



朝陽科技大學  
資訊工程系

碩士論文

使用進化式演算法解決部分故障四足型機器人的調適性移動行為

Self-Adjusting Locomotion on a Partially Broken-down Quadrupedal

Biomorphic Robot by Evolutionary Algorithms

指導教授：吳世弘 博士

研究生：邱國原

中華民國 102 年 6 月 28 日



朝陽科技大學資訊工程系

Department of Graduate Institute Computer  
Science and Information Engineering  
Chaoyang University of Technology

碩士論文

Thesis for the Degree of Master

使用進化式演算法解決部分故障四足型機器人的調適性移動行為

Self-Adjusting Locomotion on a Partially Broken-down Quadrupedal

Biomorphic Robot by Evolutionary Algorithms

指導教授：吳世弘 博士(Shih-Hung Wu)

研究生：邱 國 原(Guo-Yuan Qiu)

中華民國 102 年 6 月 28 日

June 28, 2013



傳統機器人利用動力學分析來控制機器人的行為，動力學分析最大缺點是當周遭環境變動時，無法即時編制一套行為來應付變動中的環境，進化式機器人是利用進化式演算法的優點，來彌補傳統機器人的控制缺點。進化式機器人一直是機器人學中重要的發展議題，主要是由機器人和機器人所在的環境為兩大因素構成，而進化式機器人目標，是讓機器人能自主發展行為來達到指派的任務，本篇論文採用多種進化式演算法(基因演算法、粒子群演算法、瀾集演算法)，使機器人能產生向前運動的移動行為，並劃分四足型態和三足型態的仿生機器人，針對向前運動的行為作為實驗主要方向。在考慮運動過程中，硬體可能會有誤判或耗損的風險下，以及能控制的範圍內設想機器人在行走中發生部分受損的問題，適應值篩選上必須相對嚴格，起初我們使用機器人行走的距離做為適應值，經過多次實驗後，發現效果不錯後，開始嘗試使用多目標演化做為進化的依據。

在進化式演算法方面，首先使用基因演算法做首要實驗，也套用多種經典的進化式演算法做比較(基因演算法、粒子群演算法、瀾集演算法)，實驗結果顯示，不論是四足機器人或者任意拼接的三足機器人，甚至四足機器人面臨到部分受損情況(例如:馬達故障)，也能使用進化式演算法進行即時的自我調適功能。

**關鍵字：**進化式機器人、仿生機器人、進化式演算法



## Abstract

Traditional robots use the kinetic analysis to control the robot. Kinetic analysis has big problems: it does not sense the environment when the robot works. Evolutionary robots using evolutionary algorithms to solve this. Evolutionary robots have become an interesting topic recently. Specifically, this research focuses on evolution and learning. Evolution is the adaptation of robots to the environment. Learning is a task-oriented process whereby the robot gains the ability to achieve a given goal in the environment. We use many evolutionary algorithms like GA algorithm, PSO (particle swarm optimization), and Memetic algorithm to control the robot. We use biomorphic robots to plan our experiments; biomorphic robots usually have more joints and more legs. However, the more motors on a robot the more risk that one of them might break down at an unexpected moment. Self-adjusting locomotion ability can be a help to make a partially dysfunctional biomorphic robot move. Online evolutionary algorithms are a promising way to achieve such a task. The robot receives feedback from the environment as to the fitness for its evolutionary goal.

We adopt the PSO (particle swarm optimization), GA algorithm, and Memetic algorithm as our online evolutionary algorithm and test it on a partially broken-down quadrupedal biomorphic robot. The experimental results show that the robot can adjust its actions to move even when one leg is removed.

**keyword :** Evolution Robotics, biomorphic robot, evolutionary algorithm.



匆匆三年一晃眼便已逝去，感謝求學路上，一直給予學生我許多幫助與教導的吳世弘老師，在我最需要時期鼓勵我念研究所，更與我們分享人生經驗，並時刻關懷我們。每每老師您打開研究室的門，熱切地跑來和討論關於研究方向、進度以及想法，甚至是我們生活狀況，我們瞭解，老師您不遺餘力地引領我們，只為了讓我們能夠在碩士時期，順利地、心無旁騖地完成學業。對我們來說，不僅作為碩士以及大學時期的指導老師，更是我們一直仿效的目標，憑藉著這路上的追逐，讓我們擁有掌握住未來的實力與勇氣。感謝老師，謝謝老師。同時，也謝謝我去日本三個月期間，殷切指導我的郭老師和 Saji Hameed 老師並給我在日本工讀的機會，使我這位異地學子也能夠感受到在異國的熱心關懷和經濟來源。也感謝我的口試委員莊家峰教授、秦群立教授以及鄭文昌教授，給予我許多寶貴並有用的建議，使我的論文能夠更加完善。碩士時期一直陪伴在身邊的女友和各位學長、同學以及學弟們，因為你們的鼓勵與打氣，才能成就我的論文。感謝學長，君翰；感謝好同學們，議慶、惟倫、立尹、建智、薪昊、郁翔、泰源、泉佑；感謝學弟們，厚諭、善順、士緯。銘記不忘，只為將你們刻劃於人生這段重要的路程，若缺少你們，我的碩士生涯只能稱作散亂。這三年，因為你們而完美，感謝你們。最重要的感謝給我的父母，謝謝你們給予我最大限度的支持與鼓勵，使我不必擔心家裡一切狀況以及金錢問題，



能專心且無後顧之憂地學習、努力。我的成就、我的一切，皆來自於你們，  
感恩你們。

邱國原 謹誌

朝陽科技大學資訊工程系

中華民國一百零二年六月



摘要 .....	I
ABSTRACT .....	II
致謝 .....	III
目錄 .....	V
表目錄 .....	VII
圖目錄 .....	VIII
第一章 緒論.....	1
1.1 研究動機與目的 .....	4
1.2 進化式機器人 .....	4
1.3 實驗環境.....	5
1.4 適應值 .....	5
1.5 機器人的任務 .....	6
第二章 相關研究.....	8
2.1 嵌入進化式演算法的機器人 .....	9
2.2 足型機器人 .....	9
2.3 三足機器人和四足機器人的差異.....	10
2.4 仿生機器人.....	11
2.5 相關硬體技術 .....	12
2.5.1 伺服馬達.....	12
2.5.2 超音波感測器.....	12
2.5.3 三軸加速度器.....	13
第三章 進化式演算法之機器人研究.....	14
3.1 基因演算法 .....	14
3.2 嵌入基因演算法機器人 .....	14
3.3 交叉和突變.....	16
3.4 粒子群演算法 .....	17
3.4.1 粒子群演算法原理.....	18
3.5 改良式粒子群演算法 .....	21
3.5.2 改良式粒子群演算法流程.....	25
3.6 滲集演算法 .....	27
第四章 系統架構與實驗結果.....	28



4.1 實驗一：四足機器人.....	28
4.2 實驗二：三足機器人.....	33
4.3 實驗三：嵌入粒子群演算法機器人.....	35
4.3.1 實驗環境.....	36
4.3.2 使用多種粒子設定實驗.....	37
4.4 實驗四：比較基因演算法和粒子群演算法實驗在四足機器人.....	37
4.4.1 比較基因演算法在受損的四足機器人.....	40
4.4.2 比較基因演算法和粒子群演算法實驗在受損的四足機器人.....	42
4.5 實驗五：比較多種演算法在受損機器人中的差異 .....	44
4.6 實驗六：多目標適應演化 .....	46
4.6.1 新構型機器人和感測器.....	46
4.6.2 使用多目標演化實驗設計以及結果 .....	47
4.6.3 實驗設計以及結果：使用多目標演化 .....	48
<b>五、結論與未來工作.....</b>	<b>55</b>
5.1 結論.....	55
5.2 未來工作.....	56
5.2.1 模糊控制.....	56
5.2.2 類神經網路.....	57
<b>六、參考文獻.....</b>	<b>59</b>





表一 各種任務、進化式演算法、適應值 .....	7
表二 三足機器人每一伺服器馬達可能的角度設置 .....	11
表三 三足機器人每一伺服器馬達實際的角度限制 .....	11
表四 比較基因演算法和粒子群演算法的特性 .....	39
表五 由 200 種基因組合進行基因演算法多目標演化實驗 .....	50
表六 由 200 種基因組合進行濶集演算法多目標演化實驗 .....	50

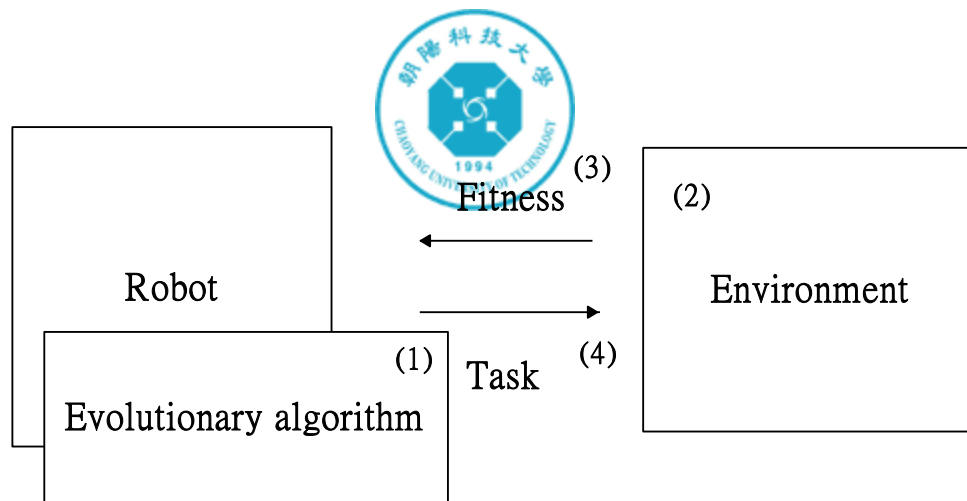


圖一 進化式機器人與環境的架構圖 .....	2
圖二 實驗環境架設圖 .....	7
圖三 (左)三足機器人(右)四足機器人.....	10
圖四 PARALLAX 公司出產超音波感測器 .....	13
圖五 ANALOG 公司生產的三軸加速度器 .....	13
圖六 基因演算法.....	15
圖七 使用十種任意的姿勢來完成一組連續的動作 .....	17
圖八 (A)一個粒子包含三個姿勢(B)五個 粒子包含三個姿勢.....	23
圖九 利用機器人的馬達角度轉換成粒子群演算法的參數 .....	24
圖十 改良式粒子群演算法.....	26
圖十一 彌集演算法.....	27
圖十二 機器人適應值曲線圖 (A)平均適應值 (B)每一次最好的適應值 .....	30
圖十三 四足機器人和制定速度比較圖(A) 人口群設定：5 (B) 人口群設定：4.....	32
圖十四 三足機器人的適應值並且和制定的速度比較圖(A) 人口群設定：5, 迭代數量：8 (B) 人口群設定 5, 迭代數量：40.....	34
圖十五 圖(左)四足機器人和實驗環境，圖(右)模擬四足機器人面臨損壞一足時候的情況 .....	36
圖十六 相同的四足機器人比較圖(A)和圖(B)，基因演算法和粒子群演算法的差異.....	38
圖十七 (A)染色體：2 迭代數量：30(B)染色體：3 迭代數量：20 (C)染色體：4, 迭代數量：15 .....	41
圖十八 相同的三足機器人比較圖(A)和圖(B)，基因演算法和粒子群演算法的差異.....	43
圖十九 比較三足機器人在基因演算法、粒子群演算法、改良式粒子群演算法的差異.....	45
圖二十 (左)新四足仿生機器人,圖(右)四足機器人中的三軸加速度計 .....	47
圖二十一 使用基因演算法進行 200 種不同基因單目標演化實驗 .....	48
圖二十二 使用基因演算法進行 200 種不同基因多目標演化實驗。圖(上) 超音波測距儀：0.9 三軸加速度計：0.1 圖(下) 超音波測距儀：0.8 三軸加速度計：0.2.....	51
圖二十三 使用基因演算法進行 200 種不同基因多目標演化實驗。圖(上) 超音波測距儀：0.7 三軸加速度計：0.3 圖(下) 超音波測距儀：0.6 三軸加速度計：0.4.....	52
圖二十四 使用彌集演算法進行 200 種不同基因多目標演化實驗。圖(上)超音波測距儀：0.9 三軸加速度計：0.1 圖(下) 超音波測距儀：0.8 三軸加速度計：0.2.....	53
圖二十五 使用彌集演算法進行 200 種不同基因多目標演化實驗。圖(上)超音波測距儀：0.7 三軸加速度計：0.3 圖(下) 超音波測距儀：0.6 三軸加速度計：0.4.....	54



進化式機器人長久以來被視為是一種有趣的議題，有學者定義進化式機器人是機器人和所在的環境所構成，且這機器人可以自主的在這環境發展並完成指派的任務，並且在 2003 年由 Dilip Kumar Pratihar 整理詳細的觀念[1]，且進化式機器人可分為進化(Evolution)、學習(learning)兩大方向來加以說明的，進化是指假設機器人的某些行為中，產生出可能會改變機器人的週遭環境的行為，並且這種改變是有跡可循的且緩慢，我們可以稱作此行為叫進化；相對的學習是指假設機器人在進化的過程中產生出適合在這一個任務的行為並且能適應這環境[2]。

因此開始有學者嘗試使用基因演算法或類神經網路來找出進化和學習的相互關係[3]，更高層次的解釋進化式機器人其目的在於使用不同的回饋機制當作進化演算法的適應值，進而完成各種指定任務，本篇論文主要在介紹使用超音波作為距離的回饋來達到進化式機器人的目標，圖一是我們認為進化式機器人各種元件的關係圖。



圖一 進化式機器人與環境的架構圖

(1) 進化式機器人(Evolutionary robot)：

從進化式機器人的角度來探討，由於進化的條件並不是進化式機器人本身，而是必須由環境和進化式機器人共同決定的[1][3]，設計者在這邊的角色就顯得不是很重要，所以我們將進化式機器人拆分成進化式演算法以及機器人兩區域來探討，這種機器人是像生物一樣，擁有自我啟發並且是有組織的生物。我們支持上述的論點認為機器人應該可以是任意建構的生物，並且具備學習功能。不同演算法都有獨特的進化方式，例如基因演算法是由選擇(choose)、交叉(crossover)、突變(mutation)這三種元素任意改變人口群排列組合[4]。

(2) 環境(Environment)：

週遭環境的變數對於機器人的影響是非常大的，當機器人執行特定任務，環境大多在實驗中都不會做任何變動的，因為這會影響整個實驗的進行及結果。



### (3) 適應方程式(Fitness function)：

適應方程式是決定能自主行動的進化式機器人在篩選基礎行為的關鍵，所以必須是要有規劃的，也是一種機器人自我評估的依據，大多適應值源自於機器人上的感測器，例如我們論文是使用進化式機器人每一次的運動行為是經由前頭的距離感測器並經由適應方程式轉換成速度作為適應值回饋的一種機制。

### (4) 機器人所需的任務(Task)：

關於機器人所需的任務的研究已經有超過二十年的歷史，我們相信這是非常成熟且有經驗的技術，根據統計足型機器人步行的學習最為普遍[4]，本篇論文也是採用足型機器人並且對其中的運動方式作為我們設計實驗的目的，實驗環境為總長 82cm 和寬 20cm 的密閉式長方形。

以上總結，圖一中(1)為機器人和被嵌入的進化式演算法所結合的機器人它會對進化式機器人所在的(2)環境造成影響，這影響可能是指派進化式機器人所要完成的(4)指定任務，並且進化式機器人也會根據自身感測器對環境做(3)適應值的自我調整行為，而進化式機器人的另一個重點是進化式機器人和環境是用甚麼方式連接關係的，在本論文使用的方法，是採用進化式演算法。就過去的文獻指出，進化式機器人所採用的機器人大多以模仿生物外型或學習生物的動作來設計的，在過去的十年中，最常仿造的生物為昆蟲類，據分析是因為昆蟲類相較於人型較於穩定[4]，而這類的機器人



通稱仿生機器人，關於仿生機器人(Biomorphic Robot)在形態的設計上，如果使用固定的形態是很難演化至理想的目標，必須朝多變化的形態來設計。

## 1.1 研究動機與目的

據過去的文獻指出，大部分的仿生機器人是由許多關節和足所組成的構型，例如仿造昆蟲步態的機器人大多是六足也有四足型態的仿生機器人，假若機器人越多足[4][5]，相對上所面臨的耗損機率也會相對提高許多，所以我們假設一種特殊構型的三足機器人，和原是四足機器人在行走時面臨到斷腿的情形中，該如何使用進化式演算法使行為較為平順許多，可以順利的達到目標是我們所關心的議題，所以根據這想法，為了使機器人能夠順利的達成目標，我們嘗試不同的進化式演算法來完成。

## 1.2 進化式機器人

假設從進化式機器人的角度來探討此議題，在先前的章節提到，進化的條件並不是機器人本身而是必須由環境和機器人共同決定，設計動作者在這邊的角色就顯得不是很重要，所以我們將進化式機器人拆分成兩區域來探討，機器人是像生物一樣，擁有自我啟發並且是有組織的生物。根據上述的論點，認為機器人應該可以是能任意拆解的個體，而這類型的機器人大多是有許多足部和關節所構成，例如六足部或四足部甚至更多足部，擁有許多足部是可以藉此克服各種障礙[6]，但相對的在控制以及移動上比輪



