

國立交通大學

機械工程學系

碩士論文

基於短期風力預測估算最大電能擷取之  
大型風機迎風轉向控制估測研究

**Study of Wind Turbine Yaw Control Algorithm Based on Maximum  
Energy-Captured Evaluation From the Short-Term Wind Power  
Forecasting**

研究生：蔡琿竹

指導教授：鄭泗東 博士

中華民國一百零二年十月

基於短期風力預測估算最大電能擷取之

大型風機迎風轉向控制估測研究

**Study of Wind Turbine Yaw Control Algorithm Based on Maximum  
Energy-Captured Evaluation From the Short-Term Wind Power  
Forecasting**

研究生：蔡琺竹

Student : Jiun-Jwu Tsai

指導教授：鄭泗東 博士

Advisor : Dr. Stone Cheng

國立交通大學

機械工程學系

碩士論文

A Thesis

Submitted to Department of Mechanical Engineering

Collage of Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of Master

In

Mechanical Engineering

October 2013

Hsinchu, Taiwan, Republic of China

中華民國一百零二年十月

# 基於短期風力預測估算最大電能擷取之 大型風機迎風轉向控制估測研究

研究生:蔡珺竹 指導教授:鄭泗東博士

國立交通大學機械工程研究所碩士班

## 中文摘要

此篇論文以 2MW 之大型風力發電機為研究載具，發展提升其發電效率之控制法則，提出以前饋倒傳遞類神經網路 (Neural Network)預測短期風向、風速，計算風能變化，整合風機迎風轉向系統的響應時間及推導風機轉向耗能與風能擷取分析，使迎風轉向反應時間較慢的大型風力發電機可依據短期風力風向預測資料，做適當的風機迎風轉向控制，進而達到在降低風機轉向耗能之下擷取最大風能之目的，以節省不必要的偏向運轉。本論文研究方法由長期的風力觀測資料庫，以前饋倒傳遞類神經網路進行短期風速與風向預測，並與歷史資料比較其準確性。由風速與風向預測結果推導在風能擷取與風機轉向耗能正負相抵間之最高電能擷取功率的轉向機制。研究結果顯示，本論文提出之迎風轉向控制法則在平均風速約為 5.924m/s 的風場下，可比隨風向迎風轉向之風機多出約 1.1kW 的淨獲能功率。

關鍵字：類神經網路、風機迎風轉向系統、風機轉向耗能

# **Study of Wind Turbine Yaw Control Algorithm Based on Maximum Energy-Captured Evaluation From the Short-Term Wind Power Forecasting**

Student : Jiun-Jwu Tsai

Advisor : Dr. Stone Cheng

Department of Mechanical Engineering  
National Chiao Tung University

## **Abstract**

Based on 2MW large-scale wind turbine platform, this research proposed feed-forward back-propagation neural network (NN) algorithm to predict short-term wind direction and wind speed for calculating the wind energy's variation. The estimation results of maximum power capture as well as the analysis of nacelle veering energy consumption are integrated with the response time of wind turbine yaw operation to derive wind energy gains and losses in order to make the appropriate wind turbine yaw control by predicting short-term wind data and finding out the best yaw angle to capture maximum wind energy which included the loss of consumption for turning orientation. This yaw control algorithm is expected to reduce unnecessary turning of large wind turbines which need longer time to change orientation. In additions, the prediction methods based on various structures of the feed-forward neural network from the values of meteorological variables are also described in this work. The result show that the proposed algorithm could have a surplus of about 1.1kW Watt net power compare to the same type specification wind turbine that yaw directly follow with wind direction under the condition of 5.924m/s wind velocity. Finally, there is a simulated human machine interface for showing the conditions of controlled turbine facing and the real wind direction and velocity.

Keywords—Short term wind speed prediction, Neural networks, Back propagation.

## 致謝

感謝指導教授鄭泗東博士循循善誘、孜孜不倦的教誨，讓學生可以學到很多課本上沒有的經驗，從一開始懵懵懂懂而後一步一步地完成研究，最終順利完成這份碩士論文。再來感謝口試委員陳宗麟博士、林昭安博士、鄭錦榮博士在口試期間給予學生意見，讓這份論文研究更加完善。

感謝研究所期間風力研究計畫的靖哲、炫錡學長，你們的觀念跟能力都很紮實，在我碩一時期給了我很多的幫助，我很高興可以當實驗室風力這塊，你們的學妹。感謝有阿宏、米奇、小游、丞哥等風趣幽默的學長，你們充實了我在新竹的食物版圖，讓我更了解新竹這個地方，也讓我的研究生生涯豐富了不少。

感謝婕安、歆葦、翔翔、士傑、燴飯同屆實驗室同學，有你們讓我在學校不孤單，雖研究領域不同但心一起為進度而打拼，希望以後工作也能遇到像你們一樣的夥伴。

感謝實驗室所有的學弟，定諺、柯偉、丞弟、嘉翔、柏霖、家暉、安翔，一起參與實驗室出遊、打球、吃飯的那些時光幫我排解了許多無聊。

最後感謝我的家人、老師、朋友一路以來的支持鼓勵，你們是我的精神支柱讓我得以完成這份論文，謝謝大家。

# 目錄

中文摘要.....	i
Abstract.....	ii
致謝.....	iii
目錄.....	iv
圖目錄.....	vi
表目錄.....	viii
第一章 緒論.....	1
1-1 引言.....	1
1-2 研究動機.....	2
1-3 文獻回顧.....	3
1-4 論文大綱.....	6
第二章 風力發電系統架構.....	7
2-1 風力發電機原理與架構.....	7
2-2 風力發電機迎風轉向系統結構.....	10
第三章 風能預測研究方法.....	14
3-1 支持向量機(Support Vector Machine regression algorithm).....	14
3-2 決策樹(Decision Tree).....	16
3-3 類神經網路(Artificial Neural Network).....	16
3-3-1 多層感知器類神經網路(Multilayer Perceptrons Network).....	18
3-3-2 倒傳遞類神經網路(Back Propagation Network).....	20
3-3-3 徑向基類神經網路(Radial Basis Function Network).....	21
第四章 前饋倒傳遞類神經網路原理與架構.....	23
4-1 批次梯度下降倒傳遞演算法(Steepest Decent back propagation).....	25

4-2 具動量的批次梯度下降演算法(Momentum back propagation) .....	25
4-3 可變學習速率倒傳遞演算法(Variable Learning Rate back propagation)...	26
4-4 有彈性的倒傳遞演算法(Resilient back propagation).....	27
4-5 共軛梯度演算法(Conjugate Gradient back propagation) .....	27
4-6 牛頓法(Newton's method) .....	28
4-7 擬牛頓法 BFGS(Quasi-Newton algorithms , BFGS) .....	29
4-8 一步階正割演算法(One Step Secant) .....	30
4-9 Levenberg-Marquardt 演算法 .....	30
4-10 前饋倒傳遞類神經網路收斂情形比較 .....	31
第五章 類神經模擬.....	35
5-1 類神經研究方法與步驟 .....	35
5-2 類神經網路短期風能預測 .....	36
第六章 風機迎風轉向控制.....	44
6-1 風力發電機組技術參數 .....	44
6-2 迎風轉向系統轉移函數 .....	45
6-3 風機面向角度控制方式 .....	47
6-4 風機實際反應之模擬 .....	51
第七章 結論與未來展望.....	57
7-1 結論 .....	57
7-2 未來展望 .....	58
第八章 參考文獻.....	59

# 圖目錄

圖 1	文獻六中的轉向指令圖.....	5
圖 2	彰化濱海風場風速風力資料[24].....	5
圖 3	輪軸式風力發電機.....	7
圖 4	直驅式風力發電機.....	8
圖 5	永磁直驅式風力發電機內部之三相繞組.....	8
圖 6	軸輪式風力機內部的輪軸箱部分.....	8
圖 7	二維 $C_p$ 圖( $\beta=0$ ).....	10
圖 8	三維 $C_p$ 圖.....	10
圖 9	迎風轉向馬達(共四台).....	11
圖 10	風機迎風轉向控制方塊示意圖.....	11
圖 11	迎風轉向系統控制架構.....	12
圖 12	風力發電機迎風轉向系統剖面示意圖.....	13
圖 13	SVM 理論的分類概念.....	15
圖 14	SVM 網路構造圖.....	15
圖 15	決策樹結構.....	16
圖 16	一個單輸入有偏權值的單神經元架構.....	17
圖 17	單層感知器的網路架構.....	19
圖 18	類神經網路訓練.....	20
圖 19	單層前饋倒傳遞網路結構.....	21
圖 20	徑向基神經元架構圖.....	22
圖 21	前饋倒傳遞類神經網路演算法分類.....	25
圖 22	各類倒傳遞前饋類神經網路演算法的收斂情況.....	34
圖 23	類神經網路輸入輸出資料形式.....	35



圖 24	龍山國小風向實際值與預測值比較圖.....	38
圖 25	龍山國小風速實際值與預測值比較圖.....	39
圖 26	風向之預測網路不同訓練筆數比較.....	41
圖 27	風速之預測網路不同訓練筆數比較.....	41
圖 28	風向之預測網路輸入個數改變比較.....	42
圖 29	風速之預測網路輸入個數改變比較.....	42
圖 30	風向之預測網路隱藏神經元改變比較.....	42
圖 31	風速之預測網路隱藏神經元改變比較.....	43
圖 32	XE93-2000 風力發電機組[12].....	44
圖 33	$\Delta M_f(.)$ 函數曲線.....	46
圖 34	決定迎風轉向角度之流程圖.....	50
圖 35	同向時，以 $t_v$ 、 $t_s$ 構成的 $W-E$ 性能函數曲面.....	51
圖 36	實際風向與能量最佳化之轉向控制比較圖.....	52
圖 37	預測風向與能量最佳化之轉向控制比較圖.....	52
圖 38	控制與實際的相差角度.....	52
圖 39	風機轉向五分鐘平均獲得功率.....	52
圖 40	風機轉向五分鐘平均消耗功率.....	53
圖 41	隨風向迎風轉向的風機面向與真實風向比較圖.....	53
圖 42	隨風向迎風轉向與經預測控制迎風轉向之風機面向比較圖.....	54
圖 43	隨風向迎風轉向與經預測控制迎風轉向五分鐘平均耗能功率比較圖....	54
圖 44	隨風向迎風轉向與經預測控制迎風轉向五分鐘平均獲能功率比較圖....	54
圖 45	隨風向迎風轉向與經預測控制迎風轉向之五分鐘平均淨獲能功率比較	55
圖 46	隨風向迎風轉向與經預測控制迎風轉向之五分鐘平均淨獲能功率差異	55
圖 47	顯示風機面向與實際風向及功能獲取的人機介面.....	56

# 表目錄

表 1 文獻[8]的誤差比較表.....	14
表 2 轉移函數及其功能.....	17
表 3 不同版本之共軛梯度法比較.....	28
表 4 各種類神經網路訓練函數預測風向誤差比較表.....	36
表 5 各種類神經網路訓練函數預測風速誤差比較表.....	37
表 6 LM 類神經網路改變訓練資料量的預測誤差比較.....	40
表 7 LM 類神經網路改變輸入個數的預測誤差比較.....	40
表 8 LM 類神經網路改變隱藏層神經元個數的預測誤差比較.....	40
表 9 XE93-2000 型直驅永磁風力發電機組主要技術參數[12].....	44



# 第一章 緒論

## 1-1 引言

風是太陽輻射熱所引起的一種自然現象，太陽光照射到地球表面，因為各處受熱不同產生溫差，造成大氣對流形成風，因此廣義上來說，風能係屬於太陽能的一種。二十一世紀全球大多數的國家均將節能減碳列為重要的國家發展項目之一，由於氣候變遷的速度已遠超過人類的想像，發展替代能源尤為重要，風力發電在替代能源方面擁有很大的潛力。目前許多國家已列為重點發展之能源方案，2008 年全球新增裝置為 27,261MW，年成長率為 29%，累計全球裝置總容量 121,187MW，2009 年全球新增風力發電容量為 37,466MW，較 2008 年成長 31%，2020 年風力發電量目標為 1,500,000MW，佔全球電力配比的 12%。[1]

目前台灣西岸一年可讓風機滿載發電的時間(滿發時數)平均約 2,500 小時，而澎湖更有高達 3,500~4,300 小時，幾乎為全世界數一數二的風力。相比之下，歐洲的風場發展潛力較弱，其滿發時數若能達 2,200 小時已屬不錯。而風機在台灣全年實際發電量佔全年滿載發電的百分比(容量因素)為 29%，即使是較無風的夏天也達 20%，澎湖風場的容量因素更高達 40~50%。總結來說台灣是一個很適合發展風力發電的地方。[30]

風力發電在再生能源當中為一種具有潛力發展的替代能源，隨著大型風力發電機發展越趨成熟，對於如何控制大型風力發電機迎風轉向的技術也越來越重要。因此，整合風場風能預測技術的迎風轉向控制系統可成為提昇風機發電效率發展方向之一。

風力發電機由擷取風能產生電力，其葉片需與風向呈最佳角度以擷取最大風能。小型風機面向只須隨著風向儀轉動即可，但是大型風機需要以系統控制的方式來轉向。為使發電的效率能提到最高及疲勞負載可以降到最低，理想上期望大型風機永遠面向風向，使風能最充分運用。但實際上，風向不停的改變，大型風

機迎風轉向系統無法隨時跟得上其改變速度。風向的轉變常常詭變不穩，若在其不穩的情況下，保持風機面向隨時使與風向一致，可能會使轉向頻繁而作功增加；此外，大型風機迎風轉向系統的響應速度較慢，若以振幅過大且頻率過多的面向指令來控制，可能會造成系統未響應完全而風向已改變的狀況。

基於以上理由，為求減少多餘的風機轉向作功，並得到更高的風機使用效率，本文以大型風力機為研究載具，利用大型風場資料來做預測並搭配風機轉向消耗做轉向控制，期待未來能在提昇大型風場能量擷取的部分有所貢獻。

## 1-2 研究動機

風力發電機迎風轉向系統可分為主動迎風轉向與被動迎風轉向。小型風力發電機因葉片截面積小、擷取之風速較低、電能產量較低維修成本不宜過高……等緣故，不需使用主動迎風轉向來控制面向；相較之下，大型風力發電機機構負載極大，若因風向改變而隨即轉動風機面向，使轉向次數過於頻繁致，不僅會使風力發電機系統耗能增加；且其剎車系統耗損速度會加快，致壽命簡短。本研究主旨為設計迎風轉向控制系統，經由風能預測與演算法設計，控制風機面向以達到最大風能擷取及最小轉向耗能平衡。

