

東海大學  
資訊工程研究所

碩士論文

指導教授：蔡清欉 博士

運用 PSO 自動化控制  
室內光照及溫濕度感測

Use PSO automation control  
indoor of light and temperature and  
humidity sensing

研究生：賴泳瑄

中華民國 107 年 1 月

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 賴 泳 瑄 所提之論文

運用 PSO 自動化控制室內光照及溫濕度感測

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

召 集 人

顏 敏 簽章

委 員

李 元 勳

周 啟 言

指 導 教 授

翁 浩 振 簽章

中華民國 107 年 1 月 18 日

# 摘要

博物館對於人們來說是一個保存歷史或具有價值文物的地方，而且保存這些重要的資產是需要注入相當的心血。然而，文物的衰敗是屬於無法避免的自然現象，所以，如何保存這些重要文化遺產就是一個挑戰。

近來IOT sensor已成為重要議題，但是要設計具備依據各種需求來自行判斷學習的IOT仍然不容易，大部分仍須要人員自行控制更改。因此為解決上述問題本論文是要設計一個具有AI自動操控多個百葉窗，使用室外的自然光源來調控室內溫濕度來達成目標。根據各個文物的保存條件進行室內環境的參數設定，以及藉由室內外各種感測器監控環境變數，透過粒子群最佳化法(Particle Swarm Optimization)訓練至最佳調控狀態，使窗簾持續更新在最佳角度，讓博物館在不需要人為操控下，可以有效保存文物又可以達到博物館的節能減碳的效果。

關鍵字：物聯網、AI、PSO、居家溫控

# Abstract

The museum preserves the history and value of cultural relics. These important assets is needed to inject considerable effort. But, the decline of cultural relics is an inevitable natural phenomenon. So, how to save these important cultural heritages is a major issue.

IOT sensor has become an important issue, but it is still not easy to design an IOT that can judge and learn from various needs. Most still need personnel to control the changes. Therefore, to solve the above problem this paper design a multi-louver though AI automatic control that use the source of outdoor natural light, temperature and humidity. In accordance with the preservation of various cultural relics automatic control the parameters of indoor environment seted by varieties of sensor monitoring of indoor and outdoor environmental. The curtains could be updated at the best point so that the museum does not require man-made control. The results show that the cultural relics could be saved and the museum's energy saved and carbon reduction.

**Keywords:** Internet of Things,AI, PSO, Home temperature control

# Table of Contents

摘要.....	I
Abstract.....	II
Table of Contents.....	III
圖索引.....	IV
表目錄.....	V
第一章 緒論.....	1
1.1 簡介.....	1
1.2 文獻回顧.....	3
第二章 電動窗簾之系統硬體架構.....	5
2.1 系統原理.....	5
2.2 控制主機(Raspberry Pi).....	7
2.3 收集感測器訊息主機(Arduino).....	8
2.4 感測器設備.....	9
2.4.1 IRremote.....	9
2.4.2 DHT-11 溫濕度感測器.....	11
2.4.3 GY-30 數位光強度檢測模組.....	12
2.4.4 GY-291 ADXL345 數字三軸重力加速度傾斜模組.....	12
2.4.5 步進馬達.....	14
2.5 RaspberryPi 開發語言.....	17
2.6 Arduino 開發語言.....	18
第三章 以 PSO 演算法做為博物館之光照及溫濕度控制.....	19
3.1 PSO 簡介.....	19
3.2 PSO 基本概念.....	21
3.3 PSO 參數在實務上的設置、調控.....	25
3.4 百葉窗 PSO 設計.....	26
第四章 實驗結果與討論.....	34
4.1 三組百葉窗 PSO 訓練實驗結果.....	34
4.2 PSO 訓練後之單一百葉窗數據結果.....	36
第五章 結論及未來發展建議.....	41
5.1 結論.....	41
5.2 未來展望.....	41
參考文獻.....	42

# List of Figures

圖2.1 單一百葉窗硬體架構圖.....	6
圖2.2 單一百葉窗硬體示意圖.....	6
圖2.3 Raspberry Pi 板.....	8
圖2.4 Ardino uno 板.....	9
圖2.5 IRremote 模組.....	10
圖2.6 DHT-11 溫濕度傳感器.....	11
圖2.7 GY-30 數位光強度檢測模組.....	12
圖2.8 GY-291 ADXL345 數字三軸重力加速度傾斜度模組.....	13
圖2.9 步進馬達.....	14
圖2.10 2相4線步進馬達.....	15
圖2.11 2相6線步進馬達.....	16
圖2.12 2相6線步進馬達激磁順序.....	16
圖3.1 魚群 PSO 概念圖.....	22
圖3.2 PSO 往最佳解移動向量圖.....	23
圖3.3 PSO 演算法.....	24
圖3.4 系統流程圖.....	27
圖3.5 文物跟各項數據毀損程度比較圖.....	28
圖3.6 三個百葉窗硬體架構圖.....	29
圖3.7 室外感測器.....	32
圖3.8 室內感測器包含控制主機百葉窗關閉.....	32
圖3.9 百葉窗開啟.....	33
圖4.1 百葉窗 a 的收斂圖(初始角度 0 速度 2).....	34
圖4.2 百葉窗 b 的收斂圖(初始角度 45 速度 2).....	35
圖4.3 百葉窗 c 的收斂圖(初始角度 -45 速度 -2).....	35
圖4.4 百葉窗 gbest 的收斂圖.....	35

# List of Table

表 4.1 百葉窗(a)位置改變記錄表格 .....	35
表 4.2 百葉窗(a)位置改變記錄表格 .....	36
表 4.3 百葉窗(a)位置改變記錄表格 .....	37
表4.4 窗簾角度在 0 的時候測量室內室外三個參數值.....	39
表4.5 窗簾角度依據 PSO 演算法更新 .....	39



# 第一章 緒論

## 1.1 簡介

居家室內溫度控制是一個重大的議題在此以博物館為例，在保存文物的時候有很多因素會影響到文物的保存時限，包括溫度、相對濕度、光線以及微生物等。這些環境因子是引發文物劣化毀損的媒介，一旦人們失去警覺或者沒有做好防備，就會對文物造成化學性或物理性的破壞。

對於博物館來說的自動化應用[1]都還是侷限在空調設備的調控，然而基本的控制溫度控制濕度對於文物來講是基本的辦法，大部分的博物館的文物都還是保存在室內空間並且陽光無法照射進去，這是因為沒辦法配置相當的人員來做窗簾的控制與設置，然而陽光無法照射進入室內代表室內的可見光源必須得依靠人造光源，雖然人造光源可以控制文物的光照強度大小來做調整，對於保存文物來說相當方便，但是這些的設置對於電的消耗來說是相當可觀的，不僅是整個博物館的光照，還有整個空間的空調設備、除濕加濕設備等，對於博物館來說，最大的經營壓力或許就是文物的保存，如果能在環境可調控下利用自

然的環境參數來調整博物館所需要的文物保存環境參數，對於博物館勢必節省長期累積下來的能源

本論文運用自然環境下的光照跟溫濕度來調整文物保存所需要的光照以及溫濕度，設計一個具有AI自動操控的多個百葉窗，利用微型電腦(Raspberry PI)來達成本專題的實驗目的，搭配各種感測器監控環境變數，並將室內測得的數據，利用PSO粒子群最佳化法(Particle Swarm Optimization)[2]來調整窗簾應有的角度，讓博物館在不需要人為操控下達到自動對於環境偵測並且改變角度的效果，一開始藉由室內室外感測器量測兩邊數值，多個百葉窗同時進行，依據當下個人最佳解跟群體最佳解數值調整角度持續收斂到最佳最終群體解，來達成室內參數接近標準。當調整到接近標準之後就持續等待直到室外參數又改變的話，再重新進行葉扇的調整，透過一連串的實驗數據，也充份證實我們的實驗結合粒子群演算法是可行的。

## 1.2 文獻回顧

溫度對於文物來講是一個相當重要的指標存在[3][4]，大部分的文物溫度越高材質變質的速度越快，例如：油畫環境溫度太高會造成油墨的溶解溶化，如果再加上環境溫度劇烈的變化，就像是大太陽完接著颱風天，對於文物來說絕對會產生不可逆的影響。濕度對於文物的影響在於[5][6]，如果環境濕度太高會造成大部分文物開始長霉菌，家裡的水泥牆壁在濕度太高的環境都會產生壁癌了，更不用說博物館裡保存的木頭材質或者紙張，但每種材質的種類也會分別產生出不同的需求濕度，也不是說濕度越低就越可以保存文物，對於獸類標本來說如果環境濕度太低會造成標本皮毛脆化因此瓦解。

當我們需要欣賞任何珍貴的文物都需要在有光的環境中[7][8]，光照對於文物來說是一個必備的東西，不管是太陽光或者人工照明，然而基於文物本身可承受的光線熱度都因材質有所不同，而以光線來講會產生不同波長的光照，像是紅外線太強的陽光會更加產生熱能，導致文物吸收熱能溶解溶化，紫外線太強的陽光會使得某些材質加速造成氧化現象，導致斷裂或彎曲等現象。

對於科博館來說幾乎所有文物都是在室內空間並使用濕溫控制空調來加以控管館內設定的濕度溫度，對於這些空調設備安裝已及電費消耗都是相當龐大

的負擔，依據博物館學季刊博館不管每天多少觀眾，每月昂貴電費是非常稀鬆平常的事情，近年來推動節能趨勢跟物聯網的概念[10][11]，如果可以使一些比較能保存在開放空間的文物，使用自動百葉窗系統來針對每天的溫濕度以及光照做調整，而不使用空調系統來強制調節溫濕度，想必可以節省許多能源。