型機器人來難控制，又例如：大多輪型機器人並不用關心平衡的問題，但對於足型機器人是嚴重的問題，尤其在套用進化式演算法的機器人上更為明顯，但相對的從進化學的角度來看基礎的進化式演算法，例如：基因演算法使用選擇、交叉、突變這三種元素來任意改變演算法中的人口數的組合並且使用循環演化，而粒子群演算法則是強調單一粒子視為動物群體中的單一個體，並且賦予這個體是擁有記憶性的，所以粒子之間會有互動來尋找最佳解，(如：具循環式的基因演算法)，是更能加快實驗上的收斂時間以及降低硬體上的成本消耗[7]。

### 1.3 實驗環境

週遭環境的變數對於機器人的影響是非常大的，過去文獻顯示，環境大多在實驗中都不會做任何變動的，但各種實驗上的設計會因為環境的複雜度不同加深實驗的難易度以及成本的消耗(例如：實驗中若放置存在太多的障礙物這會影響整個實驗失敗率)，所以要提高在複雜實驗環境中的成功率，文獻指出單使用進化式演算法是不太夠的[7][8]，比需搭配一些能使進化式演算法的結果更為平滑(例如：模糊控制理論、類神經控制)。

### 1.4 適應值

適應值是決定能自主行動的進化式機器人在篩選基礎行為的關鍵，所以必須是要有規劃的，我們論文是使用進化式機器人每一次行為的速度作適應值的依據，由此表一能看出，適應值是會依據機器人的設計以及任務的





不同要有特別的調整和改變，也會隨著賦予的任務不同而有不同的適應值上的定義，根據過去的文獻顯示，這類的改變通常會依照起始和終點的設計來調整[8]，例如本論文舉例的機器人運動方式，就考量機器人在一定的距離中每一次所走的距離和運動的瞬間速度，做機器人對於這環境的適應值調整依據如下公式(1)(2)，在公式(1)中， $D_{(n)}$ 為目前最近障礙物到機器人之間的距離， $t_{(n)}$ 為機器人目前移動距離的所需時間， $n$ 為目前實驗的次數，例如本篇論文的實驗設定在 $n$ 為10的情況，而在公式(2)中，適應值則是根據 $S_{(n)}$ 當下位置的加速度來計算出的和前一位置的加速度 $S_{(0)}$ 算出。

$$S_{(n)} = \frac{D_{(n)}}{\Delta t_{(n)}} \quad (1)$$

$$Fitness = S_{(n)} - S_{(0)} \quad (2)$$

## 1.5 機器人的任務

機器人的任務，是指機器人在面對不同的問題時，應當產生解決問題的反應，這些任務大多伴隨著較為惡劣的環境，所以機器人若想完成任務，必定要有偵查周遭環境的能力和技術。根據統計，仿生機器人以足型機器人構型佔大多數，一般為步伐的學習最普遍，本文採用足型機器人構型，並針對機器人向前運動行為做實驗的任務。以前章節為例，在環境為總長82公分寬20公分的長方形中，如圖二所示，機器人因為受限於狹窄的長條環境中，機器人能藉由左右牆壁來修正本身運動方向，優點是將實驗環境

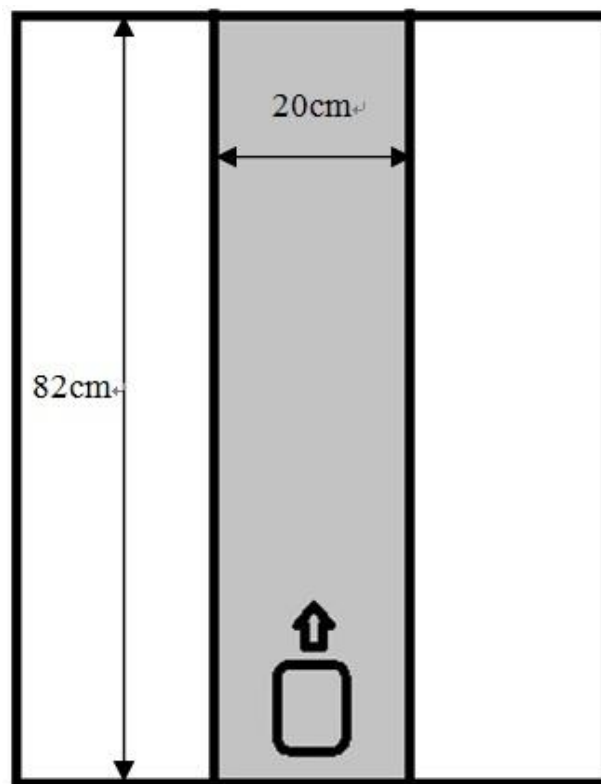




單純化，藉此降低實驗中硬體之間的耗損、成本和容錯率。

表一 各種任務、進化式演算法、適應值

Task evolved/Learned	Evolved controller type/Algorithm	Fitness
Object pushing	Neural network	$f = \text{mean}(v_l, v_r)(1 - S_{ir})$
Gait evolution	Evolvable state lookup tables	$f = \text{mean}(v_l, v_r)(1 - (v_l, v_r)^2)(1 - S_{ir})$
Locomotion	Neural network	$f = v$
Locomotion with object avoidance	Evolvable sensor-to-motor excitation mapping	$f = d$
Flying lift generation	Genetic programming	$f = d_{\max}^2 - d^2$
Locomotion with wall avoidance	Neural network	$f = v$
Gait evolution	Gait parameter set	$f = d / t$



圖二 實驗環境架設圖



## 第二章 相關研究

在過去文獻指出機器人的行為，可能被所制定的任務和環境所限制住，但是我們認為這是不合理的，因為就以生物學角度來看，物種是會因為周遭的環境變化而改變生活的方式，所以適者生存的論點是否可以應用在機器人的呢？就以前面的章節所提到，仿生機器人在當初設計目的就是仿造生物的行為所產生的，自然而然應當能適應多變的環境，然而研究生物的行為並不是本篇論文所要探討的目的及動機，反而在行為的研究著重於進化的議題上[6][9]，舉例來說，從人類學習行走的議題來看，剛出生的嬰兒是以四支爬行來前進的，經由不斷的學習及外在環境的因素，逐漸的使用兩隻腳走路，從這看似平常的生活問題，其關鍵的地方在於人這生物一直都是四隻腳的，也就是說人是可爬行或是兩隻腳行走的，並且因應環境的改變而轉換行走方式，為什麼人會知道呢？因為這是經過長時間的學習得下來的結論，我們認為這也可以相對的應用在仿生機器人的問題上，假設我們將設計機器人組裝成從來沒有看的機器人如三隻腳或是在行走中其中一組馬達壞掉，是否可以重新評估環境自我演化新的動作來達到目的，這將是我們要在後面要探討的議題。



## 2.1 嵌入進化式演算法的機器人

進化式演算法是模仿生物演化行為的演算法，其特點在於保留好的基因並且以適當的機率進行交叉和突變，我們遵循這精神，過去自主式機器人發展在進化演算法已經不是特別的例子，但仍然有受到許多人關注的，也相對有許多人將這議題實現並模擬出真實例子[10][11][12]，成功的展現出這問題的可行性，這一章節首先我們要討論的是如何亂數建構行為以便達成基因演算法初步的流程。我使用排列組合的方式產生一個未知的行為，我們將此行為視同初代基因，搭配一個行為回饋的距離和完成的時間，做為判定好或不好基因的依據。

## 2.2 足型機器人

各式各樣的機器人都會配備不同所需的感測器，而這些感測器有許多不同的種類例如：超音波、紅外線、雷射測距儀、然而從進化式機器人的角度來看最重要的事情再於是否能給予正確的回饋機制，有了這些回饋機制就能賦予機器人一種類似生物的生命力以及移動能力，並且可以完成使用者所託付的不同任務，例如：避開障礙物、夾取目標物[13]。足型機器人種類繁多，本篇論文所採用的伺服器系列的機器人為一種自由度高且易組裝的機器人，並且採用串接式伺服器馬達也是其特色之一，串接式伺服器馬達跟傳統的伺服器馬達差別在於訊號封包的傳遞上不同，傳統伺服器馬達只能一顆接收一訊號至控制器，然而串接式伺服器馬達能將多顆伺服器馬



達以串聯方式接起在統一接收主控制器的訊號，優點在於可以一次控制多顆已經串接的伺服器馬達，這對於在設計機器人是好處原因是可以比以往還有更高的自由度如圖三為本篇論文所採用的足型機器人之一。



圖三 (左)三足機器人(右)四足機器人

### 2.3 三足機器人和四足機器人的差異

不同的任務會因為實驗中適應值的需求而不同，而這些需求是來自於感測器的不同，所以本篇論文比較是機器人之間的結構不同，特別針對三足機器人以及四足機器人間的差異，我們可以知道三足機器人是比四足機器人少一足，並且由上一章節得知我們關心的平衡問題，相較不穩定許多。圖三所示，並且使用循環的模式來設計三足式機器人可能的運動模式，我們設計每一顆伺服器馬達是有 0 至 255 角度限制如所示，其中為了避免機器人機器人在運動模式中面臨受損的問題，將部分三顆伺服器馬達角度分別根據多次實驗結果定義安全的範圍如表二和表三所示。



表二 三足機器人每一伺服器馬達可能的角度設置

$m_0$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
0-255	0-255	0-255	0-255	0-255	0-255

表三 三足機器人每一伺服器馬達實際的角度限制

$m_0$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
190-215	72-91	86-91	0-255	0-255	0-255

## 2.4 仿生機器人

仿生機器人主要是用模仿自然界中的動物和昆蟲的行為以及動作，而仿生機器人的特色在於機器人本身的構型以及行為結果都是根據自身的感測或環境的變動所演化出來[12][13]，而這一連串的進化行為主要是為了發展出不同的機械構型或動作，進而完成各種時期的目標(例如：到達目的地)，大部分的仿生機器人不會被構型有所限制自身的行為，但是仿生機器人若是搭載過多的伺服器馬達，會使機器人本身變的難以控制，且硬體的損壞率也會相對的提高許多，像是三足的機器人就需要額外的動力分析以及硬體的設計，所以這類型的機器人，在做每一次動作前都必須要有較為精準的動作分析(例如：規範伺服器馬達的角度、構型)，也就因為仿生機器人的宗旨在於模仿自然界的動物和昆蟲各種行為，就以生物的觀點來看，我們認為設計機器人以及機器人應該使用各種不同功能的感測器來感知周遭環境的能力[12][13][14]，所以仿生機器人不應該被本身的構型而受限了自



身的行為，近幾年來發展了新的進化模式的仿生機器人[15][16]，這類型的機器人是多目標進化為主要的精神。

## 2.5 相關硬體技術

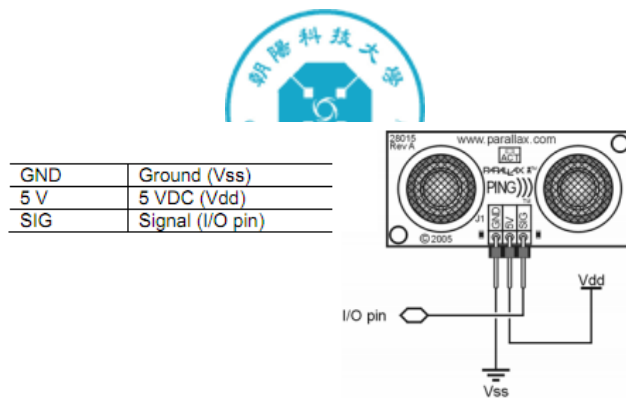
在這章節中，我們將列出使用到的硬體技術和規格，分別有串接式伺服器馬達、超音波測距感測器、三軸加速度計。

### 2.5.1 伺服器馬達

交流伺服器馬達是針對高精度位置控制之應用設計，位置、速度、轉矩、定位、通訊控制..應用領域多軸伺服器、定位、切割等應用，本文採用的馬達是屬於串列通訊控制，所以是屬於同步型伺服器馬達，它的伺服機 (WCK) 具有兩種模式切換，一種是角度控制模式 (一般伺服機的功式，另一種是連續旋轉 (DC 馬達) 的功能這樣的優點可以使你隨意產生關節或是輪胎的動作。

### 2.5.2 超音波感測器

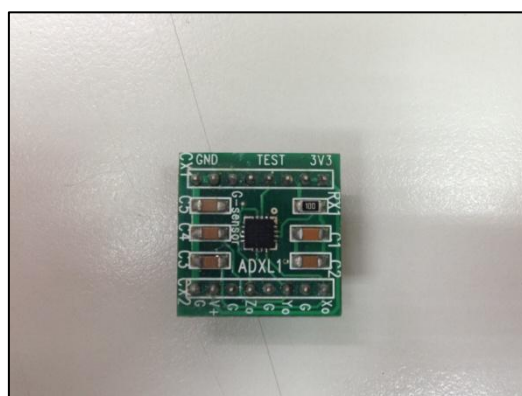
本文採用的超音波感測器為 PARALLAX 公司，所出產的 4 Pack of Pings，它有 3 公尺左右的最大測距範圍且擁有低耗電量的優點如圖四。



圖四 PARALLAX 公司出產超音波感測器

### 2.5.3 三軸加速度器

加速度計顧名思義，是一種測量加速度的感測器，特別適用於測量重力也稱為重力計，大致上能依照單軸、雙軸、三軸作為不同應用的需求，例如：本篇論文使用三軸加速度器量測機器人水平值的工具，如圖五所示，我們擷取每一次(X 軸、Y 軸、Z 軸)的起始資訊和下一目標點的三軸資訊取差值，圖五中使用的三軸加速度共能測得 3g 的加速度值，並且使用  $\text{m/s}^2$  作為穩定度的單位。



圖五 ANALOG 公司生產的三軸加速度器





## 第三章 進化式演算法之機器人研究

### 3.1 基因演算法

基因演算法是進化式演算法中複雜度較為低的演算法之一，源自於人類遺傳學的概念，並且將遺傳、突變、自然選擇的因素加入其中，如圖六所示。也有遺傳式演算法的別稱並廣泛運用在生物學、工程學、製造等等許多領域中，自 2000 年開始就有許多文獻開始使用進化式演算法控制機器人的運動模式，在基因演算法理，被優化的問題稱為個體，它是一種由變數組合成的數列集合[17][18]，這是經由實驗的需求自由訂定的，俗稱染色體或基因，染色體的大小將影響實驗的複雜度和需要花費的成本，另外大多進化式演算法中，每一個體都需要配評價，並通過適應度函數得到每一次的適應值數值，並且每一次的染色體經過一定機率的遺傳、突變、自然選擇來改變下一代的個體，例如圖七中三足仿生機器人共有六顆伺服器馬達，我們將每一次六顆伺服器馬達所產生的角度視為染色體，並設定 10 次重複不同動作來完成一組運動行為。

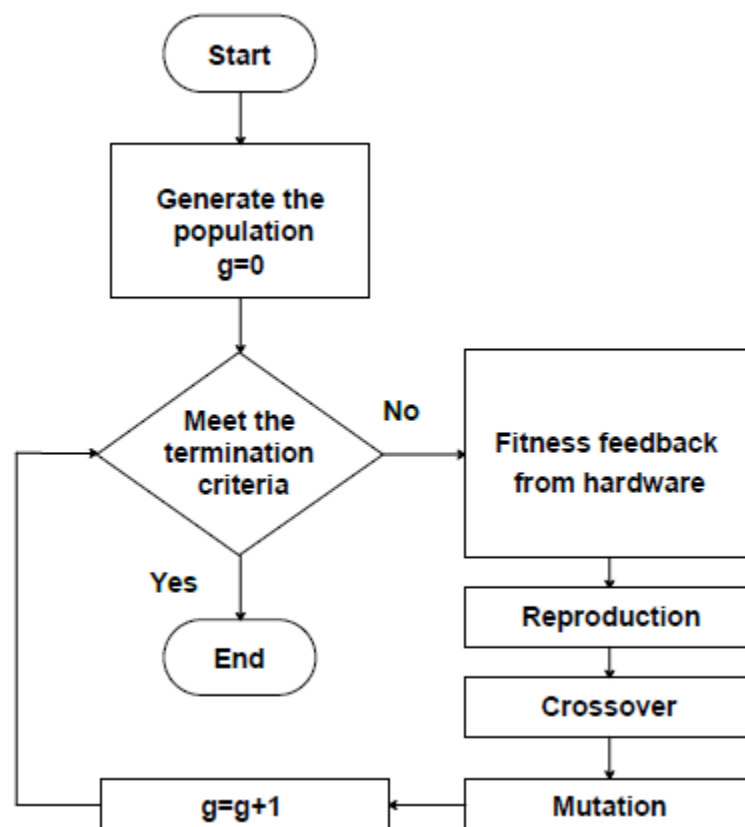
### 3.2 嵌入基因演算法機器人

實驗中我們將使用標準基因演算法如圖六所示，但是跟傳統的基因演算法不同的是，在適應值的來源並非是用數學推導出的結果，而是來自機器人本身的超音波感測器得到距離資訊來控制機器人的運動行為，但適應值





的設計因硬體關係接收的資訊並非是理想中的距離，因為實驗環境的限制如圖二所示，經過多次實驗後小於 30 公分至大於 -20 公分的範圍中，對於在這實驗環境中，是最好的容錯範圍，若超出這範圍的資訊將一律視為實驗失敗並且重新開始，而我首先我們先取出一定時間內機器人直線運動的速度，在理想的狀況下機器人應該是要不撞到任何障礙物下完成一組實驗，由於回饋機制方面是單方向，導致機器人會常有迷失方向的情形發生，經過多次的實驗。

