

國立交通大學

電機學院 IC 設計產業研發碩士班

碩士論文

眼球機器人智慧型多軸追蹤控制

Intelligent Multiaxial Tracking Control of Eyeball



研究生:林楊庭

指導教授:陳永平 教授

中華民國九十八年七月

眼球機器人智慧型多軸追蹤控制
Intelligent Multiaxial Tracking Control of Eyeball Robot

研 究 生：林楊庭

Student：Yang-Ting Lin

指 導 教 授：陳永平

Advisor：Yon-Ping Chen

國 立 交 通 大 學
電機學院 IC 設計產業研發碩士班
碩 士 論 文

A Thesis
Submitted to College of Electrical and Computer Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Industrial Technology R & D Master Program on
IC Design

July 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年七月

眼球機器人智慧型多軸追蹤控制

學生：林楊庭

指導教授：陳永平

國立交通大學電機學院產業研發碩士班

摘 要

本論文目的在於利用一個近似人類學習機制的類神經網路架構去控制四軸的眼球機器人，以追蹤移動的物體。整個追蹤策略分述如下，首先是預設眼球機器人眼睛和脖子與被追蹤物體位置之關係，其次是利用倒傳遞演算學習法則的離線學習（offline training），來訓練類神經網路以符合預設的對應關係。接著是利用影像即時擷取技術來求得欲追蹤物體的位置，當作已訓練完成神經網路的輸入訊號。最後利用類神經網路求得輸出訊號，作為眼球機器人各軸的速度修正參數。此外，為了能夠適應各種不同速度的追蹤，系統會根據物體速度來調整眼球機器人各軸轉動的速度。

Intelligent Multiaxial Tracking Control of Eyeball Robot

Student: Ying-Ting Huang

Advisor: Dr. Yon-Ping Chen

Industrial Technology R & D Master Program of
Electrical and Computer Engineering College
National Chiao Tung University

ABSTRACT



This thesis proposes a novel design method for a quarto-axis eyeball robot, based on the neural network frame and close to the human being's learning mechanism, to trace a moving object. The tracing strategy is classified into four parts. The first part is to preset the relationship between the axial speeds of the eyeball robot and the position of the moving object. Next, the offline training of the neural network based on back propagation learning rule according to the preset relationship. The third part is to retrieve the position of the moving object in real time as the input of the well trained neural network. In the fourth part, the output of the neural network is obtained as the reference data to modify the axial speeds of the eyeball robot. In addition, the eyeball robot is capable of adjusting its axial speeds in accordance with the object's speed for different velocities tracing.

致謝

時光匆匆，研究生的兩年生涯就在此告一段落了，而在研究生活中最感謝指導教授 陳永平老師的細心教導，老師不論我提出什麼天馬行空的問題，總是給予耐心指導與提供寶貴的想法，而且也分享在求學生活的過程，讓我不僅在研究方面得到了很多幫助，而且在人生與學習的態度上得到了很好的啟發。在此再次向老師致上深深的感謝。此外也感謝口試委員 楊谷洋老師與 梁耀文老師對於本論文所提出的珍貴意見與指正，讓本論文能更加的完整。

除此之外，感謝實驗室的桓展、世宏學長們，同學承育、新光和吉佑以及學弟傳源、瑋哲們在研究中所給予的幫助以及生活上的關心與包容，因為你們所提供的寶貴想法、經驗以及關心，才讓我能夠順利的完成研究，讓我在實驗室的兩年研究生活變的歡樂與溫馨。再來特別感謝我的爸爸、媽媽、弟弟們，因為有你們的支持、愛護與關心，我的兩年研究生活才能如此的無後顧之憂與順利，也很感謝關心我的親戚朋友們，因為有你們在，才有現在的我，真的非常的感謝。最後再次由衷的謝謝所有關心、幫助與支持過我的人。

林楊庭 2009.7.20

目錄

中文摘要.....	i
英文摘要.....	ii
致謝.....	iv
目錄.....	v
圖目錄.....	vii
表目錄.....	xi
第一章 緒論.....	1
第二章 倒傳遞類神經網路.....	3
2.1 類神經網路簡介.....	3
2.2 類神經相關理論.....	5
2.2.1 活化函數與學習速率.....	5
2.2.2 到傳遞連結加權值公式之數學推導.....	7
2.3 倒傳遞類神經網路.....	8
2.3.1 倒傳遞類神經網路簡介.....	8
2.3.2 倒傳遞演算法理念.....	9
第三章 系統架構.....	14
第四章 系統追蹤與控制.....	17
4.1 系統追蹤流程.....	17
4.2 影像擷取與物體位置計算.....	18



4.2.1 色彩空間.....	18
4.2.2 目標物搜尋.....	19
4.2.3 計算目標物影像中心座標.....	21
4.3 眼球機器人各軸分工.....	21
4.4 倒傳遞類神經網路的離線訓練.....	24
4.5 即時線上追蹤速度調整機制.....	26
第五章 目標物追蹤之實驗結果.....	28
5.1 定點控制 (set point control)	28
5.2 目標物追蹤結果.....	34
5.3 加入即時線上追蹤速度調整機制的追蹤結果.....	41
第六章 結論.....	45
參考文獻.....	47



圖目錄

圖2.1 人類神經細胞模型.....	3
圖2.2 單一神經元圖形.....	4
圖2.3 類神經網路圖.....	5
圖2.4 倒傳遞類神經網路圖.....	9
圖2.5 倒傳遞類神經網路訓練程式流程圖.....	12
圖3.1 硬體架構圖.....	14
圖3.2 攝影機.....	15
圖3.3 控制器.....	15
圖3.4 眼球機器人.....	16
圖4.1 眼球機器人影像追蹤系統追蹤流程圖.....	17
圖4.2 色彩空間.....	18
圖4.3 攝影機擷取到的影像.....	19
圖4.4 目標物的RGB值.....	19
圖4.5 背景的RGB值.....	20
圖4.6 欲追蹤物體的範圍.....	20
圖4.7 眼球機器人分工示意圖.....	21
圖4.8 眼球機器人分工示意圖—物體小範圍移動.....	22
圖4.9 眼球機器人分工示意圖—物體移動範圍超出眼睛轉軸的極限值.....	22
圖4.10 眼球機器人分工示意圖—物體往反方向移動.....	23
圖4.11 眼球機器人分工示意圖—再度由眼睛轉軸負責追蹤.....	23
圖4.12 眼球機器人分工示意圖—眼睛轉軸達到另一端的極限值.....	23
圖4.13 追蹤法則的區域劃分.....	24
圖4.14 目標物距影像中心誤差座標軸.....	26
圖5.1 物體位於影像左上位置的水平方向調校結果.....	28

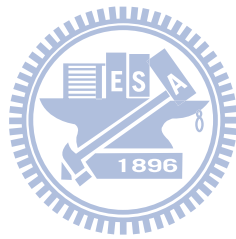


圖5.2 物體位於影像左上位置的垂直方向調校結果.....	29
圖5.3 物體位於影像左上位置調校的二維誤差軌跡.....	29
圖5.4 物體位於影像左下位置的垂直方向調校結果.....	30
圖5.5 物體位於影像左下位置的垂直方向調校結果.....	30
圖5.6 物體位於影像左下位置調校的二維誤差軌跡.....	31
圖5.7 物體位於影像右上位置的垂直方向調校結果.....	31
圖5.8 物體位於影像右上位置的垂直方向調校結果.....	32
圖5.9 物體位於影像右上位置調校的二維誤差軌跡.....	32
圖5.10 物體位於影像右下位置的垂直方向調校結果.....	33
圖5.11 物體位於影像右下位置的垂直方向調校結果.....	33
圖5.12 物體位於影像右下位置調校的二維誤差軌跡.....	34
圖5.13 欲追蹤目標物定速移動軌跡圖.....	35
圖5.14 目標物定速移動的水平方向追蹤誤差軌跡.....	35
圖5.15 目標物定速移動的垂直方向追蹤誤差軌跡.....	36
圖5.16 目標物定速移動的二維追蹤誤差軌跡.....	36
圖5.17 欲追蹤目標物規律變速度移動軌跡圖.....	37
圖5.18 目標物規律變速度移動的水平方向追蹤誤差軌跡.....	37
圖5.19 目標物規律變速度移動的垂直方向追蹤誤差軌跡.....	38
圖5.20 目標物規律變速度移動的二維追蹤誤差軌跡.....	38
圖5.21 目標物隨機改變速度移動軌跡示意圖.....	39
圖5.22 目標物隨機改變速度移動的水平方向追蹤誤差軌跡.....	39
圖5.23 目標物隨機改變速度移動的垂直方向追蹤誤差軌跡.....	40
圖5.24 目標物隨機改變速度移動的二維追蹤誤差軌跡.....	40
圖5.25 目標物定速移動加入速度調整機制的水平方向追蹤誤差軌跡.....	41
圖5.26 目標物定速移動加入速度調整機制的垂直方向追蹤誤差軌跡.....	42
圖5.27 目標物定速移動加入速度調整機制的二維追蹤誤差軌跡.....	42

圖5.28 目標物規律變速移動加入速度調整機制的水平方向追蹤誤差軌跡.....43

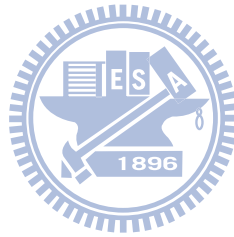
圖5.29 目標物規律變速移動加入速度調整機制的垂直方向追蹤誤差軌跡.....43

圖5.30 目標物規律變速移動加入速度調整機制的二維追蹤誤差軌跡.....44



表目錄

表4.1 目標物誤差與修正值的對應關係表.....27



第一章 緒論

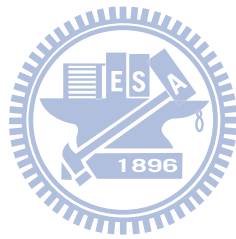
有鑑於科技之蓬勃發展，人們對於機器視覺(machine vision)的應用更加重視而在以往之視覺伺服追蹤系統中，因使用複雜的演算法則加上CPU速度不夠快導致系統無法即時的完成運算。然而近幾年來，演算平台之運算速度大幅提昇，而且影像追蹤演算法也日益精進。不僅能夠達到即時移動目標物偵測、動態視覺追蹤的應用，在性能方面也提升很多，讓本論文所要做的追蹤控制得以達成。

要控制一個多軸的機器人有很多種方法，傳統的PID控制、穩健控制和模糊控制等都可以達到控制的目的，但是大部分的控制方法都需要先知道受控體的數學模型。而對一個模仿人類行為模式的受控體來說，其數學模型通常是非常複雜而且難以獲得的。而對於模糊控制中最常應用於控制模仿人類行為模式的受控體的if-then法則來說，這樣的應用通常使得程式碼變的複雜而且龐大，相對的也減低了執行效率。近年來類神經網路的架構由於既不需要費心的去尋找受控體的數學模型，也不需要龐大的if-then法則運算一躍成為最普遍的研究主題之一。

本論文運用類神經網路架構控制一個未知數學模型的四軸眼球機器人去追蹤一個移動的物體。運用倒傳遞學習演算法(backpropagation learning algorithm)，在每一次的學習過程中會自動的調整類神經網路中的權重(weight)。經由離線學習，類神經網路會有一個大略的控制方向，控制眼球機器人的各軸使物體位置往影像中心移動。而在追蹤的過程中、即時的線上調整機制會將會調整追蹤的速度以優化整個追蹤的過程。

本論文一共分為五個章節，在本章敘述的是本論文的序論部分。類神經網路架構的概念和倒傳遞學習演算法會在第二章節作一個概略的介紹。第三章會介紹

整個系統的架構、眼球機器人四軸的控制方法和學習模式。第四章則是展示實驗的結果和數據，最後在第五章對這樣的控制方法作一個總結。



第二章 倒傳遞類神經網路

類神經網路系統，是模擬生物神經的運作方式，利用資料不斷的學習及錯誤修正以達到正確的輸出。對簡單的非線性函數進行組合，經過數次組合後，成為一複雜的函數型式，就可以解決複雜映射的問題。而倒傳遞類神經網路則有學習精度高、回想速度快、輸出值可為連續值，能處理複雜的樣本識別與高度非線性的函數合成問題等優點。

2.1 類神經網路簡介

在人腦中約有數億個神經細胞，並且每個神經元細胞都是由數千至數萬根神經連結，其模型如圖 2.1 所示[12]，在圖中呈現樹枝狀的組織我們稱之為樹突，是神經元接收輸入訊號的地方，當有訊號傳入時，可以藉由突觸，將刺激訊號傳入。另外一方則有一類似纖維束，它可以將輸出訊號傳出，我們稱之為軸索，它可以將輸出訊號傳至其它神經元。

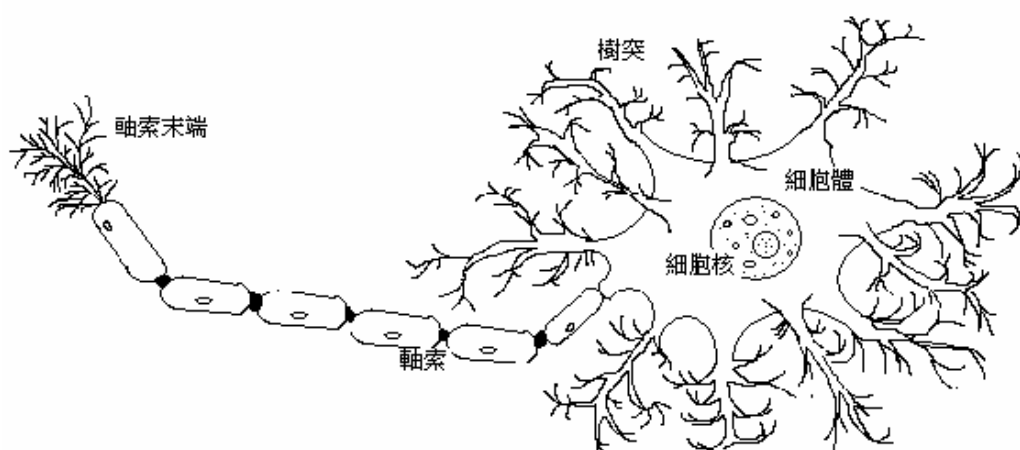


圖 2.1 人類神經細胞模型

類神經網路是「一種仿腦中樞神經系統的資訊處理技術」。此種技術是基於腦神經系統，並且由多個神經元組成，應用至估測、預測、決測、診斷...等等的技術方面。如下圖 2.2 為單一人工神經元的基本模型

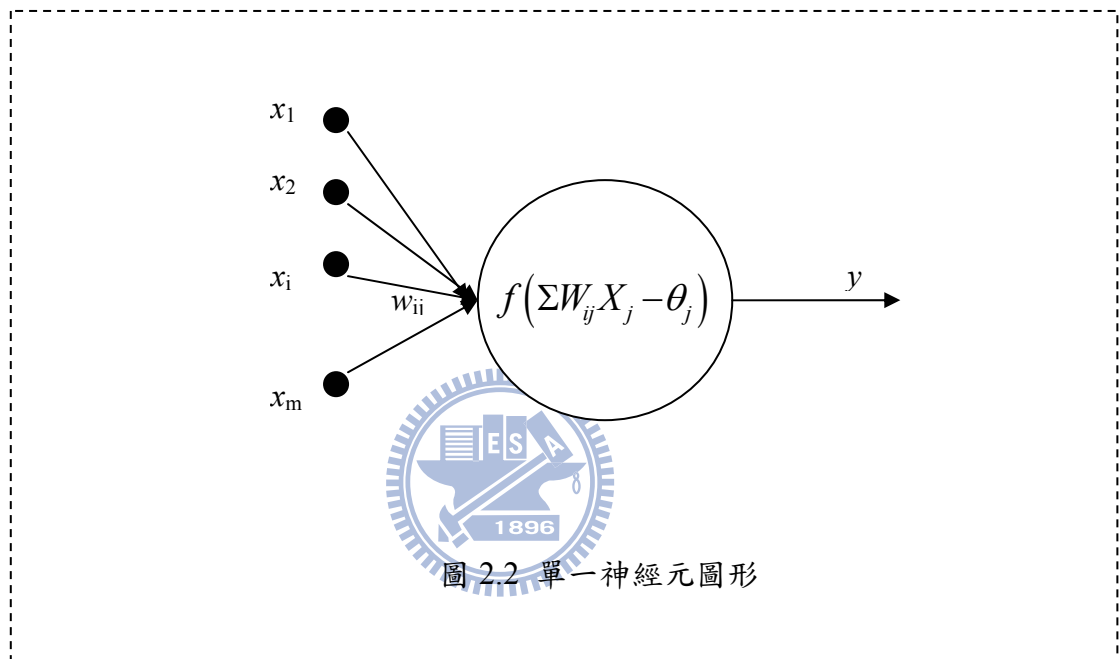


圖 2.2 單一神經元圖形

圖 2.2 中，前方有許多處理單元的輸入就好像樹突一樣，而這些輸入透過人工神經元處理單元，得到之輸出就如同軸索一樣，形成一神經元系統，在神經元處理單元中，其處理可用一種關係式表示：

$$Y_j = f(\sum W_{ij} X_j - \theta_j) \quad (2-1)$$

其中 Y_j 為輸出訊號， $f(\cdot)$ 為一活化函數 (activation function)，此函數是仿生物之非線性處理機能，由多個輸入訊號 X_i ，乘上某連結加權值 W_{ij} (模仿生物神經模型的突觸強度) 相加起來，扣除依門檻值 θ_j ，代入其中得到輸出訊號。

