳高度為 2cm。



(1)前進路徑側視圖

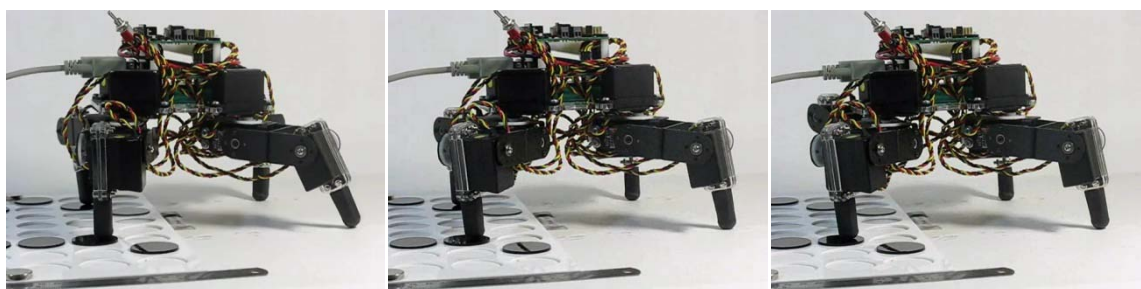
(2)預備動作



(3)偏重心

(4)右前腳抬腳

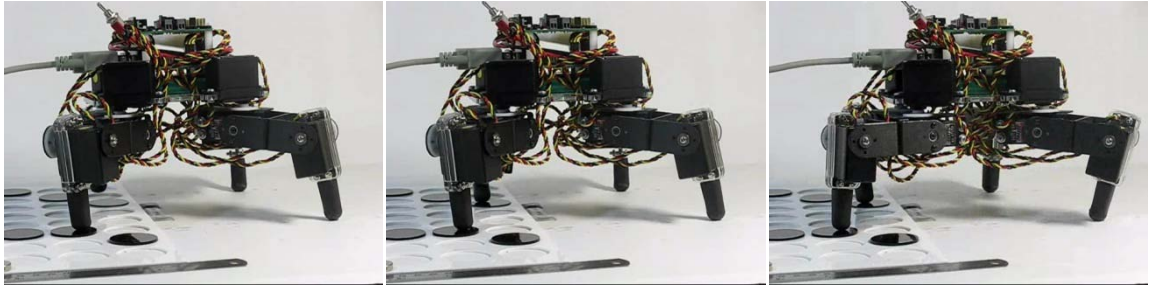
(5)移動至下一踏點上方



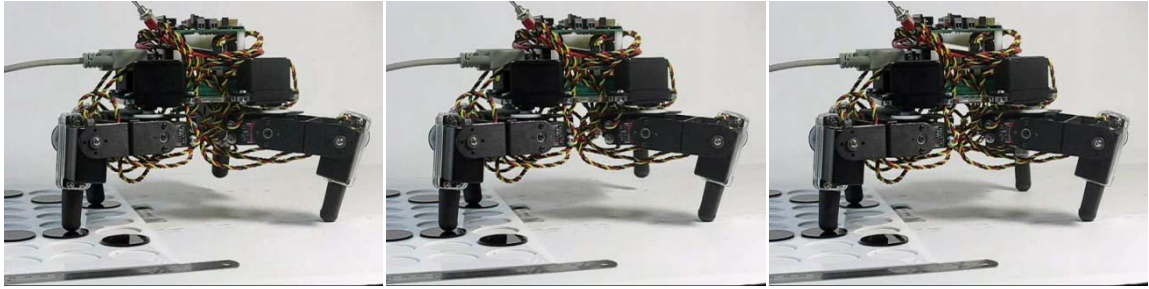
(6)踩踏目標位置

(7)偏重心

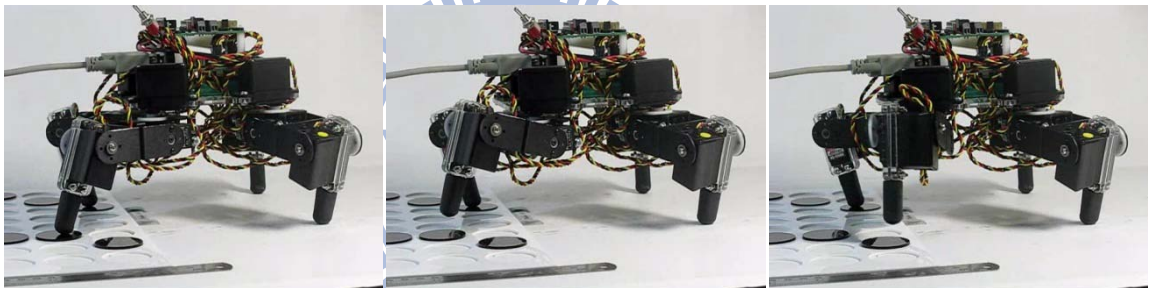
(8)左後腳抬腳



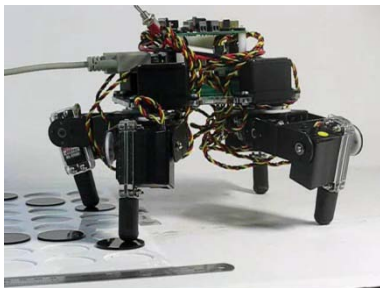
(9)移動至下一踏點上方 (10) 踩踏目標位置 (11) 偏重心



