

國立交通大學

電控工程研究所

碩士論文

串結式機器人分散式運動控制之設計與實現



Design and Realization of Distributed Motion Control on
the Chain-type Robot

研究生：陳士文

指導教授：徐保羅 博士

中華民國九十九年七月

串結式機器人分散式運動控制之設計與實現

Design and Realization of Distributed Motion Control on
the Chain-type Robot

研 究 生：陳士文

Student : Shih-Wen Chen

指導教授：徐保羅 教授

Advisor : Dr. Pau-Lo Hsu

國 立 交 通 大 學



Submitted to Institute of Electrical Control Engineering

College of Electrical Engineering

National Chiao-Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical and Control Engineering

July 2010

Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 九 年 七 月

串結式機器人分散式運動控制之設計與實現

研究生：陳士文

指導教授：徐保羅 博士

國立交通大學電控工程研究所

摘要

串結式機器人具備優越的克服複雜地形能力，於救援、監控、搜索等領域都能有所應用。為使串結式機器人可透過單節之間的合作協調，克服複雜地形，本論文首先提出了分散式行為模式多單元控制，藉由 master 與 slave 角色的合作：master 角色的攀爬豎板(riser climbing)與登上踏板(tread landing)行為，以及 slave 角色遵從 master 的命令，以克服複雜地形。

同時引入螞蟻演算法自主學習，使機器人可在運動過程中，自主調整策略，以增進克服地形的效率。經由螞蟻演算法訓練後，可縮減克服地形時間。並提出序列訓練以改善克服時間過長的問題，提出費洛蒙調整以改善學習反效果問題。

藉由在台階、斜坡上的台階，與階梯三種地形上的實驗結果驗證，經過螞蟻演算法的第一次訓練後，可縮減地形克服時間 55%。加入序列訓練後，第一次訓練的地形克服時間標準差，較未加入序列訓練時減少 77%。加入費洛蒙調整後，即使經多次訓練，也能保證地形克服時間趨於一致，證明了學習的可靠性。

Design and Realization of Distributed Motion Control on the Chain-type Robot

Student : Shih-Wen Chen

Advisor : Dr. Pau-Lo Hsu

Institute of Electrical Control Engineering
National Chiao-Tung University

ABSTRACT

The chain-type robot possesses great capability to overcome complex terrains so that it can be applied to areas like rescue, surveillance, and exploration, etc. To achieve coordination among single modules for motion on complex terrains, this thesis proposes the behavior-based distributed multi-unit control for the chain-type robot. As the master unit executes either the riser climbing or the tread landing behavior, the slave unit follows the command sent from the master. Via appropriate coordination between the master and the slave, the chain-type robot can thus successfully overcome complex terrains.

Moreover, to improve the chain-type robot motion for complex environments, the autonomous learning procedure via the ant colony optimization (ACO) is used. In addition, the sequential training is proposed to decrease the conquering time, and the algorithm of pheromone adjustment is also proposed to suppress the deviated results of the learning.

Experimental results on all steps, slope and stairs show that ACO decreases 55% of the terrain conquering time after the first learning procedure; and with the proposed sequential training, the deviation of conquering time for the first learning procedure is 77% less than that without sequential training. With pheromone adjustment, the satisfactory results in deviation of conquering time are guaranteed even after several learning procedures to prove that the present learning is reliable.

目 錄

摘要.....	i
ABSTRACT.....	ii
目錄.....	iii
表目錄.....	vi
圖目錄.....	vii
第一章 緒論	1
1.1 研究動機與目的	1
1.2 研究背景與發展現況	2
1.3 問題描述	3
1.4 研究方法	4
1.5 論文架構	5
第二章 機器人硬體與系統架構	6
2.1 串結式機器人機構.....	6
2.2 機器人平台硬體.....	8
2.2.1 系統架構.....	8
2.2.2 eZdsp™ F2812 DSK 之功能介紹.....	10
2.2.3 SCI 串列傳輸	12
2.3 感測資訊處理.....	13
2.3.1 以馬達扭矩資訊偵測機器人是否困住	13

2.3.2 以加速度計偵測傾角.....	14
2.4 機器人無線通訊.....	16
2.4.1 ZigBee 硬體.....	16
2.4.2 ZigBee 網路的建立.....	17
2.5 機器人硬體與軟體對應關係.....	20
第三章 分散式行為模式多單元控制.....	21
3.1 分散式行為模式多單元控制.....	21
3.2 以行為模式控制克服地形障礙.....	22
3.2.1 Riser Climbing 行為.....	23
3.2.2 Tread Landing 行為.....	25
第四章 螞蟻演算法自主學習.....	27
4.1 螞蟻演算法簡介.....	27
4.2 以螞蟻演算法進行 Master 角色的行為選擇.....	28
4.3 以序列訓練改善隨機選擇行為的缺點.....	30
4.4 以費洛蒙調整加強歷史經驗基礎.....	32
第五章 實驗結果與討論.....	35
5.1 兩節機器人台階實驗結果.....	35
5.1.1 原始螞蟻演算法實驗結果.....	37
5.1.2 加入序列訓練實驗結果.....	41

5.1.3 加入費洛蒙調整實驗結果.....	43
5.1.4 台階實驗結果小結.....	45
5.2 兩節機器人於斜坡上的台階實驗結果	46
5.2.1 原始螞蟻演算法.....	47
5.2.2 加入序列訓練.....	48
5.2.3 斜坡上的台階實驗結果小結.....	49
5.3 兩節機器人階梯實驗結果	50
5.3.1 原始螞蟻演算法.....	50
5.3.2 加入序列訓練.....	51
5.3.3 階梯實驗結果小結.....	53
5.4 三節機器人克服複雜地形	53
第六章 結論與未來展望	55
6.1 結論.....	55
6.2 未來展望.....	56
參考文獻	57

表 目 錄

表 4-1 預先定義好的行為序列	31
表 4-2 費洛蒙調整示例	34
表 5-1 原始螞蟻演算法克服時間資料表(單位:秒)	38
表 5-2 原始螞蟻演算法台階克服時間的統計資料, 單位: 秒	39
表 5-3 加入序列訓練後的台階克服時間	41
表 5-4 台階克服時間平均值(標準差) 單位: 秒	42
表 5-5 加入費洛蒙調整後, 台階克服時間資料 (單位: 秒)	43
表 5-6 台階克服時間中位數(標準差) 單位: 秒	44
表 5-7 斜坡上的台階克服時間(單位: 秒)	47
表 5-8 加入序列訓練後, 斜坡上的台階地形克服時間(單位: 秒)	48
表 5-9 斜坡上的台階克服時間平均值(標準差) 單位: 秒	49
表 5-10 階梯克服時間資料(單位: 秒)	50
表 5-11 加入序列訓練後, 階梯克服時間資料	51
表 5-12 階梯克服時間平均值(標準差) 單位: 秒	52



圖 目 錄

圖 2-1 單節機器人機構(北科大葉賜旭教授實驗室設計)	6
圖 2-2 串結式機器人多節機構	7
圖 2-3 串結式機器人實體圖	7
圖 2-4 硬體系統架構	8
圖 2-5 全雙工轉半雙工電路實體模組	9
圖 2-6 全雙工轉半雙工電路	9
圖 2-7 SN74LVTH245ADW 實體模組	9
圖 2-8 eZdsp 2812	10
圖 2-9 MAX3232 週邊電路	12
圖 2-10 MAX3232 電路實體模組	12
圖 2-11 馬達扭矩量測實驗環境	13
圖 2-12 馬達實體圖	13
圖 2-13 馬達扭矩量測實驗結果	13
圖 2-14 機器人困住程度的歸屬函數	14
圖 2-15 內建於 CC2430DB 模組的二軸加速度計 ADXL321	14
圖 2-16 以加速度計偵測傾斜角示意圖	15
圖 2-17 加速度與傾斜角關係圖	15
圖 2-18 Chipcon CC2430	16
圖 2-19 CC2430ZDK	16
圖 2-20 SmartRF04EB 板	17
圖 2-21 CC2430DB 板	17
圖 2-22 機器人與監控端的 ZigBee 網路架構	19
圖 2-23 機器人硬體與軟體對應關係	20
圖 3-1 串結式機器人硬體機構的多單元特性	21
圖 3-2 串結式多單元分散式控制架構	21
圖 3-3 個別單元的基於行為架構	22
圖 3-4 機器人困住於豎板(riser)前	23
圖 3-5 master 往上抬以攀爬豎板	23
圖 3-6 master 的 riser climbing 行為狀態圖	23
圖 3-7 由模糊推論推得所需馬達轉速命令	24
圖 3-8 master 困住於踏板(tread)上	25
圖 3-9 master 往下壓以登上踏板	25
圖 3-10 master 的 tread landing 行為狀態圖	25
圖 3-11 此時需執行 tread landing 行為	26

圖 3-12 此時需執行 riser climbing 行為	26
圖 4-1 螞蟻演算法示意圖	27
圖 4-2 傾斜角分區圖	28
圖 4-3 以螞蟻演算法進行行為協調示意圖	29
圖 4-4 克服台階過程	30
圖 4-5 克服台階過程	30
圖 4-6 克服台階過程	30
圖 4-7 克服台階過程	30
圖 4-8 螞蟻演算法行為協調加入序列訓練後的流程圖	31
圖 4-9 更新費洛蒙流程圖	32
圖 4-10 經由學習後，費洛蒙難以達到如初始狀態的高濃度	33
圖 4-11 螞蟻演算法結合費洛蒙調整流程圖	34
圖 5-1 台階地形實景	35
圖 5-2 螞蟻演算法學習實驗流程	36
圖 5-3 克服台階過程	37
圖 5-4 克服台階過程	37
圖 5-5 克服台階過程	37
圖 5-6 克服台階過程	37
圖 5-7 原始螞蟻算法台階克服時間散佈圖	39
圖 5-8 Tread landing 的費洛蒙衰減	40
圖 5-9 有無序列訓練台階克服時間比較	42
圖 5-10 有無費洛蒙調整時，台階克服時間比較	44
圖 5-11 斜坡上的台階尺寸	46
圖 5-12 斜坡上的台階地形實景	46
圖 5-13 有無加入序列訓練，斜坡上的台階克服時間比較	48
圖 5-14 階梯地形尺寸	50
圖 5-15 階梯地形實景	50
圖 5-16 加入序列訓練，階梯克服時間散佈圖	52
圖 5-17 困住，嘗試往上抬，依然困住	54
圖 5-18 改為往下壓，得以克服	54
圖 5-19 先前的往下壓姿態，導致困住	54
圖 5-20 頭部拉平，以克服困境	54
圖 5-21 成功通過	54

第一章 緒論

1.1 研究動機與目的

所謂的串結式機器人(chain-type robot)，係由多個單節履帶機器人串結而成。相較於單節機器人，串結式機器人可以藉由單節間的推、拉，以及抬舉關節、旋轉關節等較單節機器人多的自由度，克服更複雜的地形，因此可用於救援、搜索、監控等領域。

本論文的研究目的為設計串結式機器人的控制策略，以加速度計資訊、馬達回傳資料等感測資訊，偵測機器人的姿態，及其與環境的互動狀況，讓機器人可以由這些輸入資訊，自主決定相對應的動作，以克服台階、斜坡上的台階、階梯以及崎嶇地形。

同時，由於在複雜地形上，距離感測器或接觸感測器容易在機器人與環境的碰撞時損壞，因此設計控制策略的另一重點即為，如何在缺乏距離感測器或接觸感測器的情況下，讓機器人仍得以順利克服地形。最後，將於實際搭建的台階、斜坡上的台階、階梯以及崎嶇地形中，展示與驗證本論文所提出策略的可行性。

1.2 研究背景與發展現況

串結式機器人(chain-type robot)屬仿蛇機器人(snake-inspired robot[1])的其中一類。自然界中的蛇由於具有良好的攀爬或跨越障礙物能力、穩定的姿態(由於跟地面具有多點接觸)、多自由度(即使其中一個關節失去效用，依然可以運動)等優點，因此啟發了很多模仿蛇的機器人研究。仿蛇機器人依其機構設計，主要可分為三大類：(1)具有被動輪的機器人[2-4]、(2)具有主動輪的機器人[5-7]、(3)具有主動式履帶的機器人[8-12]。

其中(3)具有主動式履帶的機器人，由於履帶可以提供給機器人相對平坦的接觸面，同時履帶又具有主動驅動力，因此可適應極端崎嶇複雜的地形。基於此優點，本論文所使用的串結式機器人，即採用此種機構設計。

雖然串結式機器人具備先天機構上的優越克服地形能力，但多節機構使得機器人與環境的互動複雜，造成了控制的困難性。針對此控制困難性，過去的研究者大致提出以下方法。

Wang *et al.*[13]建構了分散式的串結機器人機構，各個單節的履帶機器人可以獨立運作，也可以經由電腦操控，組合而成三節的機器人。這篇論文利用紅外線測距儀、GPS、陀螺儀、影像處理模組等感測器，讓機器人可以偵測環境的狀態以及自身的姿態。這篇論文僅簡單描述了克服台階地形、從傾倒狀態回復的運動步驟，但對於如何根據感測資訊，進行進一步的姿態調整，甚至於策略的適應性，都沒有深入討論。

GAO *et al.*[14]利用力感測器的回授，機器人得以自主調整姿態，執行跨越高突的障礙物、攀爬階梯與鑽過牆上的洞等任務。Hutchison *et al.*[10]利用基因演算法與神經網路，賦予串結式機器人自主學習能力，能夠克服踏板上有石頭的階梯地形。這兩篇論文皆外加大量感測器(力感測器、距離感測器等)，但實務上，在複雜環境中，外加感測器容易因碰撞而損壞，不禁令人思考，是否有方法可以節省感測器的使用，同時又能讓串結式機器人順利克服複雜地形？

陳俊傑[15]利用擾動觀測器，僅需過馬達的速度與電流，就能估測外界地形施於機器人的干擾扭矩，設計 admittance 控制器，完全不需外加任何感測器，即能調整關節馬達轉速，克服步階、斜坡，與複雜地形。這給了本論文利用馬達資訊偵測機器人與環境互動關係的靈感，但與[15]不同的是，本論文又賦予了串結式機器人自主學習的能力，機器人透過在運動過程中的試誤學習，得以更穩健、更有效率地的克服地形。

1.3 問題描述

1、如何利用單節之間的合作協調克服地形

當遭遇到複雜地形時，往往需要將機器人從單節擴充為多節，以增進克服地形能力。此時，機器人該如何透過單節之間的合作協調，成功地克服複雜地形？

2、如何透過學習，增進機器人克服地形的效率

面對複雜地形，機器人與外界環境的互動模型往往複雜難解，設計者預先藉由經驗法則設計好的策略，不見得是最能適應環境的策略，此時機器人該如何在運動過程中，自主調整控制策略，增進克服地形的效率？

1.4 研究方法

1、如何利用單節之間的合作協調克服地形

針對此問題，本論文提出了分散式行為模式多單元控制(distributed multi-unit behavior-based control)，藉由 master 角色與 slave 角色的合作：master 執行攀爬豎板(riser climbing)與登上踏板(tread landing)行為，slave 遵從 master 的命令，達到運動的協調合作，以克服複雜地形。同時模組化的架構，使得未來可輕易地增添新的決策角色或軟體模組，具有良好的擴充性。