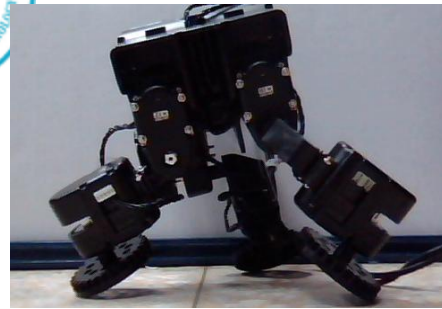
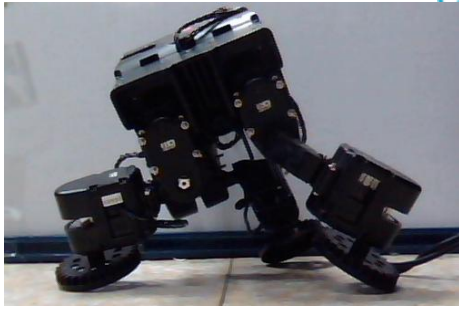


圖六 基因演算法



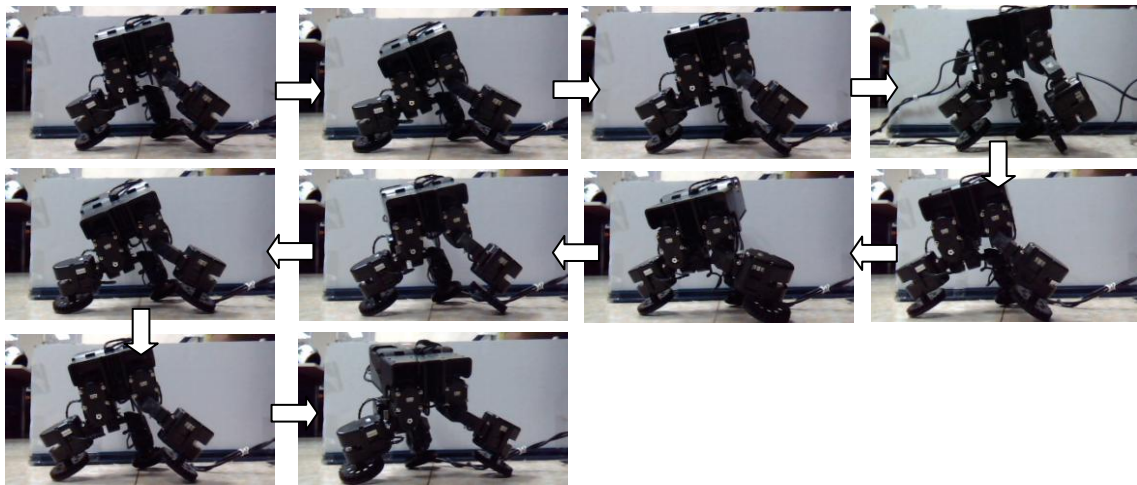
### 3.3 交叉和突變

在我們的實驗設計中，我們將基因演算法中交叉的步驟設定在，每一個染色體的中心點為交換的來產生下一代，這樣能確保下一代跟父母是不一樣，由於每一代的長度是根據機器人伺服器馬達的數量來定義的，假設實驗設計每一人口數有 10 組伺服器馬達角度並每一組由 8bits 大小的數字組合成且總共有 6 顆伺服器馬達(三足機器人)，這樣就能推論交換機率： $8/(48 \times 10) = 8/480 = 0.0167$ ，相對的，若是 8 顆伺服器馬達(四足機器人)，交換機率為  $8/(64 \times 10) = 8/640 = 0.0125$ 。



$m_0$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
190	91	86	15	13	15

$m_0$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
213	91	86	15	15	15



圖七 使用十種任意的姿勢來完成一組連續的動作

### 3.4 粒子群演算法

粒子群優化(Particle Swarm Optimization, PSO)，又稱微粒子群演算法，是由 J. Kennedy 和 R. C. Eberhart 等人於 1995 年開發的一種演化計算技術 [19]，來源於對一個簡化社會模型的模擬，其中「群(swarm)」概念來源於大自然中群體覓食得到啟發，這演算法核心「粒子(particle)」是一個關鍵的選擇，因為 J. Kennedy 認為每一個粒子是具有人工生命且具有方向性和記憶性的，演算法性質類似於(基因演算法、蟻群演算法)[20]，並且強調成員



描述為沒有質量、沒有體積的，但同時也需要描述它的速度和加速狀態[19][21]。

粒子群優化演算法最初是為了圖形化的模擬鳥群優美而不可預測的運動，並且適用於各種求最佳化的問題[22]，模擬動物社會行為的觀察每個粒子是具備獨立搜尋功能，能依照此個體最佳適應值記憶去修正下一次的搜尋方向進而改善本身的適應程度，並發現在群體中對信息的社會共享提供一個演化的優勢作為開發演算法的基礎，透過加入近鄰的速度、向量並考慮了多維搜索和根據距離的加速，形成了粒子群優化的最初版本之後引入了慣性權重  $w$  來更好的控制開發和探索，形成了標準版本[19][21][22]。

### 3.4.1 粒子群演算法原理

PSO 演算法是是一種具有群體智慧的方法，其一開始初始化產生一群隨機的粒子，並根據對環境的適應度將群體中的個體朝向好的區域學習[23]，然而它不對個體做演化，而是將每個個體看作是  $D$  維搜索空間中的一個沒有體積的微粒（點），在搜索空間中以一定的移動速度和位置，這個速度根據它本身的飛行經驗和同伴的移動速度的經驗來動態調整。第  $i$  個微粒表示為  $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ ，每一粒子經歷過的最好位置（有最好的適應值）記為  $P_i = (p_{i1}, p_{i2}, \dots, p_{id})$ ，也稱為  $pbest$ ，在群體中會有一個最接近目標的最佳並且所有微粒經歷過的最好位置的索引號用符號  $g$  表示稱為  $gbest$ ，微粒  $i$



的速度用  $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$  表示。對每一代，它的第  $d$  維 ( $1 \leq d \leq D$ ) 根據如下方程進行變化[19][22][23]：

$$V_{id}(t) = V_{id}(t-1) + c_1 \times \text{Rand}() \times (P_{id} - X_{id}) + c_2 \text{Rand}() \times (P_{gd} - X_{id})$$

$$X_{id}(t) = X_{id}(t-1) + V_{id}(t)$$

其中  $w$  為慣性權重， $c_1$  和  $c_2$  為加速常數， $\text{Rand}()$  為兩個在  $[0,1]$  範圍里變化的隨機值。

此外，微粒的速度  $V_i$  通常被一個最大速度  $V_{\max}$  所限制，如果當前對微粒的加速導致它的在某維的速度  $v_{id}$  超過該維的最大速度  $v_{\max,d}$ ，則該維的速度被限制為該維最大速度  $v_{\max,d}$ ，以減少計算成本和實驗所需時間，演算法中細分為三部分：第一部分為粒子先前行為的慣性；第二部分為「認知」部分，表示微粒本身的思考；第三部分為「社會」部分，表示微粒間的信息共享與相互合作，另外過去的研究實驗中會加入權重變素  $w$ ，通常被設置由 0.9 線性遞減至 0.4，權重主要用於找尋合適的數值設定並能可以提供局部和全域一個平衡的探索及開發能力，較大的  $w$  可使粒子具備較大的開發能力，較小的  $w$  能使粒子具備探索能力，另外  $w$  的加入也使得演算法減低了每回合需小心設定  $V_{\max}$  的需求，



「認知」部分可以由 Thorndike 的效應法則(law of effect)所解釋，即一個得到加強的隨機行為在將來更有可能出現。這裡的行為即「認知」，並假設獲得正確的知識是得到加強的，這樣的一個模型假定微粒被激勵著去減小誤差。

「社會」部分可以由 Bandura 的替代強化(vicarious reinforcement)所解釋。根據該理論的預期，當觀察者觀察到一個模型在加強某一行為時，將增加它實行該行為的幾率。即微粒本身的認知將被其它微粒所模仿。

粒子群演算法使用如下心理學假設：在尋求一致的認知過程中，個體往往記住自身的信念，並同時考慮夥伴間記憶和想法，當其察覺同事的信念較好的時候，將進行適應性地調整[23]。

所以標準粒子群演算法流程如下：

1. 初始化一群微粒（群體規模為  $m$ ），包括隨機的位置和速度。
2. 評價每個微粒的適應度。
3. 對每個微粒，將它的適應值和它經歷過的最好位置  $pbest$  的作比較，  
如果較好，則將其作為當前的最好位置  $pbest$ 。
4. 對每個微粒，將它的適應值和全局所經歷最好位置  $gbest$  的作比較，  
如果較好，則重新設置  $gbest$  的索引號。
5. 根據上面公式變化微粒的速度和位置。





6. 如未達到結束條件 (通常為足夠好的適應值或達到一個預設最大代數  $G_{\max}$ )，回到 2)。

### 3.5 改良式粒子群演算法

前面章節介紹粒子群演算法的流程以及細節，過去文獻指出，進化式演算法經過各種適當的改良是可以達到比經典的演算法效果來的好，例如由圖八和圖九中，我們設計一種計算量較低並且更適合我們機器人的一種改良式粒子群演算法，經過實驗後發現能大幅度的降低實驗上的時間消耗，因此在這章節我們將以這做為範例，並解釋如何將這演算法控制機器人的運動行為，機器人的粒子定義是根據如圖八是呈現一個循環動作的模式，圖八舉出兩個例子：分別是 1 個粒子中包含 3 種不同的機器人動作以及 1 個粒子中包含 5 種不同的動作，這是可以依照實驗的需求做調整的，接下來在每一次的動作中，根據公式(3)以及圖九可以看出我們將機器人身上每一顆伺服器馬達給予不同的角度，這些角度是一個 8bits 大小的浮動變數，並給予一個陣列做為儲存用如公式(3)所示，為了達到降低計算量，我們給予每一個動作額外三個變數做為接下來適應值的評估以及計算(分別是： $x, y, v$ )，以利降低計算的時間以及硬體的消耗如公式(4)所示，另外在適應值方程式採用超音波感測器所測得的距離資訊轉換成每一次動作的速度來做動作的篩選和演化如公式(5-6)，並且一併儲存在公式(4)的陣列中。

$$p_i = [m_0 \quad m_1 \quad \cdots \quad m_{n-1}] \quad (3)$$



每一個粒子中包含公式(4)中的所有變數以及大小為1的機器人動作。

$$P_j = [x_i \quad y_i \quad v_i \quad f_i \quad p_i] \quad i = 1 \dots l \quad (4)$$

經由超音波感測器測得的距離資訊依據公式(5)轉換成速度並對應  $j$  大小的粒子適應值。

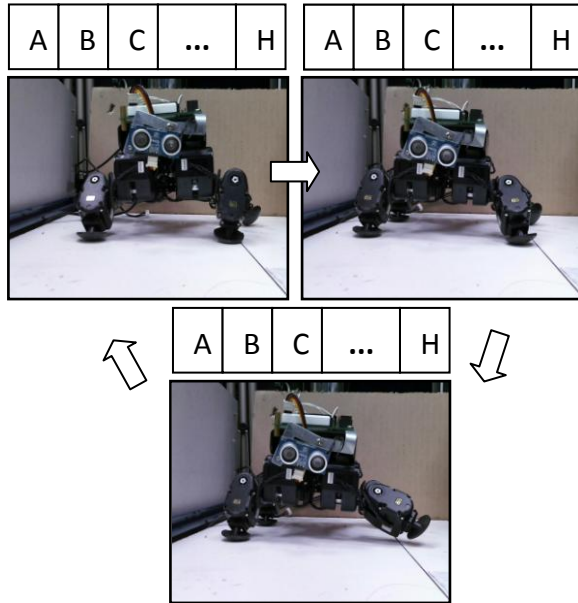
$$s_j = d_j / t_j \quad (5)$$

公式(5)中  $d_j$  代表機器人目前移動的距離另外  $t_j$  代表目前的時間(單位：秒)。

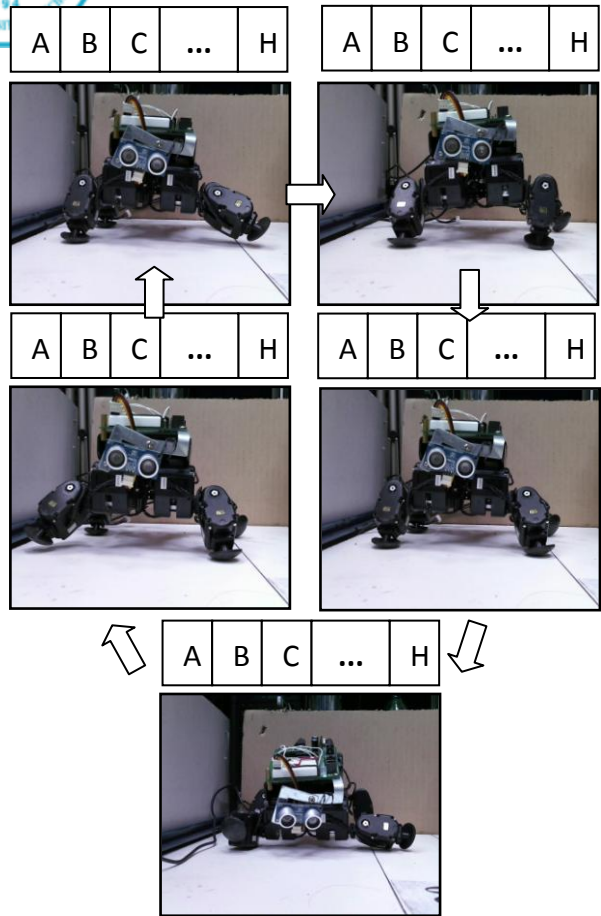
$$f_j = s_j - s_0 \quad (6)$$

在這邊  $s_0$  代表初始速度。



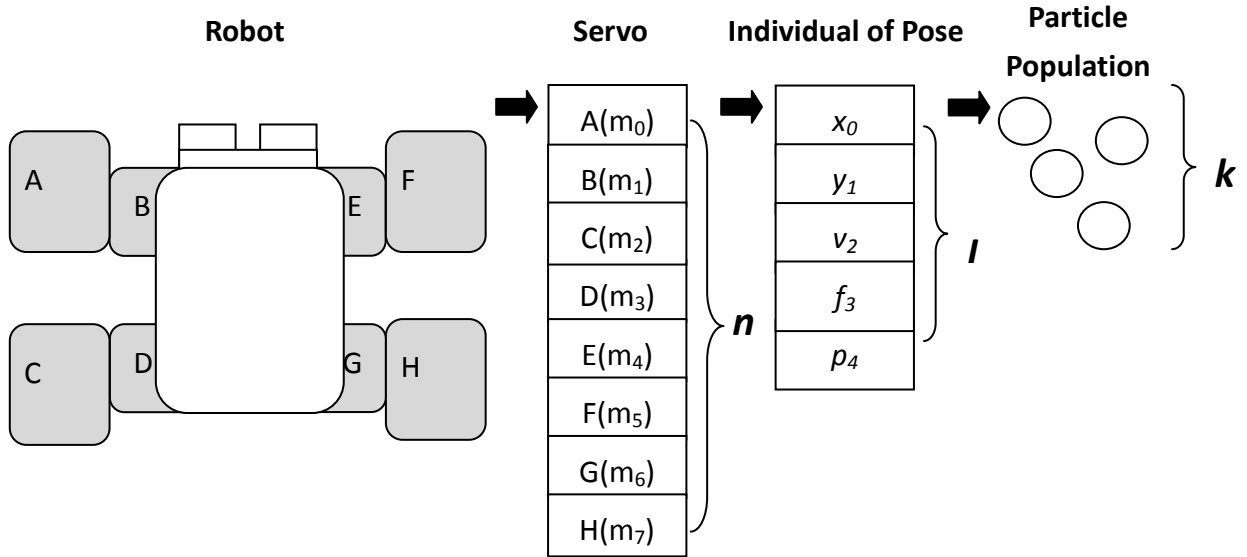


**(a)** 3 poses in one particle



**(b)** 5 poses in one particle

圖八 (a)一個粒子包含三個姿勢(b)五個 粒子包含三個姿勢



圖九 利用機器人的馬達角度轉換成粒子群演算法的參數

### 3.5.1 改良式粒子群演算法演化

粒子群演算法中最好的粒子稱為  $g\_best$  而在一群粒子中曾出現最高適應值的稱為  $p\_best$ ，所以根據  $g\_best$  和  $p\_best$  兩者重要的指標，能藉此衍生出新一代的粒子，若每一次粒子內的數值更新都必須仰賴  $g\_best$  和  $p\_best$  來計算出差值，但改良式的粒子群法除了參考這兩因素外，根據粒子來加入了其他假想的變動因素來降低計算的時間，例如延續前面章節的設定將粒子群演算法中最好的粒子稱為  $g\_best$  而在一群粒子中曾出現最高適應值的稱為  $p\_best$  並假想每一粒子存在於二維的空間中，有各自的座標  $(x, y)$  和向量  $v$  因為加入座標的概念，所以使用公式(7-8)所示更新下一位置，其中  $r1$  和  $r2$  定義為 0 至 1 之間隨機產生的數值。

$$x_j(t+1) = r1((p\_best)x_j(t) - x_j(t)) + r2\left(\left(g\_best)x_j(t) - x_j(t)\right)\right) \quad (7)$$



$$y_j(t+1) = r1((p\_best)y_j(t) - y_j(t)) + r2\left(\left(g\_best\right)y_j(t) - y_j(t)\right) \quad (8)$$