粒子群演算法(Particle Swarm Optimization, PSO)是由 Eberhart 和 Kennedy 提出[15]，演算法的部分是使用粒子群最佳化法(Particle Swarm Optimization)，PSO 演算法是根據模擬大自然中鳥群和魚群群體覓食行為所發展出的演算法[16]，粒子群的個體稱之為粒子，而每個粒子代表的是當前空間可能的答案，粒子的移動除了本身慣性還會參考自身記錄的最佳解，以及參考群體裡面的最佳解，從而得出新的移動，直到最後收斂得群體最終最佳解。

我們論文要探討的就是在開放的環境中如何利用演算法來讓百葉窗達到適合的角度跟開合度，讓百葉窗達到博物館適合的溫濕度。以博物館最常見到的恐龍骨骼標本來說，基本上恐龍骨骼標本都是大型骨骼標本，因此無法用一般小型文物的恆溫恆濕設備來加以保護，所以如果將骨骼標本放置在開放空間，勢必需要一些 AI 來操控室內的溫濕度以及光照系統，確保文物損壞率達到最小。

## 第二章 電動窗簾之系統硬體架構

在這一章節，我們將對系統所使用到的相關硬體元件以及軟體設施做一簡介說明，包括系統原理、驅動電路 L298N、硬體設備、遙控器模組 IRremote、步進馬達等。

### 2.1 系統原理

本系統使用一般家庭用的小型百葉窗當作實驗設備的硬體資源，並連接步進馬達來加以控制百葉窗開合跟葉扇角度，而百葉窗的室內室外兩側都裝有溫濕度感測器及照度感測器，還有百葉窗有一個角度感測器來偵測百葉窗移動之角度，運算主機使用的是以小型計算機著稱的樹莓派(Raspberry PI)。

開始控制葉扇角度或者百葉窗開合之前必須先收集當下室內室外資訊，收集好的資訊就可以簡略判斷當天的天氣型態以及氣候型態，當判斷完天氣型態以及氣候型態之後，就到所屬的 PSO 演算法規則中來運算百葉窗當下適合的開合情況以及葉扇角度，當百葉窗到定位之後，在依據室內資訊來跟改變前室內資訊做比較看有沒有比較符合期望值，如果不相符重新再接收當下資訊並且重新運算直到前後資訊符合期望為止。系統架構圖如下。

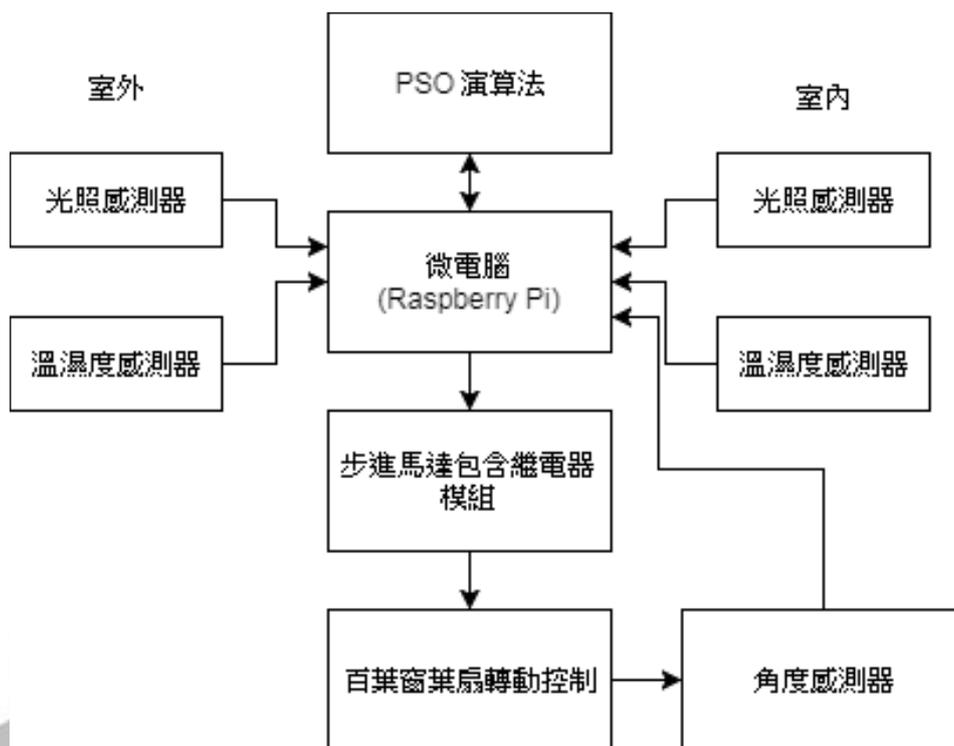


圖 2.1 單一百葉窗硬體架構圖

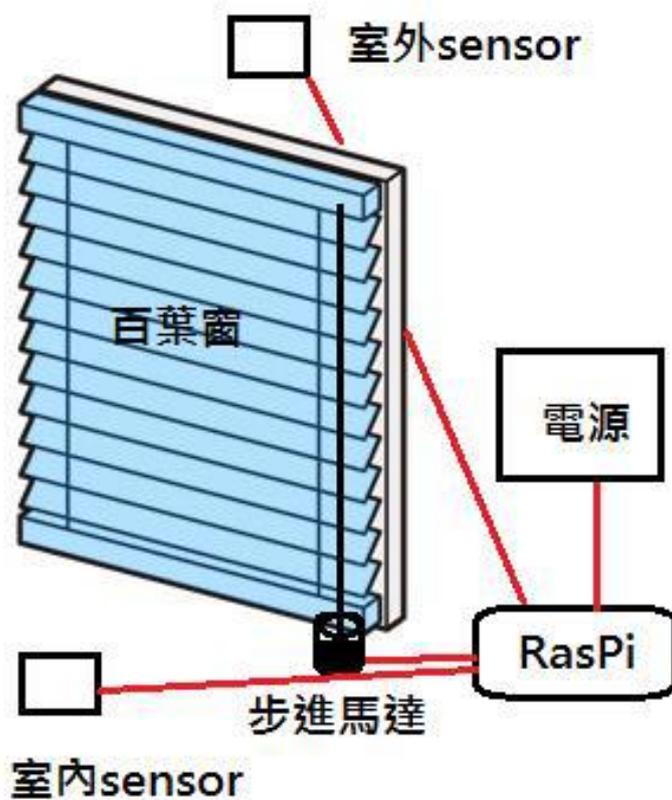


圖 2.2 單一百葉窗硬體示意圖

## 2.2 控制主機(Raspberry Pi)

Raspberry Pi 是源自一個開放源程式碼的硬體專案平台,該平台包括一塊具備簡單 I/O 功能的電路板以及一大堆的 Linux 軟體。Raspberry Pi 可以用來開發交互產品,比如它可以讀取大量的開關和感測器信號,並且可以控制電燈、電機和其他各式各樣的物理設備;Raspberry Pi 也可以開發出與 PC 一樣的周邊裝置,也可以運行在 Linux PC 上的軟體進行通信。Raspberry Pi 的硬體電路板可以自行焊接組裝,也可以購買已經組裝好的模組,而程式開發環境的軟體則可以從網上免費下載與使用。重點是它的價格非常的便宜,已今天的市價約要 1200 元台幣就可以買到。

本論文運行的作業系統是用 Raspbian,是由 Raspberry Pi 公司自行開發的作業系統上要使用 Raspberry Pi 之前必須先在 SD 卡上面安裝好作業系統,才可以做開機的動作,SD 卡相對於 Raspberry Pi 來說就像 PC 的硬碟,基本在 Raspberry Pi 上也包含 PC 各個功能,例如 HDMI 輸出,USB 插槽,網路插槽,聲音輸出等等。

Raspberry Pi 提供了基於 ARM 架構的 Debian, Arch Linux 和 Fedora 等等的發行版供大眾下載。還計劃提供支援 Python 作為主要程式語言,支援 BBC BASIC, C 和 Perl 等程式語言。



圖 2.3 Raspberry Pi 板(<https://www.raspberrypi.com.tw/>)

## 2.3 收集感測器訊息主機(Arduino)

Arduino 提供了簡單好用的 I/O 介面，並具可使用類似於 Java、C 語言等高階語言的開發環境。Arduino 是一種開放授權的互動環境開發技術，互動裝置其實無所不在，像是冷氣的恆溫裝置，便是使用感測器偵測環境溫度，進行室內溫度的自動調節；還有汽車使用的倒車雷達，過於靠近物體便會發出聲音警告駕駛者。這些裝置為生活增加不少安全及便利，互動能帶給使用者驚喜，有時會以藝術品的方式呈現在生活當中。

以往要處理相關的電子設備時，需要透過工程師，逐一由單一小元件拼湊出整個電路。大多數的設計工具都是為了工程師設計，除了電路外還需要廣泛

的知識，才有辦法完成電路。還好微處理器有了長足的進步，除了在使用上變得更加容易，價格上的減少更降低了學習的門檻。

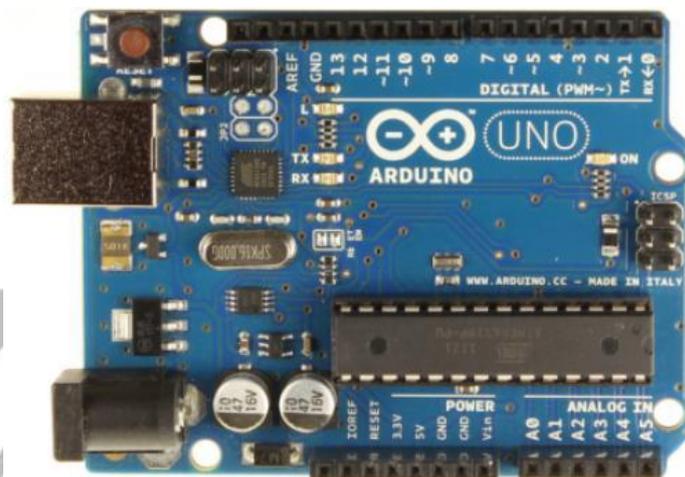


圖 2.4 Arduino uno 板(<https://www.arduino.cc/>)