2、如何透過學習，增進機器人克服地形的效率

本論文引入螞蟻演算法(ant colony optimization, ACO[20])，進行強化式學習(reinforcement learning[16])。所謂的強化式學習，係透過嘗試動作，並經由外界環境對於該動作的反饋，判斷這個動作是否恰當，不斷修正策略，更穩健地、更有效率地克服地形。而螞蟻演算法則將候選動作的機率分布以費洛蒙(pheromone)濃度表示，執行完動作後，利用環境的反饋更新費洛蒙，亦即更新候選動作的機率分布，經由反覆迭代運算後，可漸漸逼近最能適應環境的機率分布。螞蟻演算法在資料開採(data mining)、網路連線建立(networking)、計算機網路的工作排程、感測器的自我組織(self-organization of sensors)，甚至於機器人的避障學習等，都能發揮良好的效果，但未曾應用於串結式機器人的適應地形控制。因此本論文將螞蟻演算法，於串結式機器人進行嶄新的應用。機器人藉由偵測馬達扭矩判斷是否困住，以此作為學習的反饋訊號，透過試誤學習，漸漸習得最能適應環境的策略。並提出序列訓練(sequential training)，以預先定義好的行為序列，取代隨機行為選擇，改善實務上克服時間過長的問題；提出費洛蒙調整(pheromone adjustment)，適時提高費洛蒙濃度，改善學習反效果(經過多次學習後，克服時間反而增長)問題。

1.5 論文架構

本論文共分為五章，第一章旨在簡述研究動機、目的、背景與發展，並陳述所要研究的問題、方法。第二章介紹機器人平台，並說明整體系統架構，包含使用的控制核心、周邊電路與感測資訊。第三章描述串結式多單元分散式控制，包含各單元的運動控制、單元間的溝通與協調。第四章描述螞蟻演算法學習，及提出改善學習的方法。第五章為實驗結果，讓兩節機器人在台階、斜坡上的台階，及階梯地形上，驗證比較原始螞蟻演算法，與加入自行提出的改善學習方法，對於克服地形效率的影響，最後將機器人擴充成三節，驗證分散式架構的擴充性，並克服階梯與崎嶇地形所組成的複雜地形。第六章對本論文做一個結論，並提出未來可以再發展之方向。



第二章 機器人硬體與系統架構

2.1 串結式機器人機構

本論文採用北科大機電整合所葉賜旭教授實驗室所設計的串結式機器人機構。透過串結的機構設計概念，將原本已保有順應地形變化的單節機器人以串結的方式連接，即串結式機器人，使得串結式機器人的順應地形能力有效地得到提升，更能克服多種複雜的地面環境。串結式機器人由許多單節機器人所組成。如圖 2-1 所示，單節機器人共由兩組履帶傳動模組、一組旋轉機構模組及一組抬舉機構模組所組合而成。

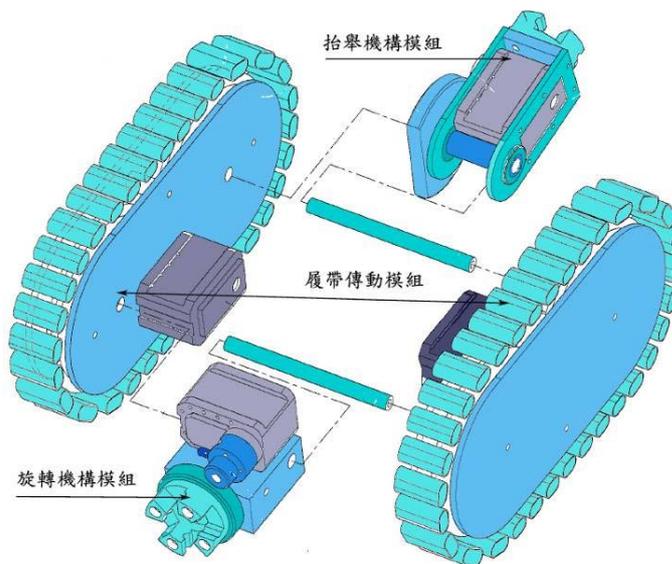


圖 2-1 單節機器人機構(北科大葉賜旭教授實驗室設計)

配合串結式結構的設計，將單節與單節之間連結設計成兩個自由度的關節，即為滾動動作(Roll)和傾角動作(Pitch)，藉由關節調整串結式救援機器人的姿態，提升順應地形的能力，對於具有較大段差及高低起伏的地面將可有效地克服，並且利用關節兩個自由度的控制，更可達到串結式救援機器人順應地形的特性。

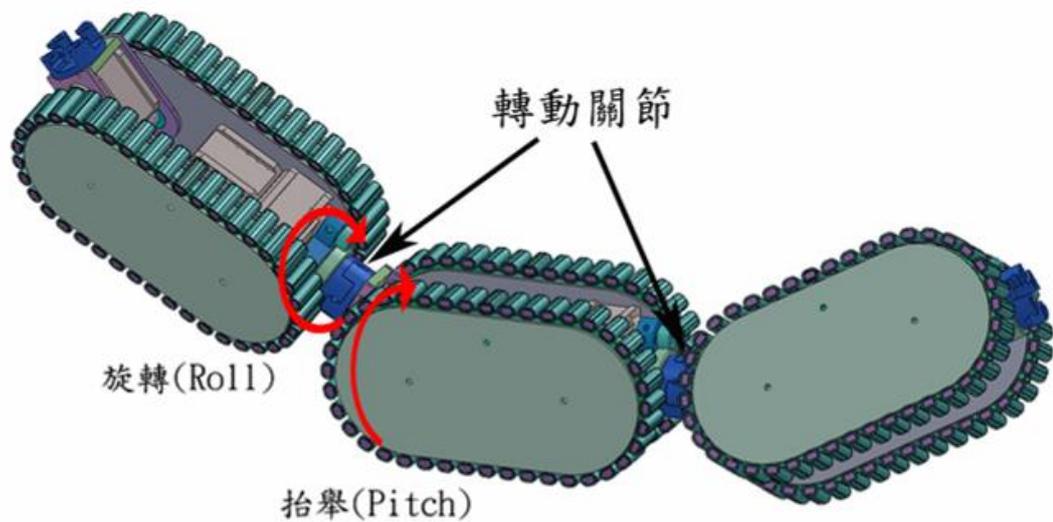


圖 2-2 串結式機器人多節機構



圖 2-3 串結式機器人實體圖

2.2 機器人平台硬體

2.2.1 系統架構

本論文採用德州儀器(Texas Instrument, TI)公司所研發的 TMS320F2812 DSP(簡稱 F2812)作為控制核心，透過串列傳輸介面 SCI-B 控制伺服機、SCI-A 與 ZigBee CC2430DB 無線傳輸模組(用以感測傾斜角及與監控端溝通)溝通。硬體系統架構如下 圖 2-4 所示。

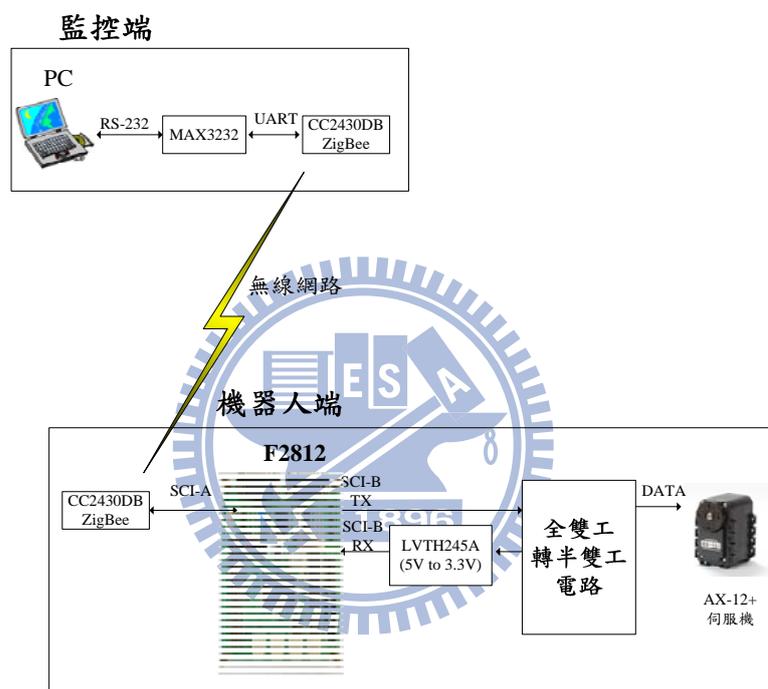


圖 2-4 硬體系統架構

機器人的控制馬達，係使用 ROBOTIS 公司出產的 AX-12 伺服機。其主要特色為半雙工 UART 數位封包通訊，與位置、速度、負載等回授資訊。由於 AX-12 伺服機的通訊協定為半雙工通訊，因此 F2812 的 SCI 需經由全雙工轉半雙工電路才能與之溝通。

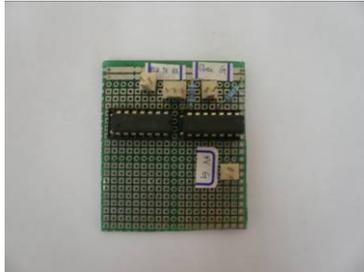


圖 2-5 全雙工轉半雙工電路實體模組

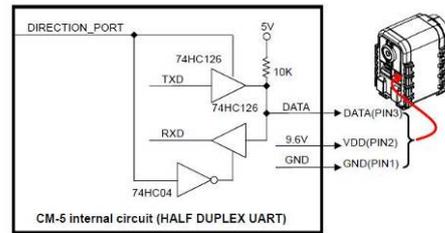


圖 2-6 全雙工轉半雙工電路

同時由於 AX-12 回傳資訊的電壓準位為 5V，需透過 SN74LVTH245ADW(具三態輸出的 3.3V ABT 八路匯流排收發器)，將電壓準位轉為 3.3V，方能將資訊回傳給 F2812。



圖 2-7 SN74LVTH245ADW 實體模組

2.2.2 eZdsp™ F2812 DSK 之功能介紹

本論文採用德州儀器(Texas Instrument, TI)公司所研發的 TMS320 F2812 DSP(簡稱 F2812)[17]作為控制核心，如圖 2-8 所示。



圖 2-8 eZdsp 2812

德州儀器公司所生產的DSP中，‘C2000系列的硬體發展平台是專為控制應用最佳化而設計的，其中F2812晶片是此系列中目前最高等級的控制器，其特點在於擁有150 MHz(6.67 ns cycle time)的快速處理能力，比TI早期出產的DSP ‘C240快7倍。在核心部分，算數邏輯單元(Arithmetic Logic Unit, ALU)、累積器(Accumulator, ACC)均採用32位元定點運算，為了提升數位訊號運算效能，以硬體方式實現乘法器、乘積位移器，可在一個指令週期(instruction cycle)內完成乘加運算。在整數計算方面，為減少數值計算所衍生如溢位(overflow)等問題，也採用硬體式的輸出倍率位移器來提高軟體執行的精確度。

記憶體容量方面，F2812(on chip)主要擁有128K*16 Flash EEPROM、兩組4K*16 Single-Access RAM(SARAM)、一組8K*16 SARAM，並採用哈佛匯流排(Harvard bus)架構。較舊型的DSP大部分將程式、資料、I/O記憶體獨自分離(定址位址重複)，但F2812卻走向單獨的記憶體空間，裡頭包含上述三種記憶體，使用更為有彈性，也可減少不同記憶體之間搬動資料的指令集。

Spectrum Digital 公司則利用 F2812 晶片結合自行開發之周邊模組，發展出 eZdsp™ F2812 初學板(DSP Started Kit, DSK)，其週邊主要有 16 個通道的 12 位元類比數位轉換器(Analog-to-digital Converter, ADC)，串列傳輸支援了四種常用的型式：SPI (Serial Peripheral Interface)、SCIs (Serial Communications Interface)、eCAN (Enhanced Controller Area Network)、McBSP (Multi-channel Buffered Serial Port)，最高可支援 56 個 GPIO(General-Purpose I/O)。



2.2.3 SCI 串列傳輸

串列傳輸標準是美國電子工業協會在1969年所頒佈之RS-232-C，同時也是目前應用最廣的串列傳輸標準。RS232串列傳輸主要是利用RXD和TXD來做資料的接收和傳送線，由於RS232與TTL所判定的高電位和低電位的電壓值不同，所以必須做位準轉換的介面，才可以讓資料由RS-232的電壓準位轉換為TTL的電壓準位。

PC 端的 RS-232 訊號首先透過 MAX3232 來做位準轉換，再與監控端的 CC2430DB ZigBee 無線傳輸模組進行溝通。周邊電路如圖 2-9 所示，而實體模組如圖 2-10 所示，透過此模組將可以將 PC 端 RS-232 串列傳輸的 9V 轉換成 TTL 電壓準位。

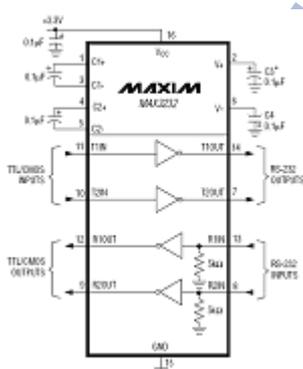


圖 2-9 MAX3232 週邊電路

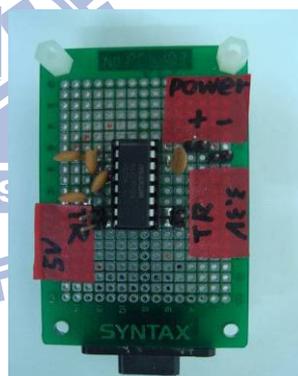


圖 2-10 MAX3232 電路實體模組

2.3 感測資訊處理

2.3.1 以馬達扭矩資訊偵測機器人是否困住

馬達扭矩回傳資訊的功用在於，偵測機器人在行走過程中，是否困住。因此，本論文做了以下的實驗。在履帶馬達轉速為 50rpm 時，分別在機器人困住(在此，困住定義為：機器因履帶打滑而致使前進速度為 0 的情況)與沒有困住的情況下，各做 5 次實驗，紀錄圖 2-11 中左上方馬達扭矩回傳資訊，每次實驗取樣 150 點，取樣週期 0.2 秒。實驗結果如下圖 2-13。

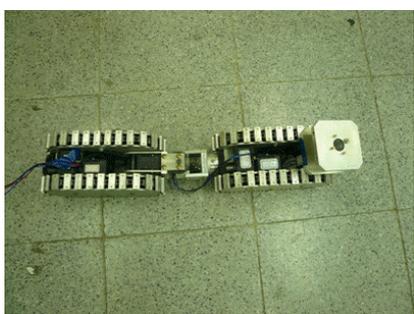


圖 2-11 馬達扭矩量測實驗環境



圖 2-12 馬達實體圖

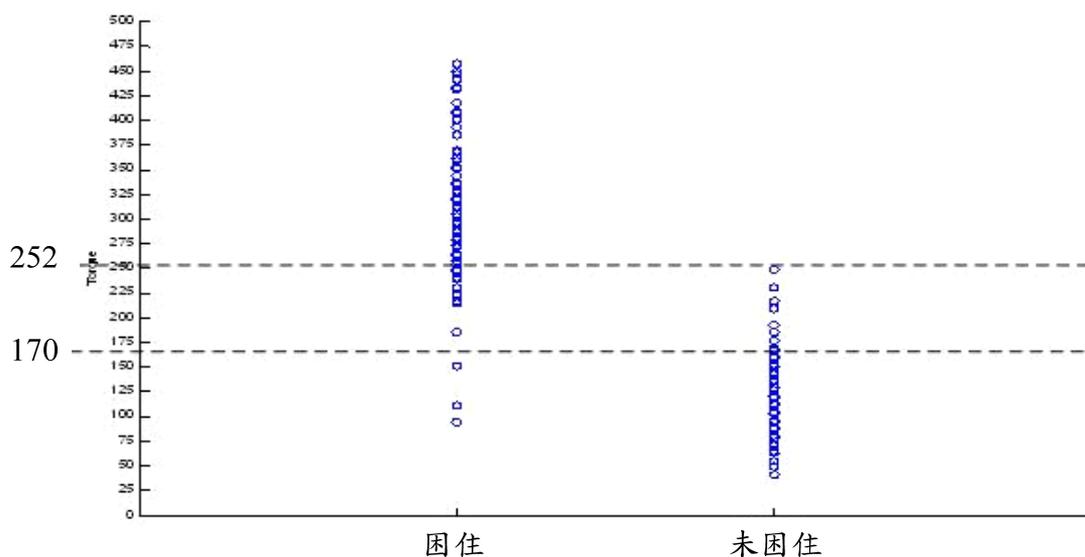


圖 2-13 馬達扭矩量測實驗結果

由上圖 2-13 可知，在馬達扭矩資訊介於 T_L :170 與 T_H :252 之間時，會造成判斷的模糊地帶。因此，我們用如下圖 2-14 的歸屬函數表示困住的程度。此歸屬函數將用於行為的設計(詳見 3.2 節)與費洛蒙更新(詳見 4.2 節)。在實際環境中，隨著地面的材質與坡度的不同， T_L 與 T_H 需加以微調。