本文利用 matlab 之類神經網路 toolbox 訓練出合適的網路架構模型，以其預測模擬出未來的風力風向資訊。將此資訊套用至轉向耗能分析後，與所能擷取之風能功率相比較以得到最適當的轉向角度，使其能降低轉向次數以兼顧大型風機轉向之疲勞耗損及耗能。將此經推導而得的風機最佳迎風轉向控制與隨風向迎風轉向的功率得失做比較以評估此預測控制的效益。最後將實際風向與風機迎風轉向結合以動畫呈現在 Matlab 人機介面上，並繪出擷取功率走勢圖來顯示控制過程。

### 1-3 文獻回顧

本文之文獻回顧的文章主要分兩部分，一是預測風力風向的資料探勘理論，二是控制風機機艙面向的應用方法。目前學術界在預測風力風向及控制風機迎風轉向上皆有相當豐富的研究發表。以下將幾項本文中參考之文獻列舉並作介紹。

1. Andrew Kusia 等人發表於 2010 年, “Short-Horizon Prediction of Wind Power\_ A Data-Driven Approach”, 此篇針對短期風能預測使用支持向量機(Support Vector Machine regression algorithm)、類神經網路(neural network algorithm)、boosting tree 等多種方法, 並列表比較其預測準確度。從文獻中可知, 只要資料來源足夠, 類神經網路之預測結果誤差值與其他方法相比皆偏低。[2]
2. J. P. S. Catalão 等人發表於 2009 年, “An Artificial Neural Network Approach for Short-Term Wind Power Forecasting in Portugal”此篇介紹使用於訓練前饋類神經網路的牛頓、擬牛頓、一步階正割等演算法。[28]
3. S. Salcedo Sanz 等人發表於2008 年, “Short-term Wind Speed Prediction by Hybridizing Global and Mesoscale Forecasting Models with Artificial Neural Networks”, 此篇介紹如何將全球性的低解析度氣象資料經由中尺度數值模式用在風機的風力預測上, 其文獻中所使用的預測方法是前饋倒傳遞類神經網路。[4]
4. C. Pérez-Llera 等人發表於 2002 年, “Local Short-Term Prediction of Wind Speed: A Neural Network Analysis”, 此篇介紹如何前置處理數據並比較找出最佳類神經網路的架構來做短期風力預測。[31]
5. WU Xin 等人發表於 2012 年, “Modified Hill Climbing Method for Active Yaw Control In Wind turbine”此篇文獻說明當風向儀有誤差時, 如何利用爬山法追蹤風能去控制風機迎風轉向。其文獻研究載具風機為 1.5MW。[29]
6. Feifei Bu 等人發表於 2009 年, “Study and Implementation of A Control Algorithm for Wind Turbine Yaw Control System” 此篇介紹如何利用風向感測

器測出風向角度後透過 DSP 及微步進控制，調整輸入至迎風轉向之步階馬達的訊號使風機機艙精確的轉向。此文獻之研究載具風機約為 800kW。 [6]

7. Han-Soon Choi 等人發表於 2010 年，“Active Yaw Control of MW class Wind Turbine”，此篇回顧了大型風力發電機的迎風轉向架構及動態響應，並且模擬風機在不同風力狀態時的扭矩負載。此文獻研究載具為 MW 以上之風機。 [32]
8. Andrew Kusiak 等人, “Power optimization of wind turbines with data mining and evolutionary computation”, 此篇利用資料探勘及進化策略演算法來做風機面向及風葉角度控制，是一篇將預測及迎風轉向結合的論文。此文獻之研究載具為 100kW 之風機[7]。

以上 1、2、3、4 篇文獻均有提到以類神經網路演算法來做風力預測，並且由以上文獻可知用學術界已有豐富關於以類神經網路去做風力預測的研究。而 5、6、7 三篇介紹中、大型風機等主動迎風轉向控制的流程圖或法則，第 8 篇則將預測風力風向及控制風機迎風轉向與風葉角度做整合，而其中預測風力風向及控制迎風轉向的部分是本論文主要探索的地方。第 8 篇論文中，雖對轉向控制有了一番理論，但模擬出的結果顯示轉向過於頻繁，並未考慮其耗能及迎風轉向系統負載之壽命，如圖 1。推斷是因為此研究使用來模擬迎風轉向的研究載具風機發電容量為 100kW，其所需的轉向耗能與本研究關注之 2MW 大型風力發電機相比甚小，故跟本研究相比，其作風力迎風轉向耗能分析的需求性相較之下不高。

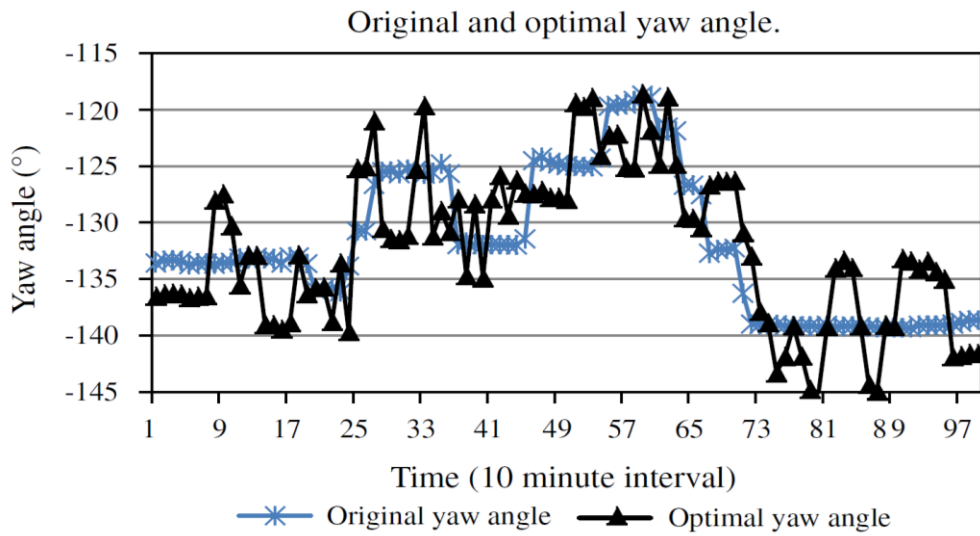


圖 1 文獻六中的轉向指令圖

下圖 2 為彰化濱海風力發電機風場之風速風力資料，由圖 2 (b)可見風力發電廠之地點其風向變化範圍約為 45 度，此風向之變動可造成風機運轉效率變化。

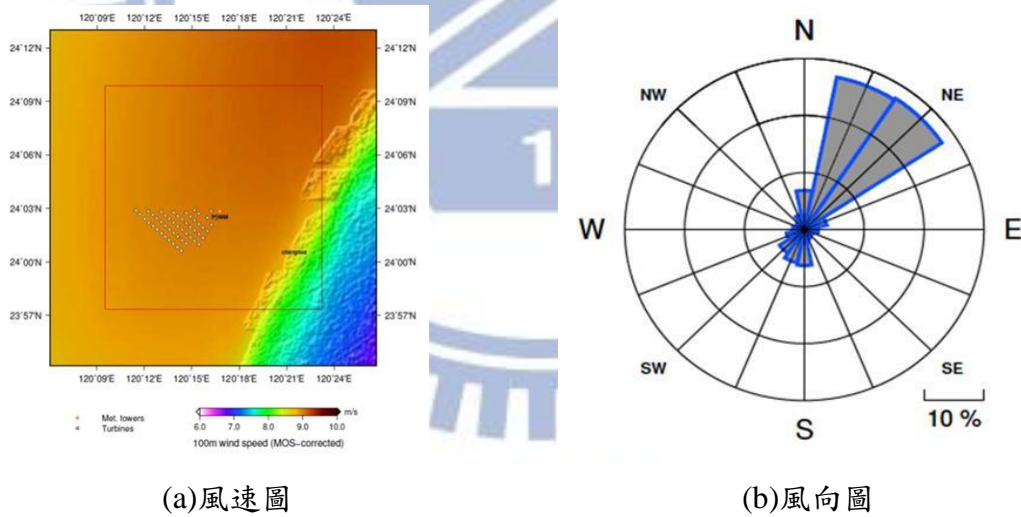


圖 2 彰化濱海風場風速風力資料[24]

未來再生能源利用的趨勢必定是更高效化，故預測風速風向後做迎風轉向控制可成為風力能源開發利用的重點之一。

## 1-4 論文大綱

本論文架構安排如下:

第一章中，說明研究動機與背景，回顧類神經網路及風機迎風轉向控制之文獻。

第二章中，介紹風力發電原理與永磁同步發電機數學模型。

第三章中，介紹預測風力風向之理論。

第四章中，介紹本文所使用之前饋倒傳遞類神經網路架構。

第五章中，說明本研究之類神經網路預測步驟與結果。

第六章中，說明本研究之風機建模及轉向控制步驟與結果。

第七章中，本文結論與未來展望。

第八章為本論文參考之文獻。





## 第二章 風力發電系統架構

### 2-1 風力發電機原理與架構

大型風力發電機通常採用水平軸型式。它由風葉輪、變速齒輪箱、發電機、迎風轉向系統、塔架等部件所組成。依照其發電原理可大略分為感應發電機、勵磁式同步風力發電機、永磁發電機三種。感應發電機與勵磁式發電機都須經由齒輪箱(如圖 6)來加速帶動發電機的轉子；永磁式發電機則不須齒輪箱即可發電(屬直驅式)，所以有重量輕、效率好、可靠度高的優點。圖 3 為彰濱線西區之輪軸式風力發電機的仰望近照，其中置有輪軸傳動箱及感應發電機，故其外型較圖 4 之機艙長。圖 4 同為彰濱線西區之直驅式風力發電機近照，因不須齒輪箱所以機艙也較短，且內部構造與勵磁式風力發電機不同(圖 5、圖 6)。[20][31]



圖 3 輪軸式風力發電機



圖 4 直驅式風力發電機

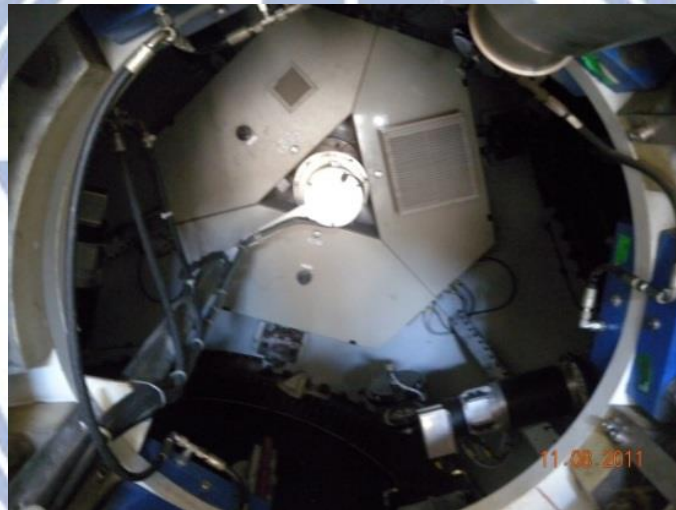


圖 5 永磁直驅式風力發電機內部之三相繞組



圖 6 軸輪式風力機內部的輪軸箱部分

風力發電利用風能推動葉扇產生動能，經轉子(感應型及同步型會再經由齒輪箱)連接至發電機，轉動發電機產生電能。而由空氣動力學可以推導出風的動能經風機轉換出來的機械功率  $P$  可以表示成

$$P = \frac{1}{2} \rho \pi R^2 V^3 C_p \quad (2-1)$$

功率單位為瓦特(Watt)，其他符號意義為

$R$ ：葉片半徑(m)。

$\rho$ ：空氣密度(kg/m<sup>3</sup>)。

$V$ ：風速(m/sec)。

$C_p$  為風機轉換效率參數，為尖端速度比( $\lambda$ )及葉片角度( $\beta$ )之函數。

尖端速度比“ $\lambda$ ”定義為

$$\lambda = \frac{R\omega_r}{V} \quad (2-2)$$

$\omega_r$  為風機旋轉角速度，根據葉片型體及角度設計， $C_p(\lambda)$  為非線性函數，其matlab內建之轉子功率係數方程式為

$$C_p(\lambda, \beta) = 0.5176 \left( \frac{116}{\lambda_i} - 0.4\beta - 5 \right) e^{-\frac{21}{\lambda_i}} + 0.0068\lambda \quad (2-3)$$

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1} \quad (2-4)$$

因此在某一固定之風速下，對照最佳的轉速，存在一個最大功率輸出點。而最大功率追蹤策略(MPPT)即是調整發電機轉速依其風速變化來控制轉子至最佳轉速，以達到捕捉最高風能效率的方法。圖7為在 $\beta=0$ 時， $\lambda$ (x軸)與 $C_p$ 值(y軸)之間的關係圖。圖8為利用(2-3)(2-4)跑出來的 $C_p$ 圖，在固定 $\beta$ (X軸)跟 $\lambda$ (Y軸)下對照z軸得到的值即為 $C_p$ 。

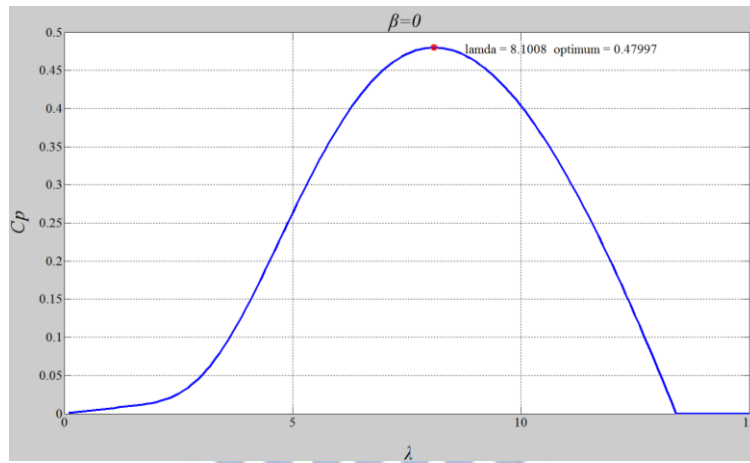


圖 7 二維  $C_p$  圖( $\beta=0$ )