每一次的演化，除了跟新座標外也同時更新向量  $v$ ，最後每一顆粒子中伺服器馬達的角度依據  $v_j$  的變動量，產生下一時間馬達角度的更新如公式(9-10)。

$$v_j(t+1) = \frac{x_j(t+1) + y_j(t+1)}{2} + v_j(t) \quad (9)$$

$$m_i(t+1) = m_i(t) + v_j(t+1) \quad (10)$$

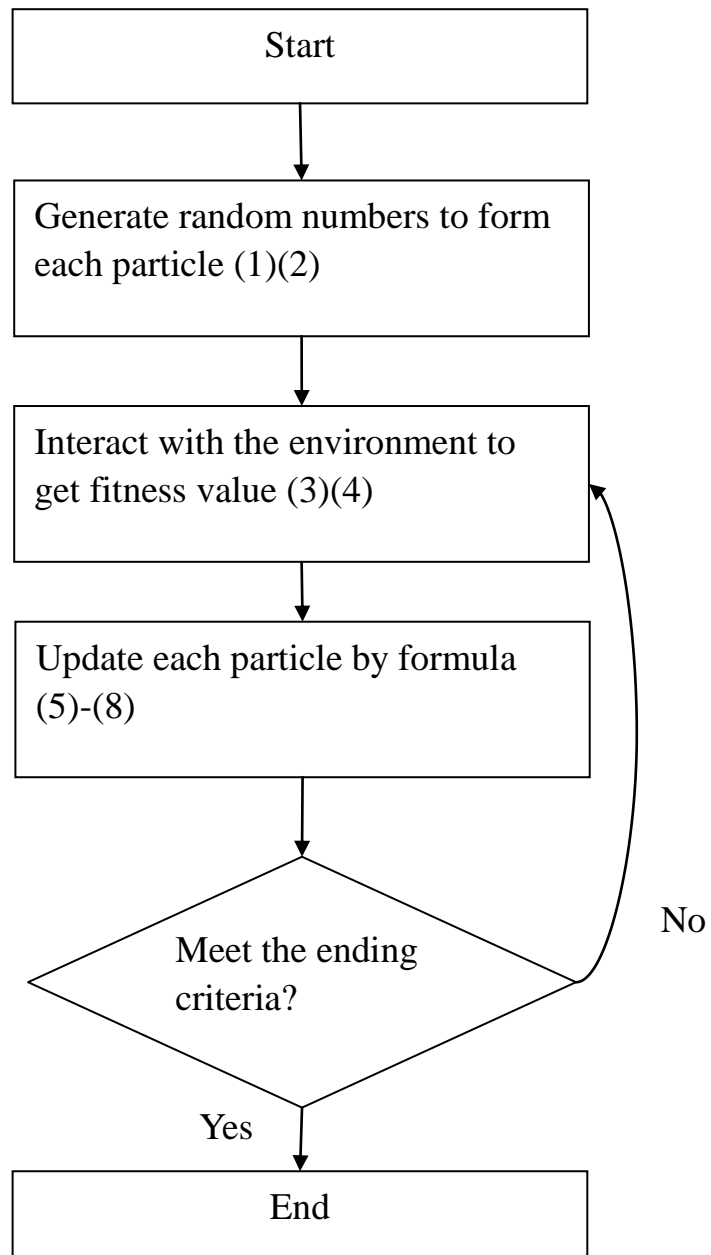
### 3.5.2 改良式粒子群演算法流程

上一章節我們解釋改良式粒子群演算法，這章節我們將展示改良式粒子群演算法的細部流程，演算法將詳細根據這演算法的細部流程並遵照以下 4 步驟完成：

- 1) 首先產生以每一顆伺服器馬達的角度( $m_1, \dots, m_n$ )並存在陣列  $p_i$  大小的粒子群若干個。
- 2) 每一次的動作根據適應值公式(3-4)將超音波感測器的測得機器人目前行走的距離，做速度的轉換，在這邊  $s_j$  代表速度：單位(公分/秒)， $d_j$  行走距離：單位(公分) $t_j$  代表花費的時間。
- 3) 更新每一顆粒子的參數：根據公式(5)-(8)更新每一顆粒子  $P_j$ ，並且同時找尋目前的  $p\_best$  和  $g\_best$ 。
- 4) 假設遇到結束條件將結束整個演算法，否則跳回步驟(2)直到條件滿足



流程圖如圖十所示。

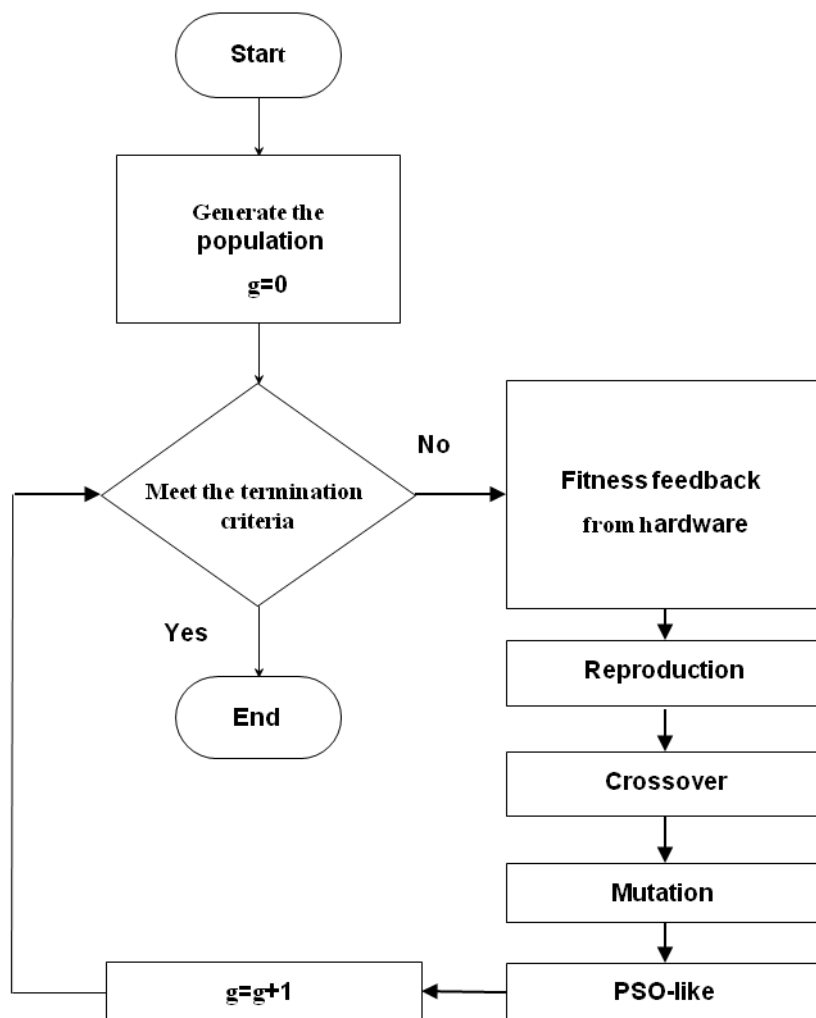


圖十 改良式粒子群演算法



### 3.6 瀾集演算法

瀾集演算法近年在進化式演算法領域中，逐漸被重視的進化式演算法之一，瀾集演算法的精神，是廣泛使用共同進化來達成演化目的[24]，例如圖十一中我們在基因演算法進行第一次演化後，加上粒子群演算法作為類似瀾集演算法的實驗流程圖。



圖十一 瀾集演算法



## 第四章 系統架構與實驗結果

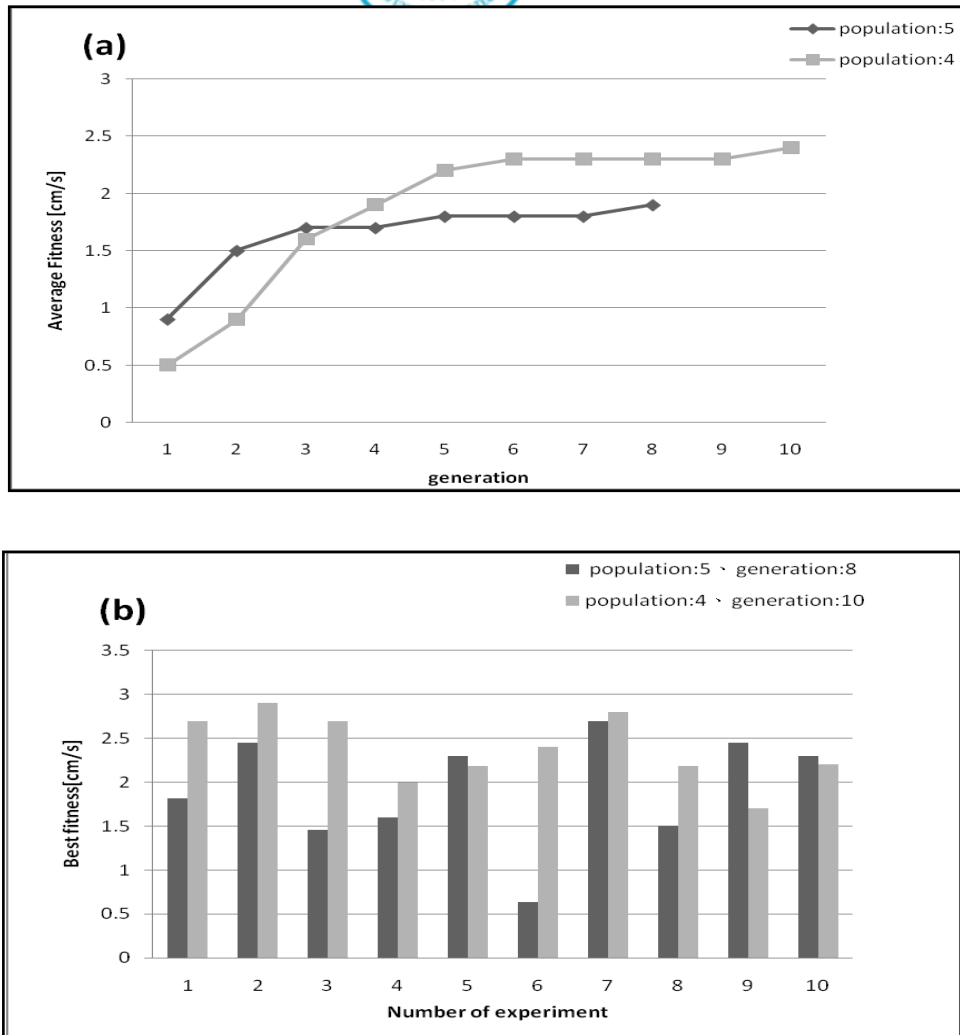
本篇論文將進化式機器人分為兩大方向實驗，並且嵌入不同的進化式演算法來達到機器人運動的是否能安全的達到目的為首要目標，在實驗環境我們如圖二所示，將機器人受限於固定的環境中，這樣是能加快實驗的時間以及耗損，我們使用進化演算法中的基因演算法來完成我們的實驗流程，基本上，不論實驗中使用哪種進化式演算法，首先我們亂數產生初代動作，原因是伺服器馬達的角度在隨機產生的，但機器人是有很高機率迷失前進方向的，若太靠近牆就會造成取抓取到錯誤的值，這時解決的方式是用人去觀察機器人，如果在實驗裡一直有撞牆的行為就將此實驗視同為失敗的實驗。

### 4.1 實驗一：四足機器人

我們根據圖二作為實驗環境以及圖三中的四足機器人的基本構型，機器人是有一個密封的長方型環境下去觀察機器人的直線移動行為，並且針對在定量下的人口數和迭代次數做為並調整其百分比所設計的實驗，首先我們使用圖二中右邊的四足機器人，並且使用預設的行走模式，測得平均速度為每秒 2 公分，我們將這適應值假設四足機器人預設行走模式，這適應值是不會被任何人口數和迭代次數的增加或減少所變動，這也是我們希望



機器人在接下來的實驗過程中能達到的目標，而在初步比較，假設在不考慮迭代的次數，人口數分別設定在 4 和 5 是否能達成我們預設的目標，由圖十二 (a)中能看出使用基因演算法是能達到將近每秒 2.5 的速度，根據初步的結論，我們將重複同樣的實驗 10 次，並且取平均如圖十二(b)所示，在圖十二(b)我們 x 軸為實驗的次數，y 軸為每一次實驗中最好的適應值，並且我們設計一種成本計算標準(人口數和迭代次數的乘積)，在接下來的我們將統稱為成本，而這種成本是將有助於人口數和迭代數比例上的調整。



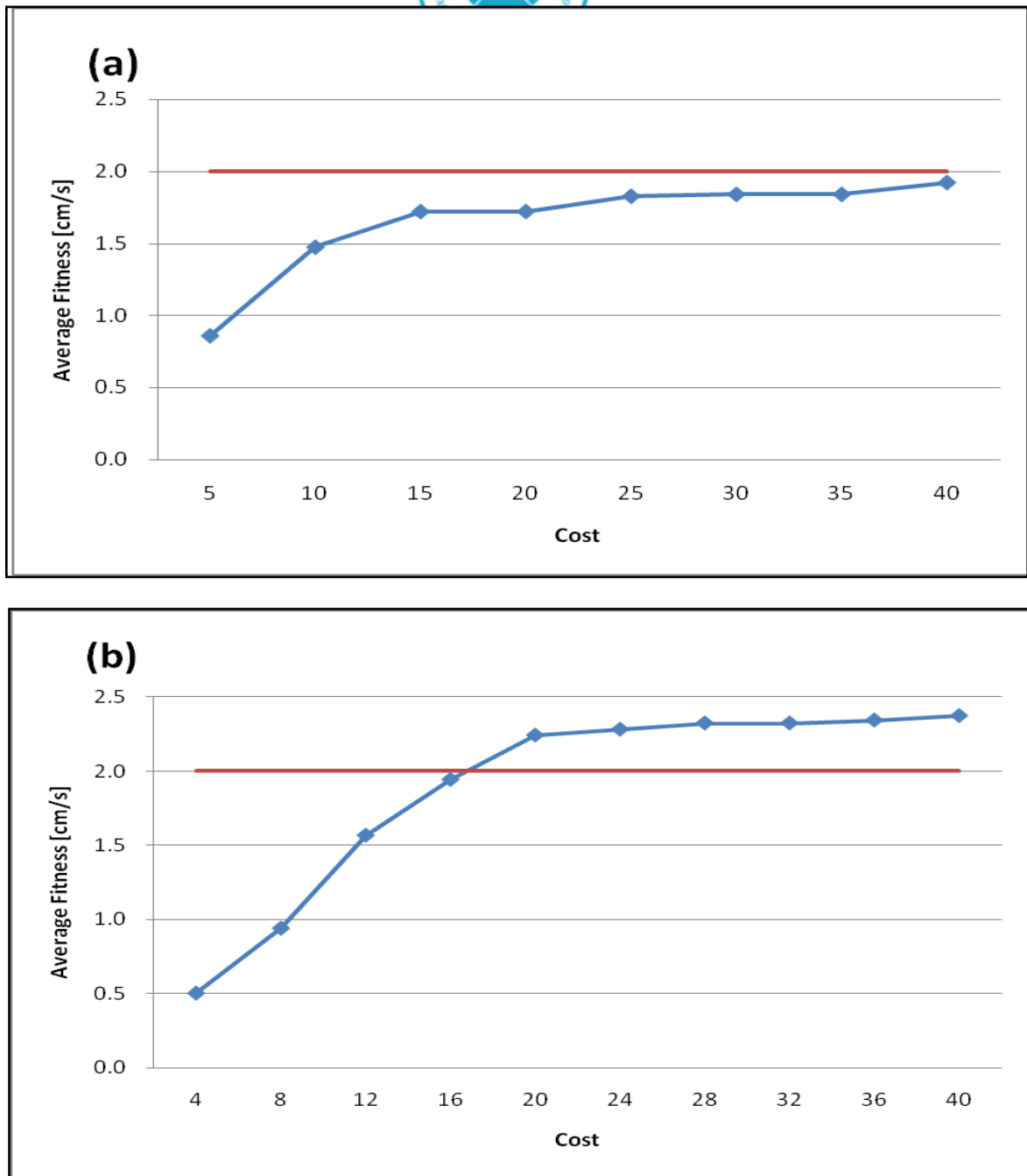
圖十二 機器人適應值曲線圖 (a)平均適應值 (b)每一次最好的適應值

根據上面的設定，在實驗一中將以人口數和迭代次數的乘積為 40 代圖十四(a)和(b)，並為了使適應值更早看出收斂的效果，將交叉率調整為 1 突變機率為 0.1，這樣可以加快實驗速度以及減少硬體的損耗，由圖十三 (a)和(b)分別是根據圖十二(b)的方法各取 10 次圖實驗流程，並取將各適應值平均，圖十三(a) 人口數：5、迭代次數：8，圖十三(b) 人口數：4、迭代次數：10，如圖可以看出在一定個消耗成本中，兩種適應值設定都是呈現線性成





長且收斂的位置也都大約在 30 代右，但圖十三(a)明顯效果沒有超過預期的適應值，相對的圖十三(b)在 16 代時就有明顯的超過，所以我們可以經由這實驗得到初步的假設及結論，人口數和迭代次數的比例會影響適應值收斂的值，但對於收斂的時間並沒有明顯的影響，接下來我們分析每一次實驗的適應值的成長幅度如圖十二模式呈現，我們發現圖十三(a)中的實驗分項 2、實驗分項 4 和實驗分項 10 的適應值是沒有任何成長的相較於圖十三(b)來說每一次實驗都有明顯的成長，依據這一次實驗結果我們無法知道人口數在多少時會影響收斂的時間，實驗結果是有達到預期的效果。