如下圖 2.3，一個神經網路，是由多個神經元與其連結權重值所組成，在圖

中分有輸入層、隱藏層、輸出層三大部分。

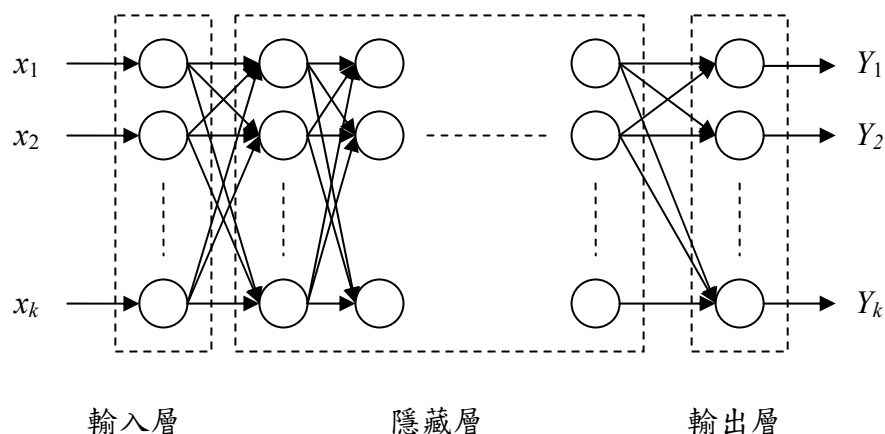


圖 2.3 類神經網路圖

輸入層為每一神經單元所需要之外在環境之輸入，輸出層為處理單元用以輸出訊息給外在環境，此外，最重要的是中間的隱藏層，此層的存在是為了提供類神經網路表現處理單元間的交互作用，與問題的內在結構的能力。

2.2 類神經相關理論

2.2.1 活化函數(Activation function)與學習速率

在實際人的神經系統中，在軸突丘部位所呈現的整體細胞膜電位，若超過閾值，則「活化電位」的脈衝會被激發，使整個傳遞而來的輸入資訊在這裡做調變的處理，因此，我們將這神經元中的處理，利用活化函數呈現。

活化函數我們可以利用下面輸入輸出之數學關係描述活化函數與神經元的關係：

$$u_j = \sum_{i=1}^p w_{ji} x_i \quad (2-2)$$

$$y_j = f(u_j - \theta_j)$$

其中 w_{ji} 代表第 i 維輸入至第 j 個類神經元的鍵結值， θ_j 代表這神經元所需觸發的閾值， $x_i=(x_1, x_2, \dots, x_p)^T$ 代表 p 維的輸入， u_j 為整體輸入量(等同軸突丘的細胞膜電位)， y_j 則代表神經元的輸出值。

通常活化函數有嚴格限制函數、區域線性函數、雙彎曲函數和高斯函數四種類型：

(1)嚴格限制函數：

$$f(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \quad (2-3)$$

(2)區域線性函數：

$$f(v) = \begin{cases} 1 & \text{if } v > v_1 \\ cv & \text{if } v_2 \geq v \geq v_1 \\ 0 & \text{if } v_2 > v \end{cases} \quad (2-4)$$

(3)雙彎曲函數：

$$f(v) = \frac{1}{1 + \exp(-cv)} \quad (2-5)$$

or $f(v) = \frac{\exp(cv) - \exp(-cv)}{\exp(cv) + \exp(-cv)} = \tanh(cv)$

(4)高斯函數：

$$f(v) = \exp\left(-\frac{v^2}{2\sigma^2}\right) \quad (2-6)$$

其中雙彎曲函數由於能根據輸入的不同而有不同的輸出，又能將輸出限制在 0 到 1 之間，故為一般類神經網路架構所採用。在本論文中也將採取此函數作為類神經網路隱藏層的活化函數。

學習速率就是設定一參數控制類神經網路的學習快慢，這參數設定依各神經系統所需皆不同，通常太大或太小對網路的收斂皆不利，這是因為若設太大，雖然有較快的收斂速度，可以快速的逼近誤差函數最小值，但由於此學習速率過大，導致類神經網路誤差修正過量，使得發生誤差振盪現象。另外，若設太小，

學習速率太慢，導致系統訓練效果不佳，無法達到快速學習的成效，故通常先設一較大的初始學習速度，再依照成果，漸漸縮小學習速率，但不超過一下限極小值，取得最佳的收斂成果。通常最大可達到10 以上，最小則到0.1 左右。

2.2.2 倒傳遞連結加權值公式之數學推導

下面主要分成兩個部分推導：第一部分為隱藏層與輸出層間之連結加權值，第二部分為輸入層與隱藏層間之連結加權值，可參考上面圖2.3。

(1)隱藏層與輸出層間之連結加權值

誤差值 E 對網路隱藏層第 j 個單元與輸出層第 k 個單元間之連結加權值 W_{jk} 的偏微分得

$$\frac{\partial E}{\partial W_{jk}} = \frac{\partial E}{\partial Y_k} \frac{\partial Y_k}{\partial net_k} \frac{\partial net_k}{\partial W_{kj}} = -(T_k - Y_k) \cdot f'(net_k) \cdot H_j \quad (2-7)$$

其中 $H_j = f\left(\sum_i W_{i,j} \cdot x_i\right)$ 令 δ_k 定義為輸出層第 k 個輸出處理單元的誤差量：

$$\delta_k = (T_k - Y_k) \cdot f'(net_k) \quad (2-8)$$

則網路輸出層與隱藏層間連結之加權值 W_{kj} 之修正量如下：

$$\Delta W_{jk} = -\eta \frac{\partial E}{\partial W_{jk}} = \eta \cdot (T_k - Y_k) \cdot f'(net_k) \cdot H_j \quad (2-9)$$

其中由於活化函數 $f(v)$ 我們在倒傳遞類神經網路中，皆使用雙彎曲函數

$\frac{1}{1 + \exp(-v)}$ ，所以其一階微分 $f'(\cdot)$ 如下：

$$\begin{aligned} f'(v) &= \frac{df(v)}{dv} = \left(\frac{1}{1+e^{-v}}\right) \left(\frac{e^{-v}}{1+e^{-v}}\right) \\ &= \left(\frac{1}{1+e^{-v}}\right) \left(1 - \frac{1}{1+e^{-v}}\right) = f(v) \cdot (1 - f(v)) \end{aligned} \quad (2-10)$$

由此可得

$$\begin{aligned} \Delta W_{jk} &= \eta \cdot (T_k - Y_k) \cdot f'(net_k) \cdot (1 - f(net_k)) \cdot H_j \\ &= \eta \cdot (T_k - Y_k) \cdot Y_k \cdot (1 - Y_k) \cdot H_j \end{aligned} \quad (2-11)$$

同理，輸出處裡單元偏權值修正量：

$$\Delta\theta_k = -\eta \frac{\partial E}{\partial \theta_k} = -\eta \cdot \delta_k = -\eta \cdot (T_k - Y_k) \cdot Y_k \cdot (1 - Y_k) \quad (2-12)$$

(2) 輸入層與隱藏層間之連結加權值

誤差值 E 對網路輸入層第 i 個單元與隱藏層第 j 個單元間之連結加權值 W_{ij} 的偏微分得

$$\begin{aligned} \frac{\partial E}{\partial W_{jk}} &= \frac{\partial E}{\partial H_j} \frac{\partial H_j}{\partial net_j} \frac{\partial net_j}{\partial W_{jk}} = \left(\sum_k \frac{\partial E}{\partial Y_k} \frac{-\partial Y_k}{\partial net_k} \frac{\partial net_k}{\partial H_j} \right) \cdot f'(net_j) \cdot x_i \\ &= \left(\sum_k -(T_k - Y_k) \cdot f'(net_k) \cdot W_{jk} \right) \cdot f'(net_j) \cdot x_i \\ &= - \left(\sum_k \delta_k \cdot W_{jk} \right) \cdot f'(net_j) \cdot x_i \end{aligned} \quad (2-13)$$

令 δ_{hj} 定義為隱藏層第 j 個隱藏處理單元的誤差量：

$$\delta_{hj} = \left(\sum_k \delta_k \cdot W_{jk} \right) \cdot f'(net_j) \quad (2-14)$$

則網路隱藏層與輸入層之間的連結加權值 W_{ij} 之修正量和偏權值修正量如下：

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} = \eta \cdot \delta_{hj} \cdot x_i \quad (2-15)$$

$$\Delta \theta_j = -\eta \frac{\partial E}{\partial \theta_k} = -\eta \cdot \delta_{hj} \quad (2-16)$$

同上面隱藏層與輸出層的活化函數，所以可以知道

$$f'(net_j) = H_j (1 - H_j) \quad (2-17)$$

使得權重修正如下：

$$\Delta W_{ij} = \eta \cdot \delta_{hj} \cdot x_i = \eta \cdot \left(\sum_k \delta_k \cdot W_{jk} \right) \cdot H_j (1 - H_j) \cdot x_i \quad (2-18)$$

由上面的修正式，在每次的訓練過程中慢慢的修正個結點間的權值，使得輸入漸漸的朝目標靠近。

2.3 倒傳遞類神經網路

2.3.1 倒傳遞類神經網路簡介

倒傳遞類神經網路(Back-propagation Neuron Network, BPNN)是監督式學習類神經網路中的一種，所謂「監督式學習」又稱為「範例中學習(Learning from examples)」，其所謂的學習是從一些範例中學習，達到所欲求的成果目的，也就是像「老師」教導學習者，什麼樣的刺激，應該要有怎樣的反應，是現今在類神經網路中應用最廣的方法。在目前類神經網路學習中，最具有代表且使用最廣泛的就是倒傳遞模式的類神經網路，其構造圖如下面(圖2.4)所示：

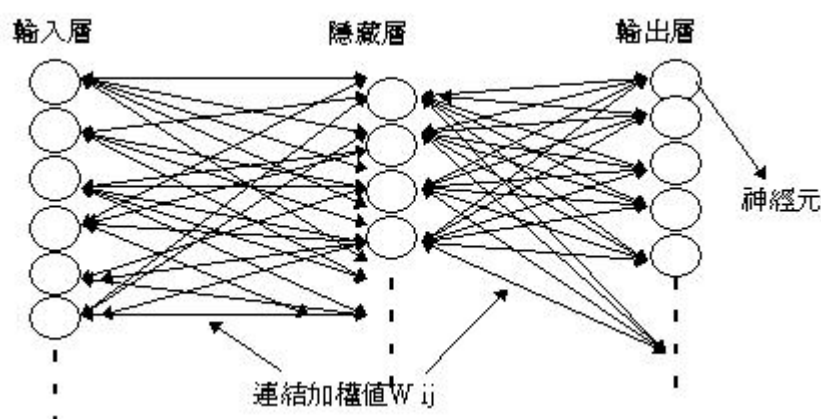


圖 2.4 倒傳遞類神經網路圖

在(圖2.4)中所顯示的意義為，根據誤差的大小，倒傳遞一些訊號來改變其類神經網路中的所有連結權重，促使連結權重改變致使逼近此種函數，使指定輸出和估測輸出誤差可以達到最小化。

2.3.2 倒傳遞演算法理念

倒傳遞演算法也可以稱為「通用差距法則(Generalized deltarule)」，是以「最陡坡降法(The gradient steepest decent method)」的觀念將誤差最小化，使得網路誤差可以達到理想目標。在演算法中包含兩個主要部份：學習和回想部份，首先是學習部分，學習部分主要是將包含輸入向量及對應的輸出向量的訓練資料輸入至網路中，將輸入向量推論得到推論的輸出值，與理想輸出值之間的誤差，以最陡坡降法算出應調整之權重值，利用疊代方式做調整，使兩者之間均方誤差可以

達到最小值，致使最後可以達到收斂的準確度指標。接下來，回想部份主要是將學習部份訓練出的達到目標的權重，代回類神經網路中，透過這權重計算出最佳的估測輸出。

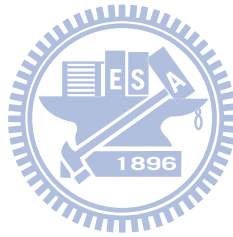
以下是學習部份和回想部分簡要的步驟如下：

1. 學習部份：

- (1) 選取神經網路輸入層、隱藏層、輸出層的神經元數目及訓練樣本的數目。
- (2) 參數初始化，以隨機亂數方式產生個層間的連接加權值向量矩陣 W_{ij} 、 W_{jk} 及偏權值向量矩陣 θ_j 、 θ_k 。
- (3) 輸入訓練樣本包含輸入向量 x_i 及目標輸出向量 T_k 。
- (4) 計算隱藏層及輸出層的輸出向量。

(i) 隱藏層輸出向量 H ：

$$\begin{aligned} net_j &= \sum W_{ij}x_i - \theta_j \\ H_j &= f(net_j) \end{aligned} \tag{2-19}$$



為運用線性整合函數將輸入資料至隱藏層的數據作統合計算所得到的隱藏層第 j 個神經元的加權乘積和，再將之代入選用之活化函數中，計算得到 H_j ，即為該隱藏層第 j 個神經元的輸出值。

(ii) 輸出層輸出向量 Y ：

$$\begin{aligned} net_k &= \sum W_{jk}H_j - \theta_k \\ Y_k &= f(net_k) \end{aligned} \tag{2-20}$$

輸出層與隱藏層的計算方式相似，改以隱藏層的輸出作為輸入 net_k 與 Y_k 分別為輸出層第 k 個神經元的加權乘積，並且接著推論出輸出值。

(5) 計算誤差量大小。

(i) 輸出層第 k 個神經元的誤差量 δ_k ：

$$\delta_k = (T_k - Y_k) \cdot f'(net_k) = (T_k - Y_k)(Y_k)(1 - Y_k) \tag{2-21}$$

其中 δ_k 為輸出層第 k 個神經元的誤差量，而 $(T_k - Y_k)$ 為目標輸出與推論輸出的誤差量。

(ii)隱藏層第 k 個神經元的誤差量 δ_{hj} ：

$$\delta_{hj} = \left(\sum \delta_{hj} W_{jk} \right) \cdot f'(net_k) = \left(\sum \delta_{hj} W_{jk} \right) \cdot H_j (1 - H_j) \quad (2-22)$$

(6)做輸出層與隱藏層的權重和偏權重之修正。

(i)輸出層權重和偏權值修正：

$$\begin{aligned} \Delta W_{jk} &= \eta \cdot \delta_k \cdot H_j \\ \Delta \theta_k &= -\eta \cdot \delta_k \\ W'_{jk} &= W_{jk} + \Delta W_{jk} \end{aligned} \quad (2-23)$$