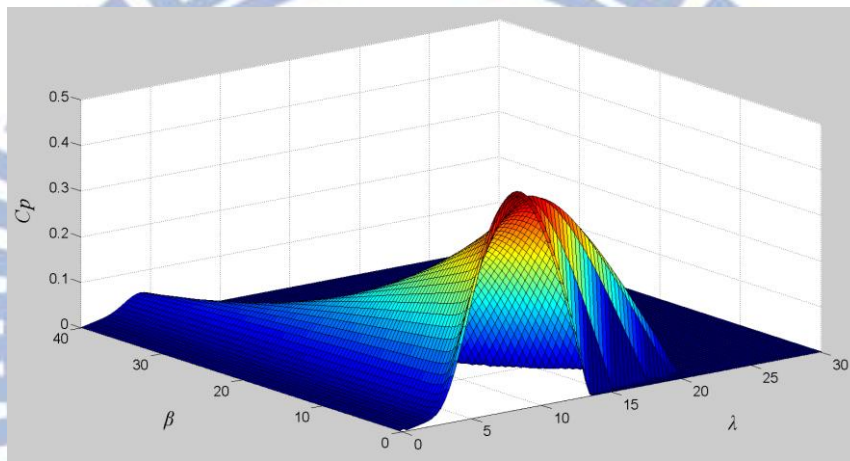


圖 8 三維  $C_p$  圖

## 2-2 風力發電機迎風轉向系統結構

發電機機艙與塔架接觸的部分有迎風轉向馬達來控制風機面向。圖9為直驅式風力發電機內部之迎風轉向控制馬達，包括四組馬達及減速齒輪組，而此部分的控制即是本文研究的主題。

有關大型風力發電機機艙的迎風轉向控制方塊圖簡略示意如圖10[6]。由風機艙上方的風力計(見圖4)得到風速風向資料後，經控制器控制迎風轉向電機使機艙轉向至適當方位。

機艙座在塔架的頂部界面處有外部齒輪滾珠軸承鋼圈，調整驅動裝置以驅動

小齒輪嚙合外部齒輪，確保風力發電機組面向達到系統要求。大部分大型風機皆有迎風轉向解纜控制系統，一旦風力發電機的控制電纜及電力電纜線扭轉2~3圈，控制系統即利用下次低風速的期間來解開扭轉的電纜線；若因風力條件不許可而造成電纜線扭轉多過3圈，則風機將不管風力條件如何，自動停機並解開扭曲，通常大部分大型風力發電機約需半小時解開扭轉的電纜線。一旦完成解開扭轉的電纜線後，風機將自動重新啟動[6]。其控制流程如圖11。

本研究重點之一為風機在正常狀態下風向改變的轉向響應，對於整體詳細的解纜程序，如風機同方向轉三圈以上及與特大強風的保護措施等特殊情況，在本迎風轉向控制的研究當中暫不考慮，僅以自動迎風轉向的部分為主。



圖 9 迎風轉向馬達(共四台)

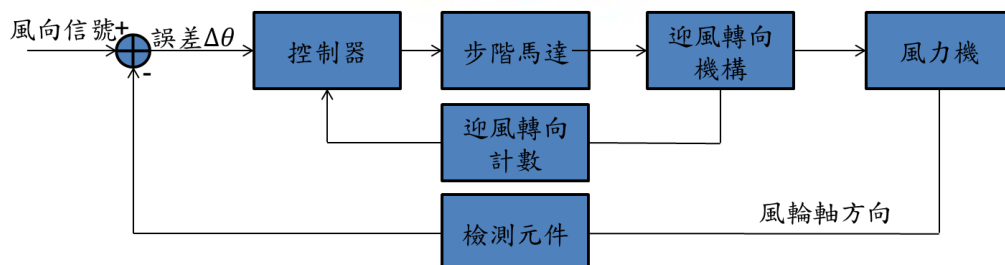


圖 10 風機迎風轉向控制方塊示意圖

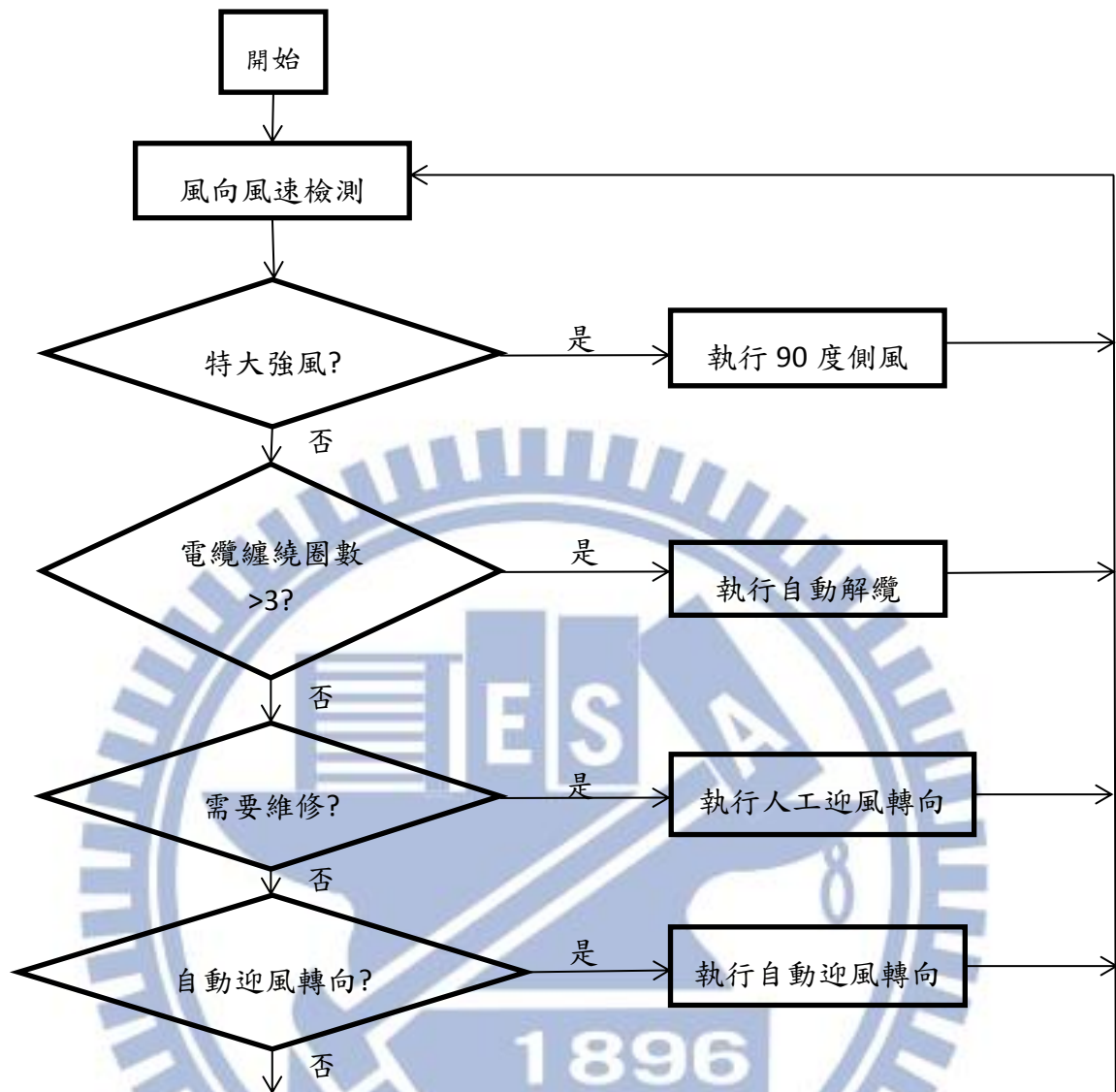


圖 11 迎風轉向系統控制架構

迎風轉向剎車機構由數個液壓控制的迎風轉向剎車盤構成(見圖 12)。迎風轉向系統有兩種壓力，分別提供迎風轉向時的阻尼和迎風轉向結束時的制動力。當迎風轉向不工作(風機不轉向)時剎車片全部抱閘鎖死；當機艙對風所有剎車盤處於半鬆開狀態，設置足夠的阻尼，保持機艙平穩迎風轉向，此時迎風轉向制動器用作阻尼器；當系統自動解纜時，迎風轉向剎車片全部釋放。

液壓控制系統當風機(指約 2MW 大型風機)面向固定時提供約 140~160bar 壓

力，使剎車片緊壓在煞車盤上，以提供足夠的制動力。迎風轉向時則液壓釋放並保持 24bar 的餘壓，以保持一定的阻尼減少風機在迎風轉向過程中的衝擊震動。

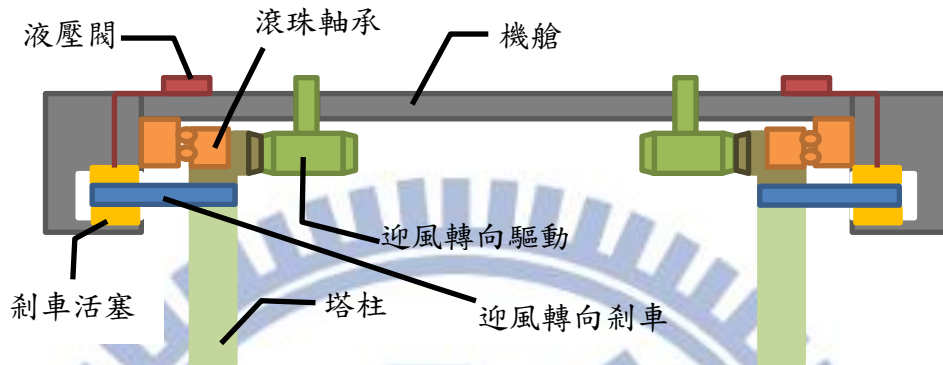


圖 12 風力發電機迎風轉向系統剖面示意圖

風力發電機擷取的風能與風葉轉動截面積成正比、風速三次方成正比，故當風機面向與風向非一致時，其公式為

$$P_{out} = \frac{\rho A}{2} \times V^3 \times \cos \theta \times C_p(\lambda, \beta) \quad (2-5)$$

其中  $\theta$  是風向與風機機艙方向之間的角度，風機運轉中迎風轉向馬達所消耗的能量為  $P_{yaw}$ ，將風力發電電能與迎風轉向馬達消耗電能相減其值最佳化即為我們所需。詳細的推論過程在第六章會有更進一步的探討。

## 第三章 風能預測研究方法

學術文獻中針對長短期風能預測已有許多研究方法。在[2]、[8]中，使用支持向量機(Support Vector Machine regression algorithm)、多層感知器(multilayer perceptron algorithm)、M5P tree algorithm、Reduced Error Pruning tree、bagging tree等方法，並列表比較其預測準確度如表1所示，其中顯示SVMreg法則預測之Mean Relative Error(MRE)最小，為3.514%。

表 1 文獻[8]的誤差比較表

Algorithm	Mean Absolute Error (m/s)	Absolute Error Std (m/s)	Mean Relative Error (%)	Relative Error Std (%)
SVMreg	0.198	0.181	3.514	5.730
MLP	0.266	0.388	4.661	6.739
M5P tree	0.206	0.267	3.953	6.035
REP tree	0.249	0.256	4.576	6.564
Bagging tree	0.202	0.198	3.665	5.723

另從文獻中可知，只要資料來源足夠，類神經網路之預測結果其誤差值與其他方法相比皆較低。以下列舉在學術文獻中所提及的風力預測方法。

### 3-1 支持向量機(Support Vector Machine regression algorithm)

Support Vector Machine regression algorithm(SVM)的原理是將資料在空間中找出最大margin來分類(如圖13)。訓練過程中，藉由調整其網路加權值( $w$ )及偏權值( $b$ )獲得兩類別間最佳分介面。也就是讓圖14的  $\frac{2}{\|w\|}$  最大，因此  $\|w\|$  越小越好。[9]

先用訓練組將各種輸入向量依得到的輸出值加以分類以調整加權值與偏權值，網路訓練完成後將更新的風力資料(輸入向量)送入此網路，經隱藏層神經元的



$K(x, x_s) + b$  的分類處理後，求得未來輸出趨勢  $y$ 。圖14顯示SVM的訓練構造，其調整過程是由類神經網路架構來做加權值與偏權值的訓練，故當分類類別越多、資料越大且非線性時，分類時的計算過程會變得越複雜。

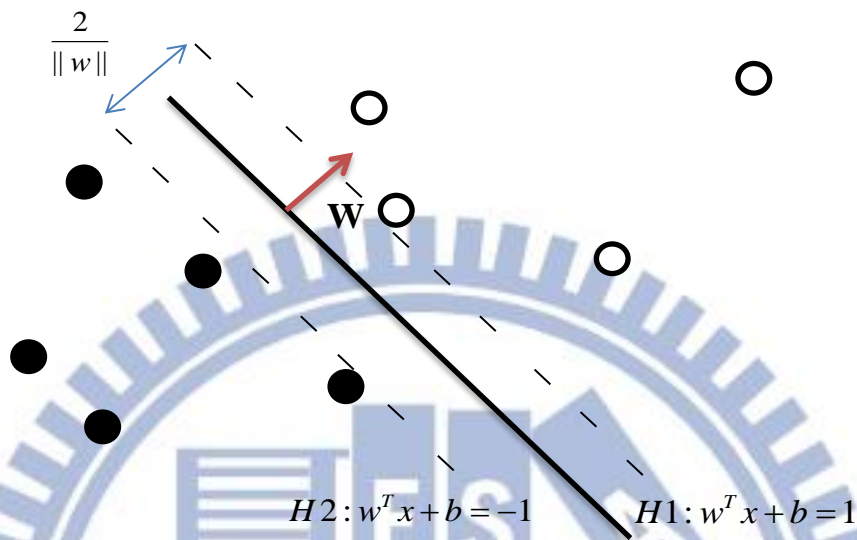


圖 13 SVM 理論的分類概念

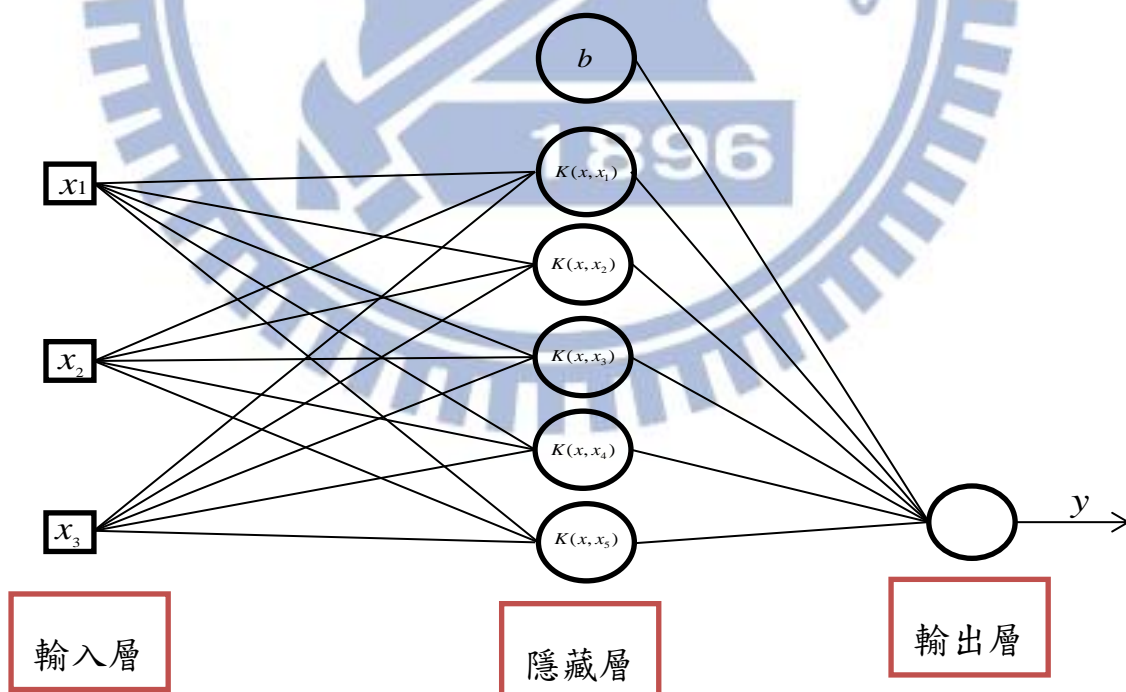


圖 14 SVM 網路構造圖

### 3-2 決策樹(Decision Tree)

決策樹是一種常見的資料探勘方法，可用於數據分析或預測。決策樹的意義代表的是對象屬性與對象值之間的一種映射關係。樹中每個節點表示某個對象，而每個分叉路徑則代表不同的屬性值。一個決策樹模型始於一個根節點，而由根節點發起散開出葉結點，其表現方式可分為圖15的(a)、(b)兩種。每個葉結點對應從根節點到該葉節點所經歷的路徑屬性值交集。此方法之優點是簡單易懂，但當遇到大量複雜的資料時，決策樹分支會變複雜、龐大，且運算時需有專業背景知識及經驗。