(12)左前腳抬腳 (13) 移動至下一踏點上方 (14) 踩踏目標位置



(15)偏重心 (16)右後腳抬腳 (17) 移動至下一踏點上方

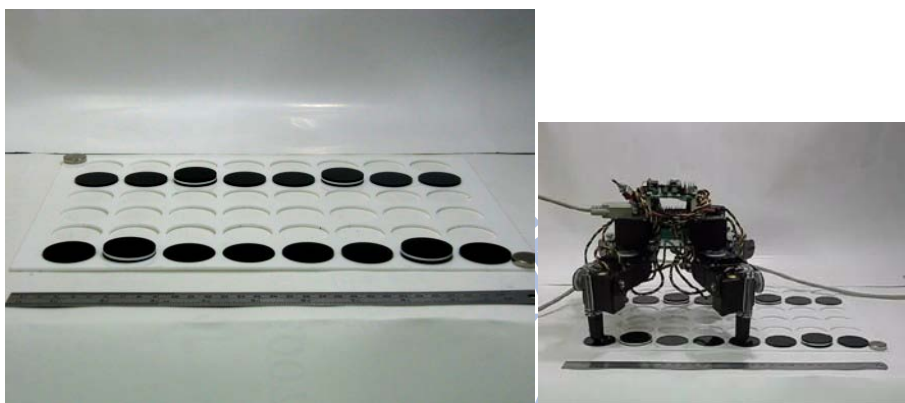


(18)踩踏目標位置

圖 5.14 平坦地形任意路徑

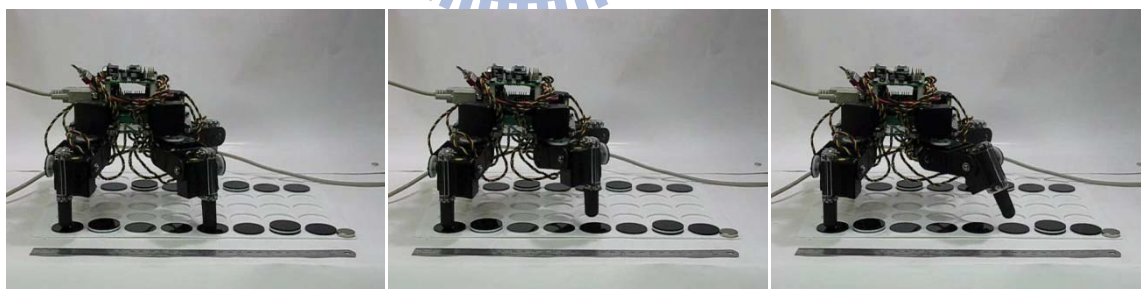
### 3. 非平坦地形→直線前進

此地形將平坦直線路徑中任意四個點位置再墊高 4mm，相對於桌面就是 8mm，為了配合地形落差變大和地形提高，此處將機身從 9.5cm 提升成 10cm，如此一來四足機器人的工作空間才能與地形相配合，礙於版面本文僅列出一個週期的過程(圖 5.15)，此外偏移重心距離仍能設定為 1cm，抬腳高度為 2cm。