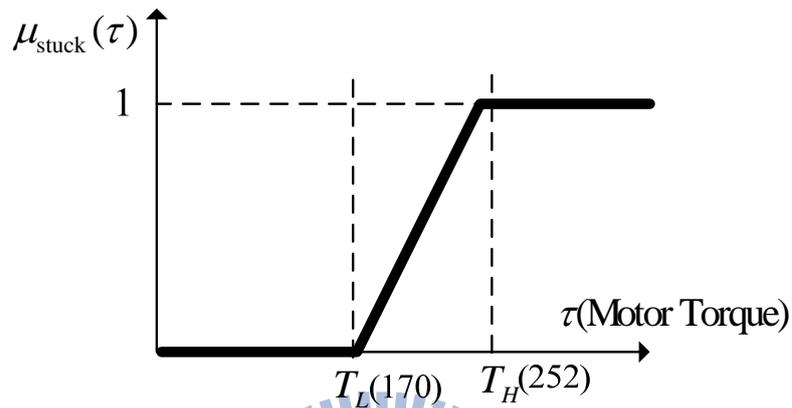


圖 2-14 機器人困住程度的歸屬函數

2.3.2 以加速度計偵測傾角

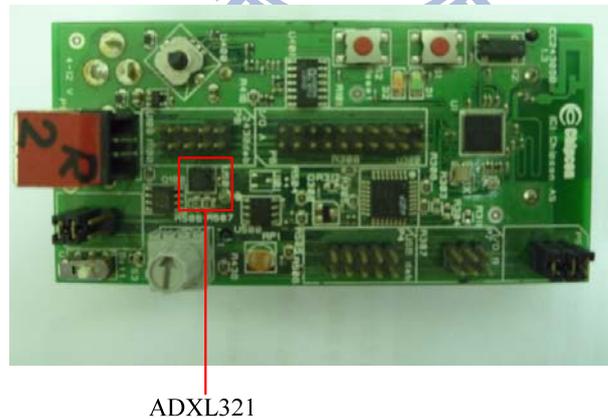


圖 2-15 內建於 CC2430DB 模組的二軸加速度計 ADXL321

本論文利用內建於 CC2430DB 模組的二軸加速度計 ADXL321 偵測「頭部」單節的傾斜角。ADXL321 可偵測 $\pm 18g$ 範圍內的加速度，支援類比輸出，輸出電壓的敏感度為 57mv/g ，並可透過 CC2430 的 A-D 轉換通道，擷取加速度資訊。

如下圖 2-16 所示，傾斜角可由重力加速度 g 在 x 軸上的加速度分量 a_x 推知。

$$\theta = \sin^{-1} \frac{a_x}{g}$$

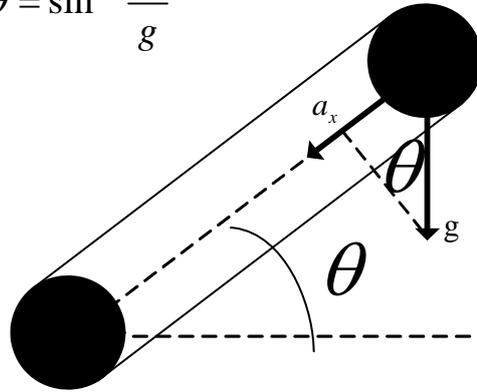


圖 2-16 以加速度計偵測傾斜角示意圖

在傾斜角為 -90° 、 -80° 、...、 90° 的情況下，各做五次的加速度量測。結果如下圖 2-17 所示。橫軸為原始加速度計資訊經由 14 位元 A-D 轉換後的數位資訊，縱軸為傾斜角度。

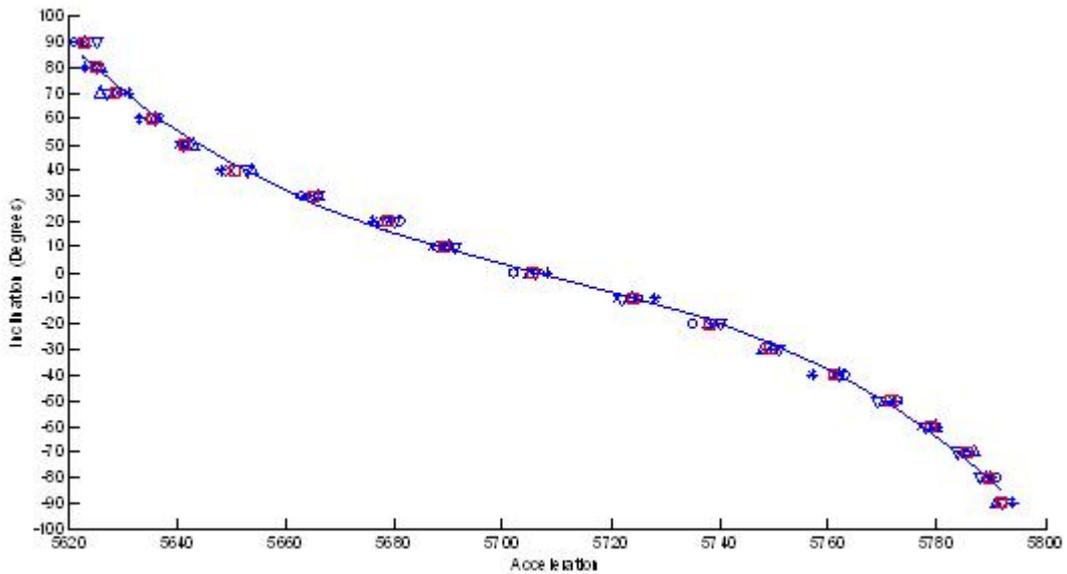


圖 2-17 加速度與傾斜角關係圖

由上圖 2-17，我們可利用各傾斜角度上所測得的平均加速度資訊，得到近似曲線方程式如下式 2-1。x 為由經由 A-D 轉換後的加速度計感測資訊，經由此式，可將感測資料轉化為傾斜角資訊 y (degrees)。

$$y = 0.0066x^3 + 1.13x^2 - 6451.96x + 1.23 \times 10^7 \quad (2-1)$$

2.4 機器人無線通訊

串結式機器人與 PC 監控端，透過 ZigBee 無線網路進行溝通。本節將介紹 ZigBee 硬體與 ZigBee 網路的建立。

2.4.1 ZigBee 硬體

以 RF 晶片為代表性，現今 2.4G Hz 的 RF 晶片以國外的 Chipcon 市場佔有率較高，其 RF 晶片 CC2420 搭配 Atmel AVR 8 bits 微處理器的平台，是大多數人接觸到 ZigBee 的第一個開發平台，而且 CC2420 在市面上已經銷售了百萬顆了。而最近也開始推出一顆 CC2430 型號的系統晶片，它主要是將 CC2420 和 8051 結合為一，有 128KB、64KB 和 32KB 三種版本的快閃記憶體空間，以及 8KB 的記憶體，可以供客戶依據需求的選擇，大大降低了 ZigBee 相關產品的成本。圖 2-18 為 Chipcon CC2430。



圖 2-18 Chipcon CC2430



圖 2-19 CC2430ZDK

本論文所使用的 ZigBee 硬體是採用美國德州儀器公司(Texas Instrument, TI)的 CC2430ZDK 模組,如圖 2-19 所示,硬體上共有 SmartRF04 和 CC2430DB 兩種,如圖 2-20、圖 2-21 所示,主要核心為 CC2430 System-on-Chip 裡面包括 CC2420 發送器、8051 控制器、FLASH 記憶體以及相關的周邊電路。

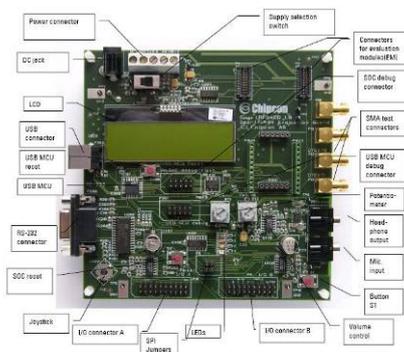


圖 2-20 SmartRF04EB 板

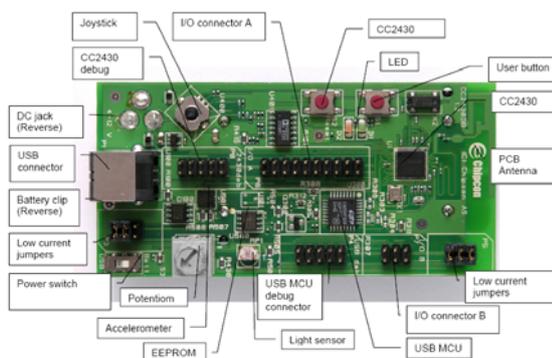


圖 2-21 CC2430DB 板

2.4.2 ZigBee 網路的建立

● ZigBee 網路位址分配演算法

當網路形成時, ZigBee 的協調者(Coordinator)必須要先定義 ZigBee 路由器最多可容許連線之裝置個數(C_m), 以及最多的子 ZigBee 路由器數量(R_m)以及網路的深度(L_m), ZigBee 規定 $C_m \geq R_m$, 因此一 ZigBee 路由器至少可供 $(C_m - R_m)$ ZigBee 終端設備連結上它。裝置的網路位址是由其父節點(Parent router)所給定的, 對於 ZigBee 協調者, 整個網路的位址空間被劃分成 $R_m + 1$ 塊, 前 R_m 塊位址空間將分配給其 R_m 個子路由器, 而最後一部份則保留給與之連線之 $(C_m - R_m)$ 個 ZigBee 終端設備。ZigBee 路由器利用 C_m 、 R_m 、 L_m 來計算一個稱為 C_{skip} 的參數, 然後再利用 C_{skip} 來計算其子路由器以及終端設備的網路位址, 假定一路由器位於網路的第 d 層, C_{skip} 的數值由下式可得:

$$C_{skip}(d) = \begin{cases} 1 + Cm \cdot (Lm - d - 1), & \text{if } Rm=1 \\ \frac{1 + Cm - Rm - Cm \cdot Rm^{Lm-d-1}}{1 - Rm}, & \text{Otherwise} \end{cases} \quad (2-2)$$

位址的分配是由 ZigBee 協調者開始，會先將自己的位址指定為 0，以及深度指定為 0，假設一個在深度 d 的父節點的位址以被指定為 A_{parent} ，該父節點將指定他的第 n 個子路由器的位址為 $A_{parent} + (n-1) \times C_{skip}(d) + 1$ ，並且指定他的第 n 個子終端設備的位址為 $A_{parent} + Rm \times C_{skip}(d) + n$ 。

● 無線區域網路建構

在網路的形成順序方面，首先透過 coordinator 建立一個 PAN(Personal Area Network)，然後 router 再加入此網路中，同時經由 bind 的程序來完成 coordinator 和 router 之間的連結。串結式機器人與監控端的網路架構如下圖 2-22 所示。Coordinator 與監控端 PC 連接，而 router 則是置於機器人的第一節，與至於第三節電路盒內的 F2812 控制核心連接，router 接收到 coordinator 傳來的監控端命令後，進一步透過 F2812 的 SCI-A 將命令轉送予 F2812。

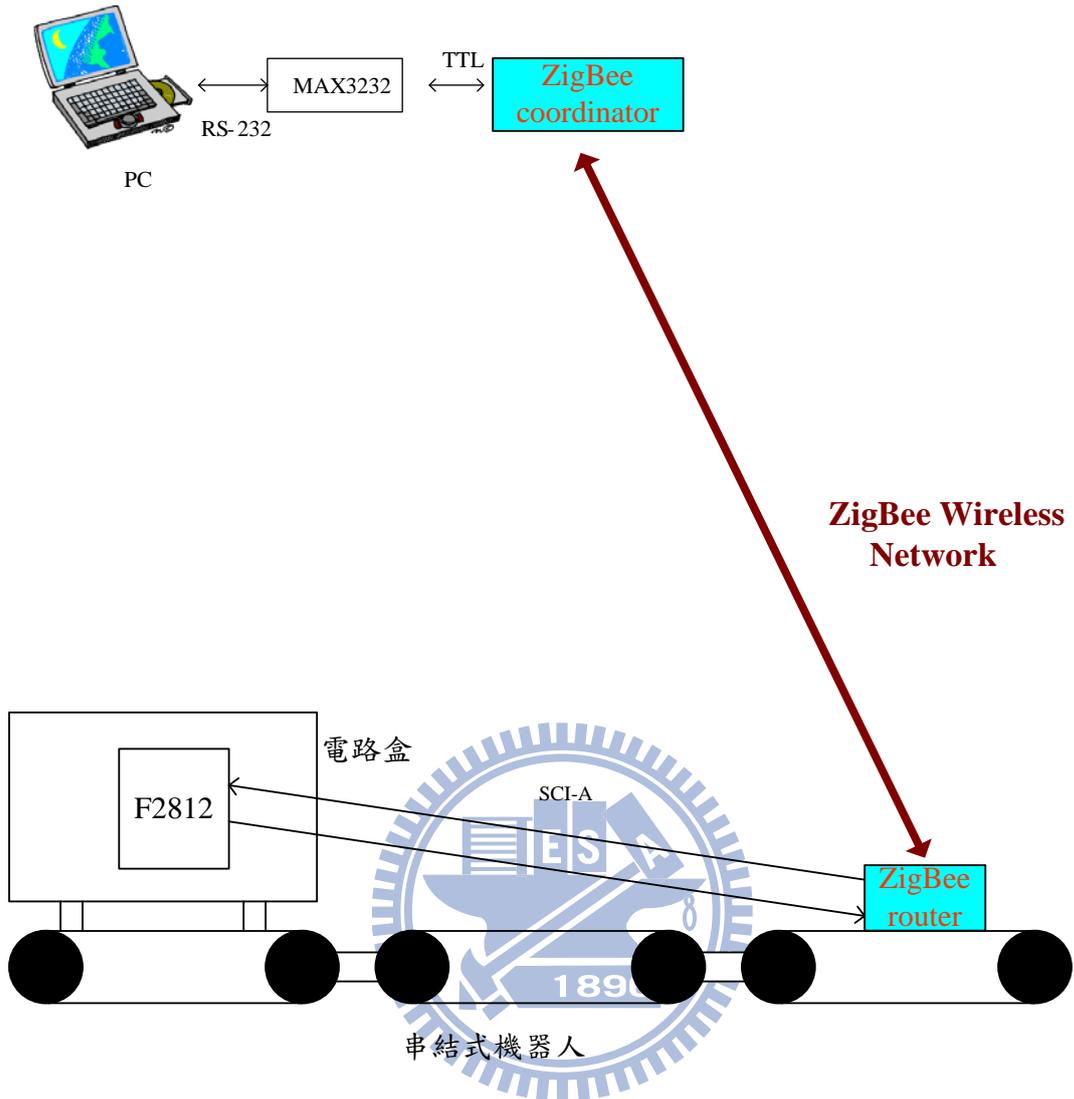


圖 2-22 機器人與監控端的 ZigBee 網路架構

2.5 機器人硬體與軟體對應關係

機器人的軟硬體對應關係則如下圖 2-23 所示。機器人的軟體模組包含分散式行為模式多單元控制(詳見第三章)、加速度計感測資訊處理(詳見 2.3.2 節)與馬達扭矩感測資訊處理(詳見 2.3.1 節)，軟體模組皆實現於 F2812 控制核心。週邊硬體主要包含 ZigBee router 與伺服馬達。ZigBee router 負責傳輸監控端命令(啟動或停止運動控制)，與加速度計感測資訊。伺服馬達則負責驅動機器人的運動，並回傳扭矩感測資訊。