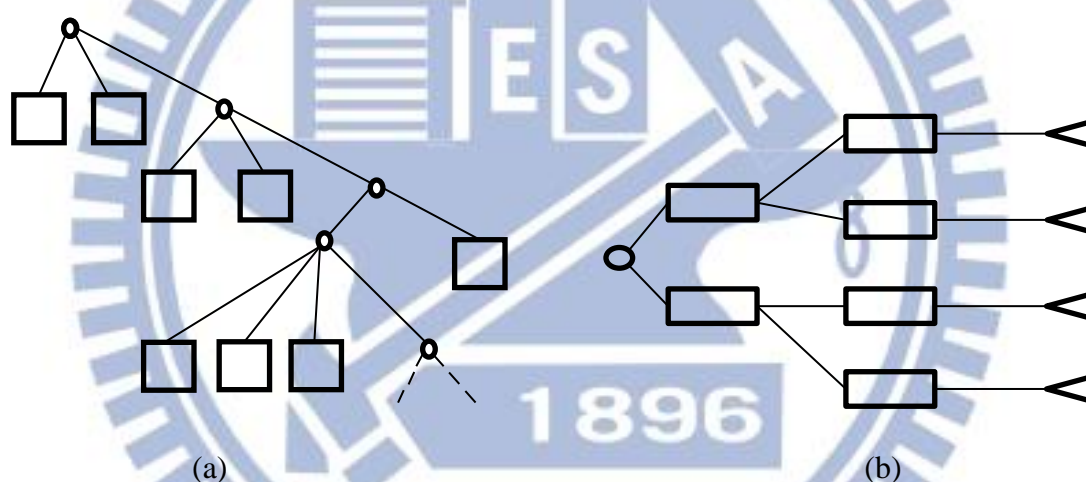


圖 15 決策樹結構

### 3-3 類神經網路(Artificial Neural Network)

類神經網路是模仿生物神經網路的資訊處理系統，是「一種基於腦與神經系統研究，所啟發的資訊處理技術」。神經元即為神經系統的基本單元，有軸突、樹突、突觸、細胞本體等部分。外界不同形式的資訊或能量訊號，經感官器官轉換成電的訊號後，進入神經細胞中，使突觸間內部電位變化並透過樹突傳送至細胞本體，再由軸突傳送到下一個神經元的樹突，成為輸入訊號。突觸對於訊號的影響強度即是生物神經網路處理資訊的地方，而生物神經網路的學習就是在調節

突觸的強度。相對於生物神經網路，人工類神經網路即是由人工神經元連結而成，用經驗模式來調整權重值與偏權值，在對研究對象之程序機構一無所知的情況下，描述實際程序數學模型的一種方式，圖 16 即為一個最簡單的單神經元架構示意圖。[10]

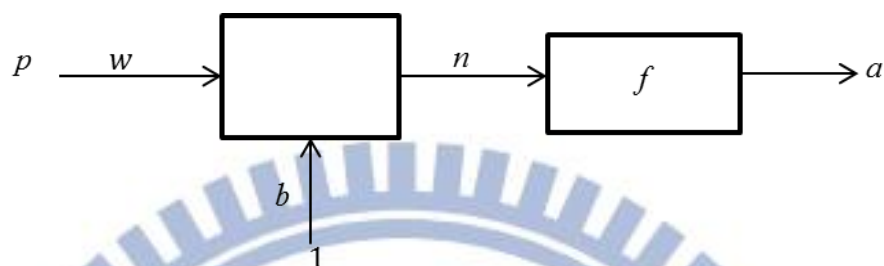
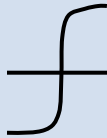


圖 16 一個單輸入有偏權值的單神經元架構

$p$  為輸入、 $a$  為輸出、 $w$  為權重值、 $b$  為偏權值、 $f$  為神經元轉移函數，轉移函數共有十幾種，不同的類神經網路可能會有不同的轉移函數，以下表 2 列出幾個轉移函數  $f$  說明。

表 2 轉移函數及其功能

名稱	輸入/輸出關係	圖示
對數雙彎曲轉移函數 (Sigmoid function)	$f(net) = \frac{1}{1 + e^{-net}}$	
線性轉換函數 (Pure Linear)	$f(net) = net$	
硬限制轉移函數 (Threshold Step)	$\begin{cases} a = 0 & n < 0 \\ a = 1 & n \geq 0 \end{cases}$	
徑向基轉移函數 (Gaussian)	$a = e^{-n^2}$	

<p>正切雙曲線轉移函數 (Hyperbolic tangent function)</p>	$f(net) = \frac{1 - \exp^{-net}}{1 + \exp^{-net}}$	
----------------------------------------------------	----------------------------------------------------	-------------------------------------------------------------------------------------

類神經網路具有三層，輸入層、隱藏層、輸出層。一般而言，在隱藏層中通常使用對數雙彎曲轉移函數或是正切雙曲線轉移函數，因為這樣可以將計算值限定在很小的範圍 (-1~1) 內作計算，以增加學習效率。而輸出層的部分，考量在使用雙彎曲轉移函數的情況下，網路的輸出會限制在很小的範圍內。故輸出層常選擇線性轉移函數(linear)，使網路輸出值可以不限定在任何範圍以逼近我們的目標值。[10]

類神經網路在實用上之運作分訓練過程及模擬過程。訓練過程中，需視各種情況選擇適當的訓練演算法。而以下介紹文獻[8][22][28]中所使用來預測的感知器類神經網路、倒傳遞類神經網路及徑向基類神經網路。

### 3-3-1 多層感知器類神經網路(Multilayer Perceptrons Network)

感知器類神經網路可說是類神經網路中最簡單的，故欲認識類神經網路要先從了解感知器類神經網路架構開始。

感知器屬監督式學習，可分為單層和多層，其不同的地方在於單層只適用線性分類，而多層可做非線性分類，原因在於單層感知器如圖17並無隱藏層，而多層感知器則在輸入層與輸出層間多了隱藏層。感知器類神經網路最適用於用在圖樣分類上，利用硬限轉移函數將輸入向量轉換成不是0就是1的輸出加以分類，在分類過程中，反覆修改權重值 $W$ 和偏權值 $b$ ，以找到最佳的決策邊界  $Wp+b=0$  使兩種不同點的輸入 $p$ 可以被完全分離開來。例如： $Wp+b > 0$ 時，輸出 $a = 1$ ； $Wp+b \leq 0$ 時， $a = 0$ 。而這樣反覆修改權重值和偏權值的過程即為類神經網路的「訓練」。

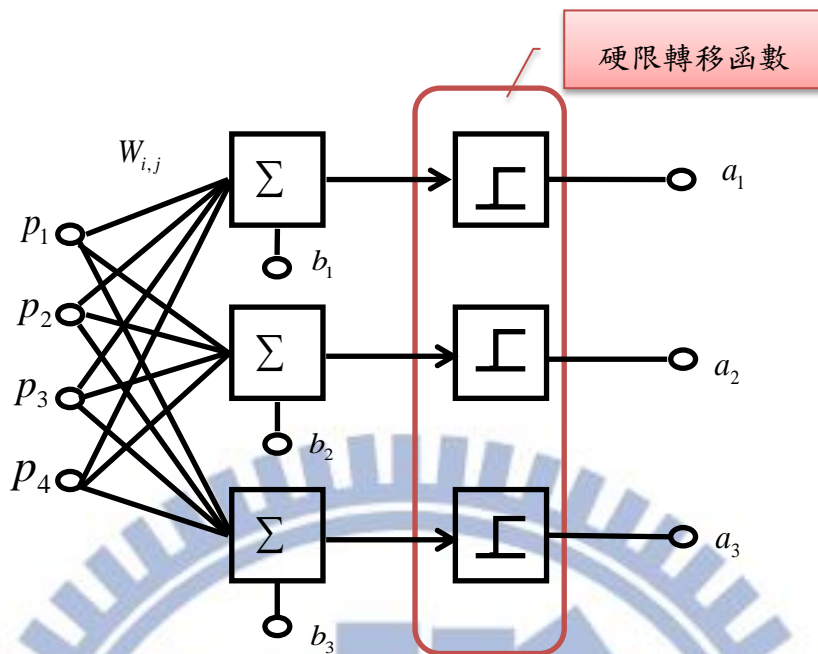


圖 17 單層感知器的網路架構

感知器修改網路權重值與偏權值的規則如下：

$$\Delta \mathbf{w} = (t - a)\mathbf{p}^T = e\mathbf{p}^T \quad (3-1)$$

$$\Delta b = (t - a)(1) = e \quad (3-2)$$

$$\mathbf{W}^{new} = \mathbf{W}^{old} + \Delta \mathbf{W} = \mathbf{W}^{old} + e\mathbf{p}^T \quad (3-3)$$

$$\mathbf{b}^{new} = \mathbf{b}^{old} + \Delta \mathbf{b} = \mathbf{b}^{old} + e \quad (3-4)$$

上述之  $t$  為目標值、 $e$  為誤差值，重複上述(3-1)~(3-4)步驟直到沒有誤差，網路即訓練完成，而此時的偏權值與權重值即構成我們所要的類神經網路，可利用此網路代入未知結果的輸入去做分類。在此要注意的是大部分類神經網路訓練出來的偏權值及權重值並不是唯一解，每一次的訓練結果不絕對一樣，尤其是複雜性較高的網路架構。[10]

### 3-3-2 倒傳遞類神經網路(Back Propagation Network)

倒傳遞演算法 Back Propagation (簡稱 BP)，屬類神經網路中較具代表性的一種。因屬於監督式學習，故需有輸入向量及其相對應的目標向量一起考量來訓練網路，直到網路能將輸出逼近於目標向量如圖 18。經過適當訓練後的倒傳遞網路，當給定此網路從不曾看過的輸入向量時，網路可以給出合理的答案。圖 19 為其結構，隱藏層神經元模型中的轉移函數  $f(\dots)$  常使用對數雙彎曲轉移函數或正切雙彎曲轉移函數，而輸出層則常用線性轉移函數。倒傳遞演算法基於不同標準的最佳化技巧，而有許多種變化的演算法並在本文第四章有詳細的介紹。[10]



圖 18 類神經網路訓練

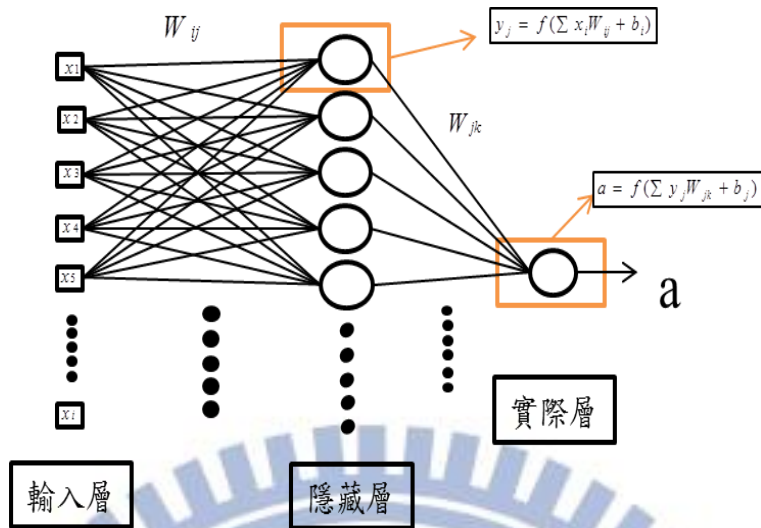


圖 19 單層前饋倒傳遞網路結構

### 3-3-3 徑向基類神經網路(Radial Basis Function Network)

徑向基網路也是屬於監督式學習，和倒傳遞網路一樣皆可近似任何連續非線性函數。他們主要差別在於倒傳遞的隱藏層神經元是使用雙彎曲函數，而徑向基網路的隱藏層神經元轉移函數稱為徑向基函數，當輸入向量越靠近徑向基函數的中心則輸出值越大。

徑向基神經網路將輸入向量 $\mathbf{p}$ 和權重值矩陣 $\mathbf{IW}$ 間的歐基里德距離 $\|\text{dist}\|$ 與偏權向量 $\mathbf{b}$ 的對應元素相乘而得徑向基函數的輸入 $\mathbf{n}$ 。經由徑向基轉移函數(3-5)而得輸出 $\mathbf{a}$ 。

$$a = \text{radbas}(n) = e^{-n^2} \quad (3-5)$$

當 $\mathbf{p}$ 與 $\mathbf{IW}$ 間距離越遠，則輸出  $\mathbf{a}$  越小趨近於0；越大則  $\mathbf{a}$  趨近於1。以此隱藏層神經元架構為出發點，搭配各種不同函數的輸出層，可使網路有函數逼近或是分類等不一樣的功能。圖20即為一徑向基神經元搭配線性輸出層的架構圖，圖中 $\mathbf{LW}$ 為線性輸出層之權重值。