(1)前進路徑側視圖

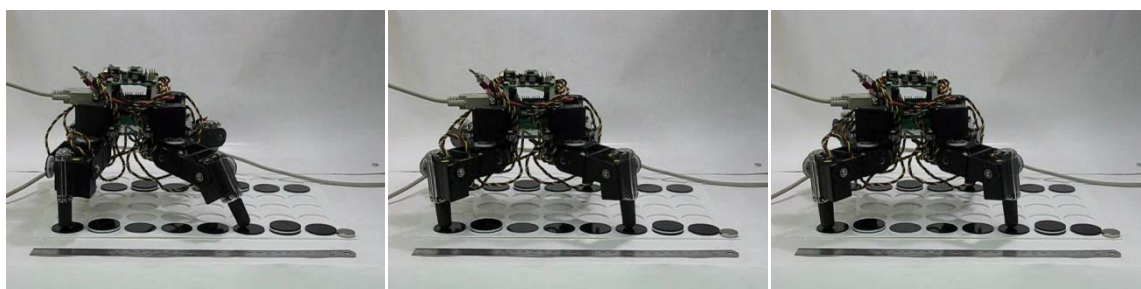
(2)預備動作



(3)偏重心

(4)右前腳抬腳

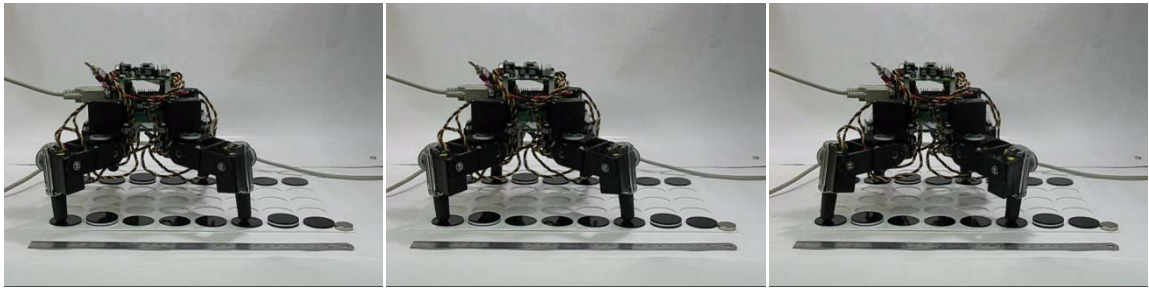
(5)移動至下一踏點上方



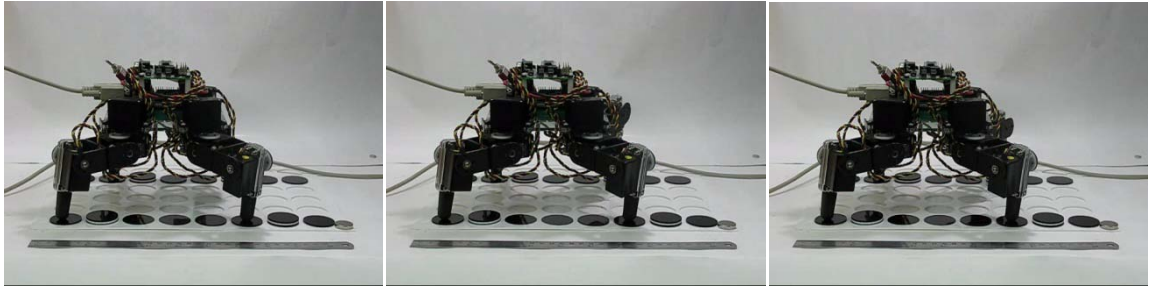
(6)踩踏目標位置

(7)偏重心

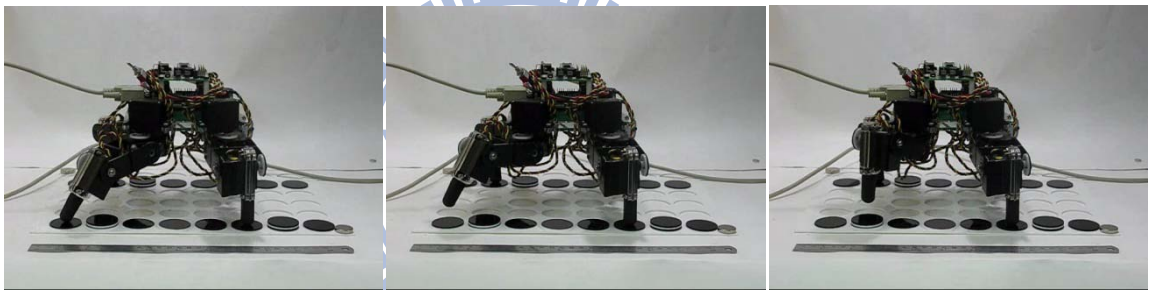
(8)左後腳抬腳



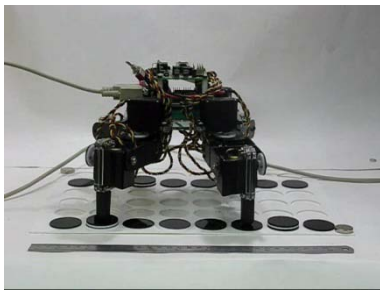
(9)移動至下一踏點上方 (10) 踩踏目標位置 (11) 偏重心