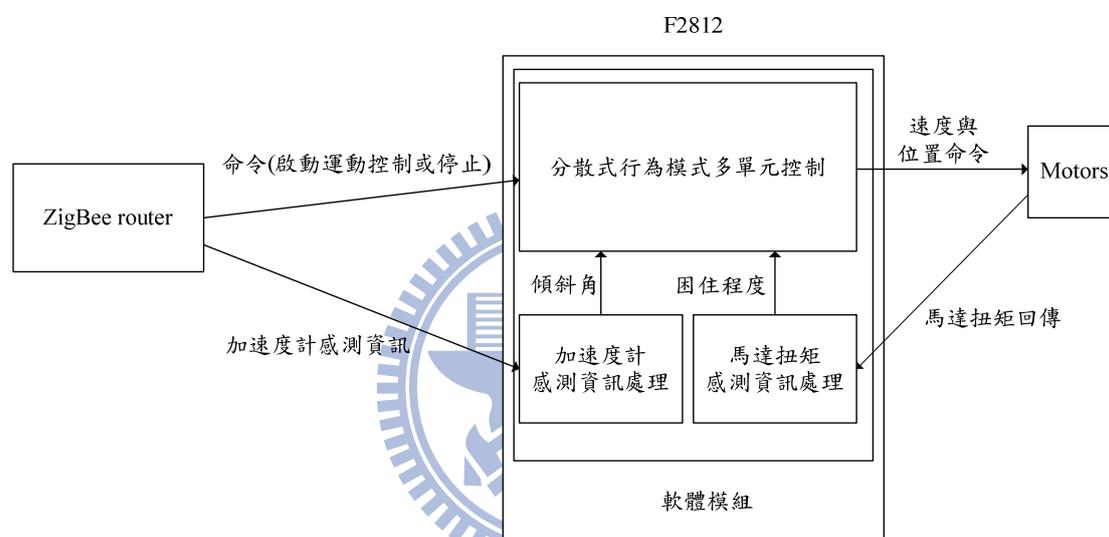


圖 2-23 機器人硬體與軟體對應關係

第三章 分散式行為模式多單元控制

3.1 分散式行為模式多單元控制

如下圖 3-1 所示，串結式機器人在硬體機構上，即具有多單元(multi-unit)的特性。

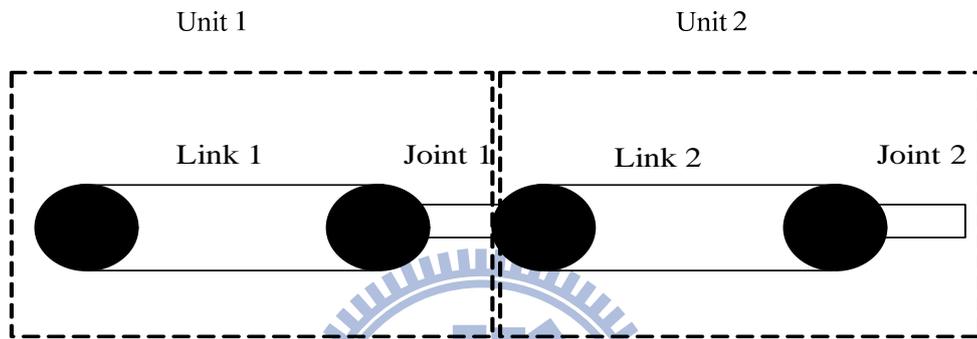


圖 3-1 串結式機器人硬體機構的多單元特性

在此，將 link 1 和 joint 1 劃分為 unit 1，link 2 和 joint 2 劃分為 unit 2，以此類推。這種多單元的特性，可以推展到軟體演算法中，成為分散式策略。軟體演算法的分散式架構如下圖 3-2 所示。在此架構中，個別單元獨立決策，單元間再透過訊息(message)的溝通達成協調。

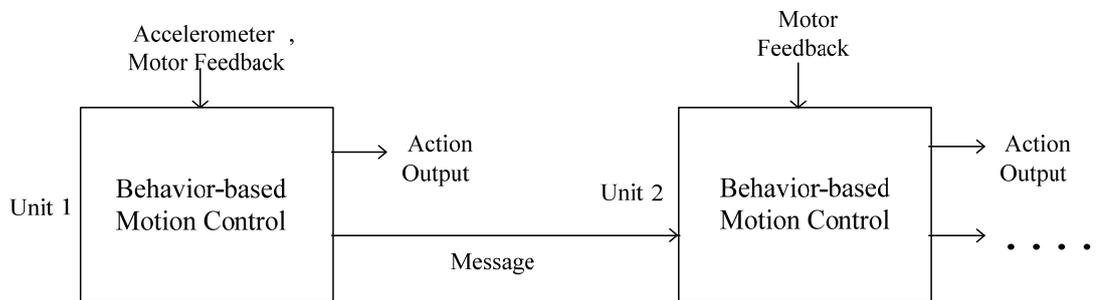


圖 3-2 串結式多單元分散式控制架構

多單元(multi-unit)的演算法架構有以下兩種優點：

- (1) 可擴充性(extensibility)：當硬體機構多了一個 unit 或少了一個 unit 時，在軟體上僅需增加或刪減處理該 unit 的模組即可，無需進行繁瑣的修改。
- (2) 簡易性(simplicity)：單一 unit 的演算法模組僅需考慮自身 unit 的決策即可，各個 unit 間再透過 message 的傳送、接收，以達成協調。

個別單元決策介紹如下：

3.2 以行為模式控制克服地形障礙

個別單元藉由基於行為架構進行運動控制。基於行為決策可在系統模型未知的情況之下，奠基於馬達扭矩資訊、三軸加速度計等感測資訊，以專家經驗所得的控制規則，控制機構的運動，以克服地形。本研究所設計的基於行為[18]架構如下圖 3-3 所示。基於行為架構的特色為，將機器人整體的動作，依照動機，劃分為幾個較簡單的行為，再經由行為的協調，決定最終的動作輸出。由於各個行為都是一個決策模組，因此將來需要增加功能時，可以輕易地增添行為，具良好的擴充性。

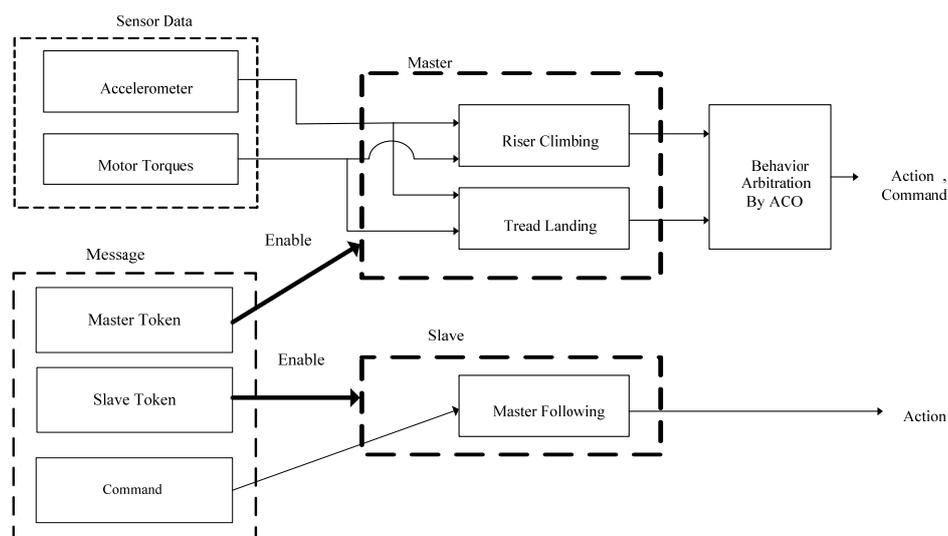


圖 3-3 個別單元的基於行為架構

根據單元的角色為 master 或 slave，將致能 master 角色的行為(riser climbing 與 tread landing)，或 slave 角色的行為(master following)。

各行為分別介紹如下：

3.2.1 Riser Climbing 行為

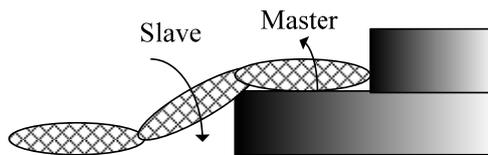


圖 3-4 機器人困住於豎板(riser)前

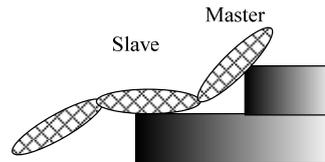


圖 3-5 master 往上抬以攀爬豎板

如上圖 3-4、圖 3-5 所示，當機器人藉由馬達扭矩資訊偵測到困住時，master 單元將往上抬昇，同時 master 命令 slave 往下壓，以輔助 master 的運動。slave 接收到 master 的命令後，將啟動 master following 行為，執行 master 的命令。狀態圖如下圖 3-6。

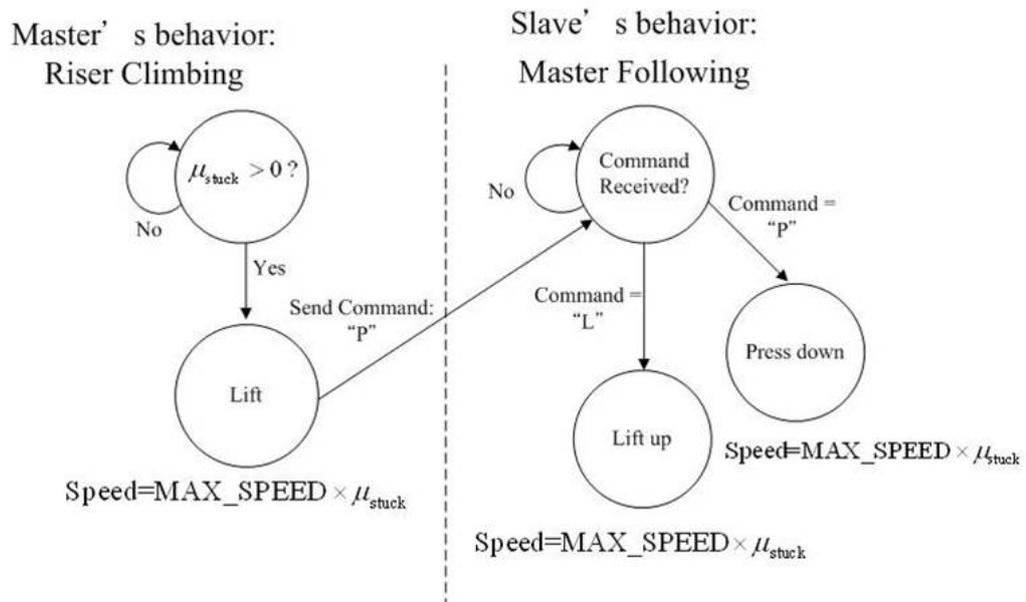
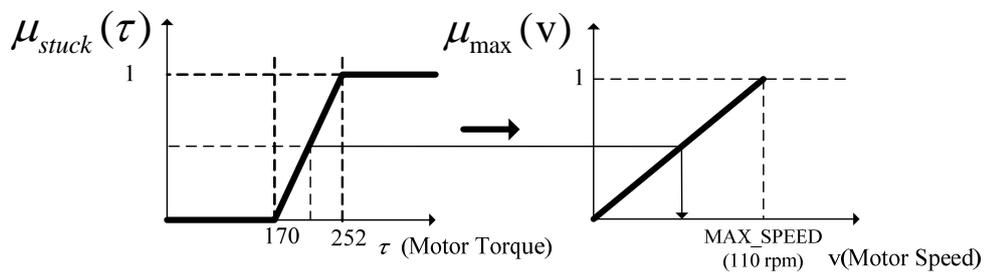


圖 3-6 master 的 riser climbing 行為狀態圖

而上抬的關節轉速 $Speed = MAX_SPEED \times \mu_{stuck}$ (MAX_SPEED 為馬達最大轉速)，則是由模糊推論(fuzzy reasoning)的第二種型態，Tsukamoto 所提出的方法 [18]，推演而來。使用此方法的好處為，可將關節馬達的轉速命令，以一簡單的式子表示，無需進行繁複的解模糊化(defuzzification[18])運算。如下圖 3-7 所示， τ 為偵測到的履帶馬達扭矩， v 為關節馬達轉速命令，利用模糊規則「IF τ is *stuck* , THEN v is *max* 」推論而得所需的轉速命令。

IF τ is *stuck*, THEN v is *max*



Fuzzy Reasoning of the Second Type: Tsukamoto's Method

圖 3-7 由模糊推論推得所需馬達轉速命令

3.2.2 Tread Landing 行為

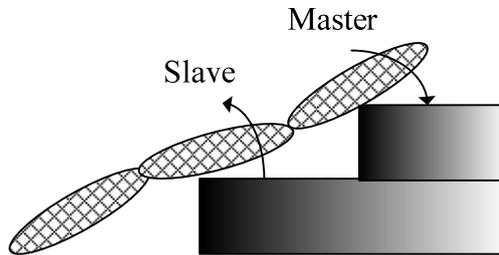


圖 3-8 master 困住於踏板(tread)上

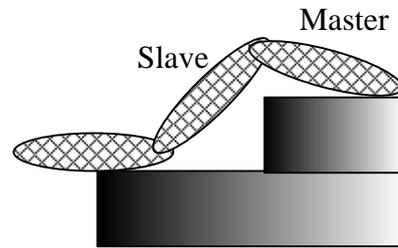


圖 3-9 master 往下壓以登上踏板

如上圖 3-8、圖 3-9 所示，當 master 困住在踏板上時，master 單元將自身往下壓，同時命令 slave 往上抬，以輔助 master 的運動。狀態圖如下圖 3-10 所示。