徑向基類神經網路與倒傳遞類神經網路相比，訓練所花費的時間較少(當前

饋類神經網路之隱藏層與輸出層皆為非線性轉移函數時)。但因輸入空間回應區域相對小，故需要更多的隱藏層神經元。[10]

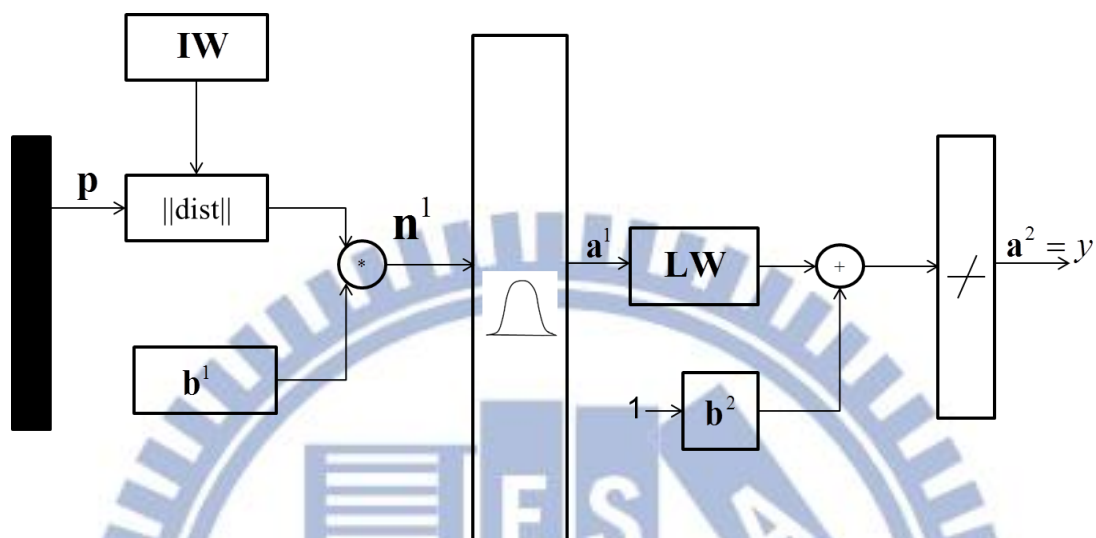


圖 20 徑向基神經元架構圖

圖 20 中之  $\mathbf{a}_i^1$ 、 $\mathbf{a}^2$ 、 $\|\text{dist}\|$  推導如下：

$$\mathbf{a}_i^1 = \text{radbas}(\|\mathbf{IW} - \mathbf{p}\| + \mathbf{b}_i^1) \quad (3-6)$$

$$\mathbf{a}^2 = \text{purelin}(\|\mathbf{LW} - \mathbf{a}^1\| + \mathbf{b}^2) \quad (3-7)$$

$$\|\text{dist}(\mathbf{W}, \mathbf{P})\| = \sqrt{\sum_i^n (\mathbf{w}_i - \mathbf{p}_i)^2} \quad (3-8)$$



## 第四章 前饋倒傳遞類神經網路原理與架構

倒傳遞類神經網路又可分為回饋式與前饋式。回饋式類神經網路其內部至少含有一個回饋迴圈，將神經元的輸出回傳給同層或前一層的神經元作輸入，此結構之類神經網路多用在時序動態訓練。前饋式類神經網路則是單純的向前傳遞架構，沒有迴圈在內部。

前饋倒傳遞類神經網路演算法是目前常見類神經用來預測的方法。最標準的倒傳遞類神經網路演算法是一種梯度下降演算法，網路權重值沿性能函數負梯度方向移動，並以均方誤差為性能指標。倒傳遞類神經網路演算法由梯度下降法為基礎做變化，梯度下降法如下[10]：

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha \frac{\partial \hat{F}}{\partial w_{i,j}^m} \quad (4-1)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial \hat{F}}{\partial b_i^m} \quad (4-2)$$

其中性能函數  $\hat{F}(k) = \mathbf{e}^T(k) \cdot \mathbf{e}(k) = [\mathbf{t}(k) - \mathbf{a}(k)]^T \cdot [\mathbf{t}(k) - \mathbf{a}(k)]$

利用鍊鎖率，令  $n_i^m = \sum_{j=1}^{S^{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m$  並定義靈敏度  $s_i^m \equiv \frac{\partial \hat{F}}{\partial n_i^m}$ ，則

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \cdot \frac{\partial n_i^m}{\partial w_{i,j}^m} = s_i^m \cdot a_j^{m-1} \quad (4-3)$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = \frac{\partial \hat{F}}{\partial n_i^m} \cdot \frac{\partial n_i^m}{\partial b_i^m} = s_i^m \quad (4-4)$$

因此(4-1)、(4-2)可寫成

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha s^m (\mathbf{a}^{m-1})^T \quad (4-5)$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha s^m \quad (4-6)$$

接著計算靈敏度  $s^m$

$$s^m = \begin{bmatrix} \frac{\partial \hat{F}}{\partial n_1^m} \\ \frac{\partial \hat{F}}{\partial n_2^m} \\ \vdots \\ \frac{\partial \hat{F}}{\partial n_s^m} \end{bmatrix} = \frac{\partial \hat{F}}{\partial n^m} = \left( \frac{\partial n^{m+1}}{\partial n^m} \right)^T \cdot \frac{\partial \hat{F}}{\partial n^{m+1}} \quad (4-7)$$

$\frac{\partial n_i^{m+1}}{\partial n_j^m}$  為 Jacobian 矩陣，可推導為：

$$\frac{\partial n_i^{m+1}}{\partial n_j^m} = \frac{\partial \left( \sum_{l=1}^{s^m} w_{i,l}^{m+1} a_l^m + b_i^{m+1} \right)}{\partial n_j^m} = w_{i,j}^{m+1} \frac{\partial a_j^m}{\partial n_j^m} = w_{i,j}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} \Rightarrow \frac{\partial n^{m+1}}{\partial n^m} = \mathbf{W}^{m+1} \mathbf{F}^m(\mathbf{n}^m) \quad (4-8)$$

將(4-8)代入(4-7)可得：

$$s^m = \mathbf{F}^m(\mathbf{n}^m) \cdot (\mathbf{W}^{m+1})^T \cdot \frac{\partial \hat{F}}{\partial n^{m+1}} = \mathbf{F}^m(\mathbf{n}^m) \cdot (\mathbf{W}^{m+1})^T \cdot s^{m+1} \quad (4-9)$$

由(4-9)可知靈敏度是由最後一層推到第一層，故稱為“倒傳遞”演算法。

而最後一層之靈敏度計算如下

$$\begin{aligned} s_i^M &= \frac{\partial \hat{F}}{\partial n_i^M} = \frac{\partial (\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a})}{\partial n_i^M} = -2(t_i - a_i) \frac{\partial a_i}{\partial n_i^M} = -2(t_i - a_i) \cdot f^M(n_j^M) \\ &\Rightarrow s^M = -2\mathbf{F}^M(\mathbf{n}^M)(\mathbf{t} - \mathbf{a}) \end{aligned} \quad (4-10)$$

前饋倒傳遞類神經網路的隱藏層通常使用雙彎曲轉移函數而輸出層則使用線性轉移函數，如此一來可允許網路正負無限的輸入及學習到輸入向量與輸出向量的非線性關係，又可使輸出能有-1到+1以外的值。只要隱藏層神經元足夠，前饋倒傳遞網路就能逼近任何非線性甚至具有有限個不連續的函數。[10]

接下來的小節由最簡單的梯度下降演算法至複雜的 Levenberg-Marquardt 演算法一步一步慢慢帶入說明，在第五章即利用此章所提及之各類類神經方法訓練

風力資料，將其結果列表比較以選取最適當用以預測本文資料的方法。

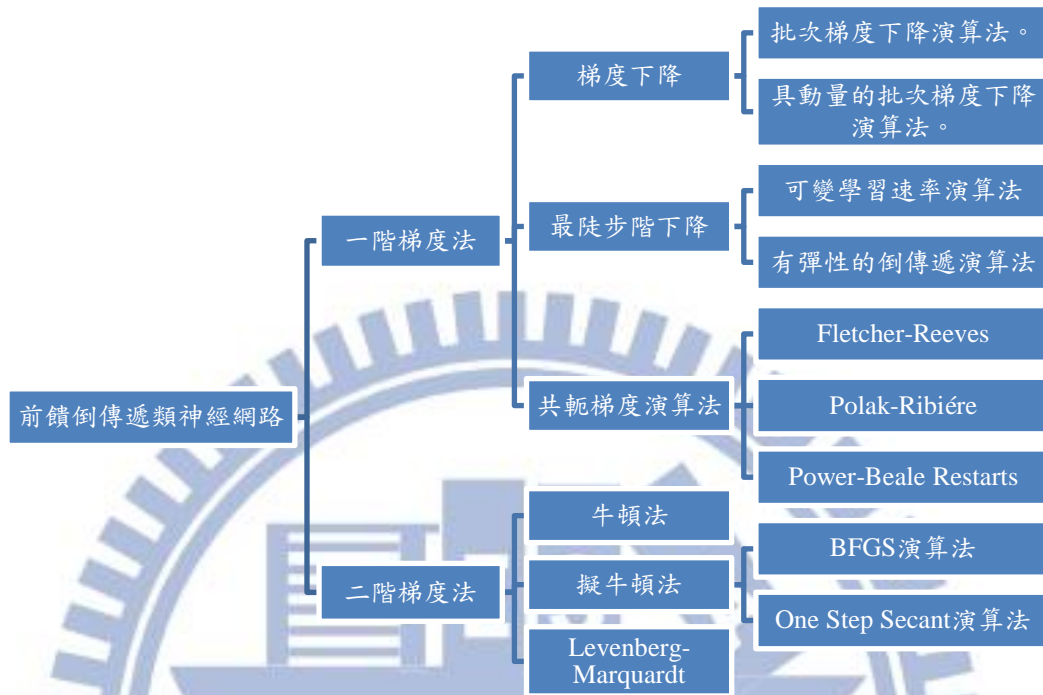


圖 21 前饋倒傳遞類神經網路演算法分類

#### 4-1 批次梯度下降倒傳遞演算法(Steepest Decent back propagation)

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k \quad (4-11)$$

$\mathbf{x}_k$  為權重值向量或偏權值向量； $\mathbf{g}_k$  是目前的梯度； $\alpha_k$  是學習速率；變數右下角小標  $k$  為疊代次數。(4-11)是梯度下降中最簡單的形式，由此疊代式衍生出之後較複雜的方法。

梯度下降法之權重值與偏權值的更新是在性能函數的負梯度方向中，以學習速率乘以負梯度來決定。過大的學習速率會導致網路的學習不穩定，但速率過小卻會使學習緩慢，並陷入局部最小值。根據經驗，學習速率通常介於 0.25~0.75。

#### 4-2 具動量的批次梯度下降演算法(Momentum back propagation)

$$\Delta W^m_k = \gamma \Delta W^m_{k-1} - \alpha(1-\gamma) \frac{\hat{\partial F}}{\partial W^m_{k-1}} \quad (4-12)$$

$$\Delta b^m_k = \gamma \Delta b^m_{k-1} - \alpha(1-\gamma) \frac{\partial \hat{F}}{\partial b^m_{k-1}} \quad (4-13)$$

其中， $W$ 為權重值、 $b$ 為偏權值、 $\gamma$ 是動量係數、 $\alpha$ 是學習速率、變數右下角小標  $k$  為疊代次數、 $\hat{F}$  為性能函數。[32]

此演算法利用動量係數使訓練所提出之新權重值改變量與前一次權重值改變量以一定比例相加得權重值改變量，此方法優於第一個方法的原因在於動量可降低網路對誤差曲面局部梯度變化的敏感度，有效抑制網路陷入局部極小值。

### 4-3 可變學習速率倒傳遞演算法(Variable Learning Rate back propagation)

可變學習速率的精神在於讓學習速率在訓練過程中變化，以改善之前標準最陡坡降演算法學習速率設太高會震盪不穩定，設太小費時且可能不會收斂的缺點，使接近性能曲面最低點的過程中可以快速收斂且保持穩定。

首先，計算初始網路輸出和誤差。在每一次訓練循環中，使用目前的學習速率來計算新的權重值和偏權值。接著，計算新的輸出和誤差。當新的誤差超越舊的誤差 1.04 倍，則如(4-14) 新的權重值和偏權值排除不用，而將學習速率減少為原來的 0.7 倍。如果新的誤差值遠少於舊的誤差，則如(4-15)增加學習速率為原來的 1.05 倍，這個過程要增加學習速率，而且不讓訓練網路時產生過大的誤差。

$$E_{new} > 1.04 \times E_{old} \Rightarrow \eta = 0.7 \quad (4-14)$$

$$E_{new} \ll E_{old} \Rightarrow \eta = 1.05 \quad (4-15)$$

$E$ : 網路輸出誤差值、 $\eta$ : 學習速率  $\alpha$  的增益值。

#### 4-4 有彈性的倒傳遞演算法(Resilient back propagation)

一般多層網路的隱藏層常使用雙彎曲轉移函數，雖然可使無限正負的輸入濃縮成-1到1的值，但當輸入極大時，會使輸出函數的斜率趨近於零( $f'(n) \rightarrow 0$ )，導致使用最陡坡降法所得到的權重值與偏權值的改變量太小造成在離最佳值很遠的地方收斂非常慢幾乎停擺。因此有彈性的倒傳遞演算法目的在除去梯度值太小造成調整參數過程幾乎停擺的影響，只使用梯度的符號來決定權重值的更新。其方法架構為：

1. 兩次權重值微分具有相同符號，權重值遞增。
2. 兩次權重值微分符號改變，權重值遞減。
3. 幾次疊代權重值在相同方向上改變，權重值變化量增加。

此演算法還有一個好處，就是他不需要儲存權重值與偏權值的更新值只要儲存梯度符號，因此對記憶體需求量不大。

#### 4-5 共軛梯度演算法(Conjugate Gradient back propagation)

前面所提的倒傳遞演算法是在性能函數下降最快的方向中調整權重，雖然性能函數能快速降低但未必能有最快的收斂。在共軛梯度演算法中，其權重值沿著共軛方向改變，反而可以產生較快的收斂。不同於之前最陡梯度法之權重值更新依學習速率來決定，共軛梯度演算法沿共軛梯度方向搜尋以決定步階大小，使性能函數可沿此方向來極小化。

第一次的疊代上都是由搜尋最陡下降方向開始如(4-16)、(4-17)，沿著目前搜尋方向決定移動的最佳距離如(4-18)。接著，決定下一次的搜尋方向如(4-19)，使此方向與先前的搜尋方向是共軛的。依照不同的共軛梯度參數計算方式，而衍生出各種不同版本的共軛梯度演算法。表3列出各種不同版本的共軛梯度法。[32]

$$\mathbf{g}_k \equiv \nabla F(\mathbf{x})_{\mathbf{x}=\mathbf{x}_k} \quad (4-16)$$

$$\mathbf{p}_0 = -\mathbf{g}_0 \quad (4-17)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad (4-18)$$