其中 ΔW_{jk} 、 $\Delta \theta_k$ 分別為輸出層與隱藏層間之權重和偏權值應該修正的值，而 η 為學習速率。

(ii)隱藏層權重和偏權值修正：

$$\begin{aligned} \Delta W_{jk} &= \eta \cdot \delta_k \cdot H_j \\ \Delta \theta_k &= -\eta \cdot \delta_k \\ W'_{jk} &= W_{jk} + \Delta W_{jk} \end{aligned} \quad (2-24)$$



2.回想過程主要是運用先前所學習的資訊，透過網路的運算可得到滿意的輸出結果。回想過程演算法的流程，其步驟如下：

- (1)選取神經網路輸入層、隱藏層、輸出層的神經元數目。
- (2)載入由學習演算法所得到的加權值向量矩陣與偏權值向量矩陣。
- (3)輸入即時測試樣本。
- (4)計算類神經網路的推論輸出向量。

倒傳遞類神經程式流程圖如下：

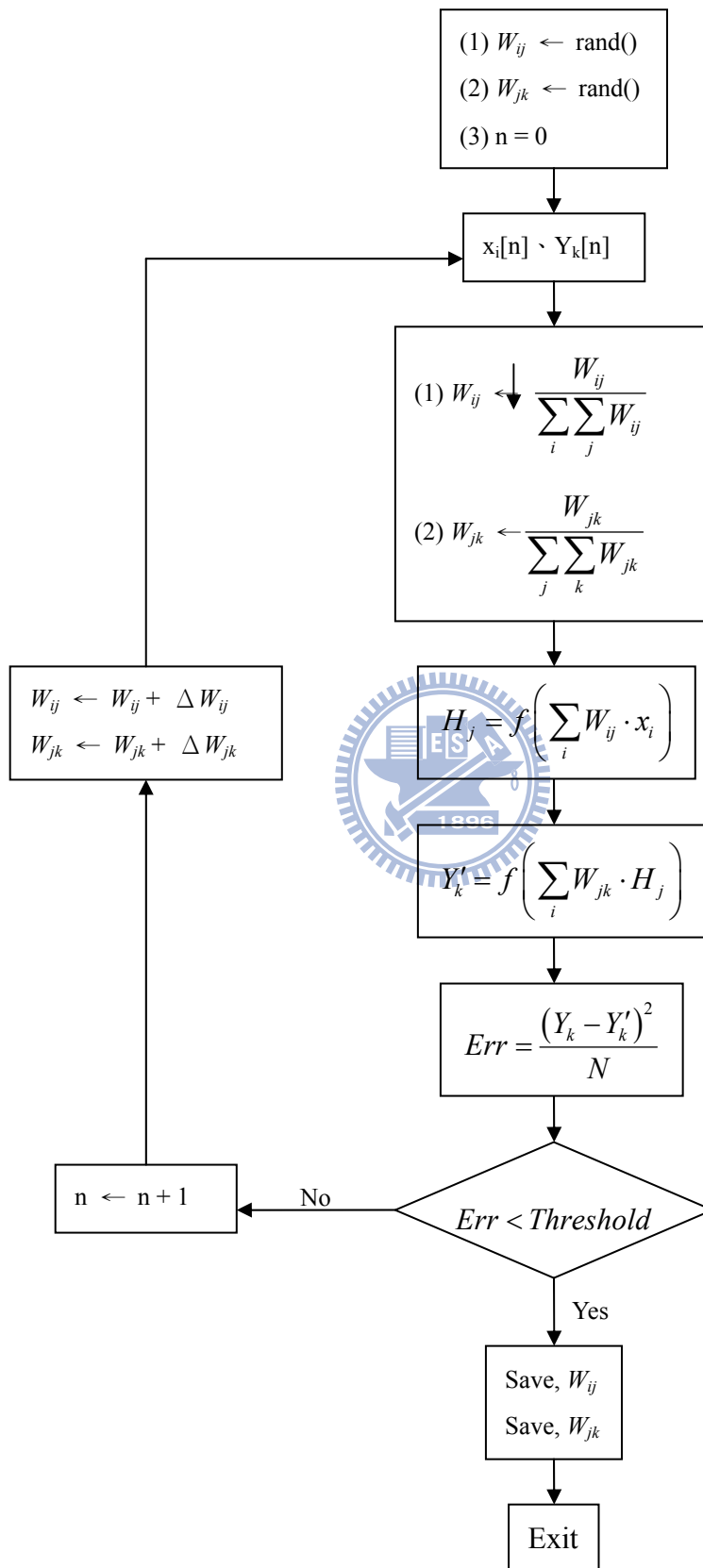
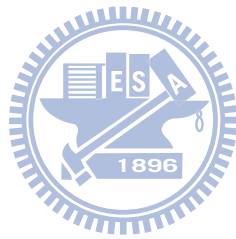


圖 2.5 倒傳遞類神經網路訓練程式流程圖

在系統的離線訓練中，採用上面的流程並設定 threshold 值為 0.0001，使用到傳遞演算法約訓練 50000 次以內可以達到要求。



第三章 系統架構

此影像追蹤系統主要分為影像的偵測、影像的追蹤以及眼球機器人各軸的控制三個部分，這些實做會用到的器材，包含軟體和硬體方面，其詳細資料如下所示：

(1) 硬體方面：其架構如下面（圖 3.1），架構中主要以攝影機、影像擷取卡、個人電腦、MCDC 控制器這幾個主要硬體構成系統迴路。

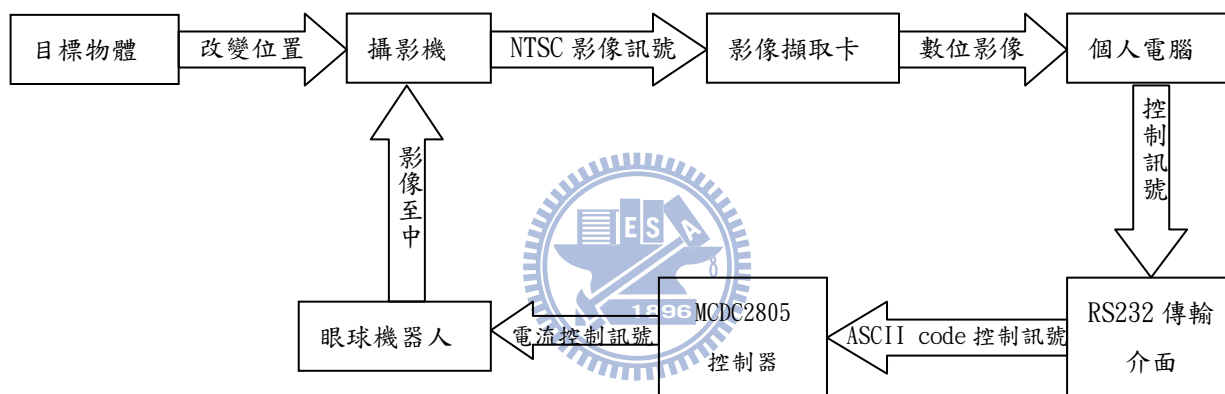


圖 3.1 硬體架構圖

1. 目標物體

欲追蹤的物體影像是以投影機投射在距離眼球機器人雙眼攝影機 80cm 的屏幕上。目標物體在屏幕上 50cmx50cm 的範圍內移動，移動的方式可由程式控制。

2. 攝影機

型號：Logitech RightLigh™2

像素：320x240 NTSC

白平衡：Auto white balance



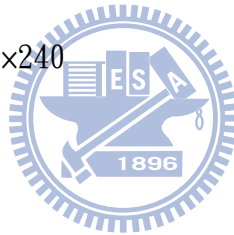
圖 3.2 攝影機

3.影像擷取卡

型號：WinFast PX8400 GS

輸入規格：NTSC

影像格式：60Hz，320×240



4.個人電腦

處理器：Intel Pentium Dual CPU E2140 @ 1.60GHz

記憶體：DDR2-667MHz 2GB

作業系統：Windows XP

5.控制器

型號：Faulhaber MCDC2805

電源電壓：12V~28V

工作頻率：62.5kHz

效率：95%

最大連續輸出電流：5A

最大峰值輸出電流：10A

可控速度範圍：10rpm ~ 30000rpm

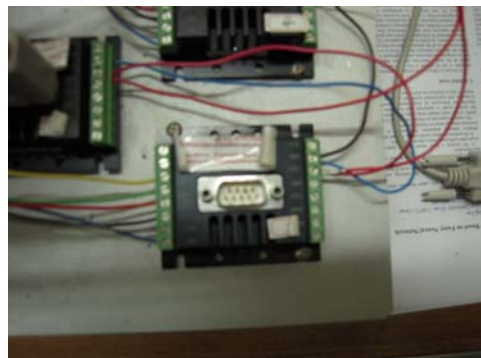


圖 3.3 控制器

輸出電壓：5V

最大輸出電流：60mA

6. 眼球機器人

為了要模擬人類頭部的運動，實驗設計的眼球機器人如下（圖 3.2）所示，一共包含了 2 台球型攝影機和 5 個可控的伺服馬達。2 台球型攝影機在實驗中負責了影像的擷取功能，對應了人類的眼睛。而五個可控的伺服馬達分別對應了人類的兩顆眼球的左右轉軸（2 軸）、眼球的上下轉軸（1 軸）和脖子的左右轉軸（1 軸）及上下轉軸（1 軸）。

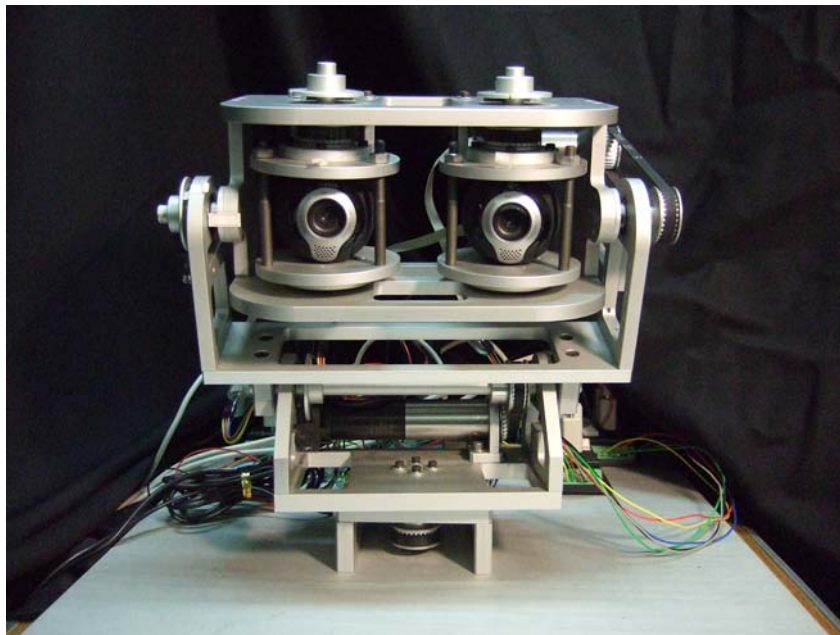


圖 3.4 眼球機器人

(2) 軟體方面：系統主要程式是使用 MATLAB 7.04 的 M-file editor 編寫，而影像擷取介面則是使用 Logitech 的 Quickcam 做影像的抓取，而個人電腦和 MCDC2805 控制器之間則用 RS232 介面作為溝通。

第四章 系統追蹤與控制

4.1 系統追蹤流程

將攝影機所擷取的影像傳送到個人電腦中，結合第二章所提到用以控制眼球機器人的類神經網路架構，整合成一個影像追蹤的系統，其流程大致如下：(1) 由攝影機擷取一張影像。(2) 當每擷取到一張影像時就開始進行欲追蹤物體位置辨識，並計算物體位置。(3) 使用類神經網路開始做追蹤控制，利用物體和影像中心的相對位置，控制眼球機器人各軸使物體移至影像中心。

眼球機器人影像追蹤系統如下所示：

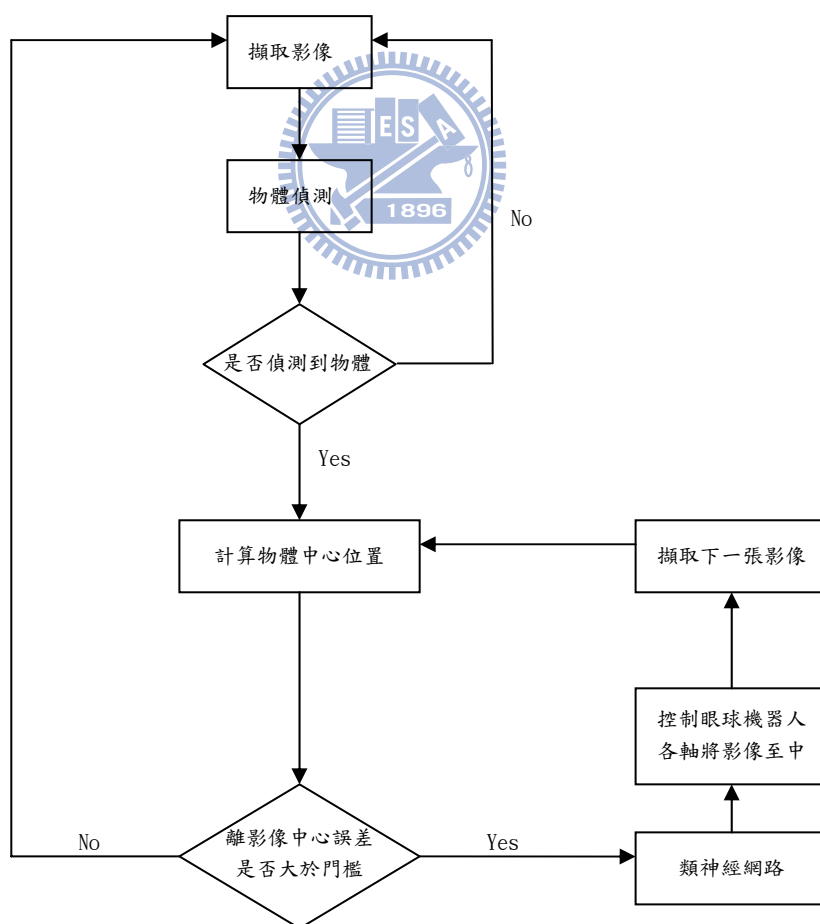


圖 4.1 眼球機器人影像追蹤系統追蹤流程圖

接下來的各結將對上述的流程中，本系統對於影像的追蹤、計算和追蹤控制的方法一一的作詳細的介紹。

4.2 影像擷取與物體位置計算

4.2.1 色彩空間

色彩空間(color space)，也可以稱為色彩模型或色彩系統，其目的是為了定義所有色彩於一個空間中，這個空間為一標準且可接受的單位正交座標系統，在系統中每一點代表一個顏色，如(圖4.2)，這色彩空間是呈現出一個單位立方體形狀，其中紅(R)、綠(G)、藍(B)值分別就是在座標軸上的三頂點，而青(C)、紫紅(M)、黃(Y)是在另外三頂點上。

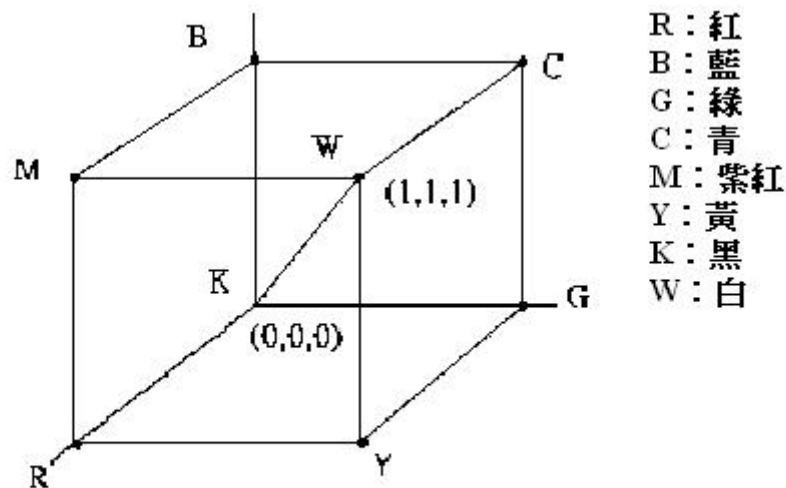


圖4.2 色彩空間

紅(R)、綠(G)、藍(B)為色彩的三原色，其分別為色彩空間的三個座標軸，在這座標軸上範圍皆在[0,1]間，並且其每個軸皆有8位元的像素深度，使的RGB影像共有24位元的像素深度，通常以全彩(full-color)這術語來表示這24位元的RGB影像空間。