(12)左前腳抬腳 (13) 移動至下一踏點上方 (14) 踩踏目標位置



(15)偏重心 (16)右後腳抬腳 (17) 移動至下一踏點上方

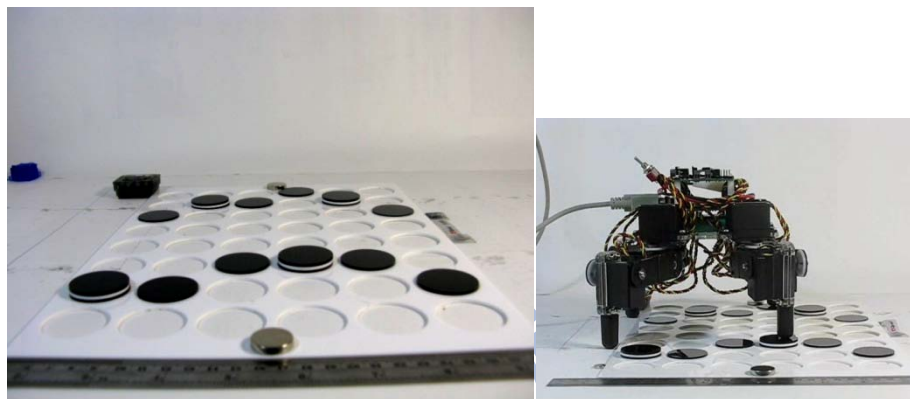


(18)踩踏目標位置

圖 5.15 非平坦地形直線路徑

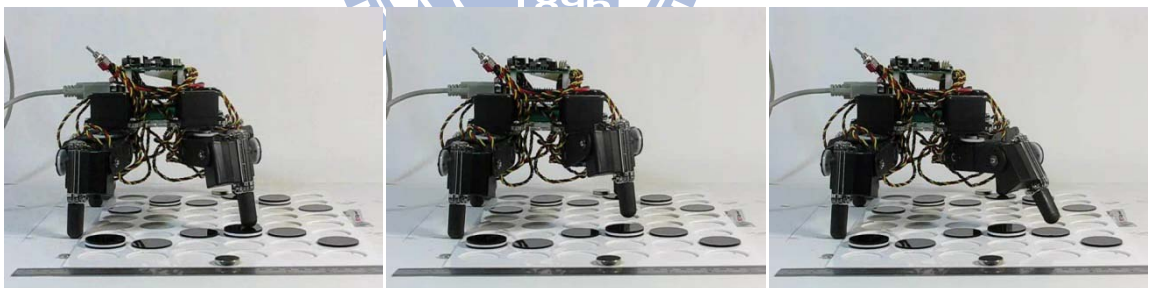
#### 4. 非平坦隨機地形

此為本文的最終地形，隨機方向且不同高度，為了配合此地形機身高度也必須提高為 10cm，偏移重心距離照理說也必須變大，但是偏移量過大會導致外側最大可踏點範圍變小，故此處還是設定為 1cm，抬腳高度為 2cm，圖 5.16 為最後實驗結果照片，礙於論文版面僅列出一個週期過程。