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1} \quad (4-19)$$

$\mathbf{g}$ 為梯度、 $\mathbf{x}_k$ 是在第k次疊代的向量參數(包括權重值及偏權值)、 $\alpha_k$ 是學習速率、 $\beta_k$ 是共軛梯度參數。

Fletcher-Reeves更新法	Polak-Ribiere更新法	Powell-Beale Restarts 更新法
$\beta_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}}$	$\beta_k = \frac{\Delta \mathbf{g}_{k-1}^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}}$	$ \mathbf{g}_{k-1}^T \mathbf{g}_k  \geq 0.2 \ \mathbf{g}_k\ ^2$ 滿足上式，搜尋方向即重新設定為負梯度
記憶體儲存需求量大	記憶體儲存需求量大	記憶體儲存需求量大 性能：良好

表 3 不同版本之共軛梯度法比較

#### 4-6 牛頓法(Newton's method)

牛頓法是相對於共軛梯度法適用快速最佳化的另一種選擇。其優點是相較於梯度下降的線性收斂，它有二次方的收斂速度。其基本步驟如(4-20)~(4-25)所示 [27]。

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x} \quad (4-20)$$

$$\Delta \mathbf{x} = -[\nabla^2 \hat{F}(\mathbf{x})]^{-1} \nabla \hat{F} \quad (4-21)$$

$$\nabla \hat{F}(\mathbf{x}) = J^T(\mathbf{x})\mathbf{e}(\mathbf{x}) \quad (4-22)$$

$$\nabla^2 \hat{F}(\mathbf{x}) = J^T(\mathbf{x})J(\mathbf{x}) + S(\mathbf{x}) \quad (4-23)$$

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_1(\mathbf{x})}{\partial x_1} & \frac{\partial e_1(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial e_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial e_2(\mathbf{x})}{\partial x_1} & \frac{\partial e_2(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial e_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_n(\mathbf{x})}{\partial x_1} & \frac{\partial e_n(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial e_n(\mathbf{x})}{\partial x_n} \end{bmatrix} \quad (4-24)$$

$$S(\mathbf{x}) = \sum_{i=1}^N e_i(\mathbf{x}) \nabla^2 e_i(\mathbf{x}) \quad (4-25)$$

$\nabla^2 \hat{F}(\mathbf{x})$  為性能指數即目前向量參數 (包括權重值和偏權值) 的二次微分、 $\nabla \hat{F}(\mathbf{x})$

是梯度函數、 $e(\mathbf{x})$  為誤差、 $\mathbf{x}_k$  是在第 k 次疊代的向量參數 (包括權重值及偏權值)。

在高斯-牛頓法中，設  $S(\mathbf{x}) \approx 0$ ，因此最終(4-21)變成

$$\Delta \mathbf{x} = [\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x})]^{-1} \mathbf{J}^T(\mathbf{x})e(\mathbf{x}) \quad (4-26)$$

(4-26) 中  $\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x})$  為 Hessian 矩陣。高斯-牛頓法收斂的比共軛梯度法快，但計算 Hessian 是複雜耗時的。因此有了擬牛頓法 (Quasi-Newton Method) 及正割法 (Secant Method) 的近似化 Hessian 替代。

#### 4-7 擬牛頓法 BFGS (Quasi-Newton algorithms, BFGS)

將 Hessian 矩陣近似如對稱矩陣  $\mathbf{B}_{k+1}$ ：

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{\mathbf{B}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{B}_k}{\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k} \quad (4-27)$$

其中  $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ ， $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ ， $\mathbf{g}_k \equiv \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$ ， $\mathbf{x}_k$  是在第 k 次疊代的向量參數 (包括權重值及偏權值)。

利用所得之近似矩陣  $\mathbf{B}_k$  以及  $\mathbf{B}_0 = \mathbf{I}$  的初始設定，一開始相當於最陡梯度下降法，經數次疊代後成 Hessian 近似矩陣，而 Hessian 的反矩陣則可近似成

$$\mathbf{B}_{k+1}^{-1} = \mathbf{B}_k^{-1} + \frac{(\mathbf{s}_k^T \mathbf{y}_k + \mathbf{y}_k^T \mathbf{B}_k^{-1} \mathbf{y}_k)(\mathbf{s}_k \mathbf{s}_k^T)}{(\mathbf{s}_k^T \mathbf{y}_k)^2} - \frac{\mathbf{B}_k^{-1} \mathbf{y}_k \mathbf{s}_k^T + \mathbf{s}_k \mathbf{y}_k^T \mathbf{B}_k^{-1}}{\mathbf{s}_k^T \mathbf{y}_k} \quad (4-28)$$

最後可得

$$\Delta \mathbf{x} = -(\mathbf{B}_k^{-1}) \nabla F(\mathbf{x})^T \Big|_{\mathbf{x}=\mathbf{x}_k} \quad (4-29)$$

經由此法可以避免計算複雜的 Hessian 矩陣，但與共軛梯度演算法相比，還是需要較多的計算和記憶儲存量。

#### 4-8 一步階正割演算法(One Step Secant)

一步階正割法是共軛梯度法與擬牛頓法之間折衷的演算法，此演算法使用割線近似，相較於擬牛頓法的二次方計算  $O(N^2)$ ，一步階正割法只需  $O(N)$  的計算。

計算方式如下：

$$\Delta \mathbf{x} = -\mathbf{g}_{k+1} + B_k \mathbf{y}_k + C_k \mathbf{s}_k \quad (4-30)$$

$$B_k = \frac{\mathbf{s}_k^T \mathbf{g}_{k+1}}{\mathbf{s}_k^T \mathbf{y}_k} \quad (4-31)$$

$$C_k = -\left(1 + \frac{\mathbf{y}_k^T \mathbf{y}_k}{\mathbf{s}_k^T \mathbf{y}_k}\right) \frac{\mathbf{s}_k^T \mathbf{g}_{k+1}}{\mathbf{s}_k^T \mathbf{y}_k} + \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\mathbf{s}_k^T \mathbf{y}_k} \quad (4-32)$$

其中  $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ ， $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ ， $\mathbf{g}_k \equiv \nabla F(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_k}$ ， $\mathbf{x}_k$  是在第  $k$  次疊代的向量參數(包括權重值及偏權值)。

此好處是不用計算逆矩陣就可計算出新的搜尋方向。因此此演算法所需的計算和記憶儲存量小於擬牛頓法而大於共軛梯度法。

#### 4-9 Levenberg-Marquardt 演算法

Levenberg-Marquardt method 中心思想就是一個混合型方法，在最陡梯度法和牛頓法中進行切換，此法在訓練權重值達幾百個的網路時是最快速的方法。



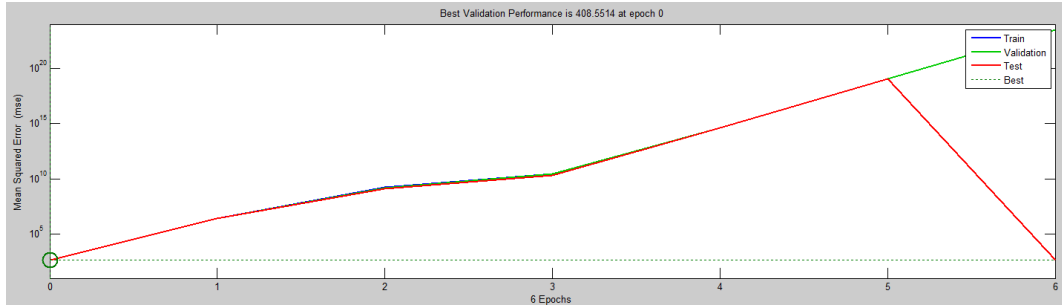
Levenberg-Marquardt method 與擬牛頓法相同不用去計算 Hessian 矩陣，其原理是將 Hessian 逼近成  $\mathbf{H} = \mathbf{J}^T \mathbf{J}$ ， $\mathbf{J}$  是 Jacobian 矩陣如(4-24)，包含網路誤差對於權重值與偏權值的一階微分。將 Jacobian 矩陣代入梯度計算得  $\mathbf{g} = \mathbf{J}^T \mathbf{e}$ ， $\mathbf{e}$  為網路輸出之誤差。利用此逼近方法將牛頓法改寫得(4-33)。

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}]^{-1} \mathbf{J}^T \mathbf{e} \quad (4-33)$$

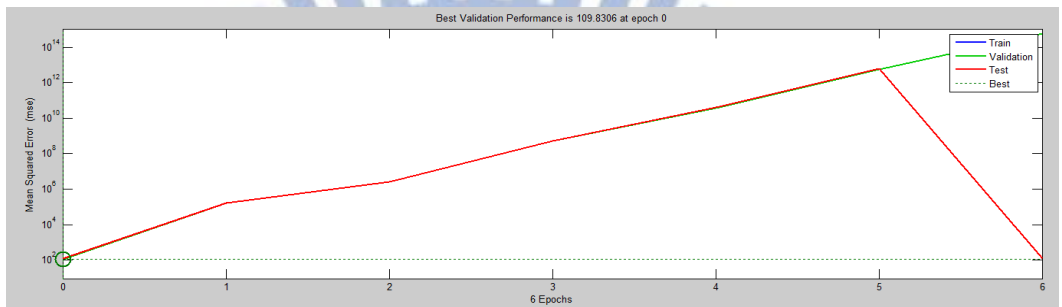
$\mathbf{x}_k$  是在第 k 次疊代的向量參數(包括權重值及偏權值)。上式(4-33)中之  $\mu$  越大則越趨近梯度下降法，越小則越近似牛頓法。每一次的疊代若性能函數有效下降則  $\mu$  降低，因此此方法最後會成近似化的牛頓法，有極小值附近較快且較準確的特性。[27]

#### 4-10 前饋倒傳遞類神經網路收斂情形比較

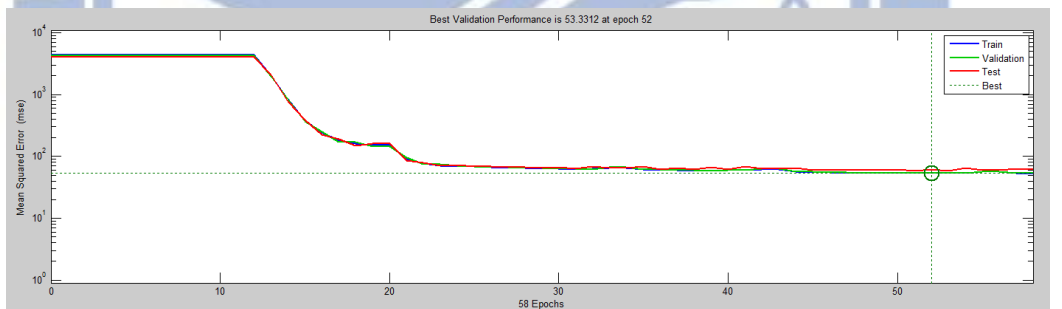
為防止會有過度訓練(over-fitting)的情形，故 Matlab nntool 在訓練時，會將訓練資料事先分成三部分，第一部分是訓練集 Train，就是專用來訓練網路的資料。第二部分是驗證集 Validation，驗證集不參與偏權值與權重值的訓練，而是每當網路訓練一輪時，將驗證集的輸入輸出代入計算誤差值。當訓練集的誤差值下降而驗證集誤差卻上升時，代表網路訓練出現 over-fitting 的情狀，表示此訓練已進入只有訓練集適用的情形，此時則停止訓練並以驗證集誤差最低點的偏權值與權重值來做為訓練結果。第三部分為測試集 Test，測試集同樣不參與偏權值與權重值的訓練。其用處在於判斷訓練集與驗證集切割的狀況是否合理，當訓練停止時，如果測試集誤差最低點與驗證集誤差最低點的疊代次數相差甚遠，則表示此數據切割得不好。以圖 22 示列出此章節所提出之演算法在訓練網路時的各種收斂情形。