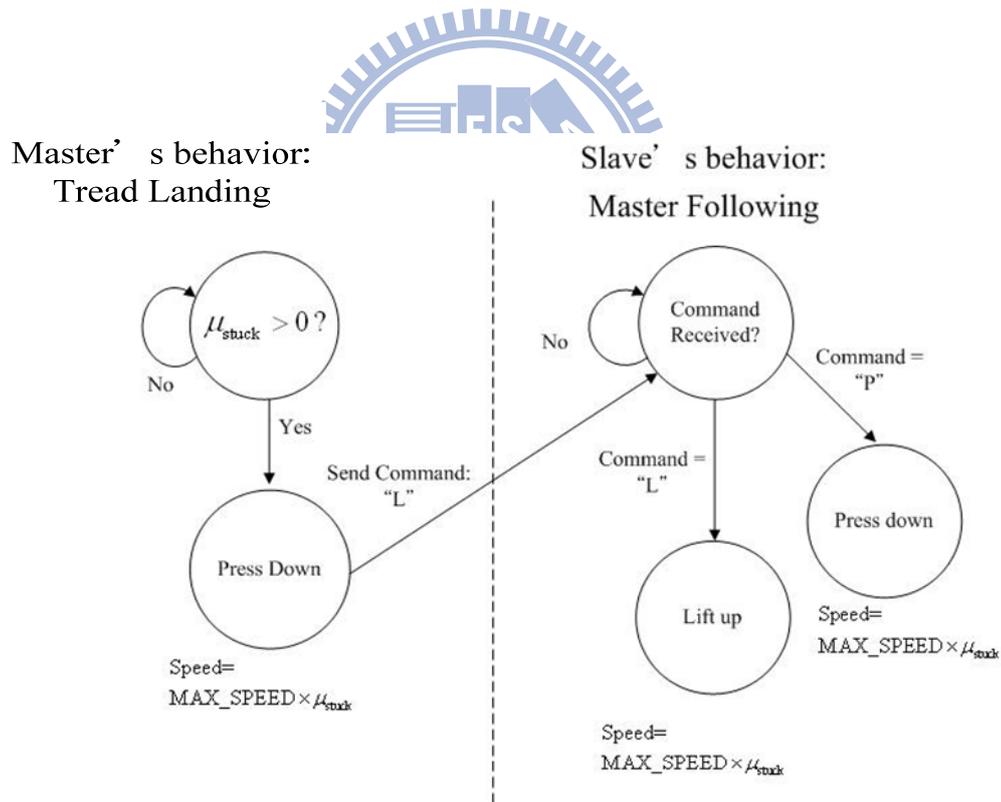


圖 3-10 master 的 tread landing 行為狀態圖

有了行為的設計後，還需要經由行為的協調，master 單元才能決定動作(action)與訊息(message)的輸出。然而，如下圖 3-11、圖 3-12 的例子所示，當機器人藉由馬達扭矩資訊偵測到困住時，因感測資訊的不足，並無法分辨圖 3-11、圖 3-12 這兩種情況，換言之，機器人並無法透過對環境的感知，清楚分辨該執行 riser climbing 或 tread landing 行為。

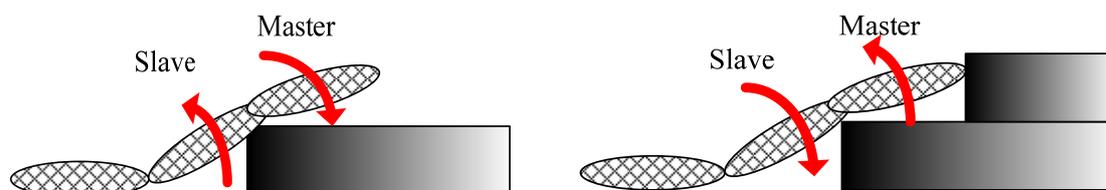


圖 3-11 此時需執行 tread landing 行為

圖 3-12 此時需執行 riser climbing 行為

針對此問題，強化式學習往往是不錯的解決方案。而螞蟻演算法[20][21]可將候選解的機率分布以費洛蒙濃度表示，費洛蒙濃度愈高的解，表示該解愈能適應環境。並利用環境給機器人的反饋訊號，更新費洛蒙濃度，調整控制策略。由於螞蟻演算法在求解過程中，並不會修改或刪除任何現有的解，僅僅改變解的機率分布，因此螞蟻演算法具有極佳的搜索能力，不易陷入次佳解中。基於搜索能力強的優點，本論文引入螞蟻演算法，進行機器人的行為協調，使得機器人可以在運動過程中，自主調整策略，增進克服地形的效率，且即使在缺乏距離感測器的情況下，依然能透過試誤學習，成功克服地形。將於第四章詳細介紹。

第四章 螞蟻演算法自主學習

4.1 螞蟻演算法簡介

螞蟻演算法(ant colony optimization[20][21])係從自然界中，螞蟻群體的行為，得到靈感，進而發展出來的一門學問。自然界中，螞蟻總能找到一條把食物搬回家的最短路徑。當發現食物時，兩隻螞蟻同時離開巢穴，分別走兩條路線到食物處。較快回來的，因費洛蒙揮發較少，會在其路線留下較多的費洛蒙(pheromone)作為記號。因此，其他同伴聞到較重的味道時，自然就會走較短的路線，然而，會有少部分的螞蟻不會依循目前費洛蒙較多的路線去走，而會憑自己的意思走新的路，因此可以漸漸地找到最短路徑。示意圖如下圖 4-1 所示。

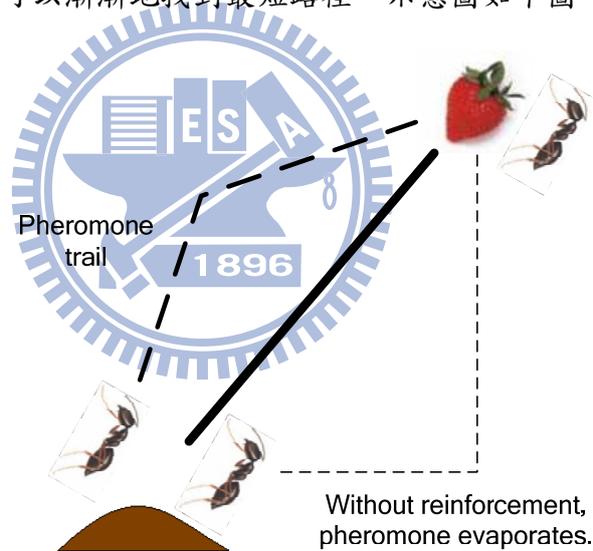


圖 4-1 螞蟻演算法示意圖

螞蟻演算法的優點在於具有良好的搜索能力(exploration ability)。螞蟻演算法在求解的迭代過程中，螞蟻演算法並不會修改或刪除任何現有的解，僅僅是更改每個可能解的機率分布(以費洛蒙表示)。由於所有可能的解，從頭到尾都被保留下來，因此不容易陷入次佳解(sub-optimal solution)中。另外，當環境產生變化，而導致最佳解改變時，搜索能力也賦予了適應性[21]。

然而，螞蟻演算法的缺點在於，當搜索空間過大時，往往會耗費很多時間求解。在串結式機器人的克服地形運動的應用上，會造成地形克服時間過長。對此，本論文提出序列訓練(sequential training，詳見 4.3 節)改善之。

4.2 以螞蟻演算法進行 Master 角色的行為選擇

在此，我們規劃將螞蟻演算法用於串結式機器人的強化式學習。首先，將 master 角色的傾斜角，劃分為 R1、R2、...、R5 五個區域，如下圖 4-2 所示。

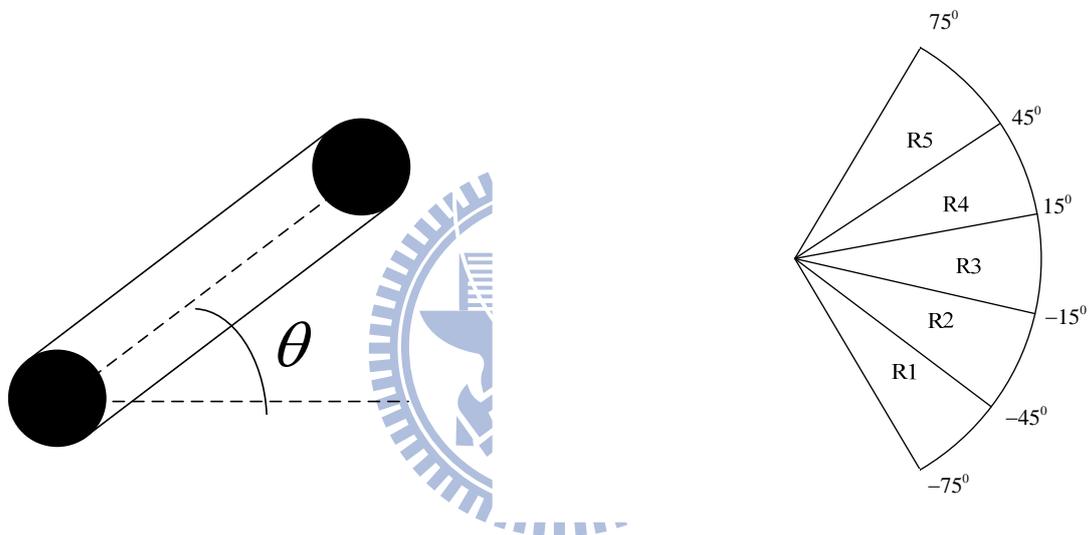


圖 4-2 傾斜角分區圖

每次的決策，就如同一隻螞蟻要離開蟻窩找尋路線。螞蟻會根據目前的傾斜角資訊，決定要去中繼站 R1、R2...或 R5。抵達中繼站後，會根據各條路徑的費洛蒙濃度，機率性地選擇行為輸出，費洛蒙濃度愈高的路徑，被選中的機率愈高，如下圖 4-3 所示。

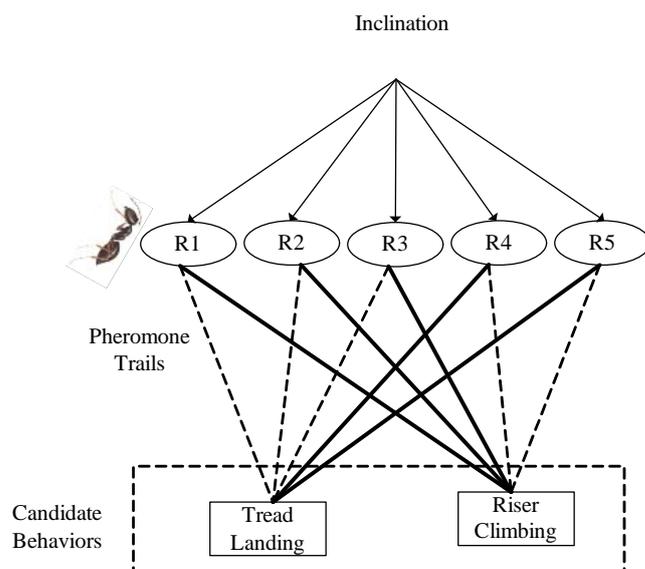


圖 4-3 以螞蟻演算法進行行為協調示意圖

在第 i 個中繼站上，選擇第 j 個行為的機率 $P_{ij}(t)$ ，與第 i 個中繼站到第 j 個行為的路徑上的費洛蒙濃度 $Ph_{ij}(t)$ ，關係如下式(4-1)：

$$P_{ij}(t) = Ph_{ij}(t) / \sum_{k=1}^n Ph_{ik}(t) \quad (4-1)$$

每次的決策循環結束後，將評估這次循環所執行的行為，對環境適應性的高低，並以此更新該路徑上的費洛蒙濃度。費洛蒙更新如下式(4-2)

$$Ph_{ij}(t+1) = Ph_{ij}(t) / b + a \times (1 - \mu_{stuck}) \quad (4-2)$$

b 為費洛蒙的衰減率， a 為費洛蒙的加強率， μ_{stuck} 為困住的程度，困住的程度愈小，代表該行為愈能適應環境。愈大的 b 與 a 會造成現在的現在的狀態(困住的程度)對費洛蒙更新的影響愈大。然而，愈小 b 與 a 卻造成歷史經驗(過去的費洛蒙濃度)對費洛蒙更新的影響愈大。為了兼顧從歷史經驗學習，以及適應目前狀態的變化， b 與 a 的值需加以權衡，在此選擇 b 為 2.31， a 為 100。

4.3 以序列訓練改善隨機選擇行為的缺點

由於螞蟻演算法的隨機選擇特性，且如上頁圖 4-3 所示，總共有 10(5 個傾斜角分區 \times 2 個候選行為)條費洛蒙路線需要進行最佳化，龐大的搜索空間會造成機器人需要花很多時間進行最佳解的搜索，而導致克服時間過長。以圖 5-7 為例 (p. 39)，在第一次訓練時，有將近 30 秒、超過 30 秒，甚至於將近 80 秒的台階克服時間分布，於實務上，這些台階克服時間都嫌過長。若再觀察機器人克服台階的過程，隨機的選擇的特性會讓機器人陷於多次反覆下圖 4-4 至圖 4-7 的姿態，困住好一段時間後，才能擺脫困境。

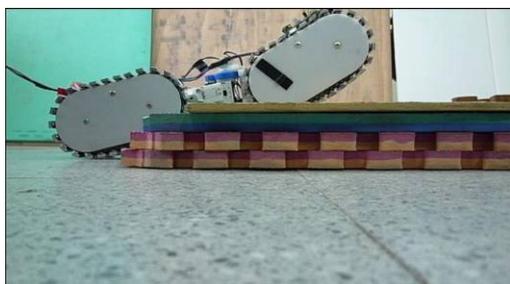


圖 4-4 克服台階過程

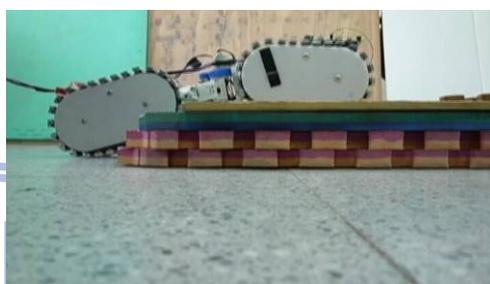


圖 4-5 克服台階過程

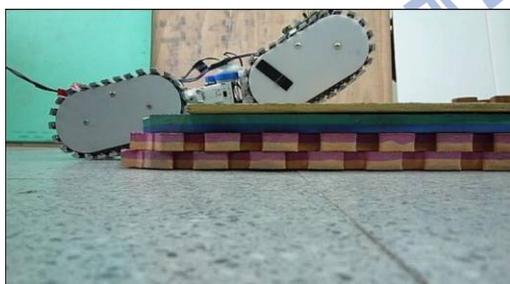


圖 4-6 克服台階過程

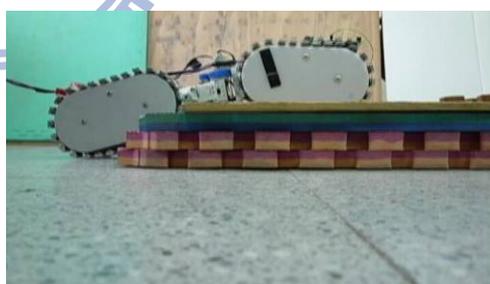


圖 4-7 克服台階過程

因此，本研究提出了序列訓練(sequential training)，以改善之。如下表 4-1 所示，利用預先定義好的行為序列，取代隨機行為選擇，如此一來，機器人不會陷入持續反覆如圖 4-4 至圖 4-7 姿態的窘境，節省嘗試姿態的時間。當機器人困住超過 10 秒鐘時，將令機器人執行此行為序列，並同時更新費洛蒙，以縮短學習過程。

表 4-1 預先定義好的行為序列

Riser Climbing	...	Riser Climbing	Tread Landing	...	Tread Landing
----------------	-----	----------------	---------------	-----	---------------