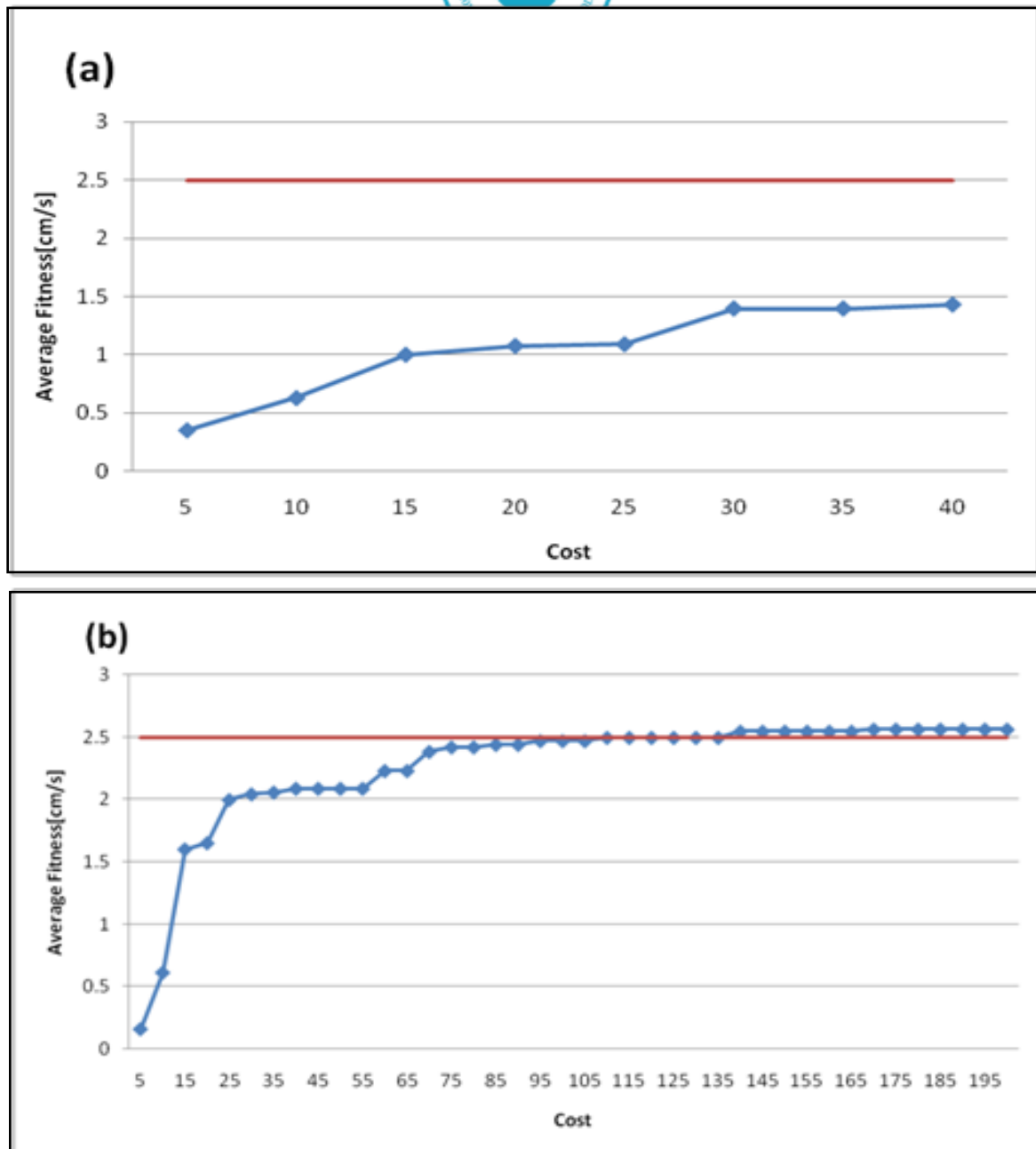


圖十三 四足機器人和制定速度比較圖(a) 人口群設定：5 (b) 人口群設定：4



## 4.2 實驗二：三足機器人

在實驗二中，我們使用同樣的模式以及實驗環境測試，在自然界中三足的生物是相對稀有許多，所以接下來我們將假設如圖三左邊的三足機器人，這類型機器人是無法從外型判斷出運動方式的，實驗步驟跟四足機器人是同樣的，先測量出一個具有公平性且不會改變的適應值當作門檻值，而我們測出的門檻值為  $2.5 \text{ cm/s}$ ，如圖十四 人口數 = 5 並且將交叉機率調整為 1 突變機率為 0.1 因素加入實驗中，在四足機器人中是如預期的可以加快實驗中適應值收斂的速度以及減少硬體的損耗，但從圖十四(a) 得知這假設是錯誤的，在第 40 代結束後適應值並沒有達到收斂的成果且適應值也沒有超過門檻值，而在接下來我們認為迭代次數的數量也是一個影響適應值收斂的關鍵點，所以將人口數 不改變並將迭代次數增加到 200 如圖十四(b)，從圖十四(b)可以看出在迭代次數為 125 時已經逐漸超過門檻值並開始收斂，並且將三足機器人每一次實驗適應值的成長幅度列出如圖十四(b)，更能證實適應值的成長幅度會直接影響，適應值的成長以及最終收斂的值。



圖十四 三足機器人的適應值並且和制定的速度比較圖(a) 人口群設定:5, 迭代數量:8 (b) 人口群設定 5, 迭代數量:40



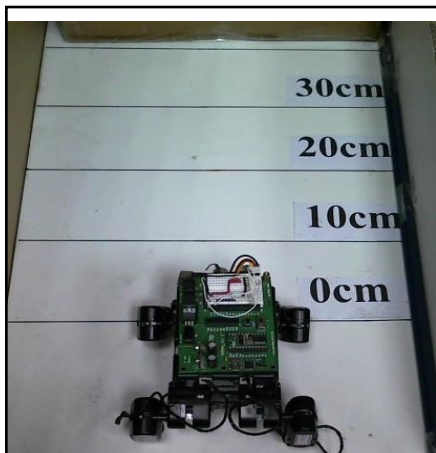
#### 4.3 實驗三：嵌入粒子群演算法機器人

大多數的文獻指出，進化式演算法的各種步驟和面臨的問題是獨立的，首先產生初代的人口群，接著根據周遭的環境因素產生相對應的適應值作為演化的根據之一，最後步驟將上述的人口群加以演化產生出新的一代，若要將進化式演算法嵌入至仿生機器人中，然而在嵌入進化式演算法至仿生機器人前，有兩個重要的且獨立的問題需要先被克服的，首先如何將設計適合這實驗環境的動作並且呈現出來，並且需要呈現出更為合理的解決方式，以及如何設計出適合每一次進化時所需要的適應方程式，以便使每一次的進化達到最佳化，這是必須先克服的，例如：實驗一和二中，我們使用標準的基因演算法作為演算法的採用，並且不論三足或者四足構型的仿生機器人，都採用隨機產生伺服器馬達角度來滿足我們演算法中所需要的人口數，在適應值方程式方面，因受限於硬體關係將採用較容易取得的每秒加速度，經由實驗一和二結果顯示，不論三足或四足的仿生機器人，經過多次的實驗次數後，皆能超出預先所定義的目標，所以在實驗三中，我們使用改良後的粒子群演算法來和基因演算法作比較。



#### 4.3.1 實驗環境

在實驗環境設定，我們遵照如圖十五(左)所示，設定仿生機器人在一個長 45 公分寬 35 公分的長方形封閉式環境中，並且我們在地上每 10 公分設置一個明顯的標註以利於實驗的進行，在機器人構型方面則選擇實驗一中四足仿生機器人如圖十五(左)所示，這機器人搭載一顆超音波感測器做計算機器人向前運動的距離使用，並且將每一次的距離使用公式 1 和 2 轉換成每秒的加速度，另外在圖十五(右)，我們拆掉四足機器人中的一足假想遇到機器人遇到不平衡狀態下是否能使用粒子群演算法超出預設的門檻值，這將是我們這一次實驗的目標。



圖十五 圖(左)四足機器人和實驗環境，圖(右)模擬四足機器人面臨損壞一足時候的情況



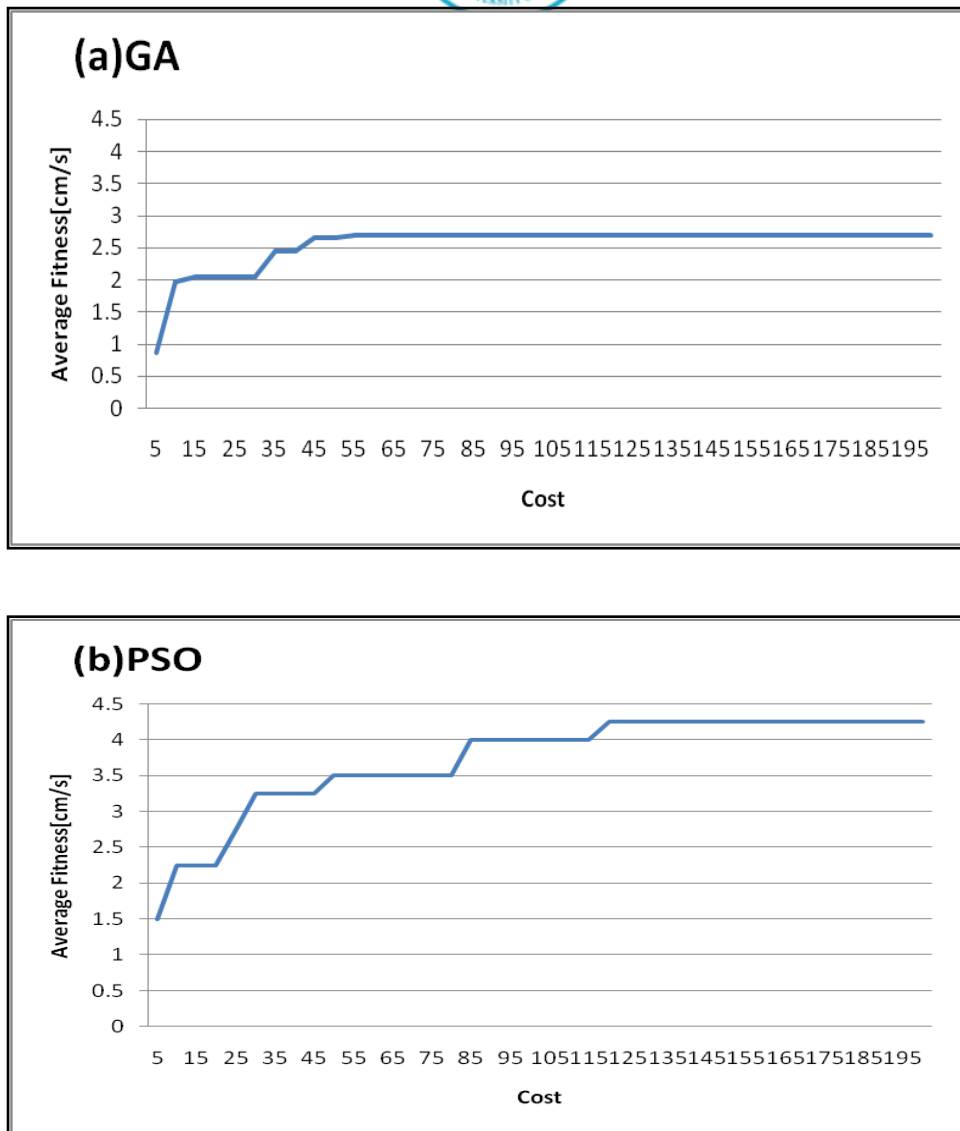
#### 4.3.2 使用多種粒子設定實驗

在粒子群演算法中，粒子的數量將會成為影響實驗效果的重要因素之一，根據文獻指出，使用的粒子數量越多效果越好但相對影響到實驗時間和成本，所以要如何使用有效的粒子量達到最大的效益將是這演算法關鍵，為了有更好的解決問題必須要有一定的取捨，為了延續我們先前設定的理論，所以我們仿造先前的實驗設定這種選擇的是有原因的，當然計算在成本計算方面也比照先前實驗設定。

#### 4.4 實驗四：比較基因演算法和粒子群演算法實驗在四足機器人

在測試損壞機器人的實驗前，於第三個實驗中首先測試了原始四足機器人如何進化算法可以在一個完全不同的方式進行機器人動力學作為參考，圖十六（a）表示對基因演算法的結果，而圖十六（b）表示其中粒子群的結果，並且人口群大小分別設定在 5 和 4 粒子並且各別迭代分 40 次。在公式（7）和（8）產生的隨機數為 0.1，圖十六（a）所示，並可以清楚看見基因演算法收斂於第 55 次生成，速度為 2.65 公分/秒。圖十六（b）表示在第 125 代，該算法的收斂速度為 4.25 公分/秒。在這種情況下，雖然粒子群演算法收斂的速度較慢，但得到了更好的結果。





圖十六 相同的四足機器人比較圖(a)和圖(b)，基因演算法和粒子群演算法的差異



所以我們相信，根據不同演算法的差異以及的兩個演算法產生下一代的方法並不相同，在進行下一步實驗前先比較基因演算法和粒子群演算法的差異如表四所示，表四明顯比較了兩種演算法不同的屬性和差異，不論使用粒子群演算法或基因演算法，這兩種演算法具有相同的隨機生成人口群開始，雖然在適應方程式是相同，然而，由於更新的公式有很大的不同，下一代也將有很大的差異。在我們的實驗三中，遺傳算法中的染色體改變是根據單點交叉和外加一個非常小的突變率(0.1)，在不同的染色體中，染色體之間的資訊是可以改變，另外粒子群演算法可被看作是一個新的粒子在粒子群中是使用和相鄰兩個粒子互相學習，也就是說當粒子之間改變每一位子都可能會認為新一代的個人行為更類似於老一代基因演算法，相對的，一個新的粒子呈現的行為是非常類似任何舊的粒子，我們將視為是一種新的行為。

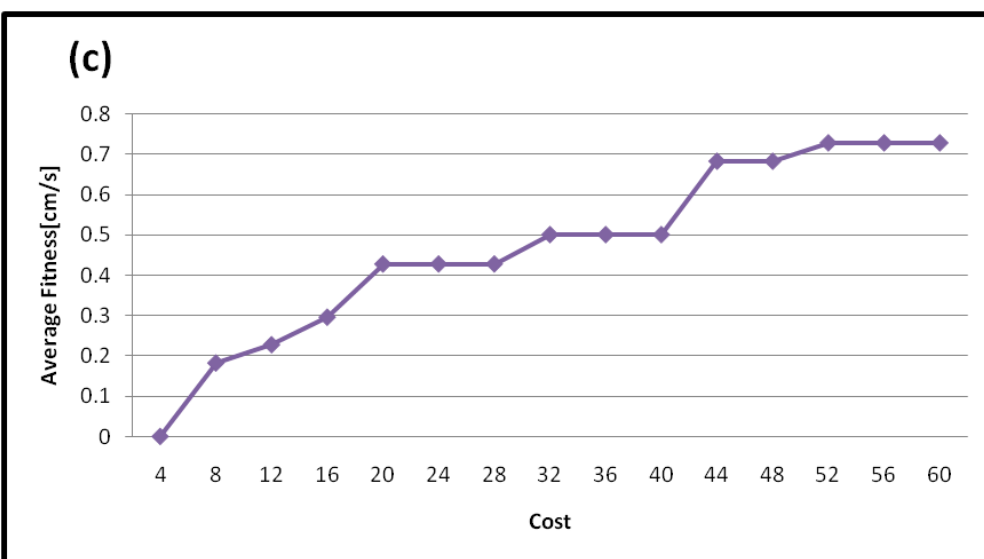
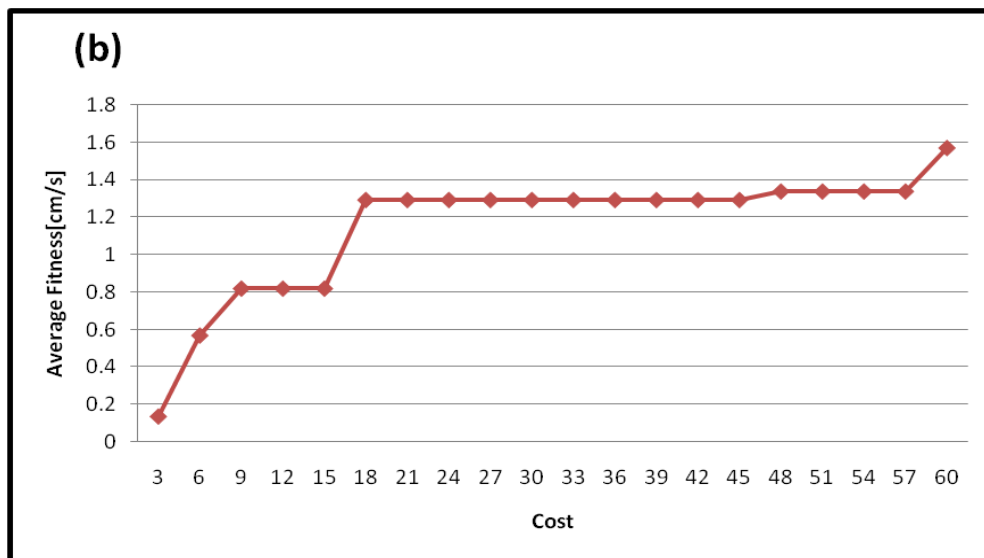
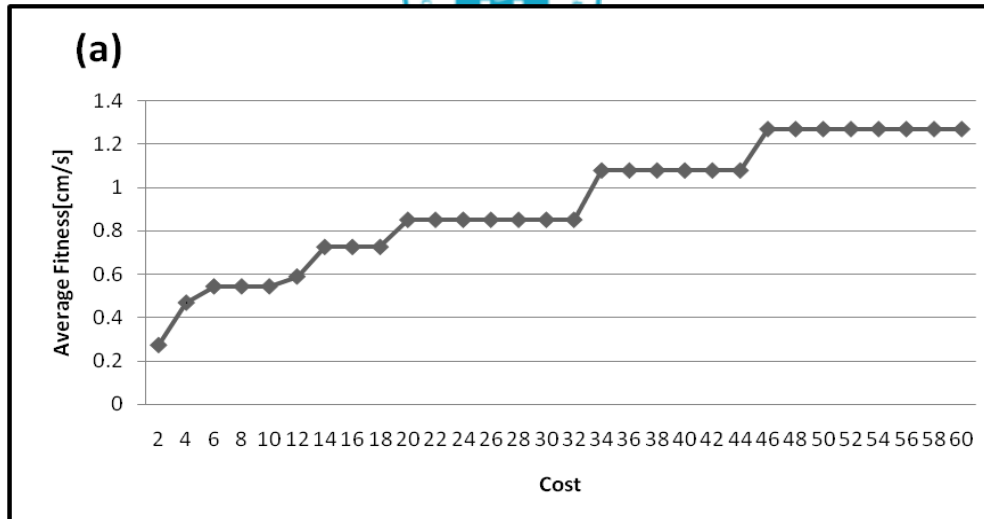
表四 比較基因演算法和粒子群演算法的特性