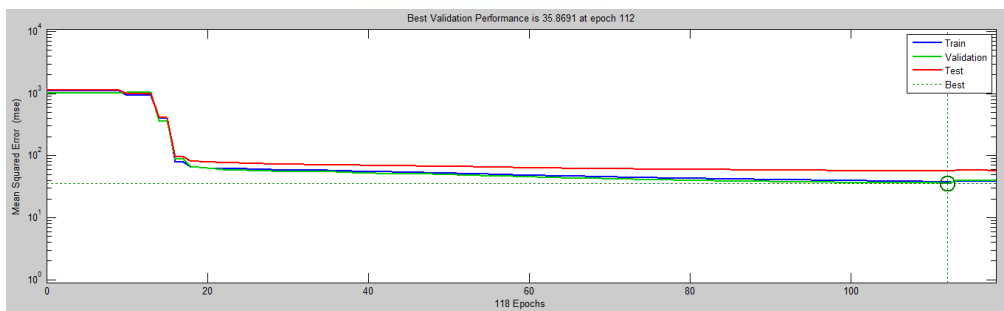
(a) 梯度下降倒傳遞



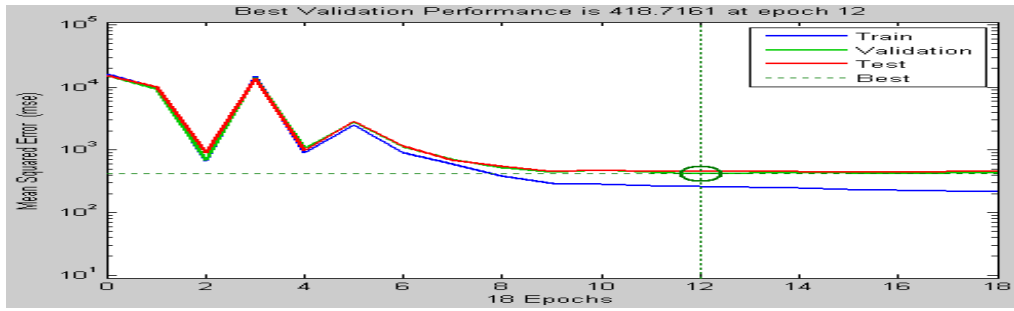
(b) 具動量的梯度下降倒傳遞



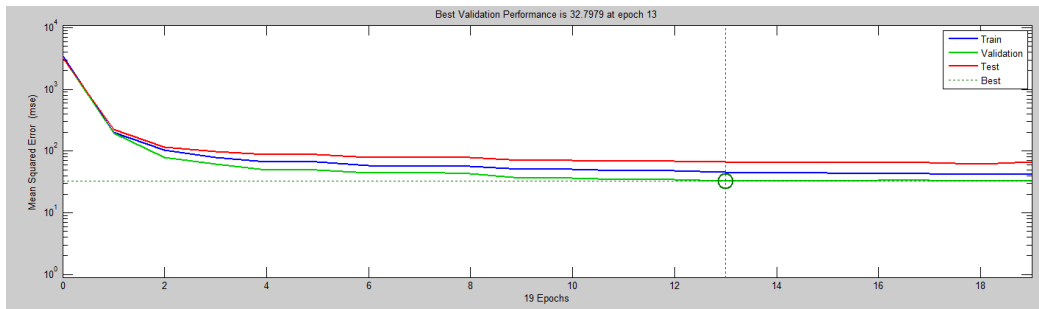
(c) 具適應性學習速率的梯度下降倒傳遞



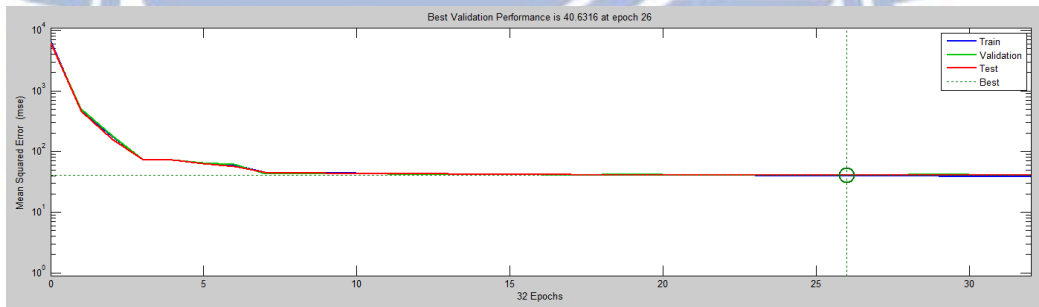
(d) 具動量和適應性學習速率的梯度下降倒傳遞



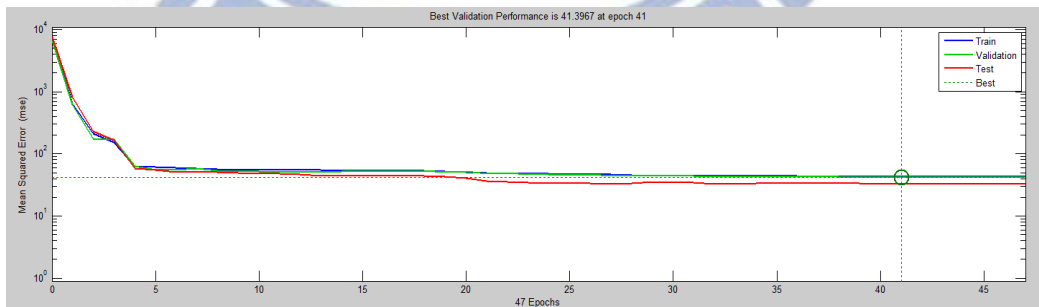
(e) 有彈性的倒傳遞演算法



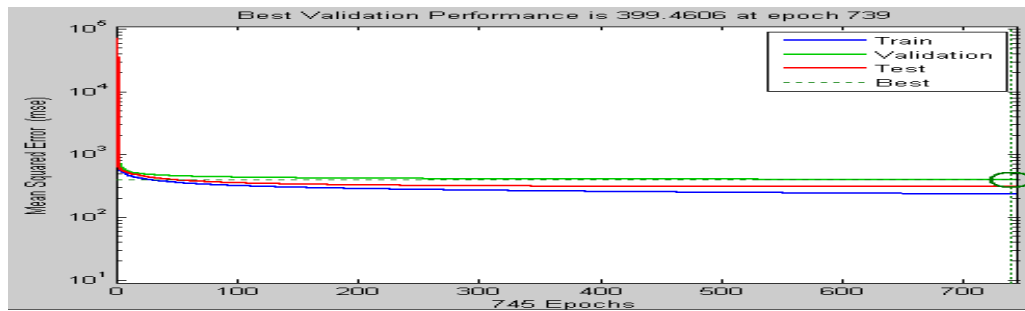
(f) Powell-Beale 共軛梯度倒傳遞



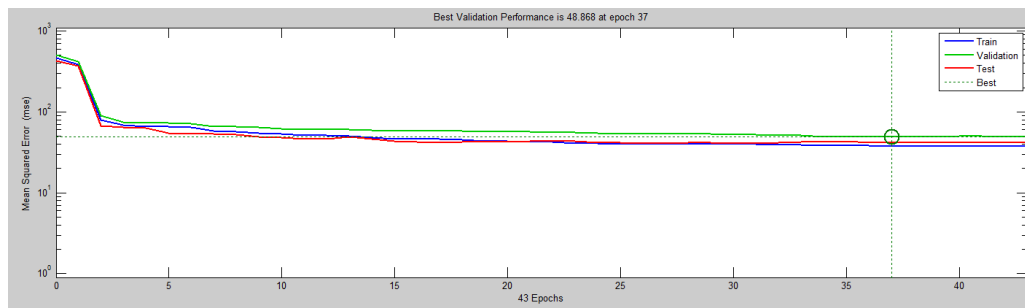
(g) Fletcher-Powell 共軛梯度倒傳遞



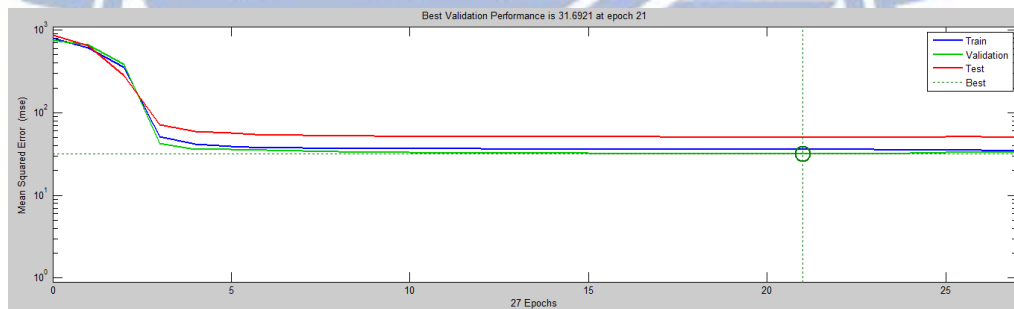
(h) Polak-Ribiere 共軛梯度倒傳遞



(i) BFGS 擬牛頓倒傳遞



(j) Ols One Step Secant



(k) Levenberg-Margardt

圖 22 各類倒傳遞前饋類神經網路演算法的收斂情況

圖 22 中所列的圖：Y 軸為性能函數(誤差)；X 軸為疊代次數；藍線為訓練誤差曲線；綠線為驗證誤差曲線；紅線為測試誤差曲線；與虛線交點的 x 軸值為網路訓練最佳疊代次數。從上列圖中曲線可見各種前饋倒傳遞訓練演算法的特色，從梯度下降、彈性倒傳遞、共軛梯度、擬牛頓最後到 LM 演算法，每個誤差曲線的走勢皆因其偏權值與權重值尋找的方式不同而有所不同。

# 第五章 類神經模擬

## 5-1 類神經研究方法與步驟

- 第一階段確定研究對象之輸入與輸出特徵後，整理分析研究樣本並將其依 Matlab nntool 之資料形式要求做前置處理，如圖 23。
- 第二階段建構初步網路模型並調整訓練範例、網路結構以及相關參數來尋找適當的訓練方法。將數據分為訓練集、驗證集與測試集，訓練集用以調整網路權重值；驗證集用以檢測訓練是否過度，一旦誤差到達最低值而開始連續遞增時就停止訓練；測試集功能則是與驗證集誤差比對，如果誤差差太多則表示數據切分得不好。
- 第三階段應用訓練出來的模型來進行預測。以歷史紀錄作為預測對象，並計算出預測值對於實際值的平均誤差。
- 重複以上步驟並將平均誤差列表，從中找出最佳的網路架構。

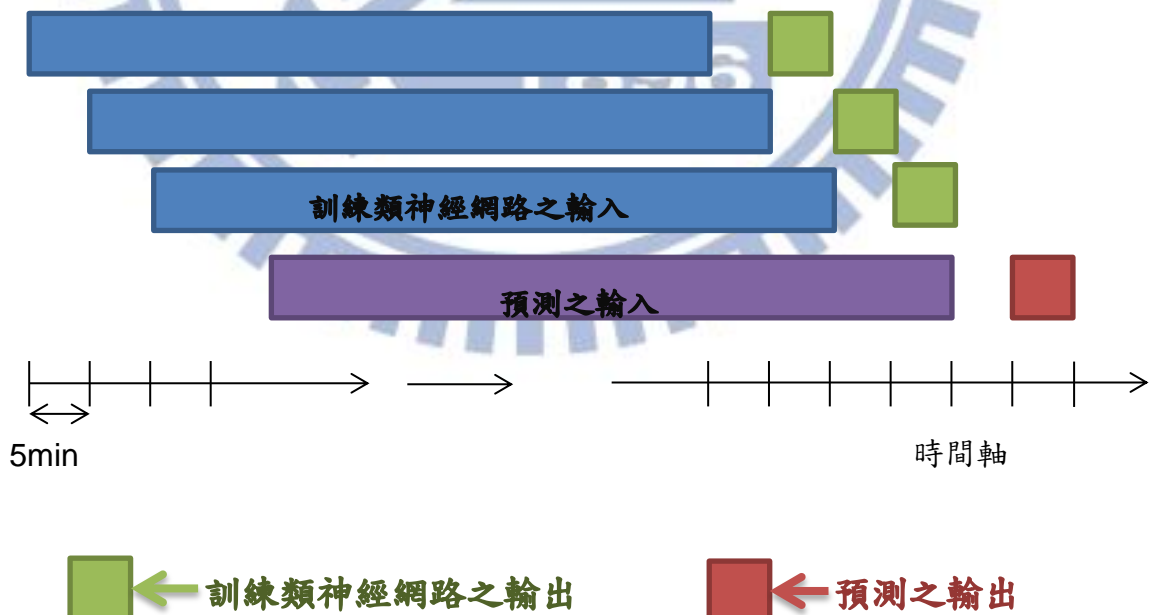


圖 23 類神經網路輸入輸出資料形式

## 5-2 類神經網路短期風能預測

本研究利用前述之各種倒傳遞前饋類神經網路在以風力資料為輸入並使用 Matlab nntool 訓練後整理如表 4、表 5，表中所使用的架構包括六個隱藏層神經元、30 個輸入端，而輸出為未來十分鐘之風力風向。結果顯示風向最低平均相對誤差 MRE 為 Fletcher-Powell 共軛梯度倒傳遞的 0.0172，風速最低 MRE 為 Fletcher-Powell 共軛梯度倒傳遞的 0.0596，但 Fletcher-Powell 共軛梯度倒傳遞的圈點式訓練時間較久。而研判表 5 風速誤差較表 3 高的原因是本研究的大型風場資料來自美國衝浪網站[25]，而文獻中的風力資料來自穩定度較高且風速較快的大型風機風場，資料的動盪程度與預測準確度有絕對的關係。經由表 4、表 5 的比較後，本文以 MRE 為主，取風向與風速之 MRE 較低者，並兼顧取訓練時間之考量，最終決定以 Levenberg-Marquardt 為本研究之訓練演算法。故本研究接下來的預測控制將統一使用 Levenberg-Marquardt method(簡稱 ML)。

表 4 各種類神經網路訓練函數預測風向誤差比較表

訓練函數	風向 MAE (註 1)	風向 SD of MAE (註 2)	風向 MRE (註 3)	風向 SD of MRE (註 4)	訓練 時間 (註 5)
梯度下降倒傳遞	76.2359	9.7403	0.2541	76.6036	1 秒
具適應性學習速率的梯度下降倒傳遞	6.2265	4.8430	0.0206	7.8719	2 秒
具動量的梯度下降倒傳遞	124.8931	11.9576	0.4149	125.0512	1 秒
具動量和適應性學習速率的梯度下降倒傳遞	5.4932	4.4448	0.0183	7.0520	5 秒
彈性的倒傳遞	5.3089	4.1001	0.0177	6.6939	1 秒
BFGS 擬牛頓倒傳遞	5.3917	4.2998	0.0179	6.8823	32 秒
Powell-Beale 共軛梯度倒傳遞	5.5626	4.3441	0.0185	7.0433	3 秒

Fletcher-Powell 共軛梯度倒傳遞	5.1487	4.1421	0.0172	6.5947	8 秒
Polak-Ribiere 共軛梯度倒傳遞	5.4990	4.1713	0.0183	6.8875	3 秒
One Step Secant	5.4859	4.0928	0.0182	6.8298	2 秒
<b>Levenberg-Marguardt</b>	5.6207	4.5722	<b>0.0177</b>	7.2310	2 秒

表 5 各種類神經網路訓練函數預測風速誤差比較表

訓練函數	風速	風速	風速	風速	訓練時間 (註 2)
	MAE (註 1)	SD of MAE (註 2)	MRE (註 3)	SD of MRE (註 4)	
梯度下降倒傳遞	4.1690	2.0125	0.3361	4.2391	1 秒
具適應性學習速率的梯度下降倒傳遞	0.8272	0.6569	0.0630	1.0077	2 秒
具動量的梯度下降倒傳遞	11.1418	1.3765	0.8645	10.3691	1 秒
具動量和適應性學習速率的梯度下降倒傳遞	22.6381	10.6925	0.0746	24.9689	5 秒
彈性的倒傳遞	0.7742	0.6108	0.0598	0.9399	1 秒
BFGS 擬牛頓倒傳遞	0.8264	0.6218	0.0636	0.9841	32 秒
Powell-Beale 共軛梯度倒傳遞	0.7799	0.6001	0.0605	0.9368	3 秒
Fletcher-Powell 共軛梯度倒傳遞	0.7716	0.6011	0.0596	0.9318	8 秒
Polak-Ribiere 共軛梯度倒傳遞	0.7904	0.6233	0.0612	0.9622	3 秒
One Step Secant	0.8317	0.6703	0.0643	1.0189	2 秒
<b>Levenberg-Marguardt</b>	0.7792	0.6601	<b>0.0602</b>	0.9761	2 秒

註 1：MAE 的計算方式如下

$$MAE = \frac{1}{n} \sum_{i=1}^n (|\hat{y}_i - y_i|) \quad (5-1)$$

註 2：SD of MAE 的計算方式如下

$$\text{SD of MAE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left( \left| \hat{y}_i - y_i \right| - \frac{1}{n} \sum_{i=1}^n \left| \hat{y}_i - y_i \right| \right)^2} \quad (5-2)$$