加入了連續性訓練(sequential training)後，整體以螞蟻演算法進行行為協調的流程，如下圖 4-8 所示。

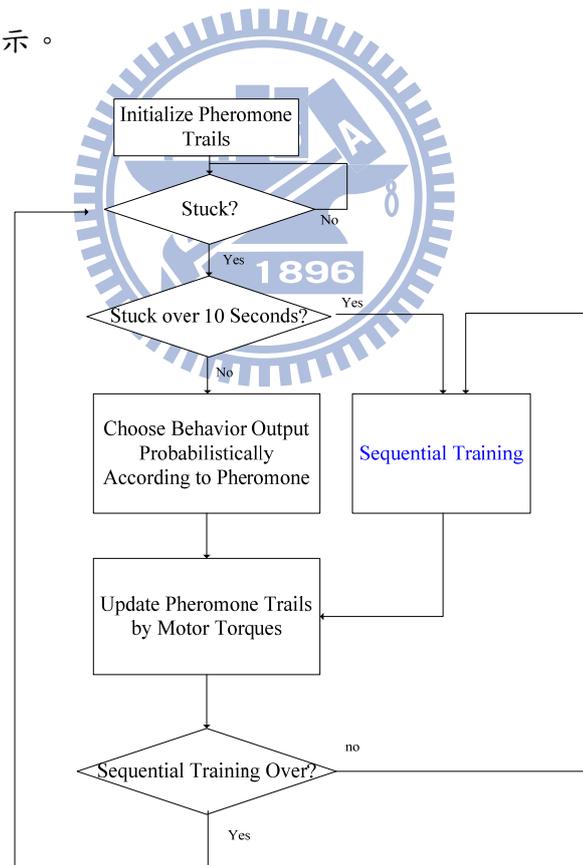


圖 4-8 螞蟻演算法行為協調加入序列訓練後的流程圖

螞蟻演算法加入序列訓練的演算步驟如下：

Step 1：初始化費洛蒙。

Step 2：判斷有無困住，若困住則繼續往下執行。

Step 3：判斷是否困住超過 10 秒，若是則跳至 step 5，反之則繼續往下執行 step 4。

Step 4：根據費洛蒙濃度機率性地選擇行為，跳至 step 6。

Step 5：執行序列訓練。

Step 6：根據馬達扭矩資訊更新費洛蒙。

Step 7：判斷序列訓練是否結束，若是則跳回 step 2，反之則跳回 step 4。

4.4 以費洛蒙調整加強歷史經驗基礎

如下圖 4-9 所示，以螞蟻演算法進行行為協調時，當機器人困住的時候，才會執行行為，接著也才會去執行費洛蒙。但若機器人沒困住，則不會更新費洛蒙。

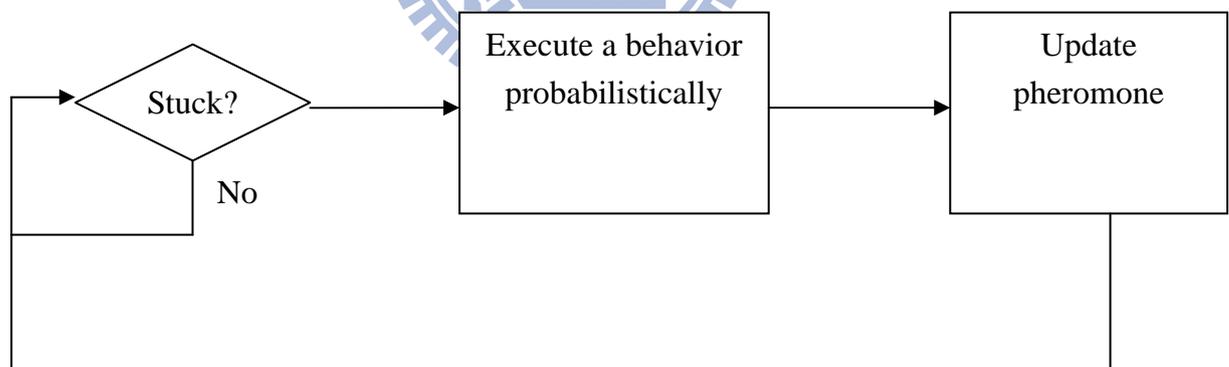


圖 4-9 更新費洛蒙流程圖

如此一來，會造成如下圖 4-10 的現象。當機器人經由長時間的試誤，終於擺脫困住的情況後，僅在由困住變為未困住的那一瞬間，費洛蒙才會增加，無法達到像初始狀態那樣高的濃度。

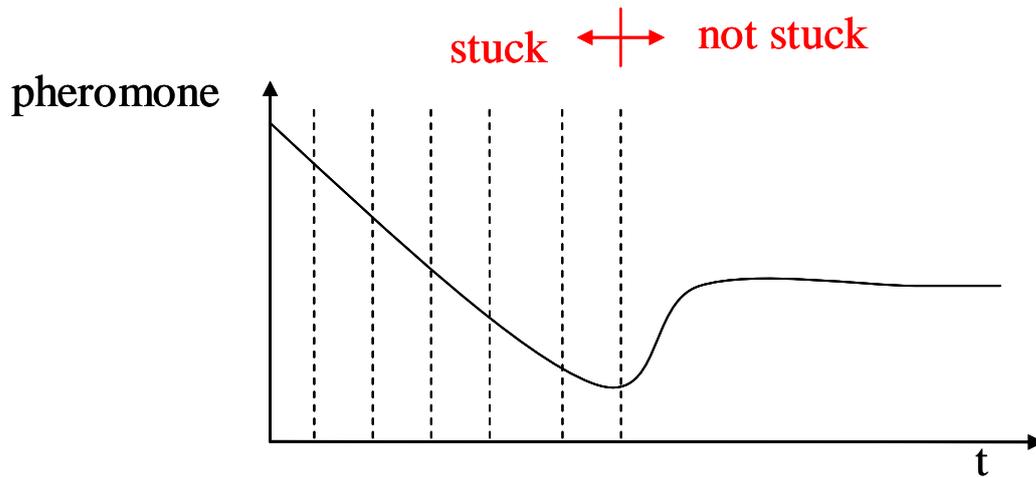


圖 4-10 經由學習後，費洛蒙難以達到如初始狀態的高濃度

再觀察上式(4-2)， $ph_{ij}(t)$ 代表的是由過去累積到現在的經驗，若 $ph_{ij}(t)$ 過低，便無法提供未來的決策一個強而有力的經驗基礎，機器人此時就往往需要重新學習試誤，因而造成反覆學了多次以後，地形克服時間的標準差反而增大。如圖 5-7(p. 39)所示，第三、四次的克服時間資料較第二次為分散。(實驗數據詳見 5.1.1 節)

針對此問題，本論文提出了費洛蒙調整(pheromone adjustment)的機制。以下表 4-2 為例，本論文所提出的費洛蒙調整方式為，將目前的傾斜角分區(在此例中，位於 R3，即 $-15^{\circ} \sim 15^{\circ}$)至 tread landing 行為路線上的費洛蒙提升至 250(250 為費洛蒙初始設定的最高濃度，代表最適應環境)，riser landing 的費洛蒙再依比例調整。將費洛蒙濃度提昇，可提昇過去的經驗對費洛蒙更新的影響力，換言之，可提供較強大的歷史經驗基礎，如此一來，原本已經能適應環境的策略較容易保存下來，經過多次訓練後，地形的克服時間趨於一致。

表 4-2 費洛蒙調整示例

inclination	R3	
behavior	tread landing	riser climbing
pheromone	50 → 250	30 → 150

將費洛蒙調整引入螞蟻演算法，流程圖則如下圖 4-11 所示。當機器人偵測到超過 5 秒的時間都沒困住時，代表目前的費洛蒙分布適合環境，因此即啟動費洛蒙調整機制，以提供未來的行為選擇，較強的歷史經驗基礎。

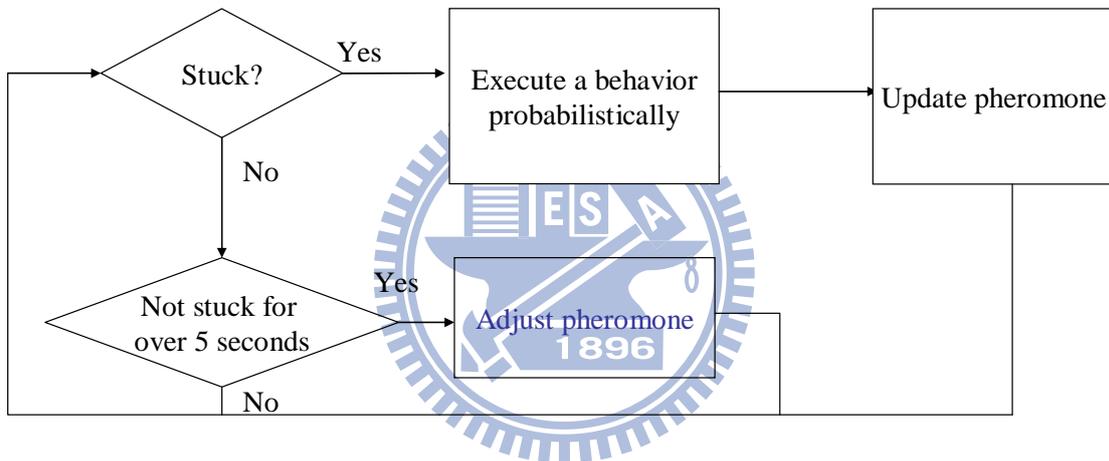


圖 4-11 螞蟻演算法結合費洛蒙調整流程圖

螞蟻演算法加入費洛蒙調整的演算步驟如下：

- Step 1：判斷有無困住，若困住則跳至 step 4，反之則繼續往下執行 step 2。
- Step 2：機率性地執行行為。
- Step 3：更新費洛蒙，跳回 step 1。
- Step 4：判斷是否超過 5 秒皆未困住，若是則繼續往下執行 step 5，反之則跳回 step 1。
- Step 5：執行費洛蒙調整，跳回 step 1。

第五章 實驗結果與討論

在本章中，首先為了驗證螞蟻演算法自主學習的效果，將兩節機器人於台階、斜坡上的台階與階梯等三種地形上，進行自主克服地形運動，並統計地形的克服時間(定義為從接觸到地形到完全克服地形所經過的時間)，驗證機器人是否能透過學習，增進克服地形的效率。同時也加入序列訓練，驗證序列訓練是否能改善克服時間過長的問題；加入費洛蒙調整，驗證費洛蒙調整是否能改善學習反效果問題。

接著，將策略擴充到三節機器人，驗證分散式架構的擴充性，並克服由階梯與崎嶇地形所組成的複雜地形。

5.1 兩節機器人台階實驗結果

如下圖 5-1 所示，在一由巧拼墊所疊成的 7 公分高台階上，讓機器人往台階前進，自主爬上台階。實驗流程如下圖 5-2 所示，第一次訓練結束後，奠基於第一次訓練的結果，再進行一次克服台階的實驗，以此類推，四次訓練為一次實驗，一次實驗結束後，即忘卻學習結果，重啟新的一次實驗，當完成 10 次實驗後，即結束實驗。



圖 5-1 台階地形實景

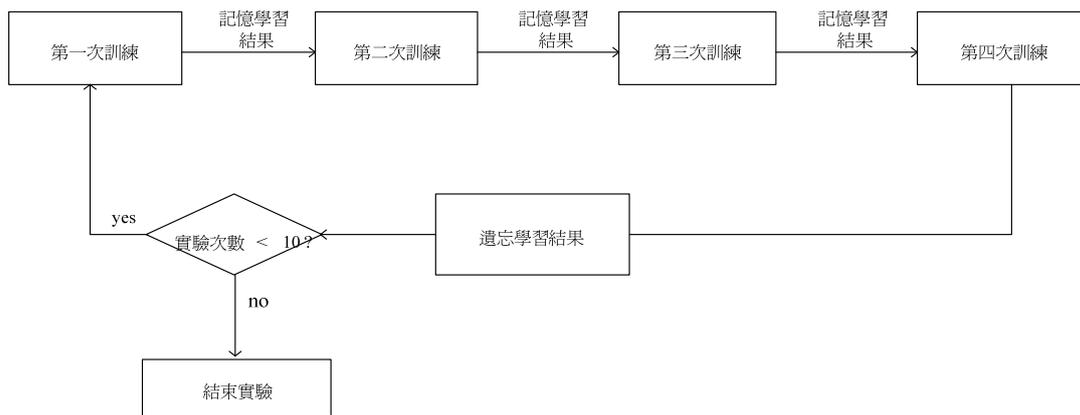


圖 5-2 螞蟻演算法學習實驗流程



5.1.1 原始螞蟻演算法實驗結果

機器人利用原始螞蟻演算法學習克服台階的過程，如下圖 5-3 至圖 5-6 所示。

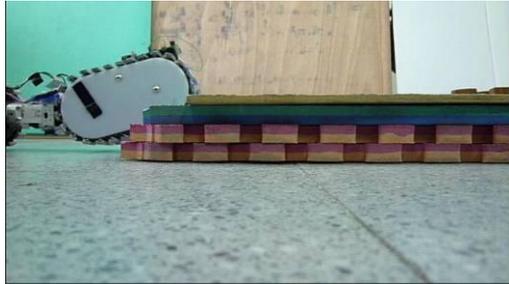


圖 5-3 克服台階過程

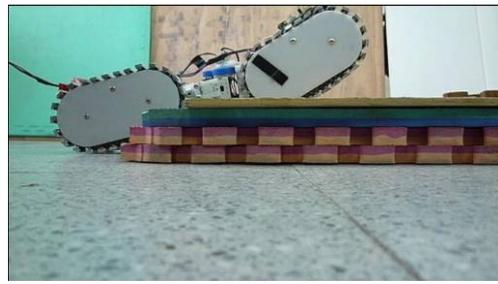


圖 5-4 克服台階過程

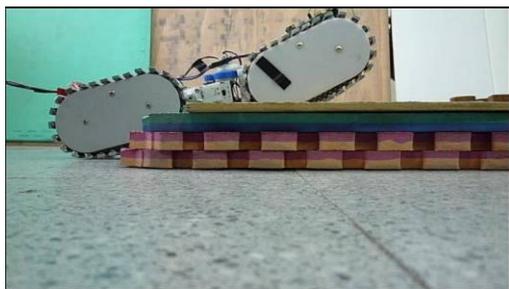


圖 5-5 克服台階過程

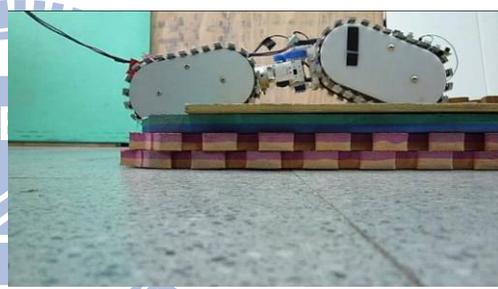


圖 5-6 克服台階過程

克服時間的資料如下表 5-1 所示。由此表，大致上可觀察到第二、三、四次訓練的克服時間較第一次訓練時短的現象。資料散佈圖則如下圖 5-7 所示。

表 5-1 原始螞蟻演算法克服時間資料表(單位:秒)

	第一次訓練	第二次訓練	第三次訓練	第四次訓練
第一次實驗	21	10	9	11
第二次實驗	77	10	11	12
第三次實驗	18	10	12	10
第四次實驗	28	13	32	18
第五次實驗	22	11	7	32
第六次實驗	17	11	9	15
第七次實驗	25	18	11	13
第八次實驗	15	10	10	9
第九次實驗	25	10	35	8
第十次實驗	35	9	20	55