(1)前進路徑側視圖

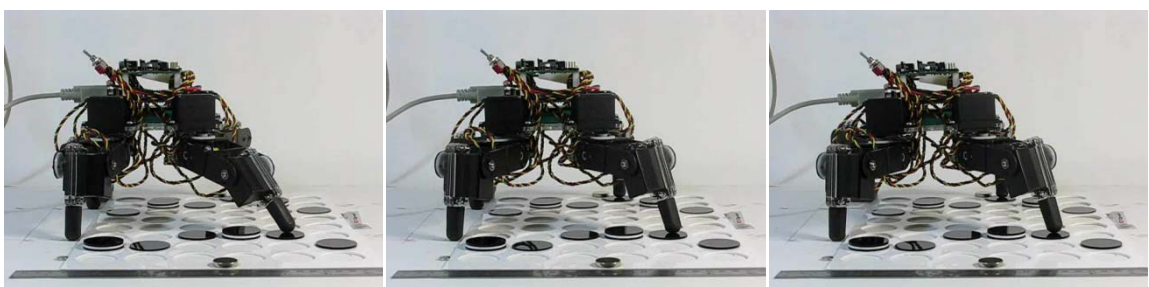
(2)預備動作



(3)偏重心

(4)右前腳抬腳

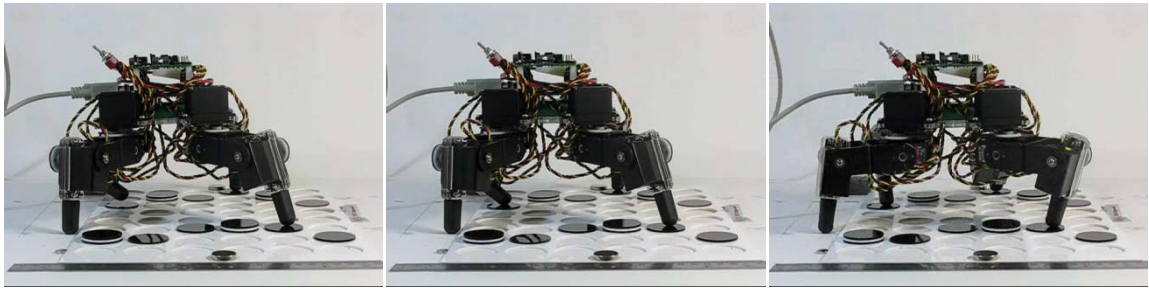
(5)移動至下一踏點上方



(6)踩踏目標位置

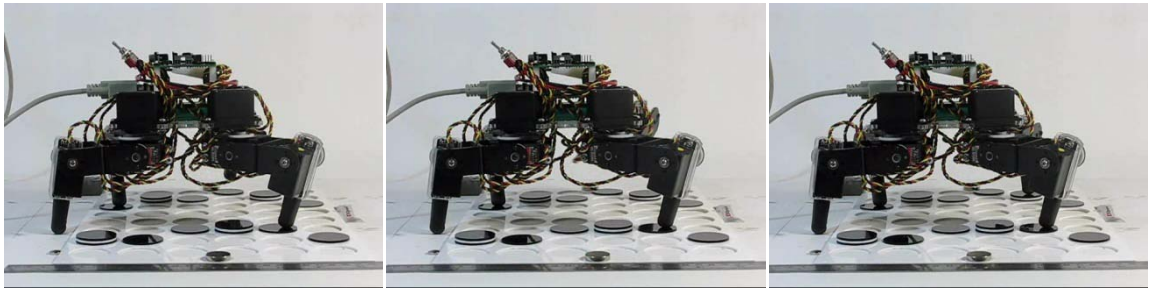
(7)偏重心

(8)左後腳抬腳



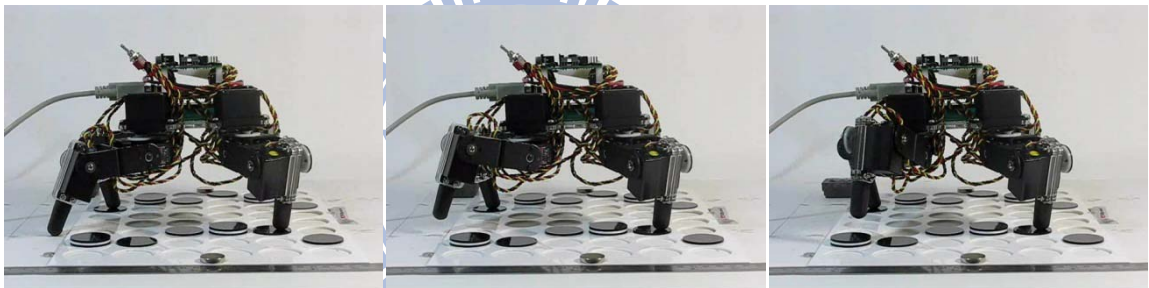
(9)移動至下一踏點上方 (10) 踩踏目標位置

(11) 偏重心



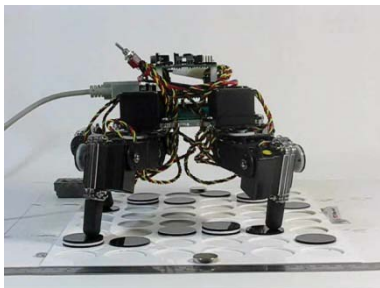
(12)左前腳抬腳

(13) 移動至下一踏點上方 (14) 踩踏目標位置



(15)偏重心

(16)右後腳抬腳 (17) 移動至下一踏點上方



(18)踩踏目標位置

圖 5.16 非平坦地形任意路徑

## 六.結論與未來展望

### 6.1 結論

自然界許多物理現象都是非線性，若用傳統控制方法，則必須知道控制系統轉移函數，可是通常非線性系統轉移函數往往都非常複雜，因此必須做假設去簡化系統，而這些假設和簡化對於實驗結果會造成誤差或不真實，甚至有些非線性系統轉移函數是無法以數學函數表達的，因此傳統控制就受到侷限，本研究結合倒傳遞神經網路和模糊理論去替代控制系統線性或非線性轉移函數，仿效人腦的思考、學習行為。