## 2.4 感測器設備

### 2.4.1 IRremote 搖控器感測器

搖控器是由發射器接收器和微處理器組成。接收器和微處理器部分都在家電主機上，例如電視機。搖控器產生不同的編碼脈衝，有決定不同的按鍵，而搖控器輸出各種以紅外線為媒介的控制脈衝訊號，這些脈衝是特定的樣式，當接收器收到之後，將紅外線訊號進行放大，限幅，檢波，整行後送到微處理器，微處理器根據不同的訊號，再做不同按箭的反應。需要注意的是，根據發射不同編碼指令脈衝設計成相應的控制功能，這些功能是生產廠商預先設計好

的，再燒錄在搖控器裡面，所以各種搖控器是不能通用的，而且各家廠商會刻意做不同的調整，以免互相干擾。發送和接收端之間的方法，目前有兩種，分別是：

- IR 紅外線
- RF(Radio Frequency)射頻

要之道搖控器是哪一種類，可以用照相機的預覽鏡頭，對著搖控器，按下時發射端可以看到這個紅外線的亮度，但一般肉眼是看不到的。而另外一種 RF(Radio Frequency)射頻，就會看不到紅外線光，這是因為 RF 射頻搖控器是用頻率來做到這些事情，所以它的範圍比較廣，並可以穿透阻礙物，不會妨礙到資料傳輸。



圖 2.5 IRremote 模組(<http://shop.cpu.com.tw/>)

接下來我們就需要撰寫 Python 的程式語言，用來顯示出搖控器按鍵按下去的訊號。在這有幾個方法可以使用，有人會推薦使用 LIRC 的函式庫來處理搖控的動作和反應，但是考量到之後我們會集合其他的硬體，所以這一個論文我

們就使用 Python 的程式語言，來達到目的。

### 2.4.2 DHT-11 溫濕度感測器

DHT-11 是結合濕度計與溫度計的感測器，包含一個高性能 8 位元單晶片，量測到的溫濕度資料，會先轉換為數位訊號，再透過 data pin 腳將資料送出。DHT-11 使用很容易，但是必須注意每次抓取資料的時間要間隔 1~2 秒，不能太快。與 DHT-22 比較，DHT-11 精準度比較差，而且溫濕度測量範圍較小，但是體積小且價格便宜是它的優點，傳輸的距離又可長達 20M 以上，因此更適合用於學習溫、濕度感測器的應用，是十分方便的簡易測試元件。

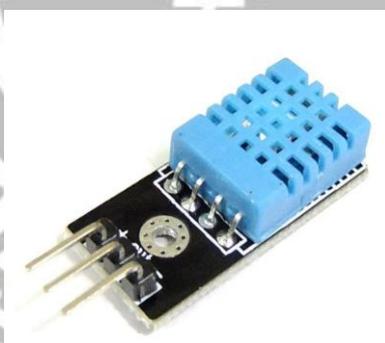


圖 2.6 DHT-11 溫濕度傳感器(<http://shop.cpu.com.tw/>)

### 2.4.3 GY-30 數位光強度檢測模組

光照感測器 GY-30 是一種用來偵測光照度的感測器單位為 Lux，包含一個 BH1750FVI 晶片，內建 16 位元的 AD 轉換器，量測到的光照資料，會先轉換為數位訊號，再透過 SDA pin 腳將資料送出。GY-30 使用前必須先導入 Wire.h 跟 BH1750FVI.h 的函式庫，然後 setAddress 設定在 0x23，並用 GetLightIntensity 來讀取光照值，光照值的範圍落在 0 到 65535。

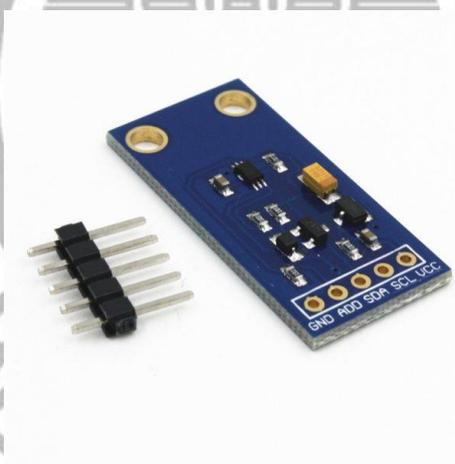


圖 2.7 GY-30 數位光強度檢測模組(<http://shop.cpu.com.tw/>)

### 2.4.4 GY-291 ADXL345 數字三軸重力加速度傾斜度模組

ADXL345 是一款小巧纖薄的低功耗三軸加速度計，非常適合移動設備應用，它可以在傾斜檢測應用中測量靜態重力加速度，還可以測

量運動或衝擊導致的動態加速度。使用 ADXL345 等數位輸出加速度計時，無需進行類比數位轉換，從而可以節省系統成本和電路板面積。低功耗模式支持基於運動的智能電源管理，從而以極低的功耗進行閾值感測和運動加速度測量。

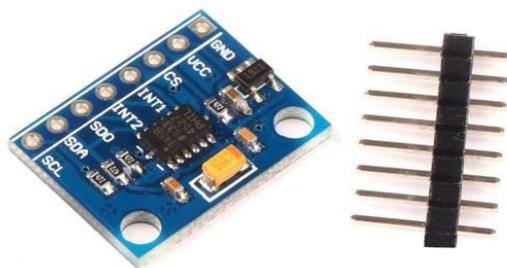


圖 2.8 GY-291 ADXL345 數字三軸重力加速度傾斜度模組

(<http://shop.cpu.com.tw/>)

### 2.4.5 步進馬達

步進馬達的種類依照結構來分可以分成三種：

- 永久磁鐵 PM 式 (permanent magnet type)
- 可變磁阻 VR 式 (variable reluctance type)
- 複合式 (hybrid type)



圖 2.9 步進馬達(<http://shop.cpu.com.tw/>)

PM 式步進馬達的轉子是以永久磁鐵製成，其特性為線圈無激磁時，由於轉子本身具磁性故仍能產生保持轉矩。

VR 式步進馬達的轉子是以高導磁材料加工製成，由於是利用定子線圈產生吸引力使轉子轉動，因此當線圈未激磁時無法保持轉矩，此外，由於轉子可以經由設計提高效率，故 VR 式步進馬達可以提供較大之轉矩，通常運用於需要較大轉矩與精確定位之工具機上，VR 式的步進角一般均為  $15^\circ$ 。

複合式步進馬達在結構上，是在轉子外圍設置許多齒輪狀之突出電極，同時在其軸向亦裝置永久磁鐵，可視為 PM 式與 VR 式之合體，故稱之為複合式步進馬達，複合式步進馬達具備了 PM 式與 VR 式兩者的優點，因此具備高精度與高轉矩的特性，複合式步進馬達的步進角較小，一般介於  $1.8^{\circ}\sim 3.6^{\circ}$  之間，最常運用於 OA 器材如影印機、印表機或攝影器材上。

步進馬達依定子線圈的數目可分成 2 相，3 相，4 相及 5 相式，小型步進馬達以 4 相式較普遍，高速型步進馬達則以 5 相式為主。在本論文使用 2 相步進馬達，雖然都是 2 相步進馬達，但是四線步進馬達跟五、六線步進馬達的驅動電路很不一樣，因為四線步進馬達是雙極馬達，雙極的意思是說，A 跟 B 相位的電流會有兩個方向。而六線式因為有共用腳，所以電流就只有單一方向了。

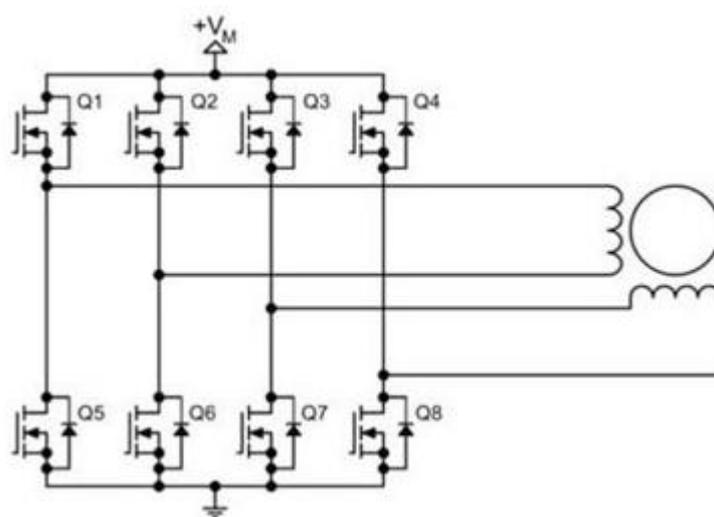


圖 2.10 2 相 4 線步進馬達(<https://www.orientalmotor.com.tw/>)