4.2.2 目標物搜尋

經由攝影機和影像擷取卡所得到的物體影像，如下圖 4.3 所示。在 MATLAB 軟體中，可以很輕易的找出目標物體(圖 4.4)和背景(圖 4.5)的 RGB 值。並根據兩者 RGB 值的不同而辨識出欲追蹤的目標物。



圖 4.3 攝影機擷取到的影像

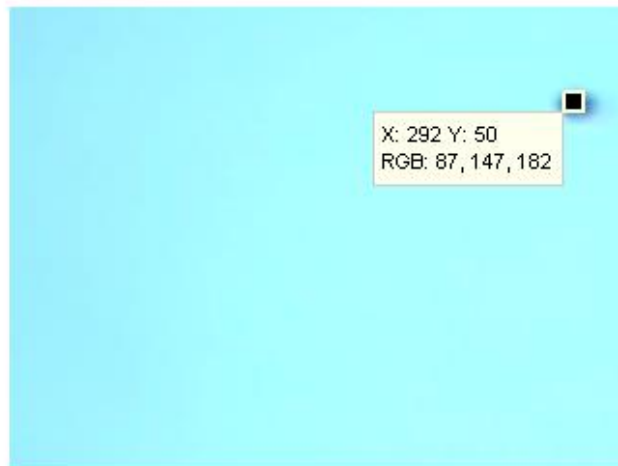


圖 4.4 目標物的 RGB 值

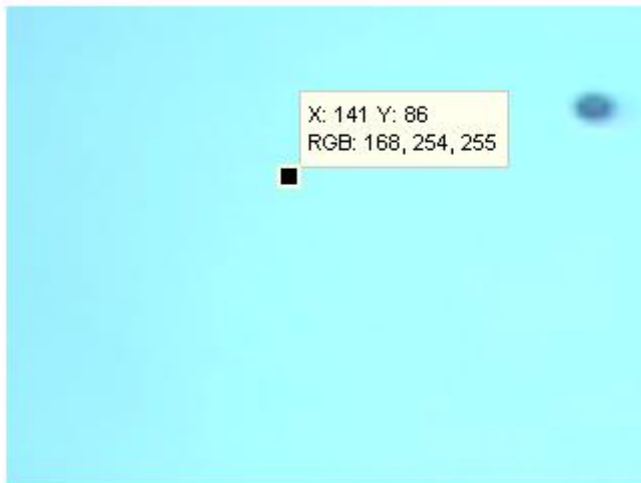


圖 4.5 背景的 RGB 值

由上圖 4.4 和圖 4.5 可以得到欲追蹤物體的 RGB 值分別落在 87、147 和 182 附近，而單純白色背景（投影布幕）的 RGB 值則在 168、254 和 255 附近。所以在追蹤過程中，系統將影像中 $70 \leq R \leq 110$ 且 $120 \leq G \leq 170$ 且 $160 \leq B \leq 220$ 的部分判定為欲追蹤的目標物並將其範圍標訂為 1，其餘部分則標訂為 0。於是，系統便得到了如下(圖 4.6)的欲追蹤物體範圍：

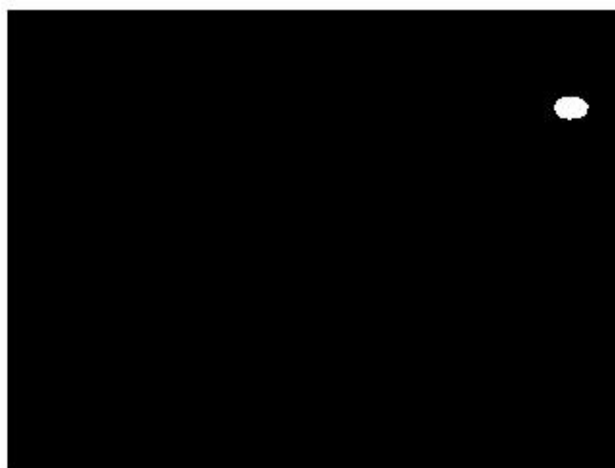


圖 4.6 欲追蹤物體範圍

4.2.3 計算目標物影像中心座標

得到欲追蹤物體的範圍之後，再計算欲追蹤目標物的重心即為欲追蹤物體應輸入到類神經網路的座標。如上圖 4.6 為一 320x240 的黑白影像，其中白色區域數值為 1、黑色區域數值為 0。首先先將影像陣列的每行全部相加成為一個 320x1 的單行向量 x_i ，再令一向量 $x_p = [1 \ 2 \ \dots \ 320]$ ，則欲追蹤目標物的影像中心橫軸座標 x 為 $x = \frac{x_i^T \cdot x_p}{1+2+\dots+320}$ 。同理將影像陣列的每列全部相加成為一個 1x240 的單列向量 y_i ，再令一向量 $y_p = [1 \ 2 \ \dots \ 240]$ ，則欲追蹤目標物的影像中心縱軸座標 y 為 $y = \frac{y_i \cdot y_p}{1+2+\dots+240}$ 。 (x, y) 即為輸入類神經網路的追蹤目標物影像座標。

4.3 眼球機器人各軸分工

在一般的影像追蹤架構中，單純追蹤物體使其位於影像中心只需要用到兩軸（水平、垂直）或三軸（水平、垂直、旋轉）。為了能夠更貼近人類的視覺學習模式，本篇論文的追蹤系統採用了四個轉軸來分別模擬人類眼睛和脖子的轉動。分工的依據為：眼睛負責小角度範圍的移動，反應速度較快。脖子則在物體移動角度過大的時後開始啟動，負責物體大範圍移動的追蹤。在追蹤的過程中系統限制眼睛部分的轉軸只有小角度的活動範圍，一旦追蹤的過程中，因為物體的移動使轉軸到達系統限制的範圍，則脖子部分轉軸開始轉動。兩者配合的模式如下圖 4.7~圖 4.12 所示：

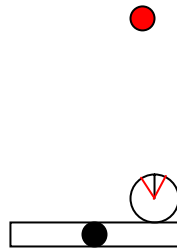


圖 4.7 眼球機器人轉軸分工示意圖

在圖 4.7 中紅色圓圈代表欲追蹤的目標物、黑色圓圈表示眼球機器人脖子部分轉軸，而白色圓圈則代表眼球部分轉軸。白色圓圈中的黑線代表攝影機面對的方向，紅色線條則代表眼睛轉軸的轉動極限範圍。

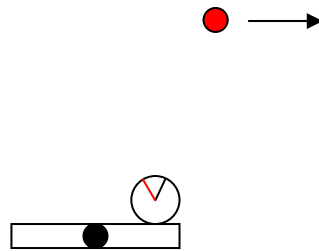


圖 4.8 眼球機器人轉軸分工示意圖—物體小範圍移動

如圖 4.8 所示當欲追蹤物體移動範圍不大的時後，只有眼睛轉軸部分轉動，脖子部分保持不動直到眼睛轉軸部分達到系統設定的極限值。

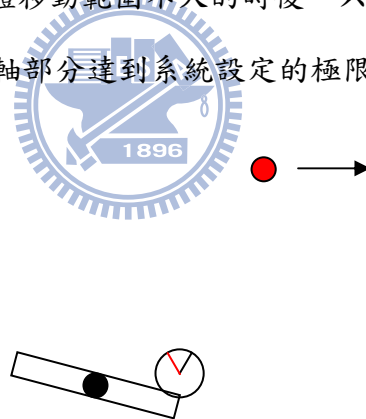


圖 4.9 眼球機器人轉軸分工示意圖—物體移動範圍超出眼睛轉軸的極限值

如圖 4.9 若物體持續往同一個方向移動，眼睛轉軸部分因為以達極限值故保持不動。這時的追蹤完全由脖子轉軸的部分負責，直到物體移動方向改變或超出追蹤範圍為止。

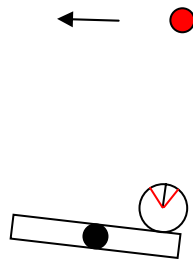


圖 4.10 眼球機器人轉軸分工示意圖—物體往反方向移動

圖 4.10 中物體改變了移動的方向，這時眼睛和脖子轉軸會一起轉動。但由於眼睛轉軸的反應速度較快，所以眼睛轉軸會先回到兩極限值的正中央。

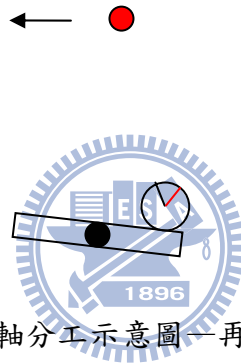


圖 4.11 眼球機器人轉軸分工示意圖—再度由眼睛轉軸負責追蹤

如上圖 4.11，當眼睛轉軸回到兩極限值的正中央之後，此時物體的追蹤再度由眼睛轉軸負責。此時脖子轉軸部分保持不動直到眼睛轉軸達到另一個方向的極限值。

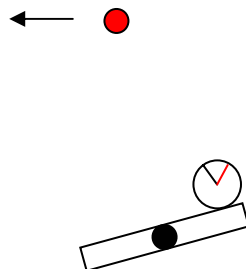


圖 4.12 眼球機器人轉軸分工示意圖—眼睛轉軸轉動達到另一端的極限值

如圖 4.12，當物體繼續移動而眼睛轉軸部分又已達到轉動的極限值。此時眼睛轉軸部分則因為以達到及現實保持固定，而追蹤再度由脖子轉軸接手，直到物

體移動方向改變或超出追蹤範圍為止。

4.4 倒傳遞類神經網路的離線訓練

本追蹤系統運用 2.3 節所提到的到傳遞演算學習法則來訓練對應各軸控制的類神經網路，並運用此類神經網路來控制眼球機器人各軸的速度，使其達到將欲追蹤目標物影像置中的目的。離線訓練的目的在於，當系統將 4.2.3 節得到的欲追蹤目標物中心座標輸入到類神經網路的時後，類神經網路要能產生對應各軸的輸出速度值。在系統中各軸的速度為：

$$V = f_v \times V_{\max} \quad (4-1)$$

其中 V 為輸出到各軸的實際速度控制值， f_v 為一個介於 0 到 1 之間的參數，由類神經網路根據輸入的欲追蹤目標物中心座標產生對應的輸出， V_{\max} 為當前狀況下轉軸的最大速度。

為了要模擬人類眼睛追蹤物體的模式，眼球機器人的眼睛轉軸和脖子轉軸必須使用不同的追蹤法則來對目標物做追蹤。而在類神經網路中，欲使得類神經網路學習一種追蹤法則需要經由一連串的訓練資料來讓系統慢慢的習得這項法則。在系統中的追蹤法則分為眼睛和脖子兩部分，並根據影像的位置將追蹤法則分為三塊區域。如下圖 4.13 所示，將影像位置正規化為 0 到 1 之間的數之後，目標物位置為 0 代表目標物位於影像的左邊界，而位置 1 則代表著影像的右邊界。追蹤法則的目的是控制各軸使目標物位於影像的正中央，也就是 0.5 的位置。

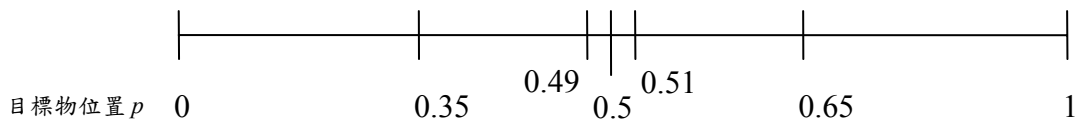


圖 4.13 追蹤法則的區域劃分

在圖 4.13 中共分為三個區域，0.49 到 0.51 為第一部份，當目標物中心位置

落在這個區域時，系統判定已經追蹤到目標物，此時眼球機器人各軸會保持靜止。而當目標物位置 p 落在 0.35 到 0.49 或 0.51 到 0.65 之間時，這部分系統判定屬於眼睛的負責範圍。除非眼睛轉軸已到達前面 4.3 節提到系統規定的極限值，不然在這範圍內都是由眼睛轉軸負責追蹤，脖子轉軸部分保持不動。第三部份則是當目標物位置 p 落在 0 到 0.35 或 0.65 到 1 之間的時候，這時因為目標物距離影像中心很遠，要旋轉的已經很大。所以直接由脖子的轉動來完成將目標物往影像中心移動的工作。以上的追蹤法則用數學方程式表示為位置對應輸出速度的表示式眼睛部份為：

$$f_{eye} = \begin{cases} 0 & 0 < p < 0.35 \\ \frac{1}{0.14}(0.49 - p) & 0.35 < p < 0.49 \\ 0 & 0.49 < p < 0.51 \\ \frac{1}{0.14}(0.51 - p) & 0.51 < p < 0.65 \\ 0 & 0.65 < p < 1 \end{cases} \quad (4-2)$$

而脖子轉軸部份則分為兩種情況當眼睛轉軸未到達系統所限制的極限時

$$f_{neck} = \begin{cases} \frac{1}{0.35}(0.35 - p) & 0 < p < 0.35 \\ 0 & 0.35 < p < 0.65 \\ \frac{1}{0.35}(0.65 - p) & 0.65 < p < 1 \end{cases} \quad (4.3)$$

而當眼睛轉軸已到達系統所限制的極限時上面的方程式則變為

$$f_{neck} = \begin{cases} \frac{1}{0.49}(0.49 - p) & 0 < p < 0.49 \\ 0 & 0.49 < p < 0.51 \\ \frac{1}{0.49}(0.51 - p) & 0.51 < p < 1 \end{cases} \quad (4-4)$$

在這邊要注意的是，類神經網路得到的輸出並非真正輸入到各軸的速度值。而是(4-1)式中的 f_i 還需和一個當前狀況下轉軸的最大速度 V_{max} 之後才是真正輸入到各軸的控制命令。

4.5 即時線上追蹤速度調整機制

在 4.4 節的(4-1)式中如果 V_{max} 一個固定的值，則只要欲追蹤目標位於影像中相同的位置，不論物體的速度快慢系統都以相同的速度將欲追蹤目標移至影像中心。本節中將介紹系統對於不同速度的物體如何去改變追蹤速度使追蹤的時間變短、誤差減少。

在本追蹤系統中，對於不同速度的追蹤目標是以改變(4-1)式中的 V_{max} 來達到調整追蹤速度的目的其調整方程式如下：

$$V_{max}(t) = V_{max}(t-1) + e \quad (4-5)$$

其中 e 是根據目前目標物距離中心的誤差 $error(t)$ 和前一時間 $error(t-1)$ 的誤差值所決定的修正值。在計算誤差的時候，以 0.5 減掉圖 4.13 中位置座標，使影像中心移到 0 的位置，使得誤差的座標系統變成如下圖 4.7 所示：

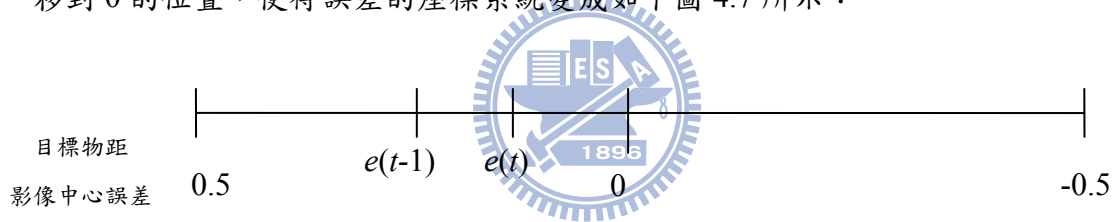


圖 4.14 目標物距影像中心誤差座標軸

而根據兩個時間誤差，可以知道此時眼球機器人追蹤目標物的速度是否太快或太慢從而調整 V_{max} 中誤差對應的 e 值，而 e 值的調整方式如下表 4.1 所示：