本研究模擬由淺入深，首先將模糊控制應用至近似平面三連桿「正向」運動學函數，接著是比較倒傳遞類神經網路和 ANFIS 應用至近似空間三連桿「反向」運動學函數，最後則是把 ANFIS 應用至四足機器人；倒傳遞類神經網路主要是依賴「學習樣本」不斷學習使架構收斂，而模糊控制主要是仰賴「經驗」來建構規則庫以推論方式得到想要的輸出，ANFIS 則是結合前兩者優點，由本文模擬結果可看出 ANFIS 相較於前兩者理論是較穩定且輸出誤差較優於前兩者的，故實驗部分最後使用 ANFIS 理論計算出四足機器人跨越非平坦河時地形所有姿態的 12 個馬達角度，搭配靜態 ZMP 穩定性分析，使機器人平穩的跨越非平坦河石地形，具體成果為以下幾點：

1. 藉由 ANFIS 理論近似反向運動學函數，有效縮短 6.4 倍電腦運算時。



2. 本研究四足機器人架構對稱，故本文使用右前腳推算其餘三足步伐，簡化運算架構和減少 ANFIS 學習的時間。
3. 判斷 ANFIS 輸入與輸出彼此相關性，排除獨立的參數，簡化 ANFIS 架構，縮短學習時間。
4. 建立圖形介面，使用者僅要簡單地點選欲前徑路徑，程式便可自動產生相對應的 12 個馬達角度數據。
5. 新一代四足機器人可穩定且精確地走過非平坦河石地形。

此外，值得注意的是類神經網路、模糊控制及 ANFIS 要設定的參數非常多，只能依靠試誤法修改或是仰賴一些經驗公式，故若是系統非常複雜時，就必須不斷修改和調整參數，而且並不是所有學習架構都會收斂，但若輸出誤差收斂至可接受時，便可取代原本複雜的公式，節省電腦運算時間。