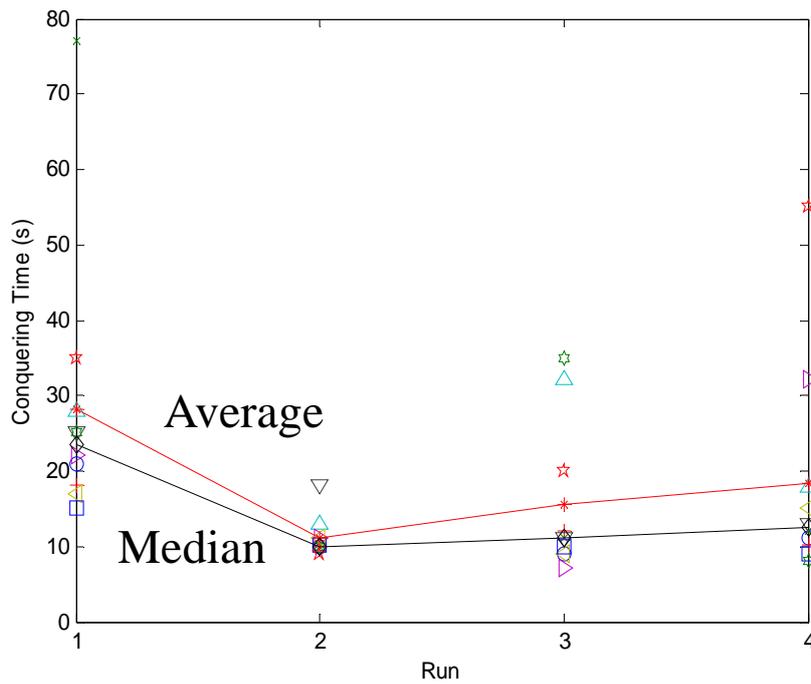


圖 5-7 原始螞蟻算法台階克服時間散佈圖

由於資料分布情況，導致平均值易受極端值影響，因此本論文另外取克服時間的中位數來分析。克服時間的中位數與標準差統計如下表 5-2 所示。

表 5-2 原始螞蟻演算法台階克服時間的統計資料，單位：秒

	第一次訓練	第二次訓練	第三次訓練	第四次訓練
平均值	28.3	11.2	15.6	18.3
中位數	22	10	11	12
標準差	19.01	2.65	10.55	7.34

由上圖 5-7 與表 5-2 可知，經由第一次訓練後，克服時間的中位數可減少 55%(由第一次訓練的 22 秒，縮減為第二次訓練的 10 秒)。但有以下兩種問題：

- (1) 偶爾克服時間過長：觀察表 5-1，克服時間偶爾會有 77 秒、55 秒、35 秒等過大(在台階地形上，將克服時間大於 30 秒的情況，定義為克服時間過長)的值。此乃由於傳統螞蟻演算法隨機選擇行為的特性，造成有時機器人耗費過多時間試誤。
- (2) 學習反效果問題：觀察表 5-1，可發現第四循環、第九循環，及第十循環，均有第三或四次訓練的克服時間大於第二次訓練的現象，觀察表 5-2，亦可發現，第三次訓練時的克服時間的標準差，大於第二次訓練。觀察其中一次實驗時，傾斜角處於 R3($-15^{\circ}\sim 15^{\circ}$)時，兩種行為的費洛蒙隨時間的變化如下圖 5-8 所示。可發現第二次訓練完成後，tread landing 的費洛蒙濃度小於第一次訓練完成後的濃度。如此一來，將造成訓練的結果無法構成足夠強大的經驗基礎(原理詳見 4.4 節)，導致經過多次訓練後，克服時間反而增長的現象。

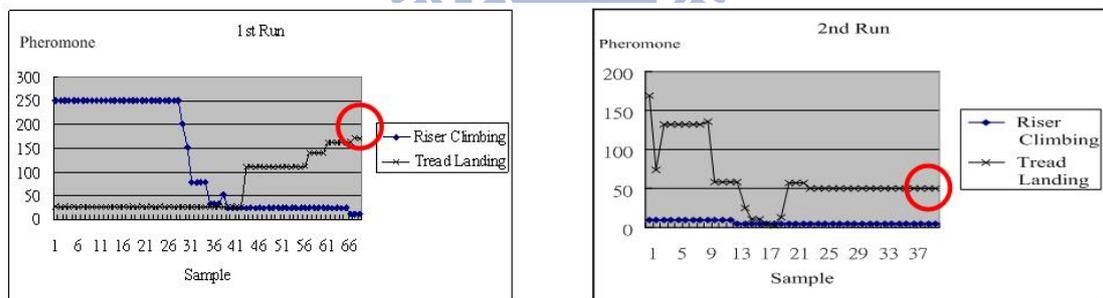


圖 5-8 Tread landing 的費洛蒙衰減

5.1.2 加入序列訓練實驗結果

加入序列訓練(sequential training, 方法詳見 4.3)後,克服時間資料如下表 5-3 所示。原始螞蟻演算法與加入序列訓練的比較,如下圖 5-9 所示。

表 5-3 加入序列訓練後的台階克服時間

	第一次訓練	第二次訓練	第三次訓練	第四次訓練
第一次實驗	26	15	11	20
第二次實驗	21	19	20	12
第三次實驗	16	21	18	21
第四次實驗	10	19	11	20
第五次實驗	17	20	11	13
第六次實驗	20	16	28	25
第七次實驗	15	16	15	20
第八次實驗	16	20	20	15
第九次實驗	15	14	20	14
第十次實驗	20	20	27	21

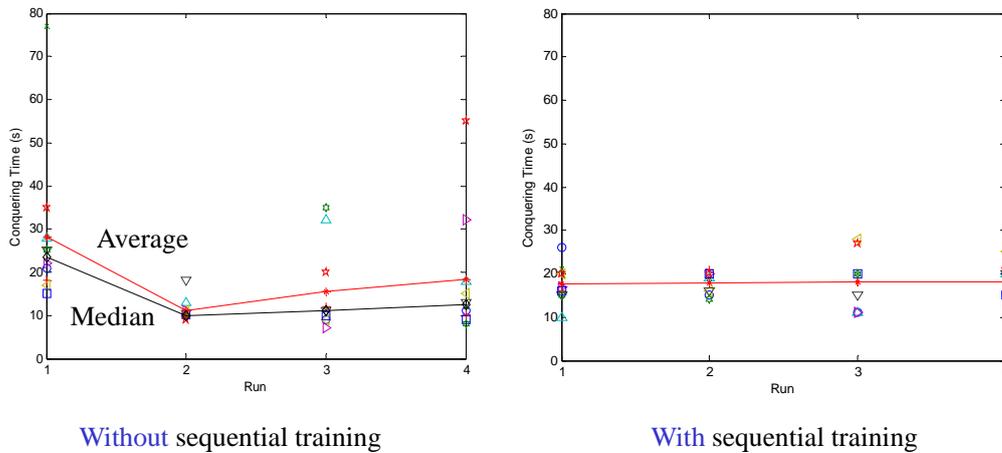


圖 5-9 有無序列訓練台階克服時間比較

由上圖 5-9 與下表 5-4 可發現，加入序列訓練後，第一次訓練的克服時間整體較未加序列訓練時短(未加入序列訓練時，第一次訓練的平均克服時間為 28.3 秒；加入序列訓練後，平均克服時間則減為 17.6 秒)，同時也由於第一次訓練的克服時間就被拉低，第二、三、四次訓練的平均克服時間均與第一次訓練差不多。同時，若觀察原始螞蟻演算法，與加入序列訓練後的台階克服時間標準差，如下所示，可發現，加入序列訓練後，第一、二、三、四次的克服時間標準差均較未加連續性訓練時小。但第三、四次訓練的克服時間標準差，依然大於第二次訓練，換言之，學習反效果問題依然存在。

表 5-4 台階克服時間平均值(標準差) 單位：秒

	1 st run	2 nd run	3 rd run	4 th run
Without sequential training	28.3 (19.01)	11.2 (2.65)	15.6 (10.55)	18.3 (7.34)
With sequential training	17.6 (4.35)	18.0 (2.49)	18.1 (6.23)	18.1 (4.28)

5.1.3 加入費洛蒙調整實驗結果

加入費洛蒙調整(pheromone adjustment，方法詳見 4.4 節)後，克服時間資料如下表 5-5 所示。原始螞蟻演算法與加入序列訓練的比較，如下圖 5-10 所示。

表 5-5 加入費洛蒙調整後，台階克服時間資料 (單位：秒)

	第一次訓練	第二次訓練	第三次訓練	第四次訓練
第一次實驗	23	12	11	13
第二次實驗	31	8	15	12
第三次實驗	36	13	15	19
第四次實驗	8	24	12	18
第五次實驗	15	12	21	14
第六次實驗	28	10	12	15
第七次實驗	14	15	15	11
第八次實驗	26	9	13	13
第九次實驗	12	15	12	11
第十次實驗	21	11	20	15

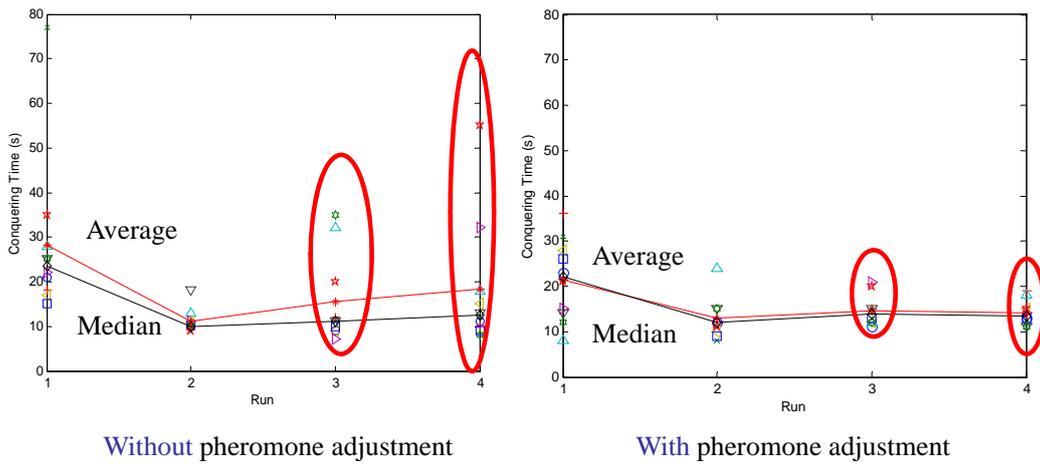


圖 5-10 有無費洛蒙調整時，台階克服時間比較

有無費洛蒙調整時，台階克服時間中位數與標準差，比較如下表 5-6 所示。可發現，加入費洛蒙調整後，第三、四次訓練的克服時間標準差，均不會大於第二次訓練的標準差；另也可發現第三、四次訓練克服時間的平均值與中位數差不多，這是由於地形克服時間經多次訓練後依然趨於一致，因此第三、四次訓練的克服時間分散程度不會增大，無極端值產生。可見學習反效果問題有效地被抑制。

表 5-6 台階克服時間中位數(標準差) 單位：秒

	1st run	2nd run	3rd run	4th run
Without pheromone adjustment	22 (19.01)	10 (2.65)	11 (10.55)	12 (7.34)
With pheromone adjustment	22 (9.05)	12 (4.53)	14 (3.44)	13.5 (2.73)

5.1.4 台階實驗結果小結

在本小節中，將台階實驗結果小結如下：

(1) 原始螞蟻演算法：

- i. 機器人克服台階地形時間的中位數，最多可縮減 55%。
- ii. 但具有偶爾克服時間過長(大於 30 秒)，與學習反效果(經過學習後，克服時間反而變長)問題。

(2) 加入序列訓練：

- i. 第一次訓練的克服時間中位數被拉低。
- ii. 與原始螞蟻演算法相較，每次訓練克服時間的標準差，均減小。
- iii. 但學習反效果問題依然存在。

(3) 加入費洛蒙調整：

- i. 第三、四次訓練的克服時間標準差皆不會超過第二次訓練，學習反效果有效被抑制。



5.2 兩節機器人於斜坡上的台階實驗結果

斜坡的台階地形尺寸如圖 5-11 所示，為增加克服地形的難度，在斜坡上又多加台階。地形實景圖則如圖 5-12 所示。

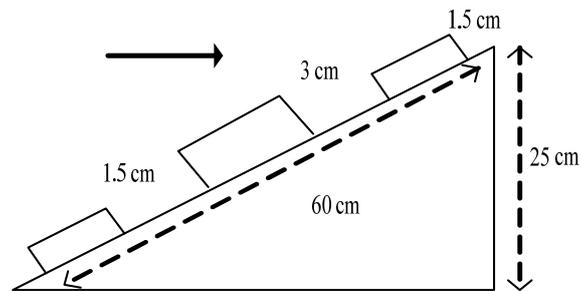


圖 5-11 斜坡上的台階尺寸

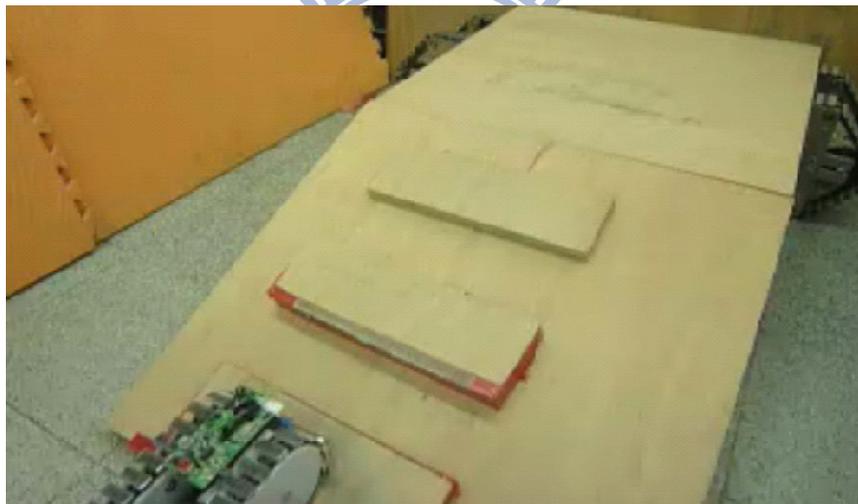


圖 5-12 斜坡上的台階地形實景

5.2.1 原始螞蟻演算法

僅使用原始螞蟻演算法，地形的克服時間資料如下表 5-7 所示。大致上，可觀察到第二、三、四次訓練時，克服時間略小於第一次訓練的現象，證實螞蟻演算法也可增進此地形的克服效率。

表 5-7 斜坡上的台階克服時間(單位：秒)

	第一次訓練	第二次訓練	第三次訓練	第四次訓練
第一次實驗	29	22	21	20
第二次實驗	16	11	12	11
第三次實驗	23	16	16	16
第四次實驗	18	20	18	21
第五次實驗	24	21	20	23
第六次實驗	21	19	20	17
第七次實驗	24	19	18	21
第八次實驗	21	18	19	24
第九次實驗	20	21	20	17
第十次實驗	24	22	20	19

5.2.2 加入序列訓練

加入序列訓練後，地形克服時間資料如下表 5-8 所示。與原始螞蟻演法比較圖如圖 5-13。平均值與標準差比較如表 5-9 所示。

表 5-8 加入序列訓練後，斜坡上的台階地形克服時間(單位：秒)

	第一次訓練	第二次訓練	第三次訓練	第四次訓練
第一次實驗	19	21	17	19
第二次實驗	18	20	21	20
第三次實驗	22	19	19	18
第四次實驗	20	17	21	21
第五次實驗	20	20	19	19
第六次實驗	18	19	19	19
第七次實驗	18	21	18	23
第八次實驗	20	21	23	18
第九次實驗	20	22	19	21
第十次實驗	21	19	21	18