$e(t-1)$	$e(t) - e(t-1)$	e
$-0.01 < e(t-1) < 0.01$	×	0
$0.01 \leq e(t-1)$	$-0.01 < e(t) - e(t-1) \leq 0$	0
$0.01 \leq e(t-1)$	$0 < e(t) - e(t-1)$	1

$0.01 \leq e(t-1)$	$e(t) - e(t-1) \leq -0.01$	-1
$e(t-1) \leq -0.01$	$0 < e(t) - e(t-1) \leq 0.01$	0
$e(t-1) \leq -0.01$	$e(t) - e(t-1) \leq 0$	1
$e(t-1) \leq -0.01$	$0.01 < e(t) - e(t-1)$	-1

表 4.1 目標物誤差與修正值的對應關係表

如同 4.4 節中所提到的當物體距離影像中心僅有 $\pm 1\%$ 的誤差的時後，系統會判定已經追蹤到物體。所以在表 4.1 中當 $-0.01 < e(t-1) < 0.01$ 時 e 值為 0，亦即 V_{\max} 不做任何調整。而為了不讓眼球機器人因為追蹤速度過快而產生 overshoot 的現象，所以在系統中限制每一個 step（系統抓取 2 張影像的間隔）中只讓物體往影像中心靠近影像範圍的 1%。而如果比較 $e(t)$ 和 $e(t-1)$ 後發現影像誤差變大，則將 V_{\max} 增加使眼球機器人能夠跟上物體的移動速度。

第五章 實驗結果

5.1 定點控制 (set point control)

為了驗證上面的控制方法能使眼球機器人將目標物正確的帶至影像中心的位置，首先將物體先將目標物固定，分別驗證當目標物位於影像的左上、左下、右上、右下的時後系統基於上面離線訓練完成的類神經網路架構對於眼球機器人的控制結果。下圖中垂直座標皆為為物體距離影像中心的誤差，而水平座標則代表系統作了幾次的追蹤流程（擷取影像、計算、送控制命令到眼球機器人各軸）。每一次的追蹤流程約耗時 0.15 ~ 0.25 秒。