這種驅動電路其實也可以驅動單極步進馬達，雖然需要八顆電晶體，比單極 2 相步進馬達所需的四顆要多。但是適合量產，所以比較常用。

H 橋電路可以放大功率，還可以讓電流有兩種方向。假設現在只用一顆功率晶體控制一顆直流馬達，那麼因為電流方向固定的關係，所以馬達只能往一個方向轉。如果用了 H 橋電路，那樣可以有兩種電流流向，所以馬達也有兩種轉動的方向了。本論文使用的是一個五線式一個六線式。

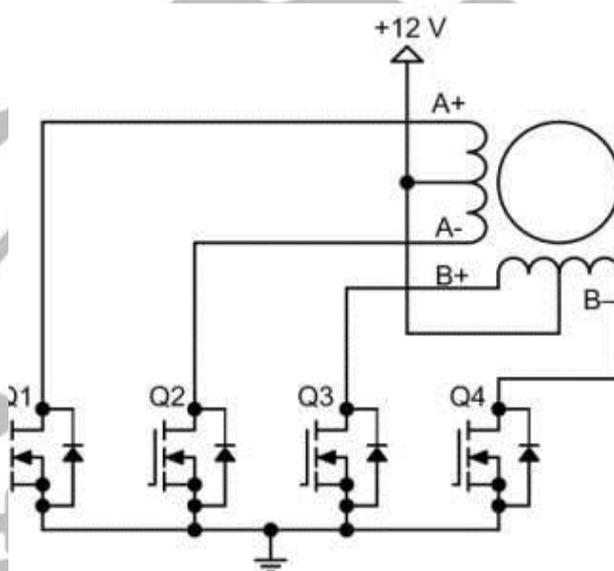


圖 2.11 2 相 6 線步進馬達(<https://www.orientalmotor.com.tw/>)

激磁順序(2相激磁)			
A+	A-	B+	B-
1	0	0	1
1	1	0	0
0	1	1	0
0	0	1	1

圖 2.12 2 相 6 線步進馬達激磁順序(<https://www.orientalmotor.com.tw/>)

## 2.5 樹莓派開發語言

本論文樹莓派是使用 Python，Python 是一套功能強大、直譯式並且物件導向的程式語言。Python 有幾項特點

### 1. 容易撰寫:

Python 具有許多物件導向的特性，然而並不要求一定得用物件導向的方式撰寫。撇開物件導向的議題不談，Python 通常可以靠比較短的程式碼完成比較多的功能，或者可以寫得比較清楚。

### 2. 功能強大:

Python 自 1990 年由 Guido van Rossum 在荷蘭的 [CWI](#) 開始發展以來，從 0.9 進步到今天的 2.4.2，不但累積了相當完整的標準程式庫（模組），更有無以計數的非標準模組，而且絕大部分都是開放原始碼的。單以內建的模組來講，從簡單的數學運算、字串處理、網際網路協定連線、網際網路資料處理、各種壓縮格式，以及 POSIX 與主要作業系統的支援功能等等，含括的範圍非常地廣泛。

### 3. 跨平台:

各種主要的作業系統都支援 Python。Python 程式常常不需要修改，便可以同時在 Linux 與 Windows 平台上執行，即使撰寫 GUI 程式（透過

PyGTK, wxPython 等 binding) 也是一樣。所撰寫的 Python 程式透過標準的 distutils (模組) 進行包裝後，用標準的方式即可安裝於各種平台；在 Windows 下更可以自動產生方便的可執行 installer。

#### 4. 容易擴充:

Python 算是執行效率不錯的直譯式語言，但畢竟比不上 C 和 Fortran。然而只要我們想，大可以 C/C++ 或 Fortran 撰寫高效率的模組；這些模組的使用方式，與內建模組以及用 Python 撰寫的模組完全一樣。最好的是，撰寫的方法並不困難。

## 2.6 Arduino 開發語言

本論文使用的開發語言是 Arduino IDE，Arduino IDE 是由官方網站上面提供的軟體系統，裡面開發的語言類似 java 跟 C 等高階程式語言。裡面可以輕鬆撰寫程式碼，並藉由 USB 傳輸線來連接 Arduino 板跟本地端電腦，只要將程式上傳至 Arduino 板就可以讓 Arduino 持續的執行所撰寫的程式。IDE 裡面也可以 include 許多已經寫好的.h 檔跟.cpp 檔，讓各種感測器跟程式撰寫更佳的方便。

# 第三章 以 PSO 演算法做為博物館之光照及溫濕度控制

粒子群最佳化法(Particle Swarm Optimization)中的每個體稱稱為「粒子」(particle)，是代表解空間(solution space)中的一個可能的解。每個粒子的移動除了出自本身的慣性(inertia)外，還參考個體本身最佳經驗移而產生認知學習(cognitive learning)的遷移，也參考群體整體最佳經驗做社會學習(social learning)，迭代演化前進，最後收斂而得到最佳解。

PSO 的發現源於自然界生物活動的觀察，模擬鳥群、魚群之覓食得群體行為(social behavior)所發展出來的演算法。這類源自觀察生物群之個體、族群間的互動行為，是一種模擬自然生物群體系統的演算法。這類源自模擬生物群體活動來找 optimal solution 的方法，稱為 swarm intelligence，有人更廣泛的稱為 Nature Inspired Computing (NIC)。



## 3.1 PSO 簡介

PSO 為 meta-heuristic search 的演算法，所謂的 heuristic 是指以(實際搜尋)的經驗進行求解的方法；以「亂數」為基礎的搜尋技巧則稱為 meta-heuristic，

並以反復嘗試增進可能解品質的方式，達到問題最佳化的運算方法。

Meta 一詞可做【後設】解，通常【xx 的 xx】我們就稱為 meta XX。如：

後設認知，是指對於自己認知知覺的認知，就是：你知不知道自己知不知道的能力就為【後設認知】。

Meta-Heuristic 可稱作【以實證方法求解的實證解法】。這裡的【實證、經驗、Heuristic】是一種【實證方法】。就是在求最佳解時，將所有可能解代入條件函數後比較而決定出(可以接受的，未必是絕對)的最佳解。換個方式來說，**Metaheuristics** 利用外層(上層、概念層)的架構來控制內層(底層、操作、實作的)的一種 Heuristic 解法。以 Nature Inspired Computing 來說，就是以自然的觀察現象，如：基因演化、魚鳥群、螞蟻群、蜜蜂群覓食等現象，做為外層的控制架構的經驗式演算法。底層的 heuristic approach 則為以隨機方式為主的 Stochastic learning。除了 NIC 之外，也有以音樂協調性為啟發的 harmony search 等。

粒子群最佳化屬於 meta-heuristic，不必對要最佳化的問題設立任何假設，就可以在空間中搜尋很大範圍的可能解。然而，如 PSO 等的 meta-heuristics 方法，卻不保證一定能找到絕對的最佳解。具體而言，粒子群不使用問題的梯度資訊(gradient)，這意味著 PSO 不需要以問題之可微分為前提，這點截然不同於

須有可微分為前提的傳統最佳化方法，如：gradient descent 和 quasi-newton methods。因此，PSO 具有更強大解決一般最佳化問題的能力，包括某些傳統最佳化法所無法處理的最佳化問題。

### 3.2 PSO 基本概念

PSO 的概念場景：假設有一群魚在池塘裡以隨機巡遊的方式搜尋著食物，這群魚所處的池塘是一個  $n$  維的解空間(solution space)，也是目標函數和適配函數(fitness function)所處的定義空間。池塘中食物的大小代表解的優劣，為位置的函數並視為適配值。池塘裡最大食物為最佳解，這群魚透過參考並分享個體最佳化的訊息，逐步收斂而搜尋到最佳值的位置。

PSO 的求解過程：任何一隻魚在巡遊搜尋最佳解的過程中，會記下自己所經歷的最大的那塊為區域最佳值(local optimal)，所有區域最佳值中的最大值為訊息分享而得知的全域最佳值(global optimal)，也會被記錄下來為群體的記憶。

PSO 藉由記憶訊息的分享，亦即經由與個別最佳(local best)、群體最佳(global best)的比較，來調整前進的方向，決定下一步搜尋的速度，包括下一步移動的方向和距離，迭代演進搜尋找到最優解。

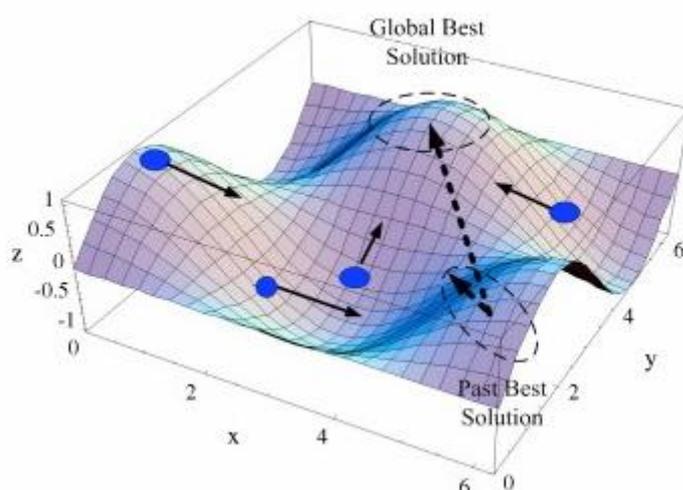


圖 3.1 魚群 PSO 概念圖(<http://blog.xuite.net/metafun/life/58295146>)

PSO 以一群隨機粒子(隨機解)為起始，迭代演進的過程中不斷產生往最佳解前進的速度。PSO 中的粒子由局部、群體的最佳經驗值的驅動牽引，在 solution space 中搜索。每次迭代中，粒子通過跟蹤兩個"極值"來更新自己的位置。粒子本身所經歷的最優解稱為個體極值 pBest；整個種群目前已知的最佳解為群體極值 gBest。