演算法	適應值公式	如何進化
PSO	$f = d/t \text{ (cm/s)}$	利用前一代馬達的角度差值，產生下一代的馬達角度
GA	$f = d/t \text{ (cm/s)}$	根據前一顆馬達角度的排列組合，來決定下一代馬達的角度排列組合



#### 4.4.1 比較基因演算法在受損的四足機器人

在這實驗中初步使用基因演算法模擬四足式機器人在壞一條腿的情形下所做的實驗，設定在相同的成本條件下分別給予不同的人口群如圖十七所示，在圖十七中橫軸部分別代表不同設定的人口群並搭配一定的消耗成本，(圖十七 (a) 染色體：2, 迭代數量：30, (b) 染色體：3, 迭代數量：20, (c) 染色體：4, 迭代數量：15)，為了使這三組實驗有一種共同的比較對象，我們設計將迭代數目設計成一種可調整的變數，在 Y 軸方面定義出一種可和 X 軸比較的參數並以 cm/s 為適應性做為評估單位做即時的演化使用，在結果方面我們發現當四足的協調機器人當面臨缺一條腿的問題時，在初代的行為是前進幅度是有很大的影響，這結果是可預期的，在三組不同人口數以及迭代 60 次後，我們發現同樣的硬體和迭代次數，但所演化出的 fitness 是有明顯的差異，如下面的實驗顯示當染色體：3 迭代數量：20 時，能達到最好的結果每秒 1.568182 cm。

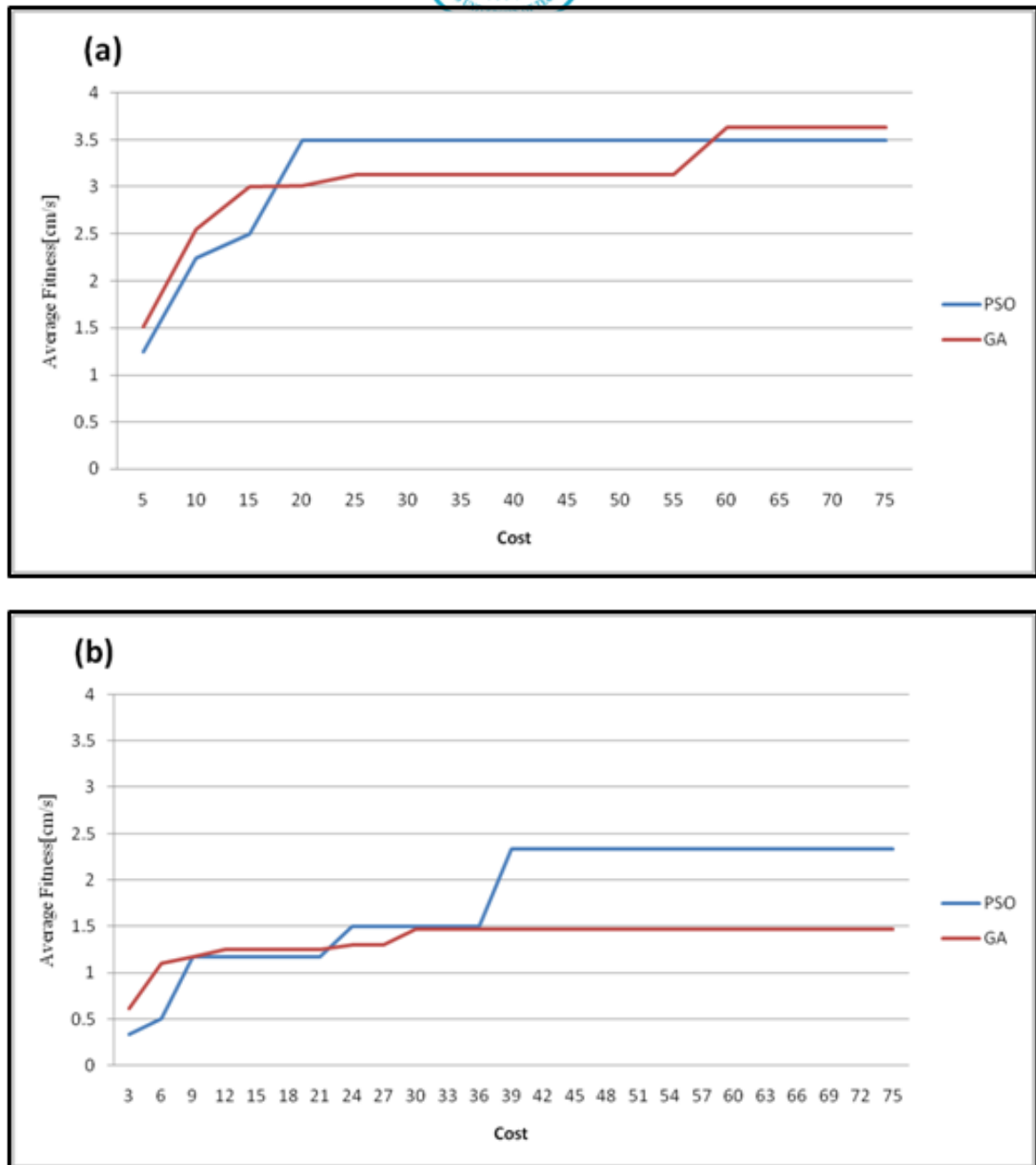


圖十七 (a)染色體：2 迭代數量：30(b)染色體：3 迭代數量：20 (c)染色體：4, 迭代數量：15



#### 4.4.2 比較基因演算法和粒子群演算法實驗在受損的四足機器人

在這個實驗中，我們採取了拆卸機器人的一條腿，模擬機器人在行走中遇到受傷情況的機器人，並假設粒子群演算法是否能在這種情況下比基因演算法收斂更快，如圖十八所示，比較粒子群演算法和基因演算法在相同的成本下有不同的設置，其結果示於圖十八（a）的設定中，人口群體設定為 5 和並且共有 15 迭代次數的，起初粒子群演算法的收斂速度比基因演算法快，但後來基因演算法在成本：60 開始收斂並且得到更好適應值，根據圖十八（a）的實驗，是沒有達到預期的結果，所以我們將調降人口群為 3 並且共迭代 25 次做為比較對象，如圖十八（b）所示人口下降到 3，和 25 的最大生成的所述第二設定的總成本仍然是 75，在此設置下，最終粒子群演算法比基因演算法得到更好的適應值然而，因為調降人口群為 3，所以收斂的結果沒有比之前的實驗好，若在有限的資源下我們寧願實驗成本最少，因此，在我們的實驗中的人口被設置為一個很小的數，所以實驗結果表明，若不考慮實驗成本因素將人口群增加仍然可以在任務中獲得更好的成績，但若在限定的實驗成本內是能證明不但能使受傷的機器人移動也能達到我們預期的效果。

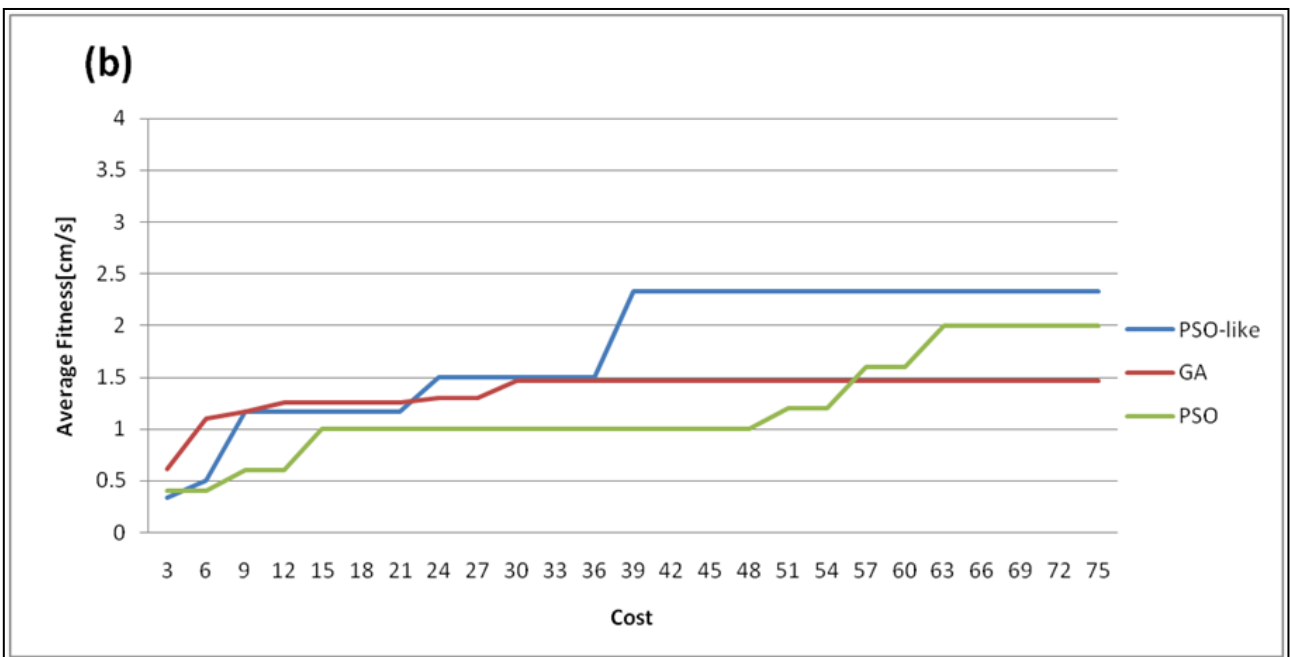
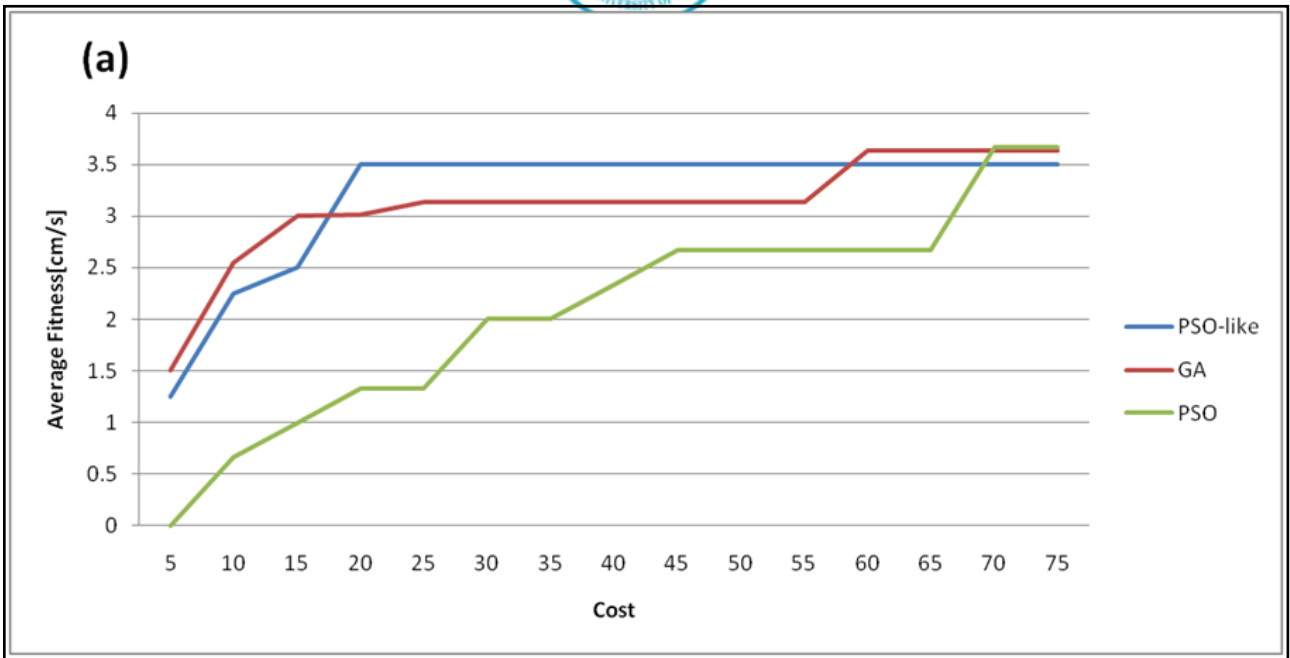


圖十八 相同的三足機器人比較圖(a)和圖(b)，基因演算法和粒子群演算法的差異



#### 4.5 實驗五：比較多種演算法在受損機器人中的差異

在實驗四中，我們比較了基因演算法、粒子群演算法、改良式粒子群演算法，並延續上一個實驗的設定，將四足機器人拆去一足觀察其運動模式，和機器人收斂的速度品質，並且比較之中的差別如圖十九所示，在圖十九(a)中，設定相同的成本：75 人口群大小為 5 並且迭代 15 次，在一開始改良式粒子群演算法收斂速度遠超過基因演算法和粒子群演算法，但是從 55 開始到 75 實驗結束後，改良式粒子群演算法最後得到的適應值並沒有基因演算法以及粒子群演算法來的好，我們推論在於改良式粒子群演算法由於強調大幅降低計算量，所以在每一次演化出下一代的複雜度就可能會大幅度的降下，在某些設定中改良式的粒子群演算法會較優於傳統的進化式演算法，根據在圖十九(a) 得到的結論我們另外實驗一樣的成本：75 中，降低人口群為 3 相較於迭代次數增加至 25 如圖十九(b)所示，在實驗開始前並沒有明顯的差距，基因演算法和粒子群演算法逐漸收斂在區域解中，但改良式粒子群演算法成本：40 時逐漸跳脫並持續領先其他演算法，我們不排除這實驗有些運氣成分在，但我們改良傳統的粒子群演算法，篩掉部分認為不需要的計算成本並經過以上的多次實驗，不論在四足型機器人或者三足型機器人都能得到不錯的期望和結果。



圖十九 比較三足機器人在基因演算法、粒子群演算法、改良式粒子群演算法的差異





## 4.6 實驗六：多目標適應演化

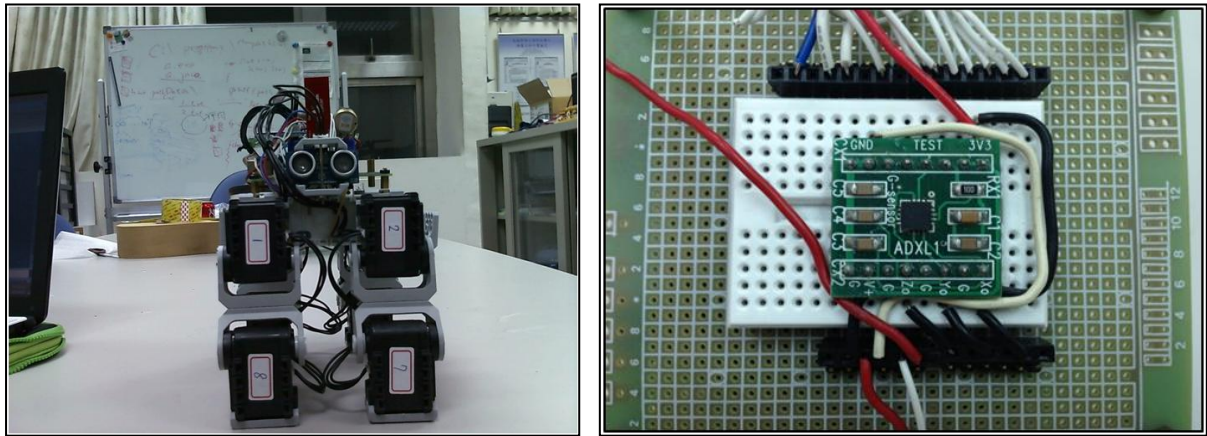
在實驗六，我們將延續先前實驗的設計加入新的感測器和影響迭代的因素，由先前的實驗我們得到一個結論，使用進化式機器人不會因為機器人本身構型而無法達成目的，甚至有很高機率能超過預期的目的，在實驗一到實驗五之間我們觀察到，不論在四足仿生機器人或三足仿生機器人又或者是部分受損的情況下的各種情形演化，常常會因為動作幅度過大而導致實驗失敗，這樣會造成時間、硬體間無謂的實驗耗損，所以我們利用三軸加速度計的特性來幫助檢測機器人在演化中的穩定度，並將穩定度和機器人行進間的速度達到多目標適應值的演化，在接下來章節我們將設計新的四足仿生機器人。