1. 物體位置位於影像之左上角

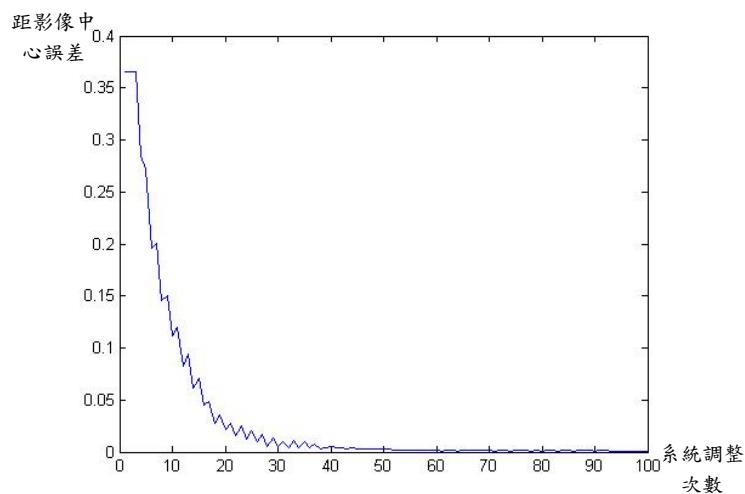


圖 5.1 物體位於影像左上位置的水平方向調校結果

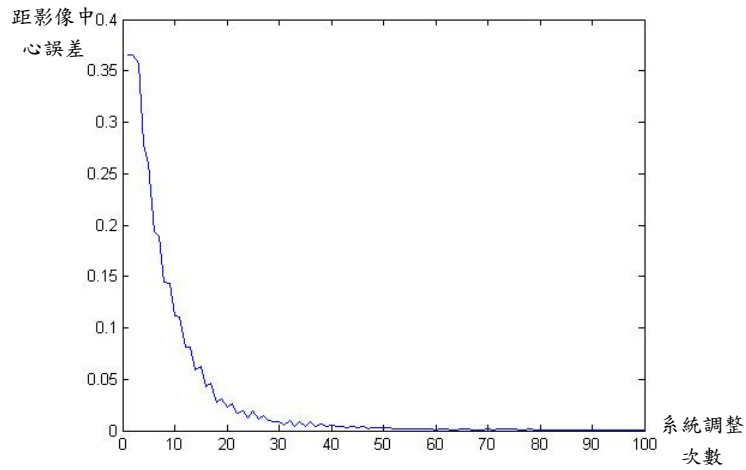


圖 5.2 物體位於影像左上位置的垂直方向調校結果

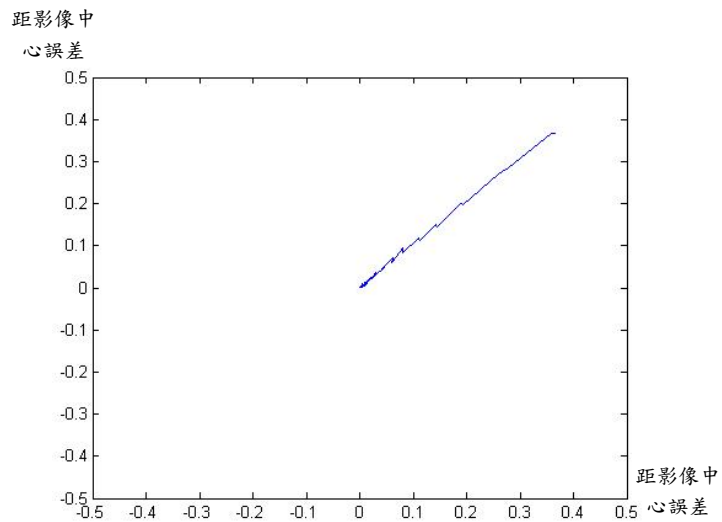


圖 5.3 物體位於影像左上位置調校的二維誤差軌跡

在這邊需要注意的是，圖 5.3 中由於是誤差軌跡圖。所以雖然影像位於物體的左上方，但是在誤差軌跡圖中看來反而是在右邊。

2. 物體位置位於影像之左下角

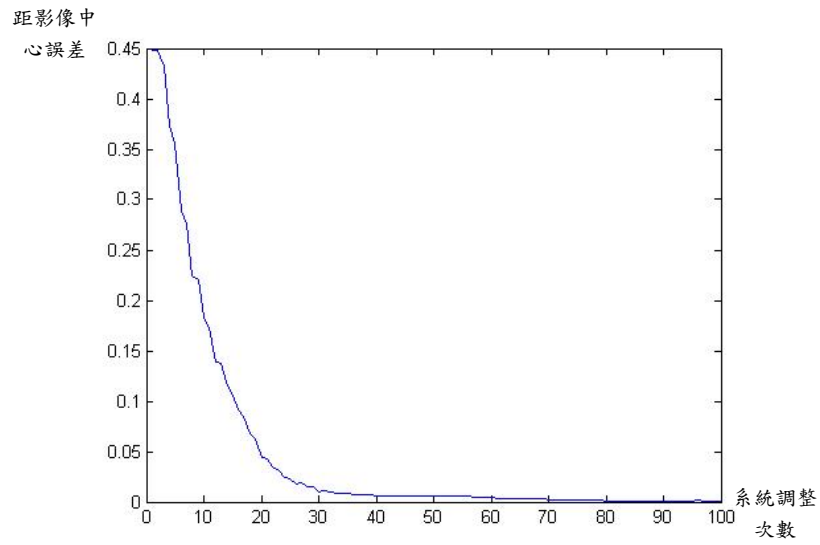


圖 5.4 物體位於影像左下位置的垂直方向調校結果

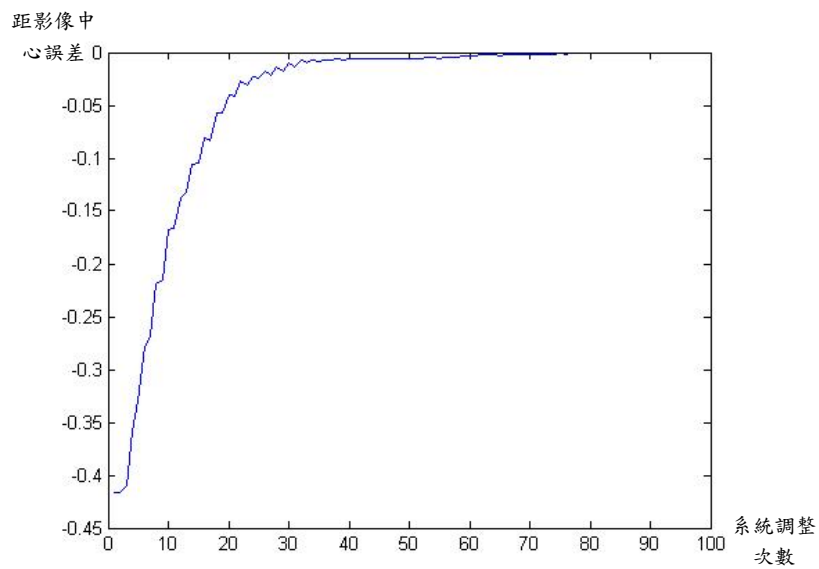


圖 5.5 物體位於影像左下位置的垂直方向調校結果

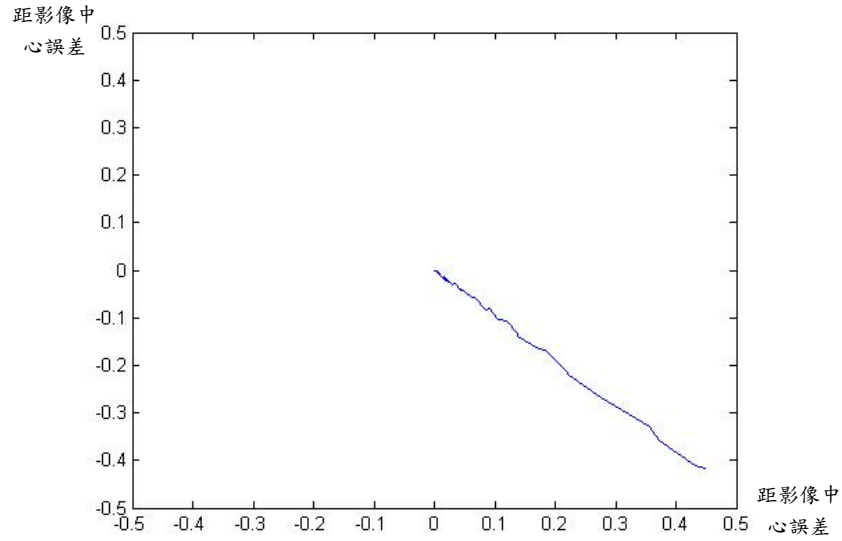


圖 5.6 物體位於影像左下位置調校的 2 維誤差軌跡

3. 物體位置位於影像之右上角

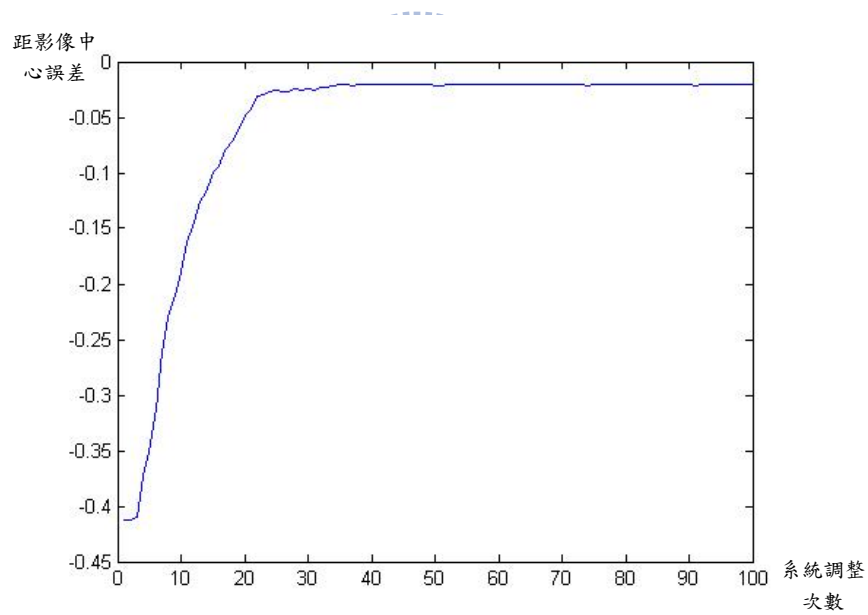


圖 5.7 物體位於影像右上位置的水平方向調校結果

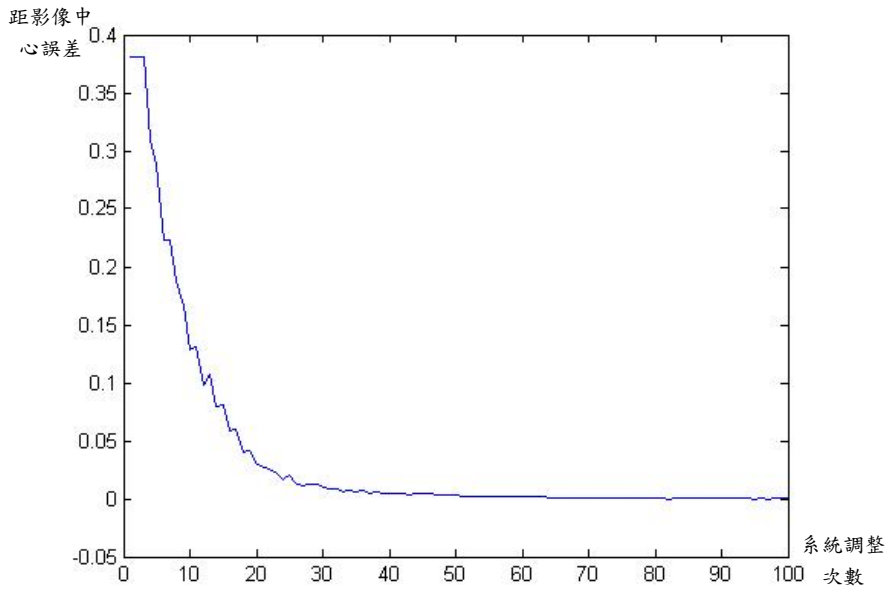


圖 5.8 物體位於影像右上位置的垂直方向調校結果

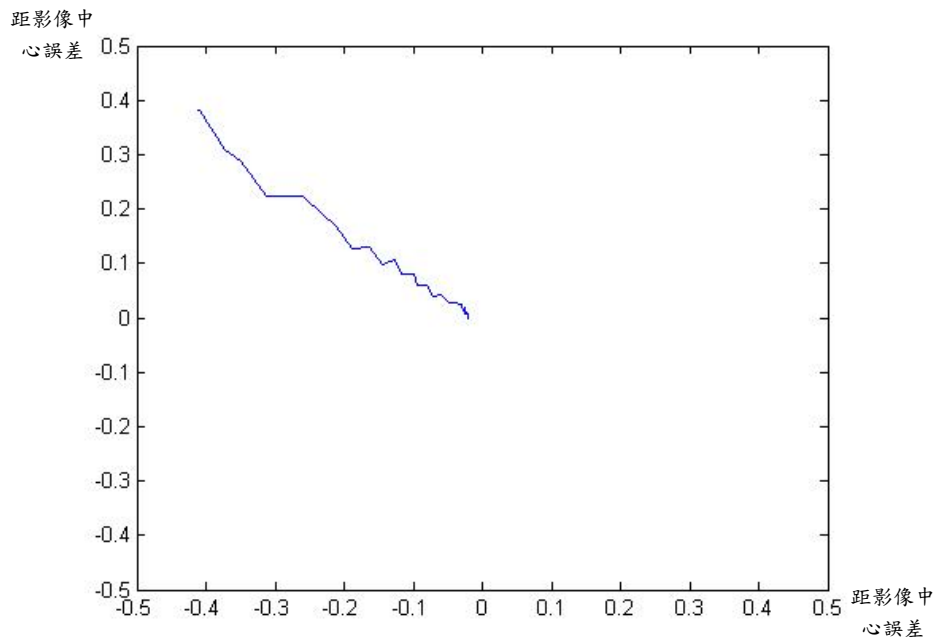


圖 5.9 物體位於影像右上位置調校的 2 維誤差軌跡

4. 物體位置位於影像之右下角

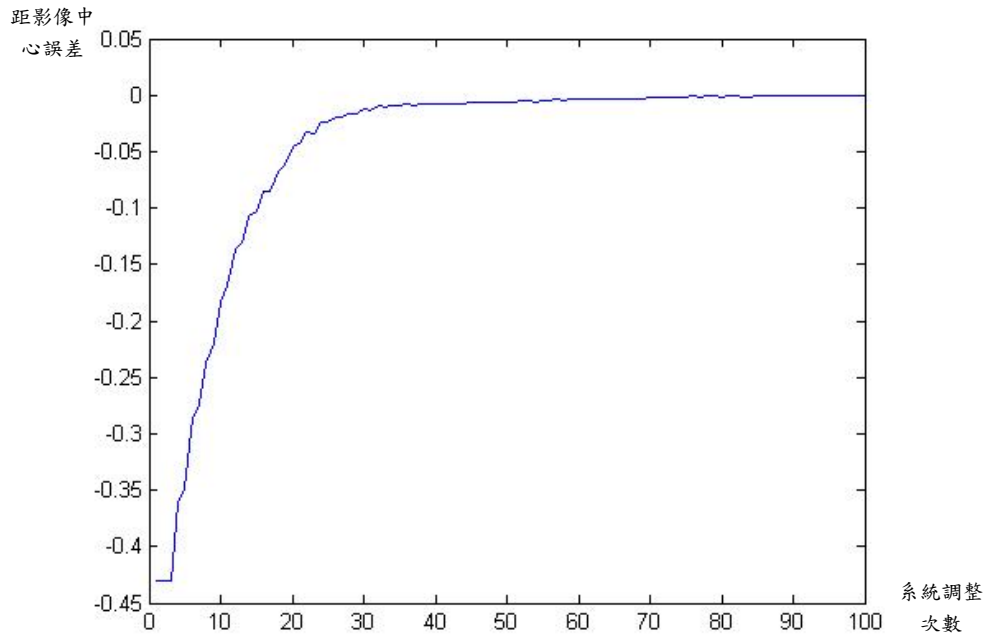


圖 5.10 物體位於影像右下位置的垂直方向調校結果

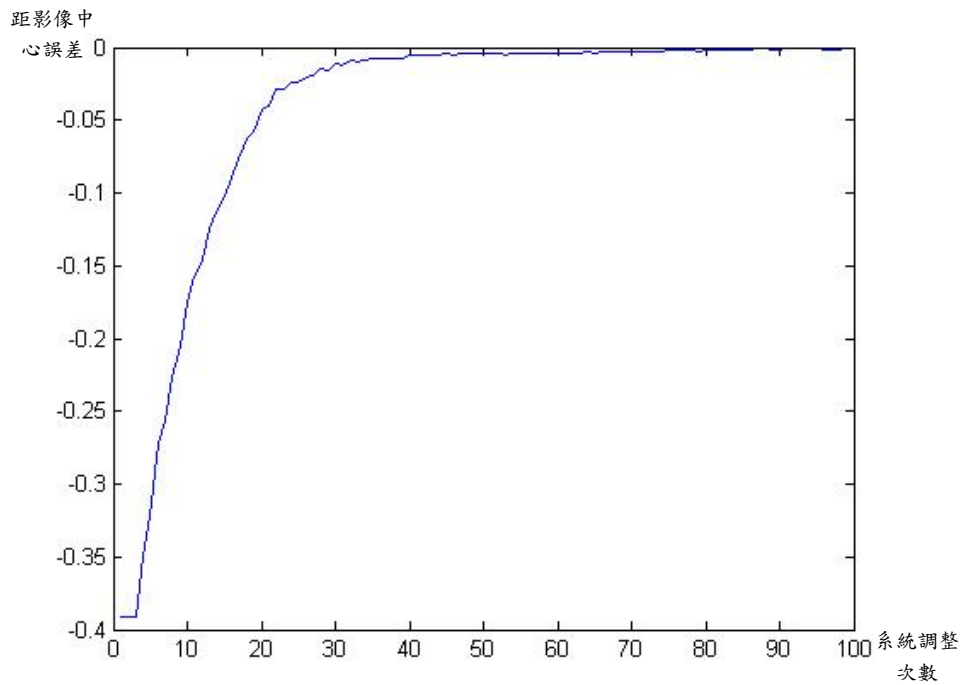


圖 5.11 物體位於影像右下位置的垂直方向調校結果

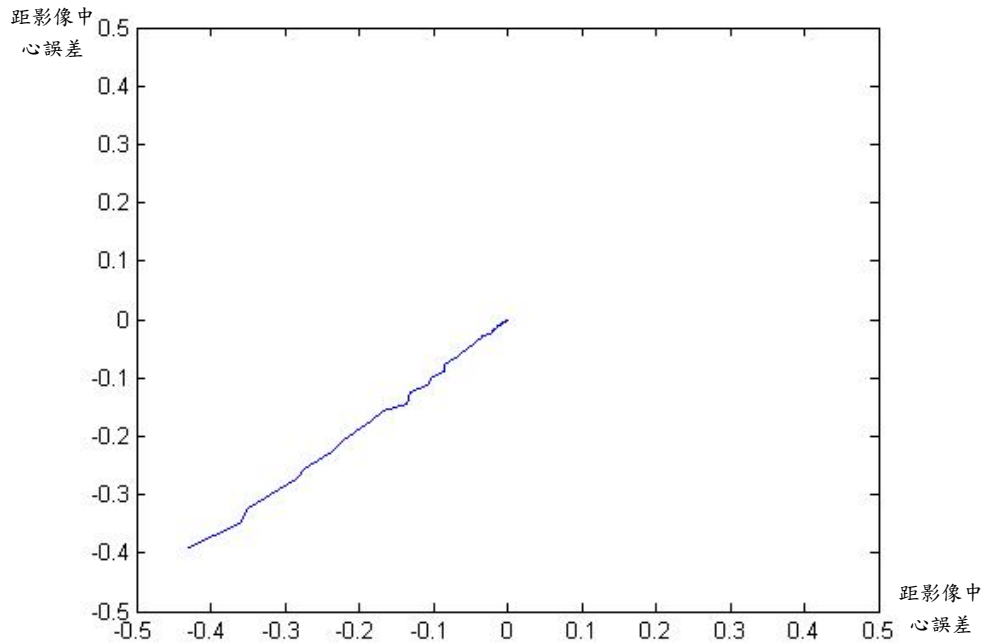


圖 5.12 物體位於影像右下位置調校的 2 維誤差軌跡

由上面的結果可以看出不管欲追蹤目標物位於甚麼位置，利用以訓練完成的類神經網路皆可成功的控制眼球機器人將欲追蹤目標物帶至影像中心。當欲追蹤目標物在角落時，系統將其帶至影像中心所花費的時間約為 8 秒。而由誤差軌跡圖也可以看出經由類神經網路的控制，欲追蹤目標物由角落到達影像中心的軌跡約和人類追蹤物體的最短路徑相仿。

5.2 目標物追蹤結果

經過 5.1 節中調校結果的驗證，確認了經過訓練後的類神經網路可控制眼球機器人將欲追蹤目標物由各角落帶至影像中心位置，接下來要驗證的是當欲追蹤目標物開始移動的時後系統在追蹤上的表現。在追蹤過程中眼球機器人脖子轉軸的最大速度 V_{\max_neck} 為 800 轉/分，而眼睛轉軸的最大速度 V_{\max_eye} 則為 100 轉/分。首先驗證的是當目標物速度固定的時候，系統對欲追蹤目標物的追蹤結果。在追蹤過程中，物體移動的軌跡如圖 5.13 所示，欲追蹤球型目標物在 50cm ×

50cm 的範圍內從右上到左下以 16 秒一次反覆的固定速度反覆的移動(單軸速度約為 0.0625m/s)。

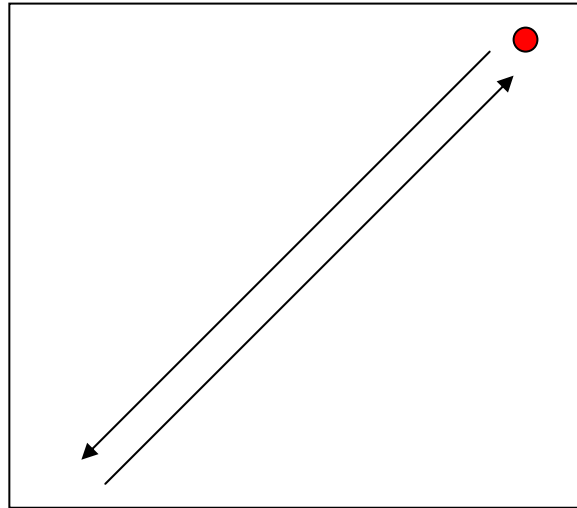


圖 5.13 欲追蹤目標物定速移動軌跡圖

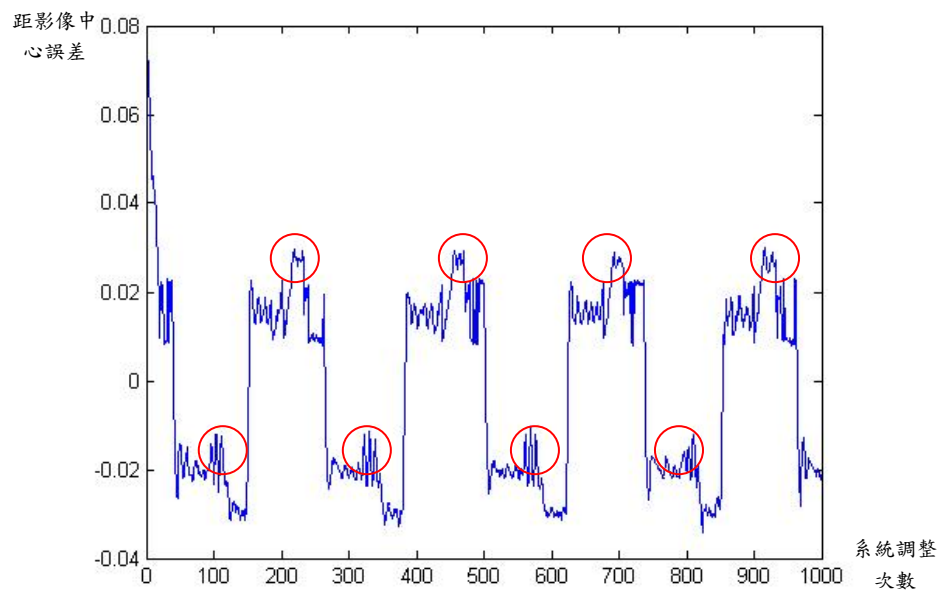


圖 5.14 目標物定速移動的水平方向追蹤誤差軌跡

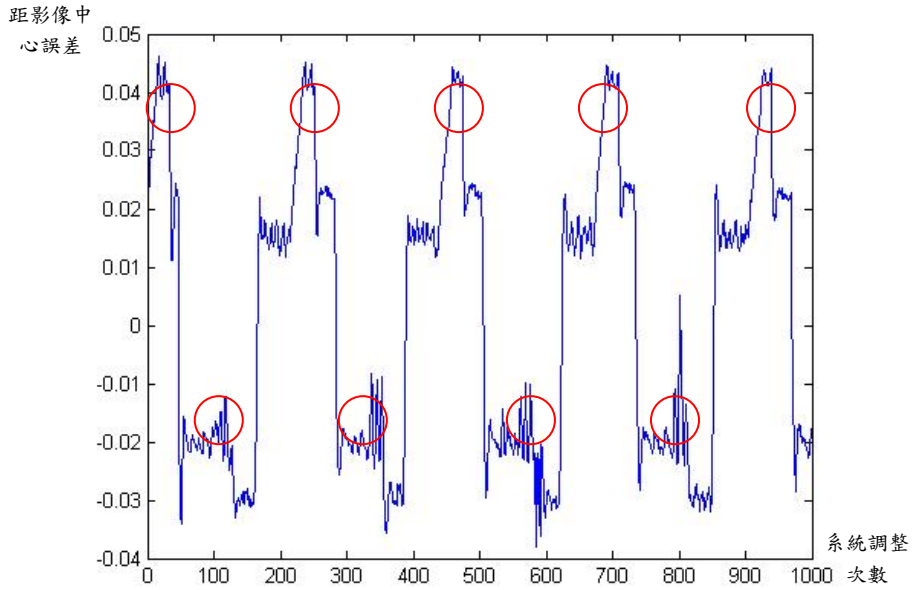


圖 5.15 目標物定速移動的垂直方向追蹤誤差軌跡

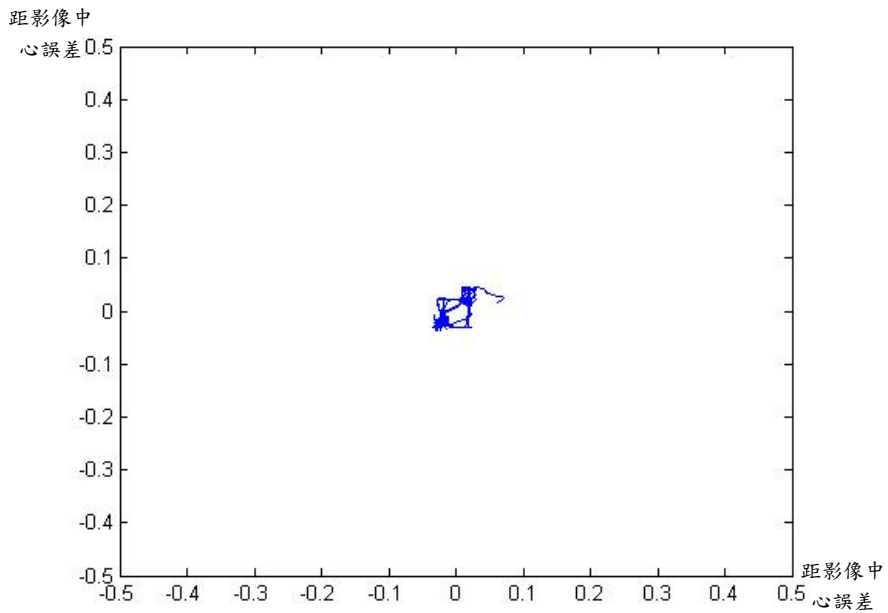


圖 5.16 目標物定速移動的 2 維追蹤誤差軌跡

追蹤的結果誤差約在 ± 0.04 ，而由圖 5.16 也可看出欲追蹤目標在整個追蹤過程中皆落在影像中心附近的一個小範圍內。在圖 5.14 和 5.15 中紅色圓圈出來的部份很明顯的看出誤差增加，這是因為此時眼球機器人因眼睛軸到達系統所限制

的極限值而做眼睛和脖子的切換動作所造成。

接下來將物體的速度隨時間規律的做改變，軌跡如下圖 5.17 以 \sin 和 \cos 函數組成一個球型的軌跡，速度為 32 秒一圈。以此來觀察追蹤系統對於目標物速度變化時的追蹤表現。