PSO 往最佳解移動的速度由三項力量共同決定，其速度內涵實乃包括：方向和距離(大小)。這三項驅使個體粒子往最佳解移動的動力為：目前速度(慣性)、區域極值、群體極值

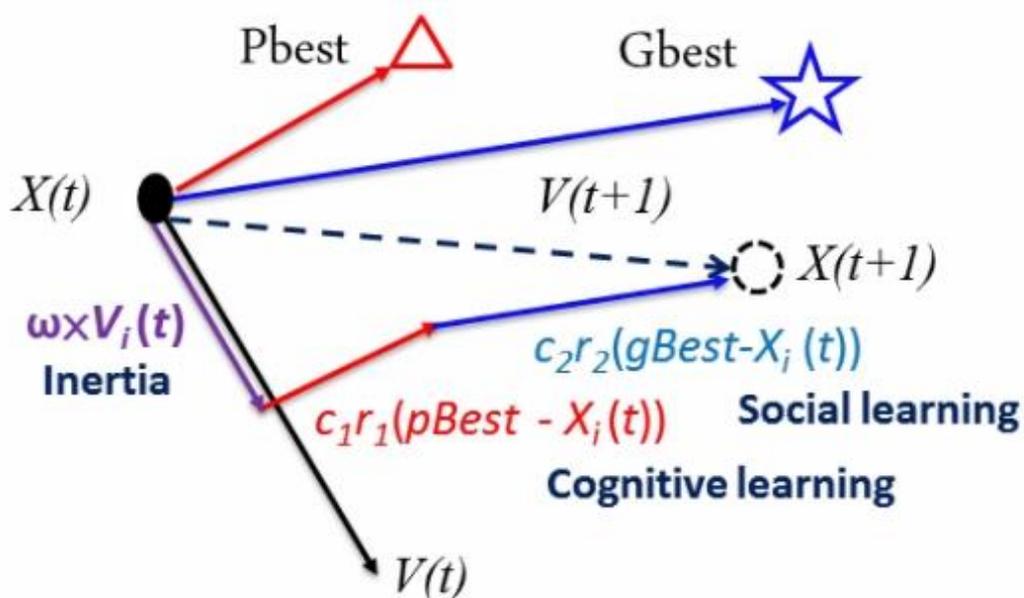


圖 3.2 PSO 往最佳解移動向量圖(<http://blog.xuite.net/metafun/life/58295146>)

1.與區域極值的差 2.與群體極值的差 3.目前的速度的慣性

這 3 者共同決定新的速度如下：

$$V_i(t+1) = \omega * V_i(t) + c_1 r_1 (pBest - X_i(t)) + c_2 r_2 (gBest - X_i(t)) \quad (3.1)$$

然後，再算出新的解(位置)  $X_i(t+1) = X_i(t) + V_i(t+1)$  (3.2)

式子中， $\omega$  為慣性權重(inertia weight)， $c_1$  為認知係數(cognitive parameter)， $c_2$  為社會係數(social parameter)。各自有的特定的值或設定值的方式，其對 PSO 效能的影響也是很多研究探討的議題。

PSO 演算法 pseudo code 如下

For each particle

    Initialize particle

END

Do

For each particle

    Calculate fitness value

    If the fitness value  $<$  (pBest)

        then (pBest) = the fitness value

End

/\* Choose the particle with the best fitness value of all the particles as the

gBest\*/

If (pBest) $<$  (gBest)

    then (gBest) = (pBest)

For each particle

    Calculate particle velocity according equation (1)

    Update particle position according equation (2)

End

While maximum iterations or minimum error criteria is not attained

圖 3.3 PSO 演算法(<http://blog.xuite.net/metafun/life/58295146>)

### 3.3 PSO 參數在實務上的設置、調控

執行 PSO 演算法必須先決定  $c_1$ ,  $c_2$  和  $v_{max}$  的值。選取的值對 PSO 的收斂速度和法能夠找到最佳值有一定影響，且不同型態的問題也許會有不同的適當的值。

1. 粒子數: 一般取 20~40. 其實對於大部分的問題 10 個粒子已經足夠可以取得好的結果, 不過對於比較難的問題或者特定類別的問題, 粒子數可以取到 100 或更大。
2. 中止條件: 設定最大循環數或最小的誤差來結束搜尋, 中止條件通常也隨問題的條件而異。
3.  $v_{max}$ : 最大速度, 決定粒子在一個循環中最大的移動距離, 通常設定為粒子的範圍寬度, 例如上面的例子裡, 粒子  $(x_1, x_2, x_3)$   $x_1$  屬於  $[-10, 10]$ , 那麼  $v_{max}$  的大小就是 20。在每一維粒子的速度都會被限制在一個最大速度  $v_{max}$ , 如果某一維更新後的速度超過用戶設定的  $v_{max}$ , 那麼這一維的速度就被限定為  $v_{max}$ 。
4. 慣性權重:  $\omega$   
若  $\omega=0$ , 則粒子的速度取決於個體最佳經驗和群體最佳經驗; 這意味著粒子可快速更改其速度, 立即從已知的最佳位置迅速改變位置到很遠的

更佳位置。因此，較小的慣性權重會有侷限於局部搜索(local search)的傾向，稱為「exploitation」探索。若設定較高的 $\omega$ ，則粒子速度的改變率較低，亦即它有明顯遵循其原來路徑的「慣性」，甚至在得知更佳的適配值 fitness values 後，仍固著於原來的的位置。因此，高慣性權重有全面搜索(global search)的傾向，稱為「exploration」探索。較大的慣性權重使 PSO 的使粒子有較大的運動慣性，粒子群因此具有較強的探索力(exploration)，傾向執行較大範圍的全面搜索(global search)。而較小的慣性權容易抑制粒子的運動慣性，則具備較強的局部搜索(local search)傾向的探勘力(exploitation)。

### 3.4 百葉窗 PSO 設計

依據文物室內溫度越接近20到25度之間越能保持骨骼材質的穩定，濕度越接近50%越能保持不脆化不受潮不發霉，光照越接近400Lux越可以降低紅外線熱能的影響，所以依據上面設定的標準值來做調整。