### 4.6.1 新構型機器人和感測器

在新的四足仿生機器人中，由於我們使用扭力較大的伺服器馬達，這能幫助我們實驗中伺服器馬達的耗損和克服環境中各種影響因素，比較圖三(左)中和圖二十(左)的不同，其中圖三(左)為舊的四足仿生機器人圖二十(左)則是新的四足仿生機器人，在同樣由八顆伺服器馬達組成機器人條件下，新的機器人除了原本裝置在前方的超音波感測器，在機器人身體裡裝置了三軸加速度計如圖二十(右)所示。和藍芽感測器，這些新的設備能不只能讓機器人更快得到環境間的變動，也能即時的透過藍芽傳輸遠端的監控加



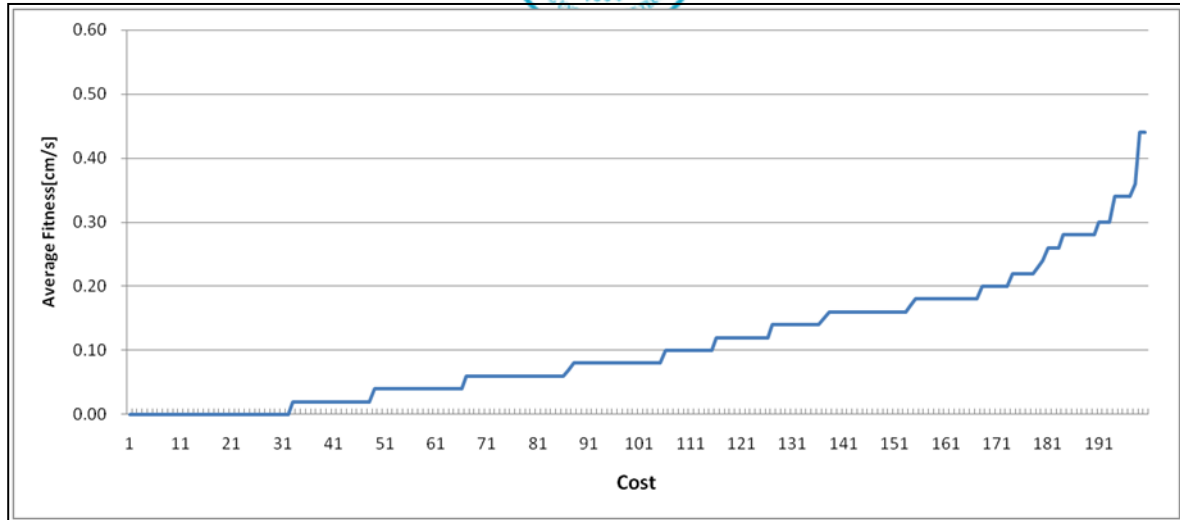
快實驗間的數據傳遞以及監控。



圖二十 (左)新四足仿生機器人, 圖(右)四足機器人中的三軸加速度計

#### 4.6.2 使用多目標演化實驗設計以及結果

在接下來多目標演化實驗中，我們採取兩種不同的感測器資訊，但基於實驗的完整性所需，我們比照之前圖六的實驗流程，進行人口群為 10 共演化 20 次總共改變 200 基因組合的實驗量，如圖二十一所示 X 軸為全部基因數目，Y 軸為超音波所轉換的瞬間速度(cm/s)，我們由實驗結果可以得知，當 X 軸為 40 和 70 左右時機器人分別得到 0.44cm/s 的速度為最好的結果，我們發現相對於前面實驗結果是相差許多，原因在於新設計的四足仿生機器人四足方面並不能產生旋轉前進的功能，所以較容易產生並設定為接下來實驗中的比較對象以及目標。



圖二十一 使用基因演算法進行 200 種不同基因單目標演化實驗

#### 4.6.3 實驗設計以及結果：使用多目標演化

$$F = W_1 \times \text{超音波測距儀} + W_2 \times \frac{1}{\text{三軸加速度計}} \quad (11)$$

在實驗流程方面，我們採用原圖六中設計的基因演算法流程圖，但唯一不同的是加入三軸加速度計的水平測量，做多目標演化並更新適應值方程式公式(11)，適應值計算是藉由超音波測距儀轉換的速度資訊(單位:公分/秒)，和三軸加速度計轉換的水平資訊(單位為加速度計回傳電壓值，電壓值正比於加速度)，公式(11)是將兩種資訊依據權重比例轉換做為新的適應值方程式的依據，主要使兩種不同資訊呈現正相關，另外新設備使用 Arduino 控制串接式伺服器馬達的方式，並採用即時運算的特性如圖二十，相較於過去舊型四足機器人圖十五(左)比較，更能完成高計算量的實驗，因此雖然



在實驗六中的實驗流程沒太重大的改變，但在實驗數據的展示上以及實驗的量會將有大幅度的成長，在實驗六中我們採取人口數：10 迭代次數 20 次共改變 200 基因的實驗量，圖二十二 X 軸和 Y 軸方面分別代表不同性質的適應值(速度以及水平值)，為了提升實驗的可信度和準確度，在每一次的演化都加入第三方的觀察確認，以確保實驗進行不必要的硬體耗損，實驗顯示出大多數的基因會使機器人在沒有任何移動的行為，圖二十二(使用基因演算法進行多目標演化實驗，並由 200 種不同基因組合而成，圖(上) 超音波測距儀：0.9 三軸加速度計：0.1 圖(下) 超音波測距儀：0.8 三軸加速度計：0.2)的設定中，並沒有得到比上一個實驗中 0.44cm/s，接著我們繼續調整公式(11)中的權重值，如圖二十三(使用基因演算法進行多目標演化實驗，並由 200 種不同基因組合而成，圖(上) 超音波測距儀：0.7 三軸加速度計：0.3 圖(下) 超音波測距儀：0.6 三軸加速度計：0.4)進行多目標演化的實驗中，我們在這一組實驗中得到超越我們實驗前設定的 0.44cm/s 更好的結果(0.5cm/s)，所以實驗結論能得到在超音波權重值設定：0.7 以及三軸加速度計權重值：0.3 時能達到最高效益。



#### 4.6.4 實驗設計以及結果：瀾集演算法

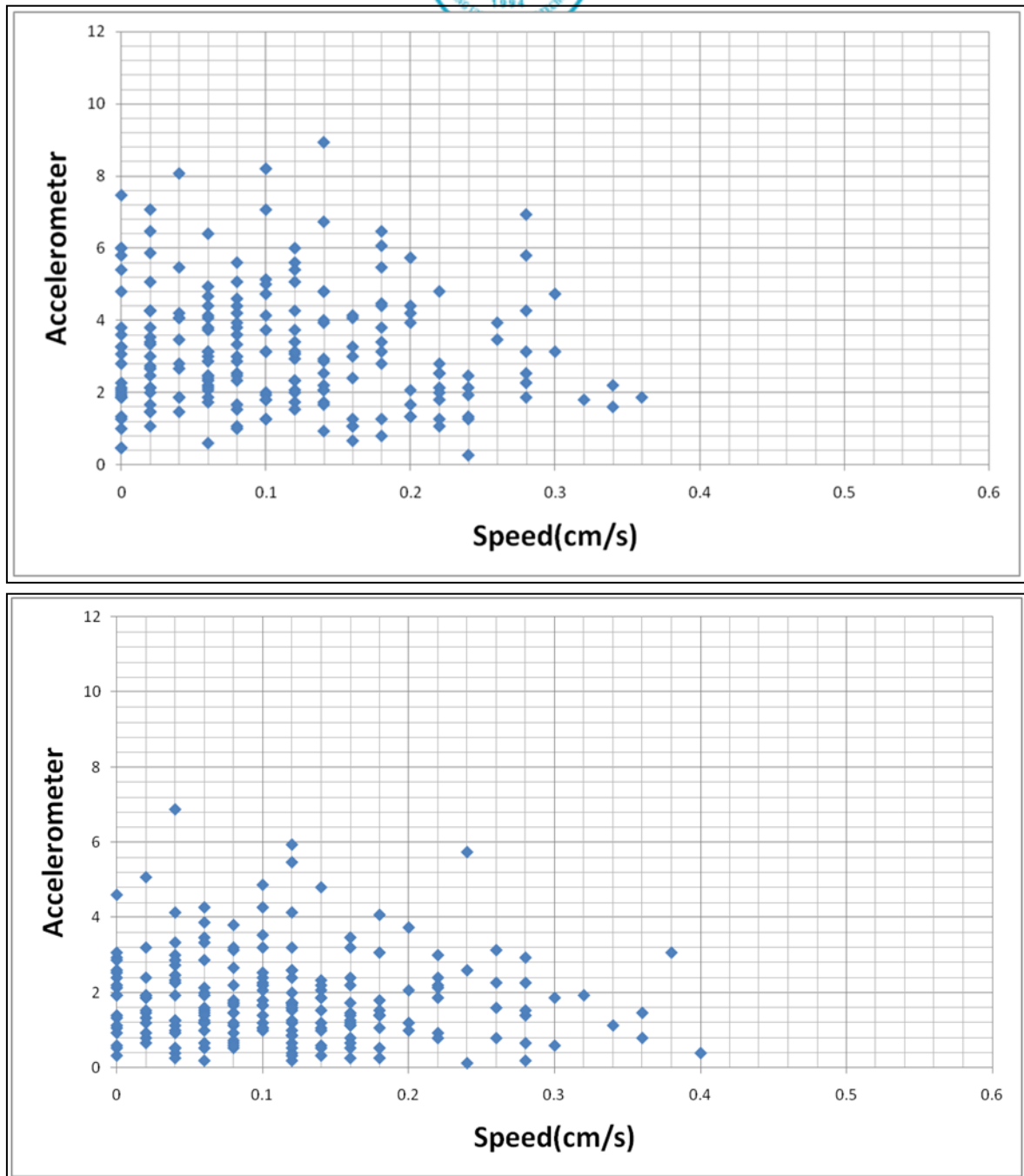
在瀾集演算法中，我們使用圖六中為基礎並加入粒子群演算法，製作出類似瀾集演算法的實驗流程，如圖十一所示，實驗流程以及參數設定依據前一實驗中的公式(11) 得到，實驗結果如圖二十四和圖二十五所示，我們可以得到由這實驗結果速度方面得到 0.5cm/s 以上的結果，雖然在穩定度方面並沒有大幅度的下降，但至少在移動方面有不錯的實驗結果，每一次實驗中最佳點(S,A) ，由(超音波測距儀，三軸加速度計)最好的數值組成，如表五表六所示。

表五 由 200 種基因組合進行基因演算法多目標演化實驗

演算法	超音波權重值 $W_1$	三軸加速度權重值 $W_2$	最佳點(S,A)	
GA	0.9	0.1	0.36	1.85
	0.8	0.2	0.4	0.4
	0.7	0.3	0.5	0.4
	0.6	0.4	0.38	4.85

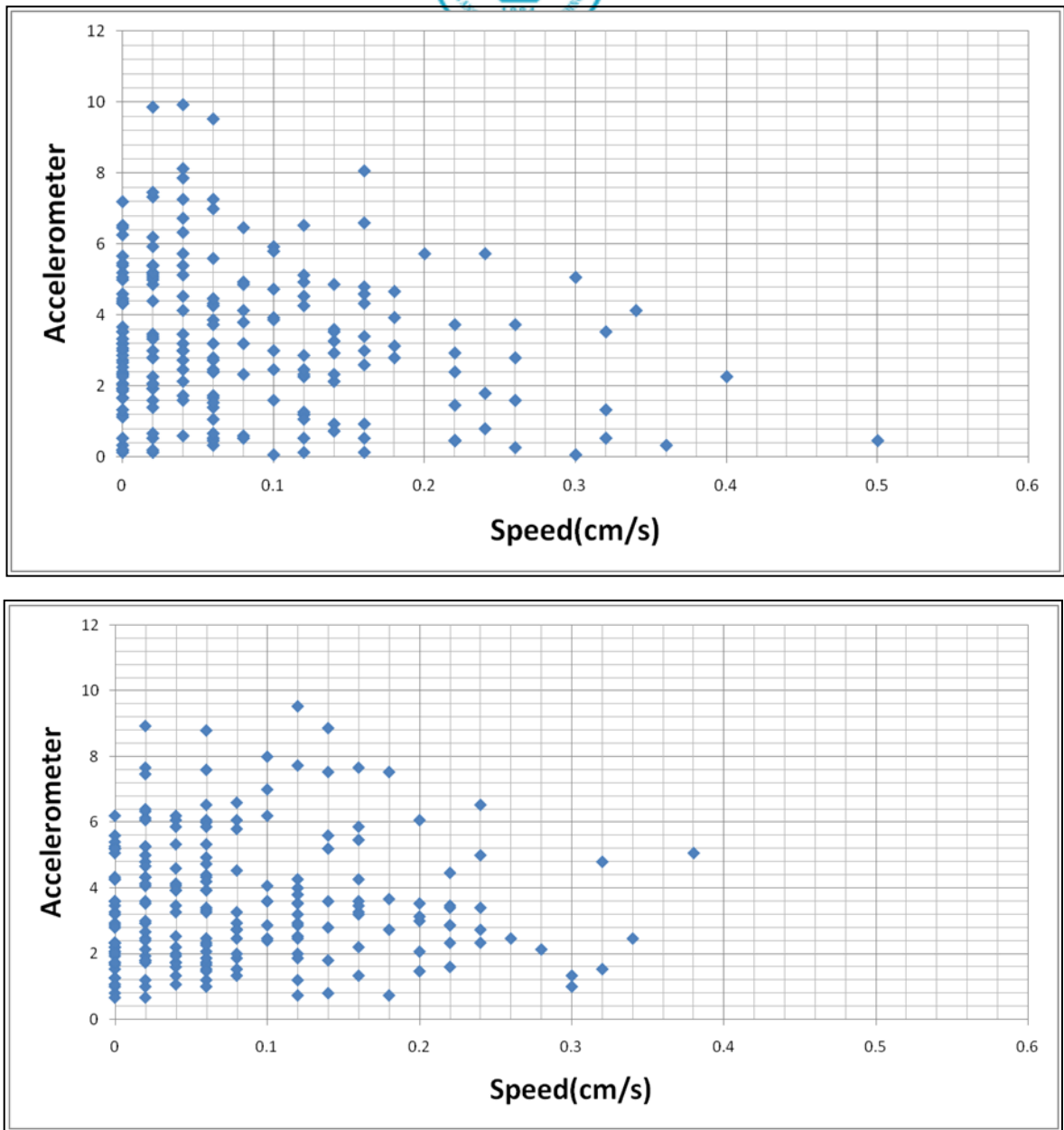
表六 由 200 種基因組合進行瀾集演算法多目標演化實驗

演算法	超音波權重值 $W_1$	三軸加速度權重值 $W_2$	最佳點(S,A)	
MA	0.9	0.1	0.5	0.4
	0.8	0.2	0.42	4.45
	0.7	0.3	0.52	4.8
	0.6	0.4	0.4	1.65

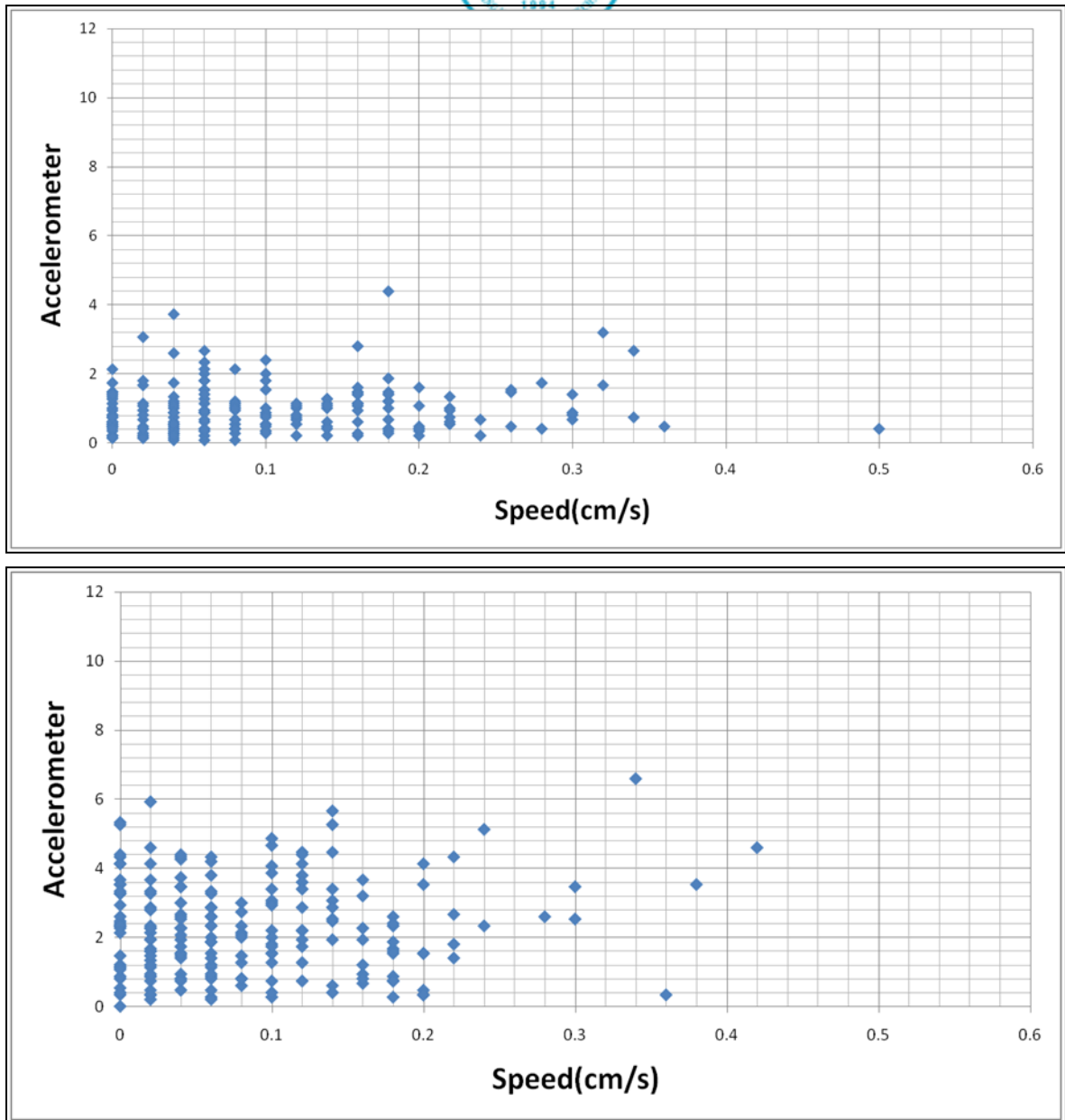


圖二十二 使用基因演算法進行 200 種不同基因多目標演化實驗。圖(上) 超音波測距儀：0.9 三軸加速度計：0.1 圖(下) 超音波測距儀：0.8 三軸加速度計：0.2