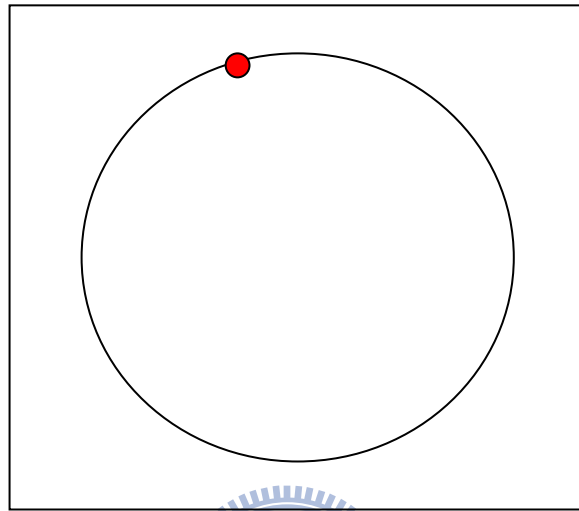


圖 5.17 欲追蹤目標物規律變速度移動軌跡圖

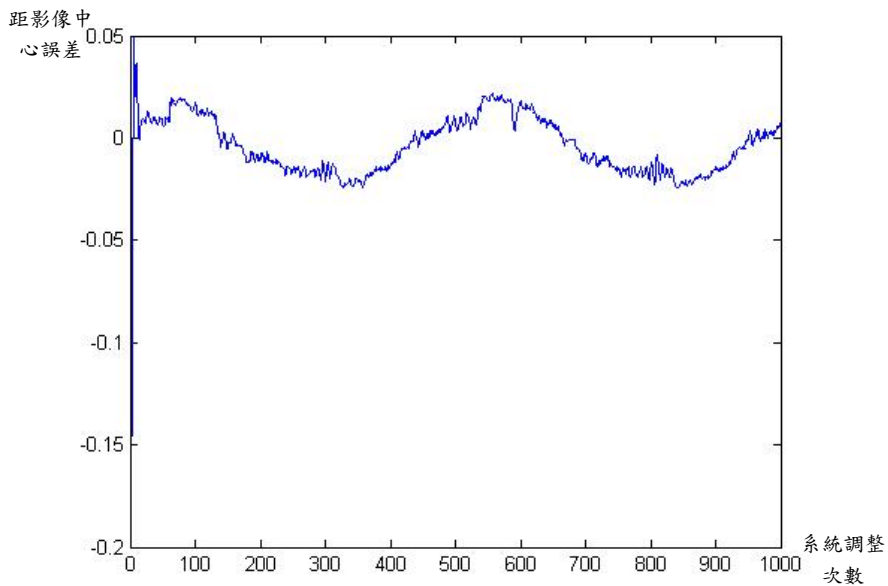


圖 5.18 目標物規律變速移動的水平方向追蹤誤差軌跡

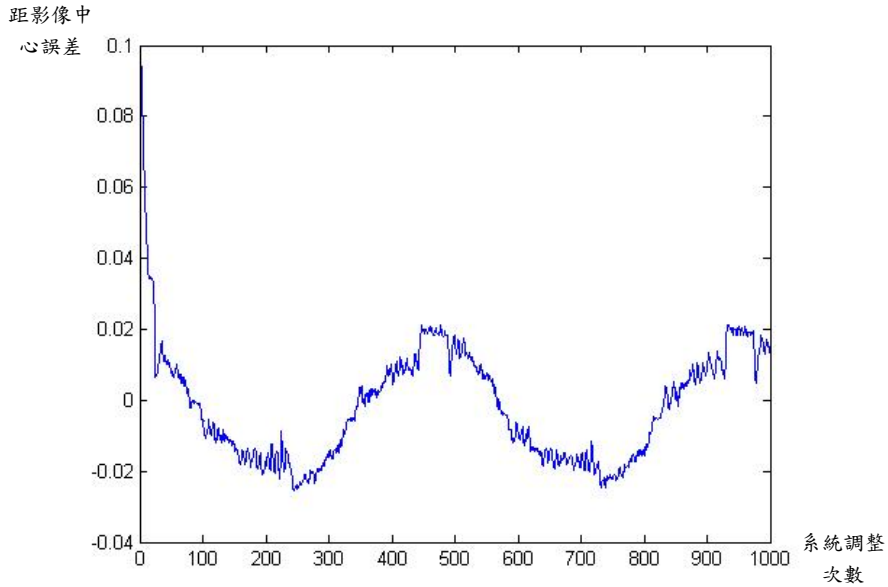


圖 5.19 目標物規律變速移動的垂直方向追蹤誤差軌跡

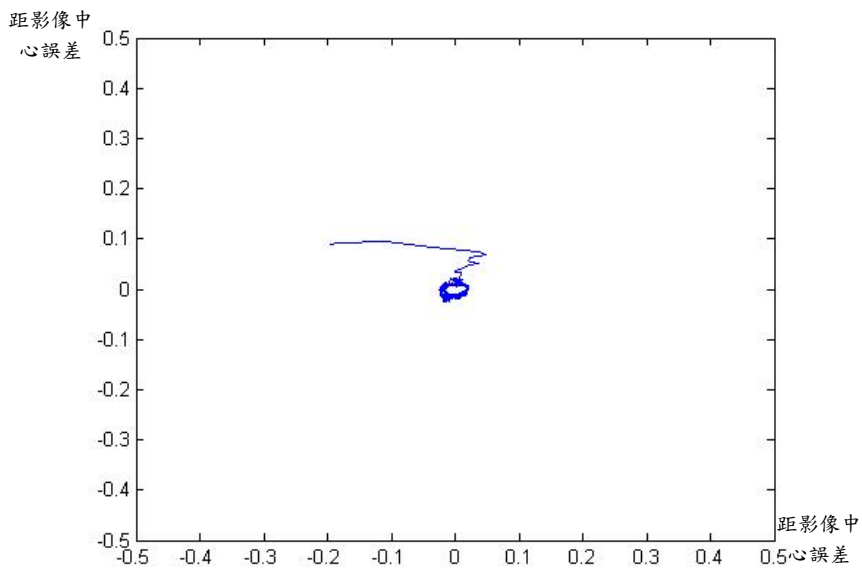


圖 5.20 目標物規律變速移動的二維追蹤誤差軌跡

在上圖中可以看出追蹤結果目標物距影像中心的最大誤差約在 ± 0.02 ，在物體速度緩慢規律變化的過程中，系統誤差也隨著物體速度而變化。物體速度越快，系統追蹤的誤差也越大，這部份將嘗試用 4.5 節中所提到的即時速度調整做修正並將結果展示在本章最後一小節。

最後測試的是當欲追蹤目標物速度隨機的作變化時，追蹤系統的反映和追蹤結果。如下圖 5.21 所示當欲追蹤目標物移動到邊框時，速度會隨機作變化，以不同的速度在 50cm × 50cm 的範圍內作直線的彈跳，從而形成不規則的軌跡。要注意的是，若目標物未碰到邊框，則速度不會作改變。

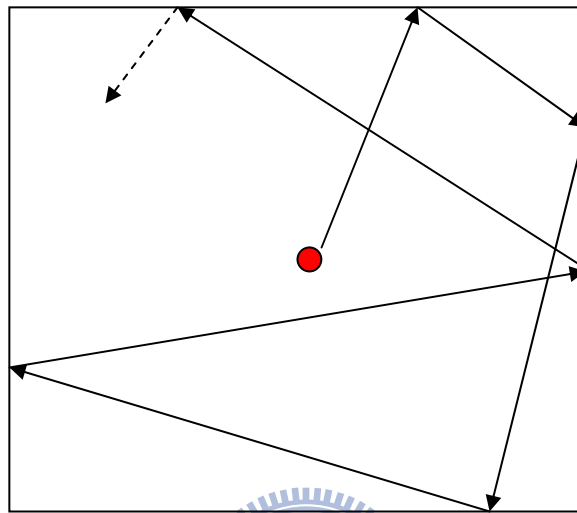


圖 5.21 目標物隨機改變速度移動軌跡圖示意圖

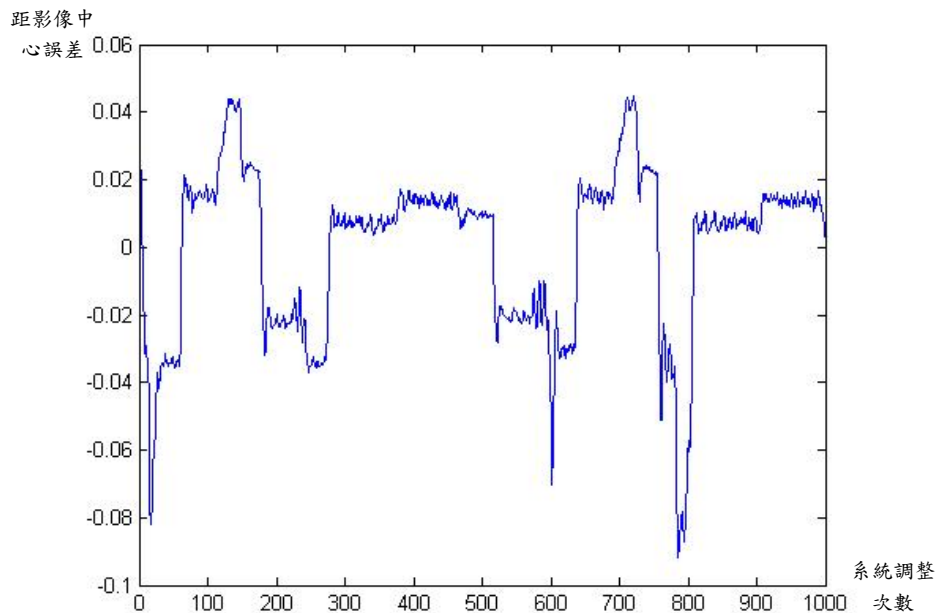


圖 5.22 目標物隨機改變速度的水平方向追蹤誤差軌跡

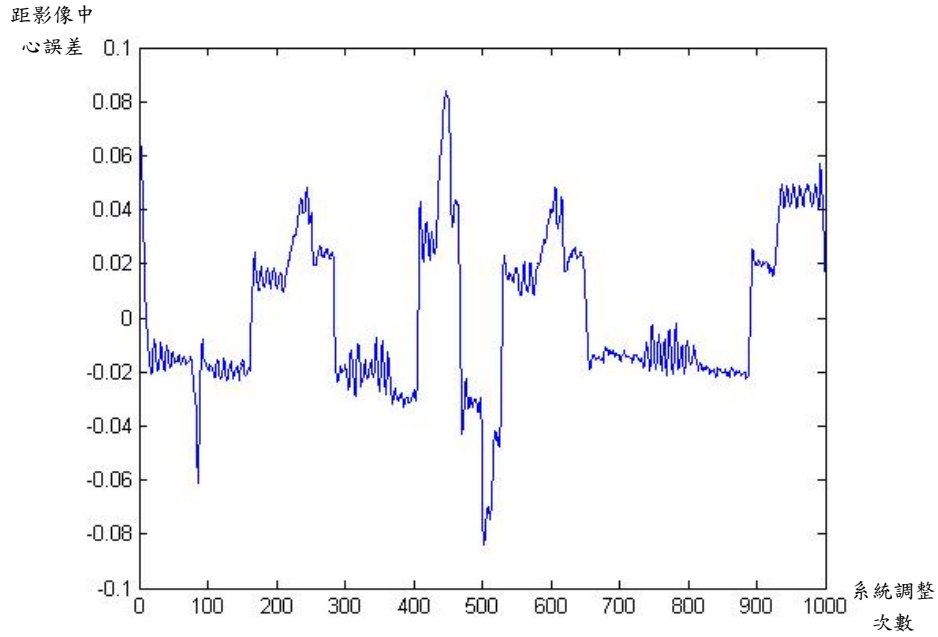


圖 5.23 目標物隨機改變速度的垂直方向追蹤誤差軌跡

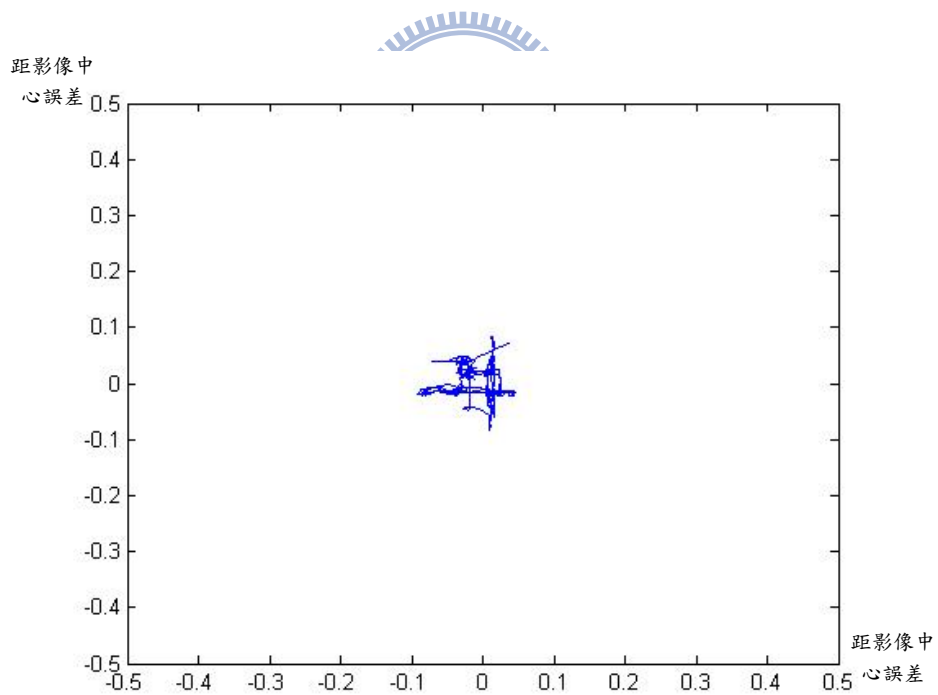


圖 5.24 目標物隨機改變速度的二維追蹤誤差軌跡

和前面兩種軌跡相比，追蹤此類型軌跡的誤差明顯的變大，最到誤差甚至到達 ± 0.08 。除了目標物速度的改變造成系統追蹤上的困難之外，因為目標物速度因

隨機的改變所達到的最大速度也比前面兩種例子還高（最高速度約為 0.1m/s），所以在結果中有較大的誤差出現。

5.3 加入即時線上追蹤速度調整機制的追蹤結果

在前面的實驗結果中因為系統的追蹤速度是固定的，所以沒辦法根據物體的速度來對追蹤誤差更進一步的作優化調整的動作。在本節中，將上面的例子重新再做一遍並將 4.5 節所提到的速度調整機制加入系統並觀察誤差的改善狀況。

為了更清楚的看出速度調整機制的的作用，本節的追蹤將眼球機器人脖子轉軸和眼睛轉軸的初始速度設為 5.2 節的一半，亦即 V_{\max_neck} 為 400 轉/分而 V_{\max_eye} 為 50 轉/分。