## 6.2 未來展望

目前已完成 ANFIS 應用於四足機器人跨越非平坦河石地形的路徑規劃，本實驗室未來仍有許多將持續進行的研究方向，茲簡述說明。

1. 加裝感測器和攝影機使機器人能隨不同環境調整步態，更具自動化。
2. 整合路徑規劃程式並模組化，使系統更具整合效益。
3. 建立隨機步伐的限制條件，以避免較不具實效的腳跟踩踏點選擇。伺服馬達最小力矩的應用，以提高機器人節能效率，增加續航力。
4. 考慮馬達速度控制，使馬達轉動時更平穩、連續。
5. 分析踏點分布使四足機器人踏步順序，可隨著不同路徑而改變。



## 參考文獻

- [1] Y. L. FU, B. JIN, H. LI, S. G. WANG “A robot fuzzy motion planning approach in unknown environments”. *Front. Mech. Eng. China* (2006) 3: 336–340
- [2] K. C. Choi , H. J. Lee and M. C. Lee “Fuzzy Posture Control for Biped Walking Robot Based on Force Sensor for ZMP”. *SICE-ICASE International Joint Conference 2006 Oct. 18-21, 2006 in Bexco, Busan, Korea*
- [3] C. L. Shih, Y. Zhu, and William A. ,Gruver  
“OPTIMIZATION OF THE BIPED ROBOT TRAJECTORY”.  
13-16 Oct. 1991 Page(s):899 - 903 vol.2
- [4] Ken. Onozato and Yutaka. Maeda “Learning of Inverse-dynamics and Inverse-kinematics for Two-link SCARA Robot Using Neural Networks”. *SICE Annual Conference 2007 Sept. 17-20, 2007, Kagawa University, Japan*
- [5] Bojun Ma, Yongchun Fang, and Xuebo Zhang  
“Inverse Kinematics Analysis for a Mobile Manipulator with Redundant DOFs” .*Proceedings of the 26th Chinese Control Conference July 26-31, 2007, Zhangjiajie, Hunan, China*
- [6] Farzad Pourboghrat “Neural networks for learning inverse kinematics of redundant manipulators”. *Volume ii, 8-14 July 1991 Page(s):1004 vol.2*
- [7] P. Vadakkepat, O. C. Miin, X. Peng, and T. H. Lee “Fuzzy Behavior-Based Control of Mobile Robots”. *Fuzzy Systems, IEEE Transactions on Volume 12, Issue 4, Aug. 2004 Page(s):559 – 565*
- [8] Prabu D, Member, IEEE., Surendra Kumar and Rajendra Prasad  
“Dynamic Control of Three-Link SCARA Manipulator using Adaptive Neuro Fuzzy Inference System”. 6-8 April 2008 Page(s):1609 – 1614
- [9] L. X. Wei, H. R. Wang, and Y. Li, “A new solution for inverse kinematics of manipulator based on neural network”. *Proceedings of the Second International Conference on Machine Learning and Cybernetics, Wan, 2-5 November 2003*
- [10] Amar KHOUKHI, Luc BARON, Marek BALAZINSKI “Fuzzy-Neuro Optimal Time-Energy Control of a Three Degrees of Freedom Planar Manipulator”. 3-6 June 2006 Page(s):247 – 252
- [11] 鄭璧瑩，「創意機器人系統設計備忘錄 2009」，交大 CIDM Lab，2009
- [12] 朱明輝、彭增榮，類神經網路控制系統，新文京開發出版，民國九十

七年

- [13] 楊英魁、孫宗瀛、鄭魁香、林建德、蔣旭堂，模糊控制理論與技術，  
全華科技圖書出版，民國九十一年
- [14] 王進德，類神經網路與模糊控制理論入門與應用，全華科技圖書出版，  
民國九十六年
- [15] 張智星，MATLAB 程式設計與應用，清蔚科技出版，民國九十二年
- [16] S.M. Song. , C.D. Zhang 1993 “A study of the stability of Generalized  
Wave Gaits” Mathematical Biosciences 115 1-32
- [17] A. P. Bessonov and N. W. Umnov,1976,”Configuration of Autonomous  
Walkers for Extreme Terrain” ,International Journal of Robotics  
Research,535-559
- [18] McGhee, R. B. and Frank A. A.,1968, “On the stability Properties of  
Quadruped Creeping Gaits” ,Mathematical Biosciences 3,331-351”