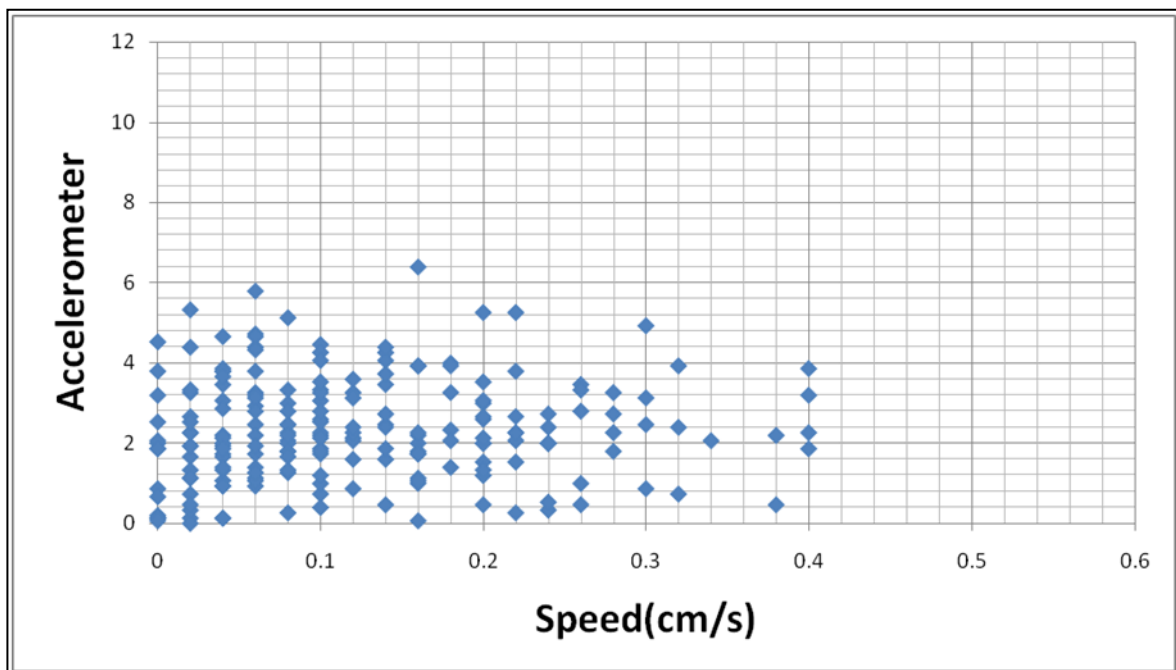
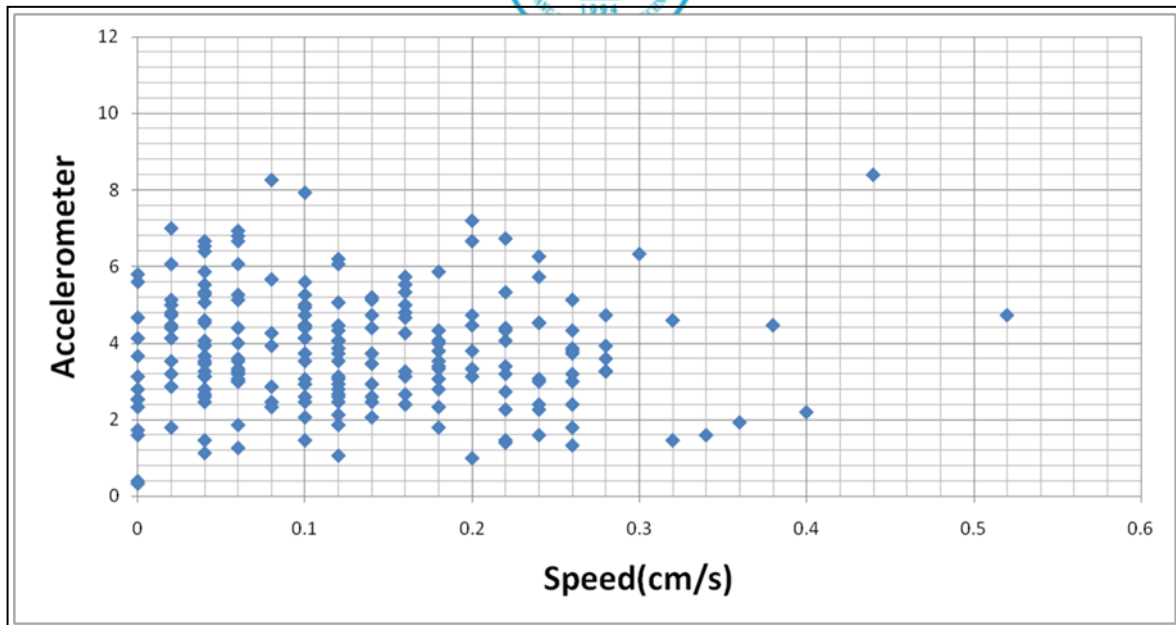


圖二十三 使用基因演算法進行 200 種不同基因多目標演化實驗。圖(上) 超音波測距儀：0.7 三軸加速度計：0.3 圖(下) 超音波測距儀：0.6 三軸加速度計：0.4



圖二十四 使用瀾集演算法進行 200 種不同基因多目標演化實驗。圖(上)超音波測距儀：0.9 三軸加速度計：0.1 圖(下) 超音波測距儀：0.8 三軸加速度計：0.2





圖二十五 使用瀾集演算法進行 200 種不同基因多目標演化實驗。圖(上)超音波測距儀：0.7 三軸加速度計：0.3 圖(下) 超音波測距儀：0.6 三軸加速度計：0.4



## 五、結論與未來工作

### 5.1 結論

本研究目標是以發展一套能使機器人產生向前運動行為，為此我們針對機器人提出三種不同構型，進行實驗和評斷，第一種為常見的四足機器人系統，根據機器人本身的八顆伺服器馬達設計多種不同機器人姿勢排列組合，第二種為較不常見三足機器人系統，根據機器人本身的六顆伺服器馬達設計多種不同機器人姿勢排列組合，第三種則設定一種機器人處於不穩定的三足狀態，並加以分析和實驗。在機器人控制方面，不使用動力學分析控制機器人的運動，而採用近年流行的進化式演算法來完成機器人複雜行為控制，我們平均給予不同構型可能所需的演算法，並且朝著降低機器人在控制方面，使用最少成本(例如:染色體複雜度、時間複雜度)來達到最大移動效益為目標，實驗中強調機器人於不穩定的運動行為和自我修復能力為主要貢獻，首先我們使用標準基因演算法做為實驗首選，搭配裝載於機器人前方的超音波感測器，做為演算法中需要的回饋資訊，實驗中將同樣構型下，使用運動分析控制的機器人做為比較對象，發現在一定時間量可以逐漸超越使用運動分控制的機器人，實驗二，將同樣實驗流程複製於不常見的三足機器人也能得到相同效果，實驗三至六依序使用基因演算法、粒子群演算法、改良式粒子演算法、改良式彌集演算法，在最後實驗中，



加入三軸加速度計做為另一種進化式機器人多目標演化量測依據，我們得到結論是，使用進化式演算法在取代機器人運動控制方面是可行的，不論機器人是四足或者三足甚至是任意拼接的構型下，都能使用進化式演算法達到自我運動的前進行為，以下我們使用條列式詳細列出我們的貢獻。

1. 我們嘗試多種演算法去完成進化式機器人的移動控制。
2. 根據實驗環境的需求，改良粒子群演算法、濶集演算法達到進化式機器人在移動上的控制。
3. 嘗試使用任意拼接的三足式機器人，以及設定四足式機器人面臨部分受損情形時，機器人能產生自我調適的移動行為。
4. 使用超音波、三軸加速度計實驗多目標的演化。

## 5.2 未來工作

在未來工作裡，我們將嘗試進化式機器人其他的汰換可能性，例如：構型、進化式演算法、感測器等等。另一方面，我們將使用模糊控制、類神經網路等其他演算法，優化每一次演化的可調參數。

### 5.2.1 模糊控制

過去文獻指出，模糊控制已經大量使用在機器人控制的議題上[25]，模糊控制是由連續的歸屬函數所構成的最先是美國加州大學查德（L. A. Zadeh）教授於1965年所發表的[26]，主要目的是降低問題在開發時的複



雜度，特別解決非線性以及不確定性較高的問題時效果更高，這想法源自於真實世界中存在於許多不確定性的問題，例如：一群動物中有狗、馬、貓、鳥等等許多選擇性，所以在這種情形"一群動物"中存在許多種物體，並稱為模糊集合，二擇一的解決方式將不適用，模糊控制設計理念這將問題朝向人類的思維模式，基於這個優點，在過去許多文獻將此問題用於解決進化式機器人上感測器上的判定以及優化機器人的行為[26][27]，模糊控制分為四個步驟：(1)將問題模糊化並繪製出問題的歸屬函數圖、(2)將問題訂定出模糊規則庫、(3)將問題使用模糊規則推導、(4)將問題解模糊化。

### 5.2.2 類神經網路

類神經網路已被研究多年，並已經使用進化式機器人學的領域[8][28]，類神經網路起源於 1940 年起，科學家即著手從事此方面的研究，仿造最簡單的神經元模式，開始建立最原始的類神經網路(Artificial Neural Network ANN)，其原因在於電腦的功能雖然普及，但卻不能因此就完全取代人腦。因人腦可輕易的聽、說、讀、寫、看，電腦卻力有未逮，要如何模擬這種複雜的人類行為進而建立一部可以和人腦匹敵的電腦，可說是電腦科學家的一大夢想，類這些類神經網路的模型主要是嘗試著去模仿人類的神經系統，因為人類的神經系統在語音、聽覺、影像和視覺方面均有很完美的表現，也期望電腦利用這些模型能夠在這些方面有出色的成果，所以類神經



的架構就是來自於現今對人類神經系統的認識[29]。現今的類神經網路是由很多非線性的運算單元(我們叫神經元 Neuron)和位於這些運算單元間的眾多連結所組成，而這些運算單元通常是以平行且分散的方式在作運算，如此就可以同時處理大量的資料，由這樣的設計就可以被用來處理各種需要大量資料運算的應用上，比如說機器人控制。



## 六、參考文獻

- [1] Dilip Kumar Pratihara, "Evolutionary robotics—A review, "SADHANA Volume 28, Part 6, pp.999-1009, 2003.
- [2] Dawkins, Richard, "Universal Darwinism". In Bendall, D. S. Evolution from molecules to man. Cambridge University Press, 1983.
- [3] Thomas Schmickl, Jürgen Stradner, Heiko Hamann, Karl Crailsheim "Major Feedbacks that Support Artificial Evolution in Multi-Modular Robotics", International Workshop Exploring New Horizons in Evolutionary Design of Robots at IROS'09, 2009.
- [4] Gary Parker , David W. Braun , Ingo Cyliak "EVOLVING HEXAPOD GAITS USING A CYCLIC GENETIC ALGORITHM" Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing , 1997.
- [5] R.A. Watson, S.G. Ficici, J.B. Pollack, "Embodied evolution : Distributing an evolutionary algorithm in a population of robots", Robotics and Autonomous Systems 39 (1) 118, 2002.
- [6] D. Floreano and F. Mondada, "Evolution of homing navigation in a real mobile robot," IEEE Transactions on Systems, Man, Cybernetics Part B: Cybernetics, Vol. 26, 1996, pp. 396-407.
- [7] Shi, Y. and R. C. Eberhart "Empirical study of particle swarm optimization. "Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Washington, DC, 1999.
- [8] Andrew L. Nelson, Gregory J. Barlow, Lefter Doitsidis, "fitness functions in evolutionary robotics : A survey and analysis", Robotics and



Autonomous System 57, 345-370, 2009.

- [9] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, "Self-Organizing Biological Systems", Princeton University Press, 2001.
- [10] R. A. Watson, S. G. Ficici, and J. B. Pollack, "Embodied evolution: Distributing an evolutionary algorithm in a population of robots," *Robotics and Autonomous Systems*, Vol. 39, pp. 1-18, 2002.
- [11] R. Stoen and K. Price, "differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, Vol.11, pp. 341-359, 1997.
- [12] S. Nolfi and D. Floreano, "Co-evolving predator and prey robots: Do ‘arms races’ arise in artificial evolution?" *Artificial Life*, Vol. 4, pp.311-335, 1998.
- [13] B. Andersson, P. Svensson, M. Nordahl, and P. Nordin, "On-line evolution of control for a four-legged robot using genetic programming," in *Proceedings of EvoWorkshops, 2000*, LNCS, pp. 319-326, 1803.
- [14] I. Harvey, P. Husbands, and D. Cliff, "Seeing the light: Artificial evolution, real vision," D. Cliff, P. Husbands, J.-A. Meyer, S. Wilson, eds., *From Animals to Animates 3*, *Proceedings of 3rd International Conference on Simulation of Adaptive Behavior*, MIT Press/Bradford Books, Boston, pp. 392-401, MA, 1994.
- [15] N. J. Rodriguez, J. H. Garzon, and J. J. Soriano, "Morphology, kinematics and stability of a *Scarabaeus* as bases for design of robots," in *Proceedings of the Electronics, Robotics and Automotive Mechanics Conference*, Vol. 2,





pp. 151-156, 2006.

- [16]A. E. Eiben, G. Karafotias, and E. Haasdijk, "Self-adaptive mutation in on-line, onboard evolutionary robotics," in Proceedings of the 4th IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshop, pp. 147-152, 2010.
- [17]S. Cagnoni, et al., "Real-World Applications of Evolutionary Computing", Springer-Verlag Lecture Notes in Computer Science, Berlin, 2000.
- [18]J. Bongard,"The 'What', 'How', and the 'Why' of evolutionary robotics," in S. Doncieux et al., New Horizons in Evolutionary Robotics, SCI 341, Springer-Verlag, Berlin, pp. 29-35, 2011.
- [19]Eberhart, R. C. and J. Kennedy A new optimizer using particle swarm theory. Proceedings Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995.
- [20]M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," IEEE Transactions on Evolutionary Computation, Vol. 1, pp. 53-66, 1997.
- [21]J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of IEEE International Conference on Neural Networks IV, pp. 1942-1948, 1995.
- [22]Eberhart, R. C. and Y. Shi "Comparison between genetic algorithms and particle swarm optimization." 1998 Annual Conference on Evolutionary Programming, San Diego, CA, 1998.
- [23]Shi, Y. and R. C. Eberhart "A modified particle swarm optimizer." Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Piscataway, NJ, 1998.





- [24]Chen, X. S.; Ong, Y. S.; Lim, M. H.; Tan, K. C. "A Multi-Facet Survey on Memetic Computation". IEEE Transactions on Evolutionary Computation 15 (5): 591–607, 2011.
- [25]Khairulmizam Samsudina, Faisul Arif Ahmadb, Syamsiah Mashohorb, " A highly interpretable fuzzy rule base using ordinal structure for obstacle avoidance of mobile robot", Applied Soft Computing, Volume 11, Issue 2, Pages 1631 – 1637, March 2011.
- [26]L.A. Zadeh, "Fuzzy sets", information and Control 8, pp. 338-353, 1965.
- [27]Shi, Y. and R. C. Eberhart "Fuzzy adaptive particle swarm optimization." Proceeding of the 2001 Congress on Evolutionary Computation, Piscataway, NJ, 2001.
- [28]Chia-Feng Juang, Senior, Yu-Cheng Chang, and Che-Meng Hsiao, "Evolving Gaits of a Hexapod Robot by Recurrent Neural Networks With Symbiotic Species-Based Particle Swarm Optimization", IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 58, NO. 7, JULY 2011.
- [29]J. J. HOPFIELD, "Neural networks and physical systems with emergent collective computational abilities." Proc. NatL Acad. Sci. USA Vol. 79, pp. 2554-2558, April 1982.