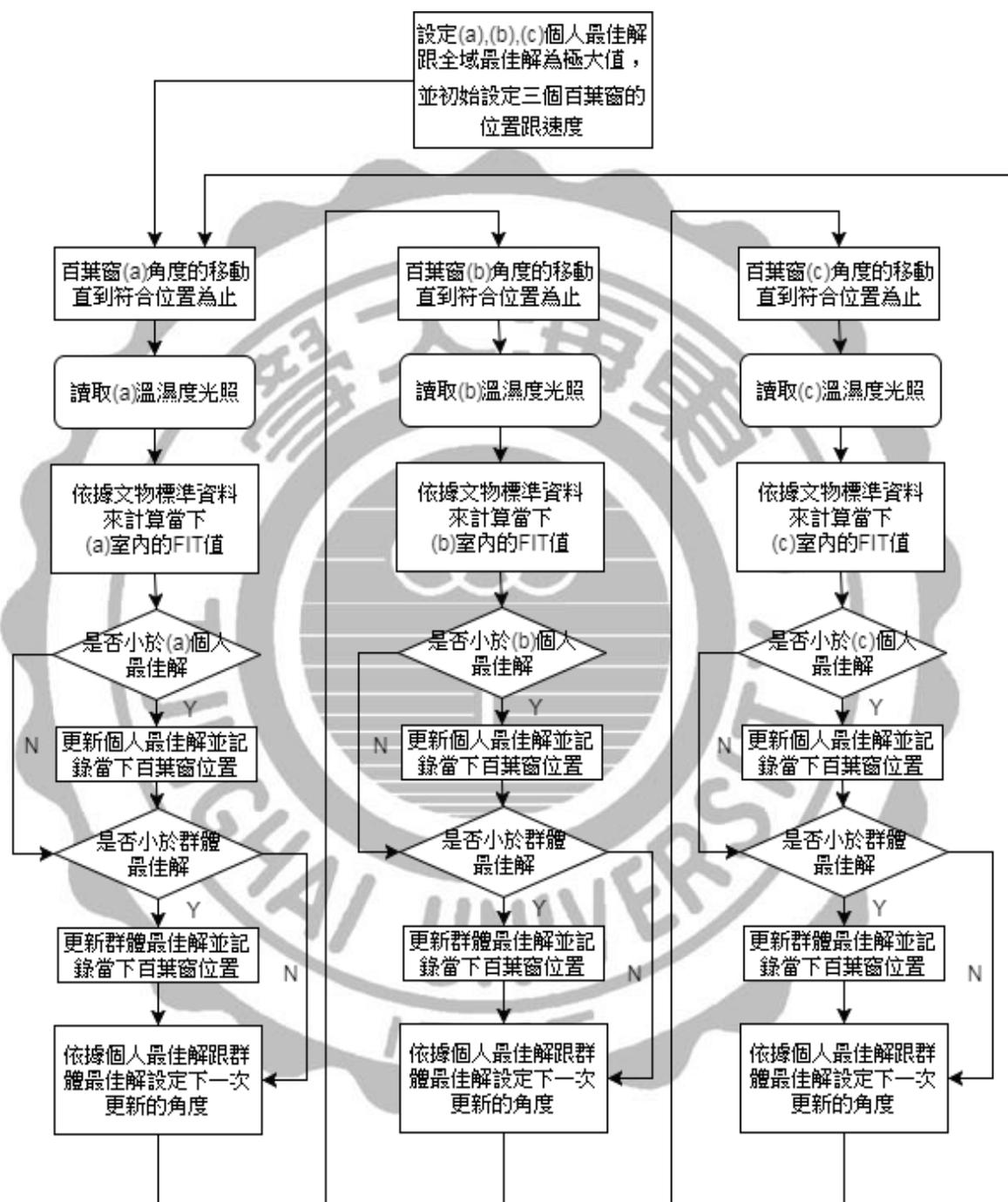


圖 3.4 系統流程圖

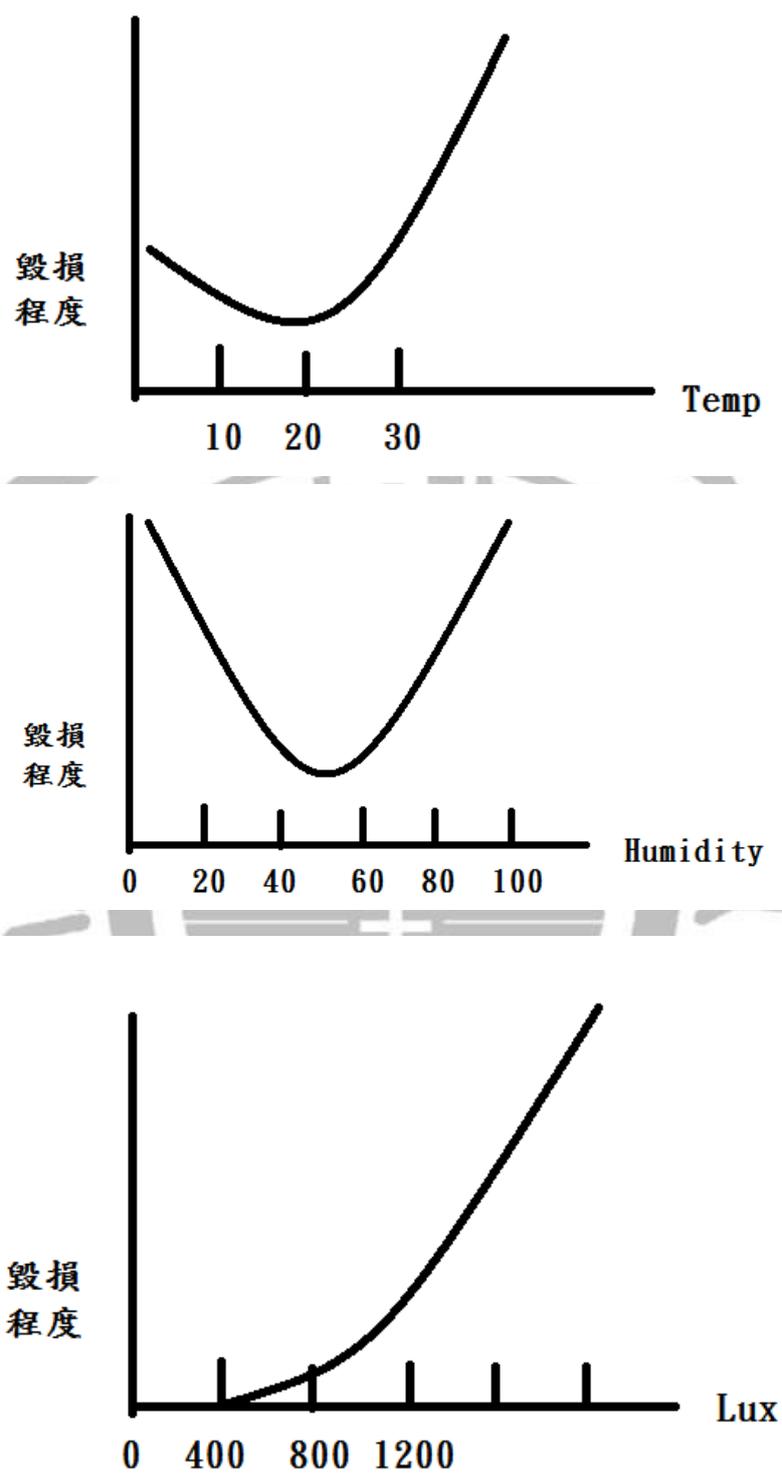


圖 3.5 文物跟各項數據毀損程度比較圖

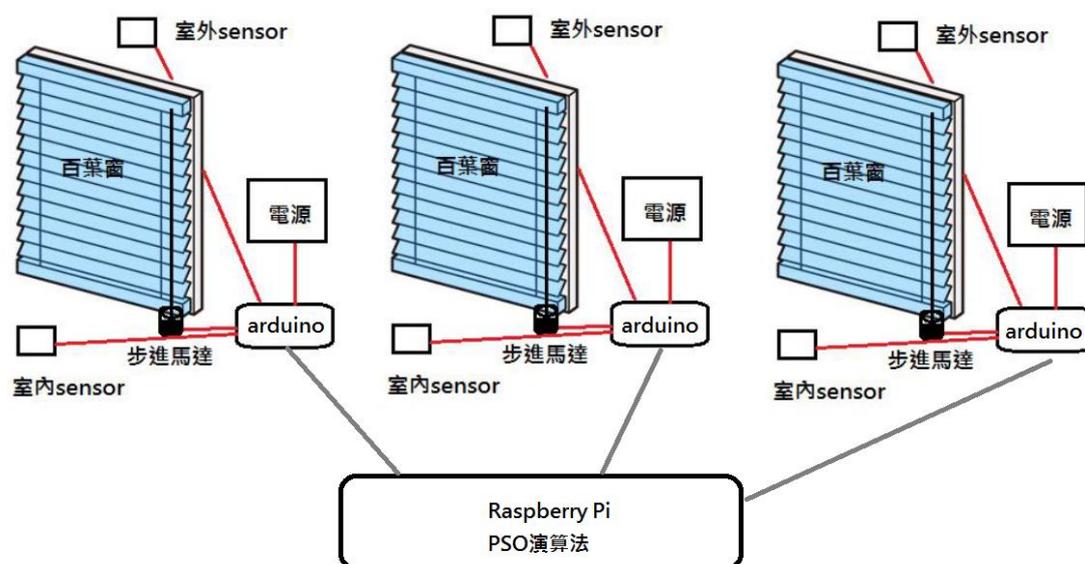


圖 3.6 三個百葉窗硬體架構圖

三組百葉窗各自使用 Arduino 來接收室內室外的感測值，並且傳入樹莓派來進行參數運算，由左至右依序傳室內外感測值給樹莓派，一開始先將角度傳感器數值傳給樹莓派，樹莓派先檢視當下回傳的角度是否為初始角度，如果不是就將角度調整到初始設定角度為止。在各個百葉窗未運算 PSO 前先設個人起始最佳解為 10000、群體最佳解也為 10000，當第一次執行之後就會依據當下的解開始跟個人最佳解和群體最佳解進行比較，10000 為相當大的數值基本來說不太可能達到，所以第一次就會將個人最佳解跟群體最佳解更新，接著每個百葉窗都會執行一次，並且判斷哪個百葉窗初始角度的解為最佳值並更新群體最佳解。

百葉窗 PSO 演算法的部分，三個百葉窗同時進行並且設定百葉窗的起始角

度各為 0 度、45 度、-45 度，然後角度改變初始速度為 2 度/秒、2 度/秒、-2 度/秒，藉由當下室內溫度跟文物標準溫度 20 度的差距值，如果當下溫度大於標準溫度就乘上 100 的權重數值，如果當下溫度小於標準溫度就乘上 50 的權重數值，這樣就可以得出溫度的 FIT 函數值。然後再用當下室內濕度來跟文物標準濕度 50% 的差距值，再乘上 20 的權重數值可以得出濕度的 FIT 函數值。最後如果當下室內光照值跟/文物可承受光照值小於 1000lux 的差距就將當下室內光照值跟文物可承受光照值的差距值直接當作光照的 FIT 函數值，反之如果當下室內光照值跟文物可承受光照值大於 1000lux 就將當下室內光照值跟文物可承受光照值的差距值乘上 2 當作光照的 FIT 函數值，將三個函數值加相即可得到完整 FIT 值，如果這次是第一次動作就將 FIT 值設成個人最佳解並且記錄角度，也是全域最佳解並且記錄角度。

$$\text{FIT} = \text{abs}(\text{light} - 400) + (\text{temptrue} - 25) + (\text{humility} - 50) \quad (3.3)$$

接下來計算窗簾的調整角度[10]，將每次計算的完整 FIT 值來跟個人最佳解的值和群體最佳解的值來做比較。窗簾的調整角度  $X_{i(t+1)}$  藉由個人最佳解角度 (pBest) 減去當前角度  $X_{i(t)}$  算出跟個人角度的差距值，再算出群體最佳解角度 (gBest) 減去當前角度  $X_{i(t)}$  的差距值，將兩個差距值各乘上像對應的系數再相加並加上當前葉扇速度  $V_{i(t)}$  乘上權重為新的速度  $V_{i(t+1)}$ ，然而窗簾的調整角度

$X_i(t+1)$ 為當前的角度  $X_i(t)$ 加上新的速度  $V_i(t+1)$ 。

$$\begin{aligned} V_i(t+1) = & \omega * V_i(t) + c1r1(pBest - X_i(t)) \\ & + c2r2(gBest - X_i(t)) \end{aligned} \quad (3.4)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (3.5)$$

做完窗簾調整角度，重新計算FIT值，之後來跟個人最佳解角度的FIT值做比較來判斷，如果FIT值比個人最佳解角度的FIT值還要小，就更新個人最佳解，然後再跟全域最佳解的FIT值比較判斷是否要更新全域最佳解的FIT值，之後再重新計算窗簾的調整角度持續循環下去。