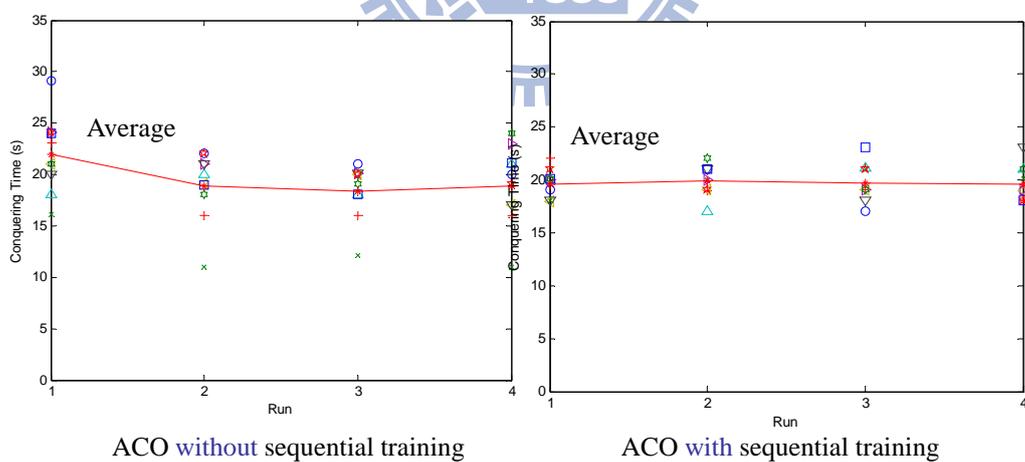


圖 5-13 有無加入序列訓練，斜坡上的台階克服時間比較

表 5-9 斜坡上的台階克服時間平均值(標準差) 單位：秒

	1 st run	2 nd run	3 rd run	4 th run
Without sequential training	22 (3.65)	18.9 (3.35)	18.4 (2.67)	18.9 (3.81)
With sequential training	19.6 (1.35)	19.9 (1.45)	19.7 (1.77)	19.6 (1.65)

由圖 5-13、表 5-9 可知，加入序列訓練後，可將第一次訓練的平均克服時間拉低，且每次訓練的克服時間標準差，均小於未加序列訓練。

5.2.3 斜坡上的台階實驗結果小結

本小節對斜坡上的台階實驗結果小結如下：

- (1) 原始螞蟻演算法：
 - i. 由於此地形較簡單，故第一次訓練時就不會花太多時間克服地形。
 - ii. 但學習依然能發揮效果，第二次訓練後，克服時間的平均值依然會減小。
- (2) 加入序列訓練：
 - i. 第一次訓練的平均克服時間被拉低。
 - ii. 每次訓練的克服時間標準差均較原始螞蟻演算法小。

5.3 兩節機器人階梯實驗結果

5.3.1 原始螞蟻演算法

台階地形尺寸如下圖 5-14 所示。台階地形實景照片則如 圖 5-15 所示。克服時間資料如表 5-10 所示。由表上，大致可發現只有在第一、二、四、十次實驗時，才能發現第四次訓練的克服時間較第一次訓練時短，其他幾次實驗的克服時間資料則較看不出學習的效果。這時由於此地形較為複雜，原始的螞蟻演算法不易突顯出效果。

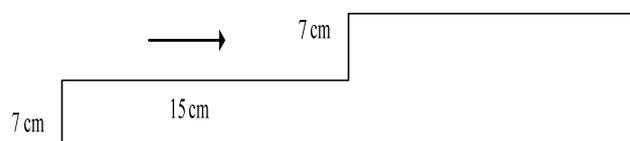


圖 5-14 階梯地形尺寸

圖 5-15 階梯地形實景

表 5-10 階梯克服時間資料(單位：秒)

	第一次訓練	第二次訓練	第三次訓練	第四次訓練
第一次實驗	40	50	36	23
第二次實驗	39	49	36	18
第三次實驗	21	14	49	30
第四次實驗	21	31	17	13
第五次實驗	18	24	36	28
第六次實驗	26	30	13	37
第七次實驗	14	12	12	14
第八次實驗	22	17	26	35
第九次實驗	18	16	15	30
第十次實驗	62	37	47	15

5.3.2 加入序列訓練

加入序列訓練後，克服時間資料如下表 5-11 所示。由表可發現，大致上第二、三、四次訓練的克服時間，比第一次訓練時短。換而言之，加入了序列訓練後，使得螞蟻演算法可以較明顯地增進克服階梯地形的效率。散佈圖比較則如圖 5-16 所示。

表 5-11 加入序列訓練後，階梯克服時間資料

	第一次訓練	第二次訓練	第三次訓練	第四次訓練
第一次實驗	23	16	25	20
第二次實驗	20	17	13	15
第三次實驗	27	25	20	20
第四次實驗	32	29	15	21
第五次實驗	26	16	23	15
第六次實驗	28	17	23	26
第七次實驗	14	16	26	25
第八次實驗	24	16	15	14
第九次實驗	24	21	21	18
第十次實驗	24	30	18	25

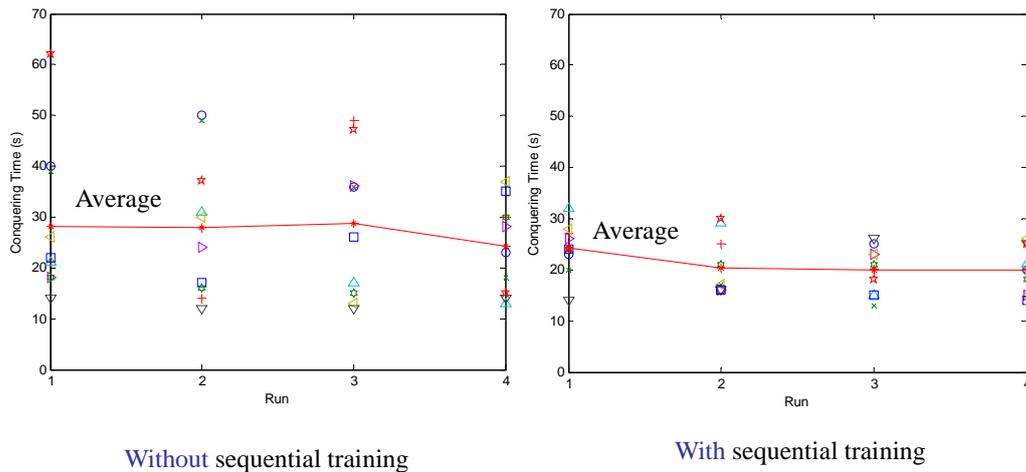


圖 5-16 加入序列訓練，階梯克服時間散佈圖

有無加入序列訓練時，階梯克服時間的平均值與標準差統計如下表 5-12 所示。

表 5-12 階梯克服時間平均值(標準差) 單位：秒

	1st	2nd	3rd	4th
Without sequential training	28.1 (14.74)	28 (13.95)	28.7 (14.00)	24.3 (8.92)
With sequential training	24.2 (4.83)	20.3(5.66)	19.9 (4.51)	19.9 (4.43)

觀察圖 5-16 與表 5-12 可發現，加入序列訓練後，會有以下兩種現象。

- (1) 加入序列訓練後，在第二次訓練的時候，即可觀察到平均克服時間減少的現象。換言之，序列訓練可縮短學習過程，讓學習效果較快展現出來。
- (2) 加入序列訓練後，克服時間的標準差，均小於原始螞蟻演算法。亦即克服時間過長的問題被改善。

5.3.3 階梯實驗結果小結

本小節對階梯實驗結果小結如下：

(1) 原始螞蟻演算法：

- i. 由於階梯地形較台階複雜，因此在第四次訓練時，才能觀察到克服地形時間的平均值，與標準差減小的現象。
- ii. 依然存在偶爾克服時間過長(大於 40 秒)的問題。

(2) 加入序列訓練：

- i. 在第二次訓練時，即可觀察到克服時間的平均值與標準差減小的現象。換言之，序列訓練加快了學習效果。
- ii. 每次訓練的標準差，皆小於未加入序列訓練時的標準差，有效改善克服時間過長的問題。

5.4 三節機器人克服複雜地形

本論文所提出的串結式多單元分散式控制，使得當硬體機構擴充時，軟體演算法亦能輕易擴展。在此小節中，將原兩節機器人擴充為三節機器人，並於階梯與崎嶇地形所組成的複雜地形中進行測試。如下圖 5-17 至圖 5-23 所示。圖 5-17 中，機器人困住，機率性地選擇將第一節往上抬。此時依然困住，接著 圖 5-18 中，改將第一節往下壓，終於可以克服階梯。然而圖 5-19 中，當接觸到崎嶇地形時，由於先前的下壓姿態，使得機器人困住。因此 圖 5-20 中，將第一節上抬、拉平，以克服崎嶇地形。最後，如圖 5-23 所示，終於成功通過複雜地形。

在此過程中，可發現，所提出的策略，讓串結式機器人於運動過程中，嘗試動作，並自主判斷該動作是否適應地形，以調整動作的機率分布，藉由即時調整策略，最終能順利克服地形。



圖 5-17 困住，嘗試往上抬，依然困住

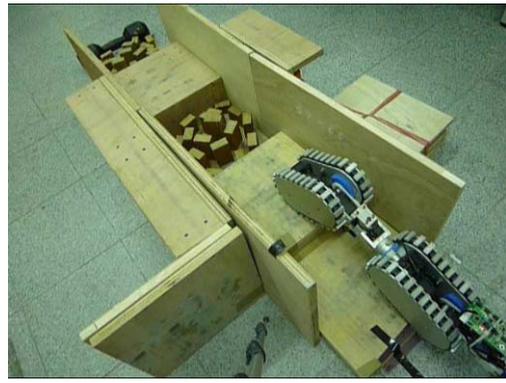


圖 5-18 改為往下壓，得以克服



圖 5-19 先前的往下壓姿態，導致困住



圖 5-20 頭部拉平，以克服困境



圖 5-21 成功通過

第六章 結論與未來展望

6.1 結論

在此為本論文做出以下幾點結論：

1. 以串結式機器人單節之間的合作協調克服複雜地形：

提出串結式機器人的串結式多單元分散式控制，透過 master 與 slave 角色的合作：master 的攀爬豎板(riser climbing)與登上踏板(tread landing)行為、slave 遵從 master 的命令，得以克服複雜地形。雖然目前實現在三節串結式機器人上，只需兩種角色，但未來機器人擴充為更多節時，可增添更多角色，具有良好的擴充性。

2. 透過學習，增進機器人克服地形的效率：

引入螞蟻演算法，賦予機器人自主學習能力，藉由運動過程中的試誤，自主調整策略，增進機器人克服地形的效率。在台階、斜坡上的台階，與階梯三種地形上，兩節串結式機器人實驗結果，可得知，經由螞蟻演算法第一次訓練後，可縮減整體地形克服時間 55%，但存在克服時間過長，與學習反效果(多次學習後，克服時間反而變長)的問題。因此提出以下兩種方法以解決之。

(1) 序列訓練：

以預先定義好的行為序列，取代隨機行為選擇。可縮短學習歷程，加快克服時間的收斂，並降低克服時間的分散程度，實驗結果證實，加入序列訓練後，第一次訓練的克服時間標準差，較未加入序列訓練時縮減 77%，有效改善克服時間過長問題。

(2) 費洛蒙調整：

當行為適合環境時，適時調高費洛蒙。實驗結果證實，經多次學習後，克服時間標準差並不會隨著增大，有效壓抑學習反效果。(詳見圖 5-10 與表 5-6)

本論文已完成軟體控制程式的設計於 F2812 控制核心，以及硬體週邊電路的整合，實現控制策略於兩節串結式機器人，並能輕易地將策略擴充至三節串結式機器人，驗證了分散式策略的擴充性。

6.2 未來展望

本論文設計分散式控制策略，使得串結式機器人，能夠發揮本身機構的優點，克服台階、斜坡上的台階、階梯與崎嶇地形，未來仍有可繼續研究發展之處，以下列出幾點未來可研究的方向：

(1) 更複雜的地形：

目前不使用距離感測器，僅使用加速度計與馬達資訊偵測，即使機器人能透過學習克服地形，但難免走得跌跌撞撞，未來可加入雷射測距儀、超音波測距儀、碰撞感測器等感測器，以克服更複雜的地形。

(2) 結合定位技術以達到導航功能：

結合 GPS(室外)、慣性導航(室內)與 ZigBee 定位(室內)等技術，讓機器人在克服困難地形的同時，可以往目的位置前進。

(3) 搜尋方面：

結合影像辨識或熱源偵測等，讓機器人可以偵測目標物或災區罹難者，能夠執行搜尋任務。

参考文献

- [1] J. K. Hopkins, B. W. Spranklin, and S. K. Gupta , "A Survey of Snake-inspired robot designs," *Bioinspirations & Biomimetics*, vol. 4, no. 2, 2009.
- [2] S. Hirose, "Biologically Inspired Robots: Snake-like Locomotors and Manipulators," Oxford: Oxford University Press, 1993.
- [3] M. Mori and S. Hirose, "Development of Active Cord Mechanism ACM-R3," *Proceedings of of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 1552-1557, 2001.
- [4] A. Crespi, A. Badertscher, A. Guignard, and A. J. Ijspeert, "Amphibot I: An Amphibious Snake-like Robot," *Robotics and Autonomous Systems*, vol. 50, no. 4, pp. 163-175, March 2005.
- [5] S. Hirose, A. Morishima, S. Tukagosi, T. Tsumaki, H. Monobe, "Design of Practical Snake Vehicle: Articulated Body Mobile Robot KR-II," *5th Conference on Advanced Robotics*, vol. 1, pp. 833-838, 1991.
- [6] B. Klaassen and K. L. Paap, "GMD-SNAKE2: a Snake-like Robot driven by Wheels and a Method for Motion Control," *IEEE/RSJ International Conference on Intelligent Robotics and Automation*, vol. 4, pp. 3014-3019, 1999.
- [7] H. Yamada and S. Hirose, "Development of Practical 3-deimensional Active Cord Mechanism ACM-R4," *Journal of Robotics and Mechatronics*, vol. 18, no. 3, pp. 305-311, 2006.
- [8] S. Hirose, http://www-robot.mes.titech.ac.jp/robot/snake_e.html
- [9] J. Borenstein, G. Granosik, and M. Hansen, "The Omitread Serpentine Robot-Design and Field Performance," *Proceedings of SPIE Defense and Security Conference, Unmanned Ground Vehicle Technology VII*, pp. 324-332, February

2005.

- [10] W. R. Hutchison, B. J. Constantine, J. Borenstein, and J. Pratt, "Development of Control for a Serpentine Robot," *Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robots and Automation*, pp. 149-154, 2007.
- [11] H. Zhang, W. Wang, Z. Deng, G. Zong, and J. Zhang, "A Novel Reconfigurable Robot for Urban Search and Rescue," *International Journal of Advanced Robot Systems*, vol. 3, pp. 359-366.
- [12] H. Zhgnag, W. Wang, Z. Deng, G. Zong, and J. Zhang, "Locomotion Capabilities of a Novel Reconfigurable Robot with 3 DOF Active Joints for Rugged Terrain," *Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5588-5593, 2006.
- [13] J. Gao, X. Gao, W. Zhu, J. Zhu, B. Wei, "Design and Research of a New Structure Rescue Snake Robot with All Body Drive System," *Proceedings of 2008 IEEE International Conference on Mechatronics and Automation*, pp. 119-124, 2008.
- [14] W. Wang, H. Zhang, G. Zong, and J. Zhang, "Design and Realization of a Novel Reconfigurable Robot with Serial and Parallel Mechanisms," *IEEE International Conference on Robotics and Biomimetics*, pp.697-702, 2006.
- [15] 陳俊傑, "串結式救援機器人之設計與控制," 國立台北科技大學機電整合研究所碩士論文, 中華民國98年
- [16] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: a Survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237-285, 1996.
- [17] "TMS320F2810, TMS30F2811, TMS320F2812, TMS320C2810, TMS320C2811, TMS320C2812 DSPs (Rev. O)," *Texas Instruments*, 2007.
- [18] R. C. Arkin, "Behavior-based Robotics," *The MIT Press*, 1998.

- [19] C. T. Lin and C. S. G. Lee, “Neural Fuzzy Systems,” *Prentice Hall.*, 1996.
- [20] M. Dorigo, M. Birattari, T. Stutzle, “Ant Colony Optimization,” *Computational Intelligence Magazine, IEEE*, vol. 1, no. 4, pp.28-39, Nov. 2006.
- [21] A. Costa, “Ants, Stochastic Optimisation and Reinforcement Learning,” *The Australia Mathematical Society*, vol. 32, no. 2, pp.116-123, 2005.