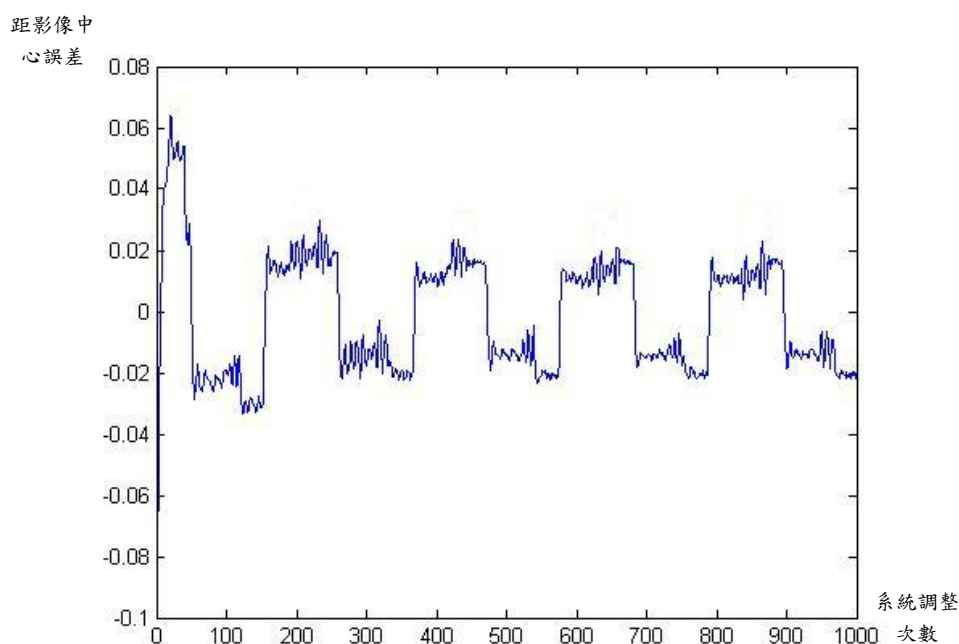


圖 5.25 目標物定速移動加入速度調整機制的水平方向追蹤誤差軌跡

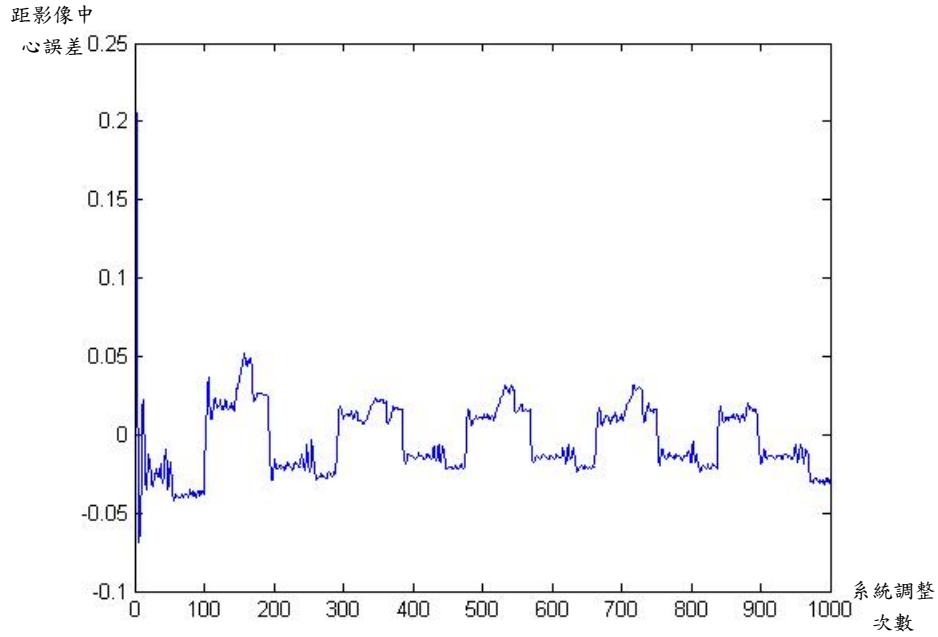


圖 5.26 目標物定速移動加入速度調整機制的垂直方向追蹤誤差軌跡

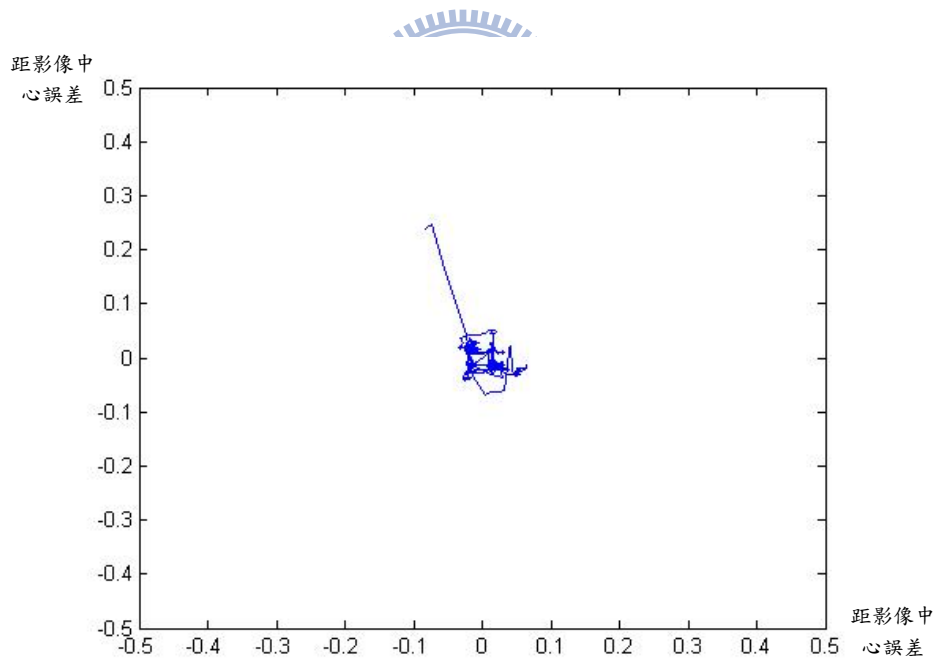


圖 5.27 目標物定速移動加入速度調整機制的二維追蹤誤差軌跡

由圖 5.25 和圖 5.26 可以看出水平和垂直的誤差皆在慢慢的變小，到最後水平的誤差約在 ± 0.02 左右而垂直的誤差約在 ± 0.03 附近。和 5.2 節中 ± 0.04 的誤差相

比較，明顯的加入速度調整機制後可以有更佳的追蹤軌跡。由圖 5.27 也可以看出軌跡圖由外慢慢往內繞，離影像中心的誤差越來越小。

再來是目標物作規律變速移動的狀況，在系統加入速度調整機制後的追蹤結果。

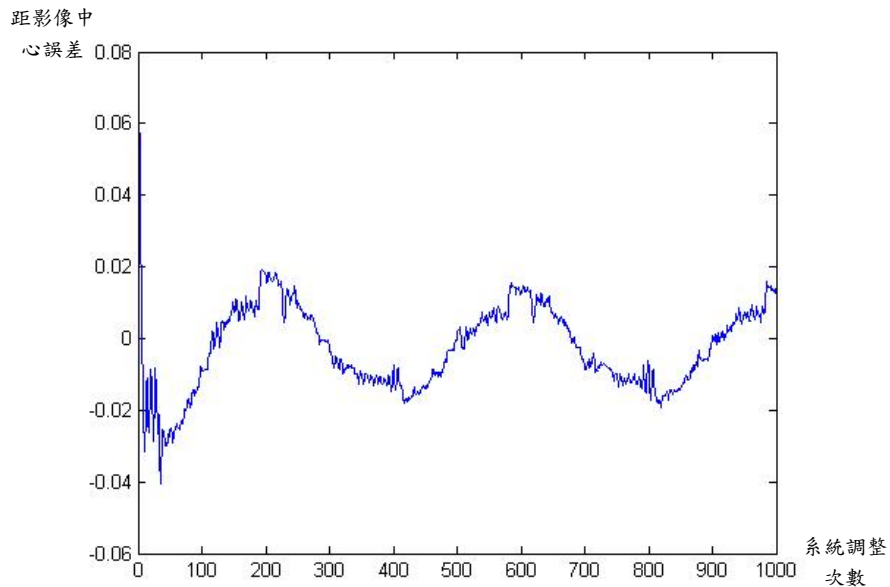


圖 5.28 目標物規律變速移動加入速度調整機制的水平方向追蹤誤差軌跡

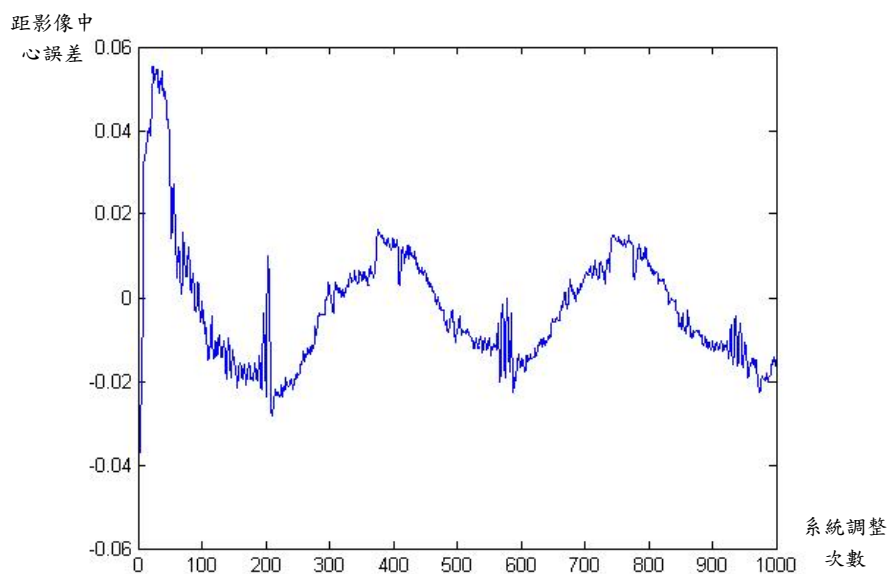


圖 5.29 目標物規律變速移動加入速度調整機制的垂直方向追蹤誤差軌跡

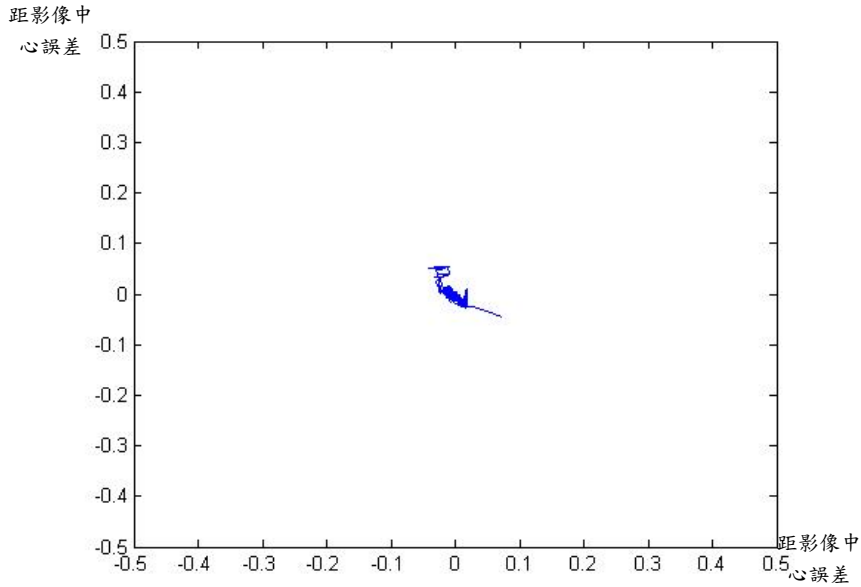


圖 5.30 目標物規律變速移動加入速度調整機制的二維追蹤誤差軌跡

由圖 5.28~5.30 中還是可以看出不管在水平或垂直的軌跡誤差上都有由大到小慢慢改善的狀況。加入速度調整機制後最終的誤差約在 ± 0.018 上下，可是和 5.2 節目標物規律變速移動的最終誤差 ± 0.02 比較，並沒有目標物定速移動的狀況中改善得這麼多。這是因為在本追蹤狀況中，目標物的移動速度已經相當的慢（最高速度約為 0.04m/sec），原本 5.2 節中眼球機器人的初始設定（脖子轉軸的最大速度 V_{\max_neck} 為 800 轉/分，眼睛轉軸的最大速度 V_{\max_eye} 則為 100 轉/分）已經貼近了該速度的最佳追蹤轉速。故在本狀況中雖然誤差還是由大到小，但和未加入速度調整機制的狀況相差不大。但是在圖中還是可以發現，若原本的初始設定使誤差過大，加入速度調整機制後是可以將誤差軌跡慢慢修正回來的。

最後將目標物速度隨機變化的狀況加入速度調整機制後，因為速度調整機制的反應時間不夠快，而速度變化的間隔又太短。所以在加入後反而造成比原本固定機器人各軸追蹤速度還要差的軌跡誤差，這將於結論中作總結說明。

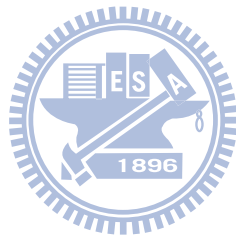
第六章 結論

在本論文中，將影像的偵測、追蹤和眼球機器人控制，整合成一個影像追蹤系統，並以類神經網路配合倒傳遞演算法作為眼球機器人控制的核心，試圖使系統的追蹤結果盡可能的接近人類頭部追蹤影像的模式。

從 5.1 節的實驗結果中可以發現，用類神經網路為控制核心的追蹤系統在目標物固定的狀況下，將目標物移至影像中心的軌跡幾乎是直接由目標物初始位置直接走最短路徑的到達影像中心。這和離線訓練時要求的接近人類頭部追蹤影像的模式相符合，驗證了用類神經網路來控制本影像追蹤系統的可行性。在實際追蹤的表現方面則跟目標物的移動速度、目標物速度是否改變和速度改變的方式有很大的關連性。在本影像追蹤系統中，從擷取一張影像、辨識目標物、計算目標物位置、計算各軸速度到下命令給各軸這樣的一次流程約花時 0.15~0.25 秒，所以當追蹤的目標物速度越快的時後所造成的誤差也就相對的增加。雖然在 5.3 節中系統加入了速度調整的機制，由實驗結果中也可以看出經過速度調整後的系統在固定速度和目標物速度規律改變的狀況下都有著較好的追蹤結果。但是受限於系統反應時間的關係，對於速度變化太快的目標物此調整機制無法有效的對於變化的速度做出相應的調整。

本系統成功的將類神經網路應用在眼球機器人上，且對追蹤和速度的調整也有了一個初步的架構。在未來的發展上可以結合影像辨識的部份，使系統先辨識出目標物的類型並由目標物類型來決定眼球機器人中各軸的初速。這樣不但可以使系統更加的具有人類的智慧，而且在追蹤上面也避免了一開始過大的誤差使系統反應不及而導致追蹤的失敗。另外一方面，跟人眼每秒 16 張影像的最低要求比起來，系統每秒 4~6 張的影像擷取速度顯然過慢。這造成了追蹤系統對於速

度太快的影像會因為反應時間跟不上而追丟，而速度調整機制也來不及反應的狀況。對於這部份來說 4.5 節所提到的速度調整機制顯然還有很大的改善空間，而在程式的編寫上也可以轉往 C code 上面編寫以提升系統的反應速度。期許將來本系統能朝這幾方面作改善，使得本系統可以應用在各個場合的追蹤上。



參考文獻

- [1] A. Arsenio and J. Santos-Victor, "Robust Visual Tracking by an Active Observer," IEEE IROS 97, Vol 3, pp.1342-1347, 1997.
- [2] D. H. Nguyen and B. Widrow, "Neural networks for self-learning control systems," *IEEE Control Systems Magazine*, pp. 18-23, Apr. 1990.
- [3] A.M. Baumberg, and D.C. Hogg, "An Efficient Method for Contour Tracking using Active Shape Models," Motion of Non-Rigid and Articulated Objects, Proceedings of the IEEE Workshop , pp.194-199, Nov. 1994.
- [4] J.S.R. Jang, "Self-learning fuzzy controller based on temporal backpropagation," *IEEE Trans. Neural Networks*, Sept. 1992.
- [6] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Reading, MA: Addison-Wesley, 1991.
- [7] O. Omidvar and D. L. Elliott, *Neural Systems for Control*. New York: Academic, 1997.
- [8] W.T. Miller III, R.S. Sutton, and P.J. Werbos, Eds., *Neural Networks for Control*. Cambridge, MA: MIT Press, 1990.
- [9] B. C. Kuo and F. Golnaraghi, *Automatic Control Systems*, Wiley, 2002.
- [10] Buchanan, J. L. and Turner, P. R., *Numerical methods and analysis*, McGRAW-Hill, 1992.
- [11] Matlab 程式設計與應用，張智星著，清蔚科技出版。
- [12] 應用類神經網路，葉怡成著，儒林圖書公司，1997。
- [13] 人工神經網絡模型分析與應用，閻平凡，黃端旭著 安徽教育出版社，1993。
- [14] 數位影像處理，Woods R.E., Gonzalez R.C.著；繆紹綱譯，普林斯頓國際有限

公司，2002。

[15] 倒傳遞類神經網路在冰水機節能控制上之應用，古國松，賴炎生，林俊民著，國立台北科技大學電機工程研究所，2004。