停止角度更新條件有 FIT 值完全為 0 代表的意思等同於完全符合設定的溫濕度以及光照度，令一個條件為前兩次葉扇都有被更新方向代表葉扇往兩個方向移動都是把 FIT 值加大越離標準越遠，葉扇停止之後持續監測 FIT 值直到 FIT 值被更改並且 FIT 值加大就代表環境的溫濕度跟光照被更新需要重新設定葉扇角度。

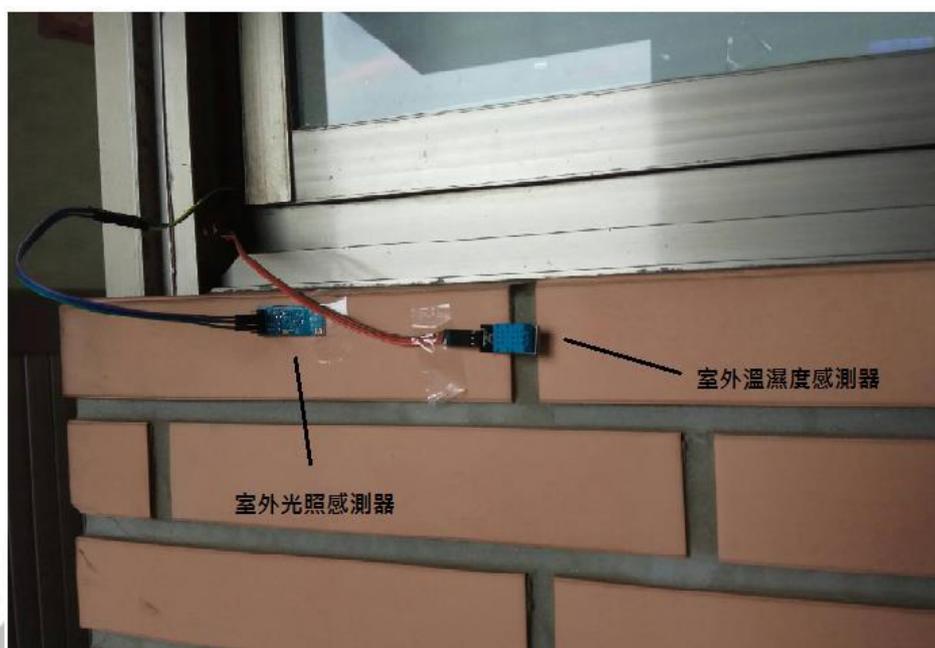


圖 3.7 室外感測器



圖 3.8 室內感測器包含控制主機百葉窗關閉



圖 3.9 百葉窗開啟



# 第四章 實驗結果與討論

## 4.1 三個百葉窗 PSO 訓練實驗結果

對於三個百葉窗來做 PSO 的訓練，環境設定在早上八點，然後將三個百葉窗依據個人學習跟互相學習來達到最佳解的位置  $P_a(i)$  為百葉窗 a 的疊代、 $P_b(i)$  為百葉窗 b 的疊代、 $P_c(i)$  為百葉窗 c 的疊代， $i$  為各個百葉窗的疊代次數，FIT 值越低代表越接近索設定的標準環境光照以及溫濕度，初始角度跟速度都隨機設定。

由下面三表格跟收斂圖可以得知，個人最佳解和全域最佳解都被持續更新，從百葉窗(a)第一次執行開始，更新自己的個人解跟全域解，再來換百葉窗(b)來更新自己的個人解，接著判斷 FIT 值是否有比剛剛被更新的全域解還要低，如果有就更新沒有就不更新，再來換百葉窗(c)來更新自己的個人解，再持續判斷是否 FIT 值有比全域解還要低，經過三個百葉窗更新完之後就可以知道，哪個初始位置的解為全域最佳解，並且三個百葉窗都是第一次執行所以當下位置都是個人最佳解，做完第一圈判斷後，再重新回到百葉窗(a)依據算出來的下一個更新位置來重新計算 FIT 並且重新判斷所有百葉窗的各個解是否要更新個人最佳解跟全域最佳解，

表 4.1 百葉窗(a)位置改變記錄表格

X(t)	V(t)	Pb	Gb	光照	溫度	溼度	FIT
0	2	0	0	641	26	52	381
1	1	1	-45	638	26	52	378
-44	-45	-44	-44	400	25	53	60
-66	-22	-44	-44	276	25	54	204
-33	33	-44	-44	500	25	53	160
-38	-5	-44	-44	460	25	53	120
-52	-14	-44	-44	350	25	54	130
-43	9	-44	-44	410	25	53	70
-41	2	-44	-44	425	25	53	85
-46	-5	-44	-44	385	25	53	75
-45	1	-44	-44	390	25	53	70
-43	2	-44	-44	410	25	53	70
-44	-1	-44	-44	400	25	53	60
-44	0	-44	-44	400	25	53	60
-44	0	-44	-44	400	25	53	60
-44	0	-44	-44	400	25	53	60

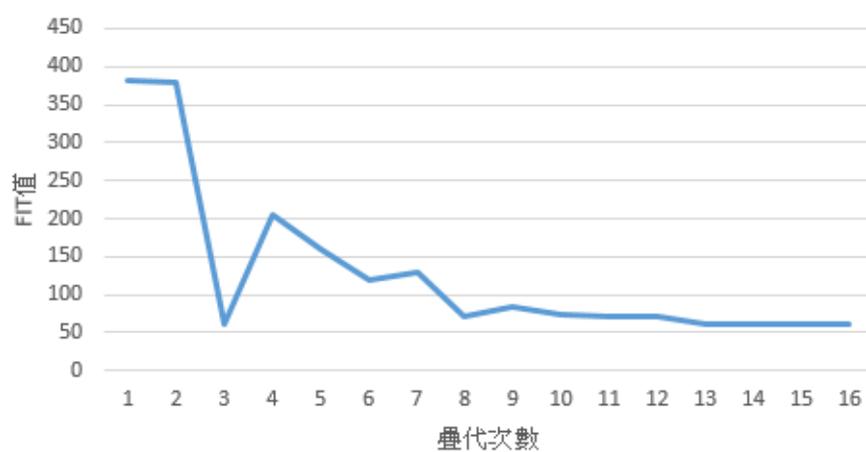


圖 4.1 百葉窗 a 的收斂圖(初始角度 0 速度 2)

表 4.2 百葉窗(b)位置改變記錄表格

X(t)	V(t)	Pb	Gb	光照	溫度	溼度	FIT
45	2	45	45	276	25	54	204
46	1	45	-45	270	25	54	210
-45	-91	-45	-44	390	25	53	70
-80	-44	-45	-44	180	25	54	300
-31	49	-45	-44	530	25	53	190
-33	-2	-45	-44	500	25	53	160
-57	-24	-45	-44	320	25	54	160
-44	13	-44	-44	400	25	53	60
-38	6	-44	-44	460	25	53	120
-47	-9	-44	-44	380	25	53	80
-46	1	-44	-44	385	25	53	75
-42	4	-44	-44	417	25	53	77
-44	-2	-44	-44	400	25	53	60
-45	-1	-44	-44	390	25	53	70
-44	1	-44	-44	400	25	53	60
-44	0	-44	-44	400	25	53	60

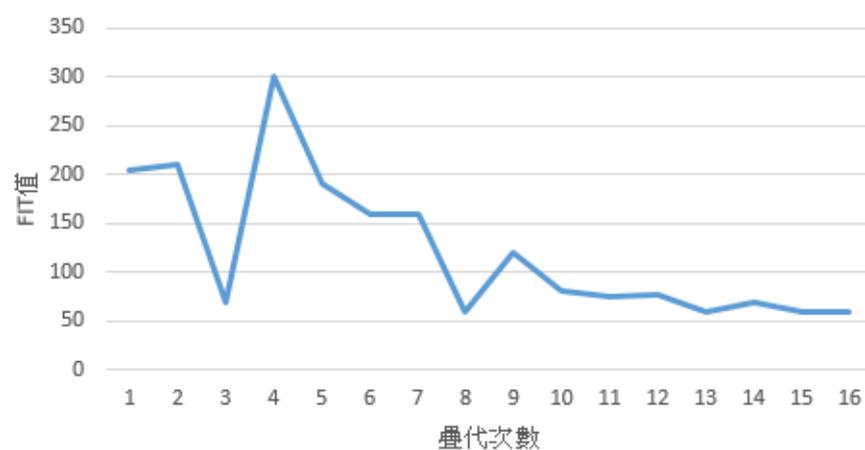


圖 4.2 百葉窗 b 的收斂圖(初始角度 45 速度 2)



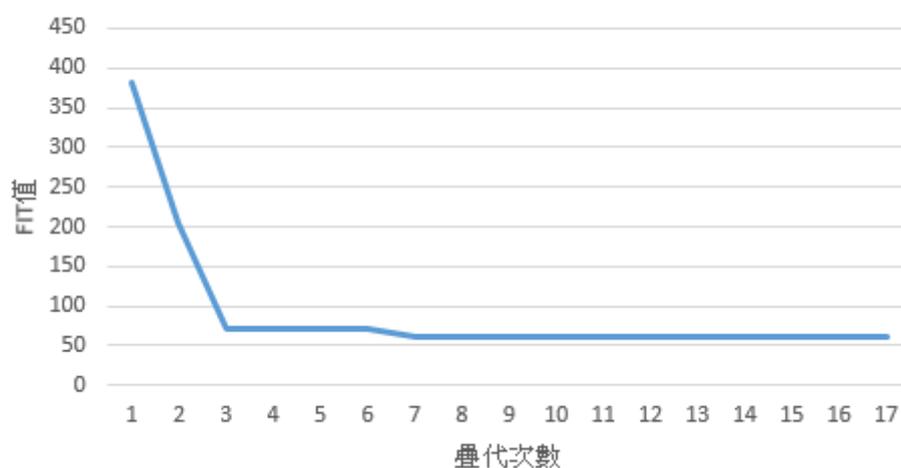


圖 4.4 百葉窗 gbest 的收斂圖

經過一連串的动作可以看出百葉窗(c)是最快收斂完成，因為百葉窗(c)一開始的位置是最接近最佳解的位置所以可以很快的收斂，雖然百葉窗(a)一開始經過兩次疊代剛好就已經達到全域最佳解位置，但是他的初始速度還是存在，所以他還是會依據前一次經驗來持續更新相同動作，結果慣性動作讓百葉窗(a)又離開了最佳解的位置，但是百葉窗(a)還是率先找到全域最佳解，代表著最先找到全域最佳解的百葉窗也不一定是最快收斂的百葉窗，只是因為他剛好訓練過程中經過並得到全域最佳解。

## 4.2 PSO 訓練後單一百葉窗之數據結果

表 4.4 窗簾角度在 0 的時候測量室內室外三個參數值

時間	室外			調整後的室內			FIT 值	角度
	光照	溫度	濕度	光照	溫度	濕度		
5:00	253	25	60	0	25	55	100	0
6:00	1548	25	59	42	25	54	80	0
7:00	4522	26	57	348	26	55	100	0
8:00	12797	27	53	641	26	54	421	0
9:00	16248	29	50	812	26	52	552	0
10:00	15483	30	48	806	27	50	606	0
11:00	16812	31	46	874	27	48	714	0
12:00	21574	31	45	942	28	46	882	0
13:00	14267	30	48	684	27	47	514	0
14:00	9754	30	50	572	26	50	372	0
15:00	7841	29	52	461	26	52	201	0
16:00	8452	28	51	477	26	52	217	0
17:00	3741	28	53	316	25	53	60	0
18:00	447	27	55	25	25	54	80	0

表 4.5 窗簾角度依據 PSO 演算法更新

時間	室外			調整後的室內			全域 FIT 值	角度
	光照	溫度	濕度	光照	溫度	濕度		
5:00	253	25	60	0	25	55	100	32
6:00	1548	25	59	2	25	54	80	37
7:00	4522	26	57	146	25	55	100	43
8:00	12797	27	53	398	25	54	80	64
9:00	16248	29	50	424	26	52	164	68
10:00	15483	30	48	431	26	50	131	73
11:00	16812	31	46	442	26	50	142	82

12:00	21574	31	45	453	27	49	273	85
13:00	14267	30	48	421	26	50	121	76
14:00	9754	30	50	412	26	51	132	46
15:00	7841	29	52	391	25	53	60	46
16:00	8452	28	51	389	25	52	40	42
17:00	3741	28	53	125	25	53	60	45
18:00	447	27	55	3	25	54	80	60

由上兩表格可以得知開窗簾的差異大部分都落差在太陽升起的部分早上 6 點到下午 6 點之間，陽光對於 FIT 值影響最為大，但是光線的強度也會間接影響環境中的溫度以及濕度，在兩個表格沒陽光的部分幾乎都一模一樣，會有差異的基本上都是因為當時空氣流動造成的影響，但是可以藉由兩個比對發現窗簾角度設為 0 的溫度變化相較於有角度變化的大，是因為太陽光被遮蔽的較少，造成感測器連帶的被熱輻射給拉高溫度，有時候因為陽光角度或者折射的關係感測器被直射可能突破 40 度以及濕度可能低於 10%，這些些微的因素都可能造成文物損壞，對於 PSO 百葉窗的 FIT 值來說是全域最佳解，所以不一定每個百葉窗都是依照最佳解的角度，可能因為陽光照射的範圍或角度的依據各自移到最佳解，角度的落差可能會差到 20 到 30 度，原因是因為窗戶玻璃光的折射以及附近地板的反射造成的。

# 第五章 結論及未來發展

## 5.1 結論

在現今的科技進步環境下，一般傳統物品結合環境參數的IOT產物已經越來越多，然而結合AI的產物卻是更加需要的，讓物品可以隨著人的習性去自動產生應有的對應措施，在將生物特性得出的演算法加入原本實體物品，就可以得到新的AI產物，然後傳統的物品使用習慣一時之間要替換掉是一個相當不容易的事情，所以本論文選擇在傳統的物品上加上微型電腦和感測器，加上感測器就可以得知所有想得知的环境參數，微型電腦是用來執行大量的演算法運算，微型電腦或許效能還是沒有一般電腦效能好但是對於演算法的運算已經相當足夠，然而AI的設備並不是僅僅作用於方便操控，現在更多的議題都是在談論節能，本論文的選擇就是將所有議題納入考量之中，選擇博物館也是因為大型公共建築基本上最急迫的需求都是節能的問題，對於博物館來說保存文物須要耗費相當大的能源是不可避免的，所以基於無法避免的問題來做相當的研究。

## 5.2 未來展望

對於未來展望是希望能結合所有冷氣、窗戶、除濕機加濕機來做環境以及參數的控制，這樣對於博物館來說只需要控制主機情境或文物設定就可以對於文物來做適當的處理，不需要花費大量人力就可以達到利用自然環境節能的成果。

## 參考文獻

- [1]. Lee, A., Angeles, C., Talampas, M., Sison, L., and Soriano, M., “MotesArt: Wireless Sensor Network for Monitoring Relative Humidity and Temperature in an Art Gallery”, IEEE International Conference on Networking, Sensing and Control (ICNSC 2008), April 2008, pp. 1263-1268.
- [2]. J. Kennedy, and R. C. Eberhart, “Particle swarm optimization”, Proc. IEEE International Conf Center, Piscataway, NJ, 1995, pp. IV: 1942-1948.
- [3]. Rodríguez, L., Brito, L., and Gouveia, B., “The WISE-MUSE Project: Environmental Monitoring and Controlling of Museums based on Wireless Sensors Networks. Electronic Journal of Structural Engineering (EJSE), Special Issue on Sensor Network for Building Monitoring: From Theory to Real Application”, October 2009, pp. 46 –57.
- [4]. Brito, L., Rodríguez, L., Santos, F., and Fernandes, R., “Environmental Monitoring of Museums Based on Wireless Sensor Networks”, Proc. 4th International Conference on Wireless and Mobile Communications (ICWMC 2008), IEEE Computer Society Press, Athens, Greece, July 27 -August1, 2008, pp. 364-369.
- [5]. 梁光余，2004，博物館建館容易經營難  
[http://web2.nmns.edu.tw/PubLib/Library/quarterly/200404\\_139.pdf](http://web2.nmns.edu.tw/PubLib/Library/quarterly/200404_139.pdf)

- [6]. 王鵬揚，2002，保存環境體檢—談國美館的溫溼度環境診測  
[https://www.ntmofa.gov.tw/ntmofapublish\\_1047\\_384.html](https://www.ntmofa.gov.tw/ntmofapublish_1047_384.html)
- [7]. 張琳，2012，文物保存溫溼度規範因應節能趨勢之探討  
[http://web2.nmns.edu.tw/PubLib/Library/quarterly/201210\\_183.pdf](http://web2.nmns.edu.tw/PubLib/Library/quarterly/201210_183.pdf)
- [8]. Hu, A., and Servetto, S., "Algorithmic Aspects of the Time Synchronization Problem in Large-Scale Sensor Networks", *Mobile Networks and Applications*, 2005 Springer Science + Business Media Inc., Vol. 10, No. 4, August 2005, pp. 491-503.
- [9]. 交通部中央氣象局  
[http://www.cwb.gov.tw/V7/knowledge/encyclo\\_pedia/me005.htm](http://www.cwb.gov.tw/V7/knowledge/encyclo_pedia/me005.htm)
- [10]. Rodríguez, L., Brito, L., and Gouveia, B., "The WISE-MUSE Project: Environmental Monitoring and Controlling of Museums based on Wireless Sensors Networks. *Electronic Journal of Structural Engineering (EJSE)*, Special Issue on Sensor Network for Building Monitoring: From Theory to Real Application", October 2009, pp. 46–57.
- [11]. Ciolfi, L. and L. Bannon (2003), "Learning from Museum Visits: Shaping Design Sensitivities", in Jacko, J. and C. Stephanidis (eds) *Proceedings of HCI International 2003-Vol.1*, Crete, June 2003
- [12]. J.-Y. Son, J.-H. Park, K.-D. Moon and Y.-H. Lee, "Resource-aware smart home

- management system by constructing resource relation graph," IEEE  
Transaction on Consumer Electronics, vol. 57, no. 3, pp. 1112-1119, 2011.
- [13]. O. Elma and U. S. Selamogullari, "A new home energy management algorithm  
with voltage control in a smart home environment," Energy, vol. 91, pp. 720-731,  
2015
- [14]. Artificial Intelligence (Chipsets) Market by Technology (Deep Learning  
Robotics Digital Personal Assistant Querying Method Natural Language  
Processing Context Aware Processing) Offering End-User Industry and  
Geography - Global Forecast to 2022. MarketsandMarkets 2016/11.
- [15]. R. C. Eberhart, and J. Kennedy, "new optimizer using particle swarm theory",  
Proc. Sixth International Symposium on Nagoya, Japan, 1995, pp.39-43.
- [16]. Particle Swarm Optimization  
<http://jeremy.fix.free.fr/Softwares/pso.html>