註 3：MRE 的計算方式如下

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^n \left( \left| \frac{\hat{y}_i - y_i}{y_i} \right| \right) \cdot 100\% \quad (5-3)$$

註 4：SD of MRE 的計算方式如下

$$\text{SD of MRE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left( \left| \frac{\hat{y}_i - y_i}{y_i} \right| - \frac{1}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \right)^2} \quad (5-4)$$

註 5：訓練停止的狀況如下

1. 當訓練誤差低於目標值以下，即停止訓練。
2. 當疊代次數到設定之上限點即停止訓練。
3. 當驗證數據集的誤差遞增次數達上限時，即停止訓練並取驗證誤差極小值之權重值與偏權值。

在套用大型風力場的風力資料之前，本研究先利用龍山國小氣象站[11]所發布每五分鐘更新一次的資料來測試 Matlab 類神經網路預測的正確率。

圖 24 與圖 25 為風速風向與實際值之比較圖，利用 matlab 的類神經網路 toolbox 使用一層隱藏層內建十二神經元並以 Levenberg-Marquardt 演算法訓練，其中中間層使用雙曲活化函數，輸出層是線性活化函數。

由兩張圖可見 matlab 的模擬預測與實際值走向相去不遠。

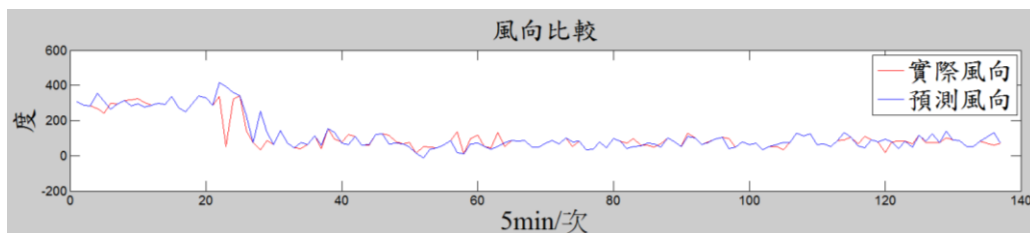


圖 24 龍山國小風向實際值與預測值比較圖



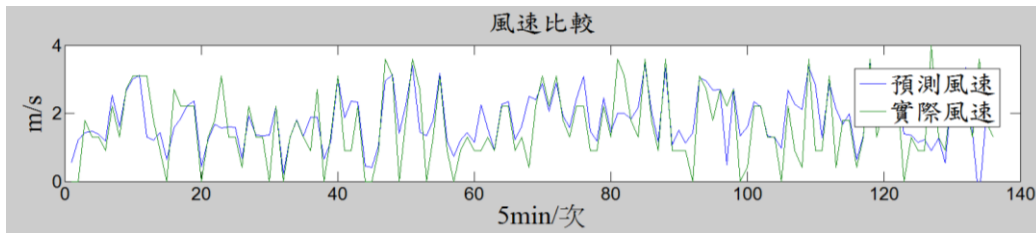


圖 25 龍山國小風速實際值與預測值比較圖

因美國風帆網站[25]風向風速資料量豐厚且時間間格小，故本研究利用其所提供的風力歷史資料做為大型風場資料來預測，將五分鐘平均風向、平均風速資料以不同輸入形式做未來十分鐘風力之預測，並將預測結果繪圖及列表比較，從中選擇出最適合的輸入資料形式。

以下為訓練架構比較之順序

1. 固定輸入層輸入端個數及隱藏層神經元個數後，調整網路預測的資料量做風速風向預測，比較其誤差以決定最適宜的訓練資料量，如表 6。而其所有訓練筆數所預測之風向、風速與實際比較的走勢圖則如圖 26、27 所示。
2. 決定訓練資料筆數後，以此訓練資料筆數為基準，調整輸入層元素個數，而隱藏層神經元則以輸入層元素數量開根號後取整數為神經元數量作為網路架構。不同輸入端數量之預測結果誤差整理列如表 7。而其所有不同的輸入層元素個數所預測之風向、風速與實際比較的走勢圖則如圖 28、29 所示。
3. 決定訓練資料比數及輸入層元素以後，調整隱藏層神經元個數，以順序 2 中使用的隱藏層神經元數量附近的數值作為其餘隱藏層神經元個數來比較，其預測誤差結果列表如 8。而其所有不同的隱藏層神經元個數所預測之風向、風速與實際比較的走勢圖則如圖 30、31 所示。

經過以上的參數選取以後，決定所需的短期未來風力資料以上列步驟的選擇設定為訓練預測網路的架構，最終目的為提供風機淨能量最佳化所需之迎風轉向動態式計算。

表 6 LM 類神經網路改變訓練資料量的預測誤差比較

	訓練資料量	輸入層元素個數	隱藏層神經元個數	風速 MRE	風向 MRE
1.	270 筆	30	6	0.0850	0.0215
2.	450 筆	30	6	0.0819	0.0186
3.	750 筆	30	6	0.0664	0.0185
4.	1000 筆	30	6	0.0696	0.0174
5.	1250 筆	30	6	0.0641	0.0193

由表 6 統計表中可發現風速預測誤差隨訓練資料量的增加而遞減，而風向雖有動盪但一直維持在誤差 2% 的水準以內，因此固定訓練筆數為 1250 筆，做改變輸入個數的比較。

表 7 LM 類神經網路改變輸入個數的預測誤差比較

	訓練資料量	輸入層元素個數	隱藏層神經元個數	風速 MRE	風向 MRE
1.	1250 筆	45	7	0.0633	0.0192
2.	1250 筆	30	6	0.0641	0.0193
3.	1250 筆	15	4	0.0568	0.0175
4.	1250 筆	10	3	0.0596	0.0171
5.	1250 筆	9	3	0.0682	0.0174

由表 7 統計表中選取誤差最小的輸入個數來預測，結果以 15 及 10 表現最佳，擇一選擇固定輸入個數為 15，去做隱藏層神經元改變的比較。

表 8 LM 類神經網路改變隱藏層神經元個數的預測誤差比較

	訓練資料量	輸入層元素個數	隱藏層神經元個數	風速 MRE	風向 MRE
1.	1250 筆	15	6	0.0614	0.0174

2.	1250 筆	15	5	0.0589	0.0191
3.	1250 筆	15	4	0.0568	0.0175
4.	1250 筆	15	3	0.0586	0.0163

經由以上篩選後，最終決定以訓練資料量 1250 筆、輸入層元素 15 個、隱藏層神經元 4 個來訓練網路以預測風向風速，並套用至迎風轉向控制以求達到省能的最佳狀態。

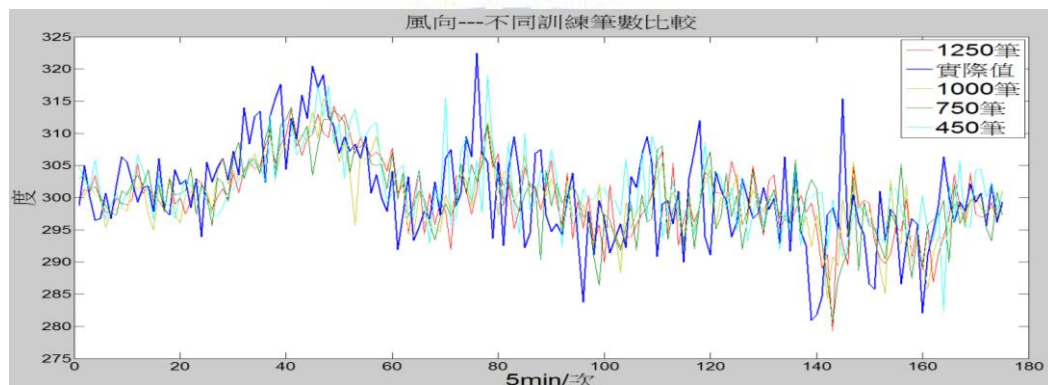


圖 26 風向之預測網路不同訓練筆數比較

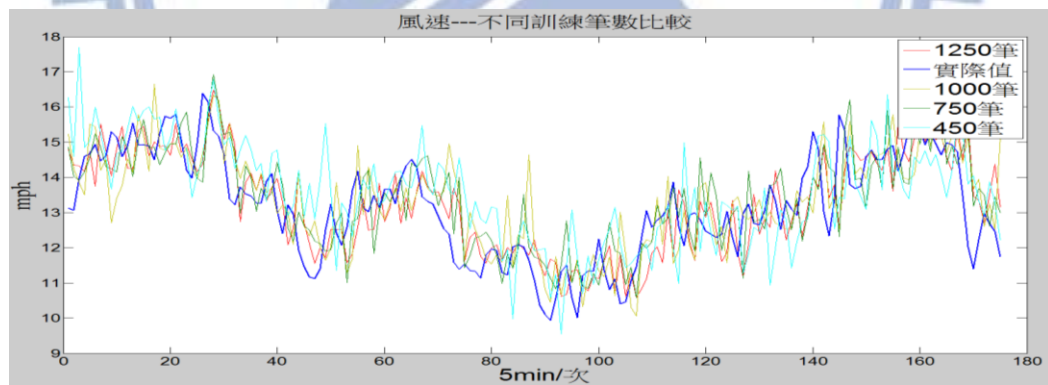


圖 27 風速之預測網路不同訓練筆數比較

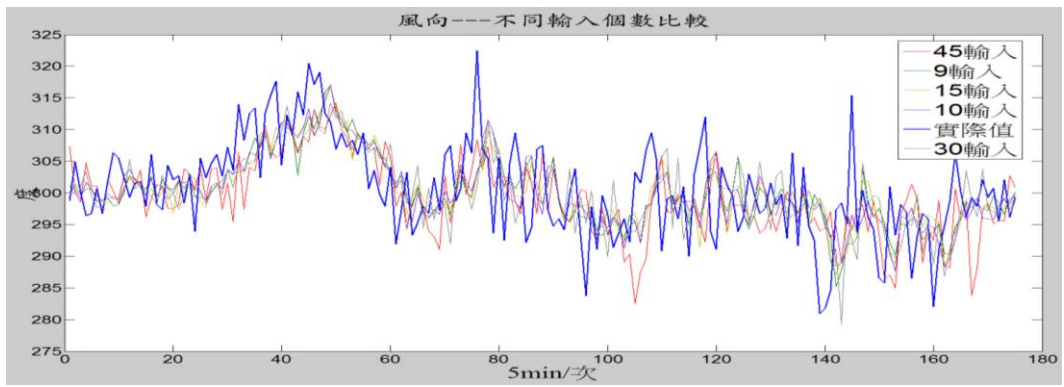


圖 28 風向之預測網路輸入個數改變比較

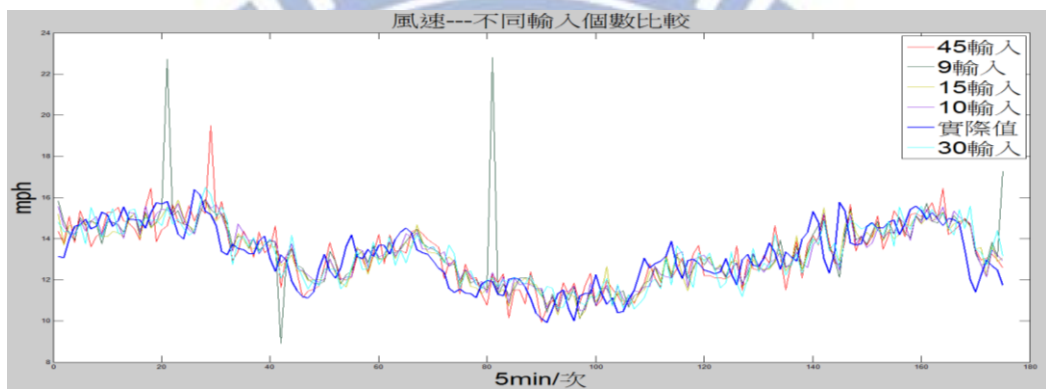


圖 29 風速之預測網路輸入個數改變比較

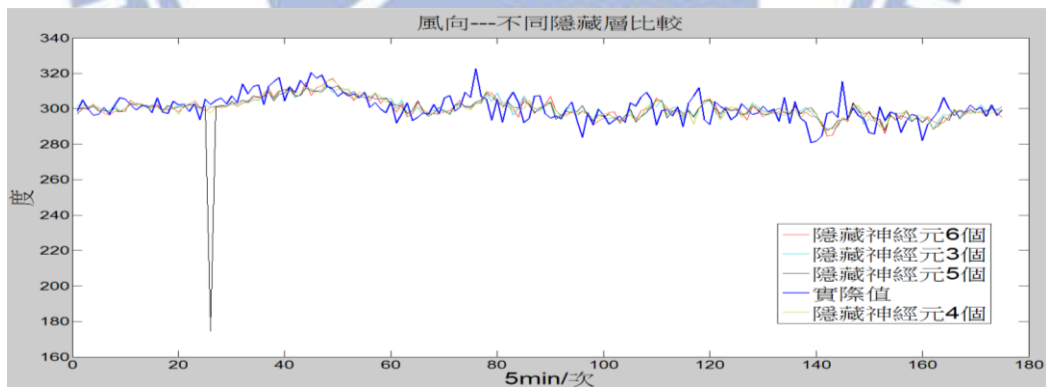


圖 30 風向之預測網路隱藏神經元改變比較

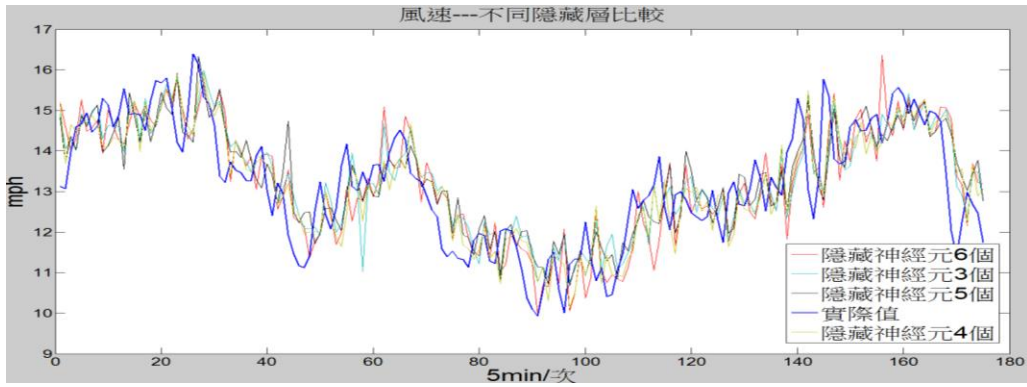
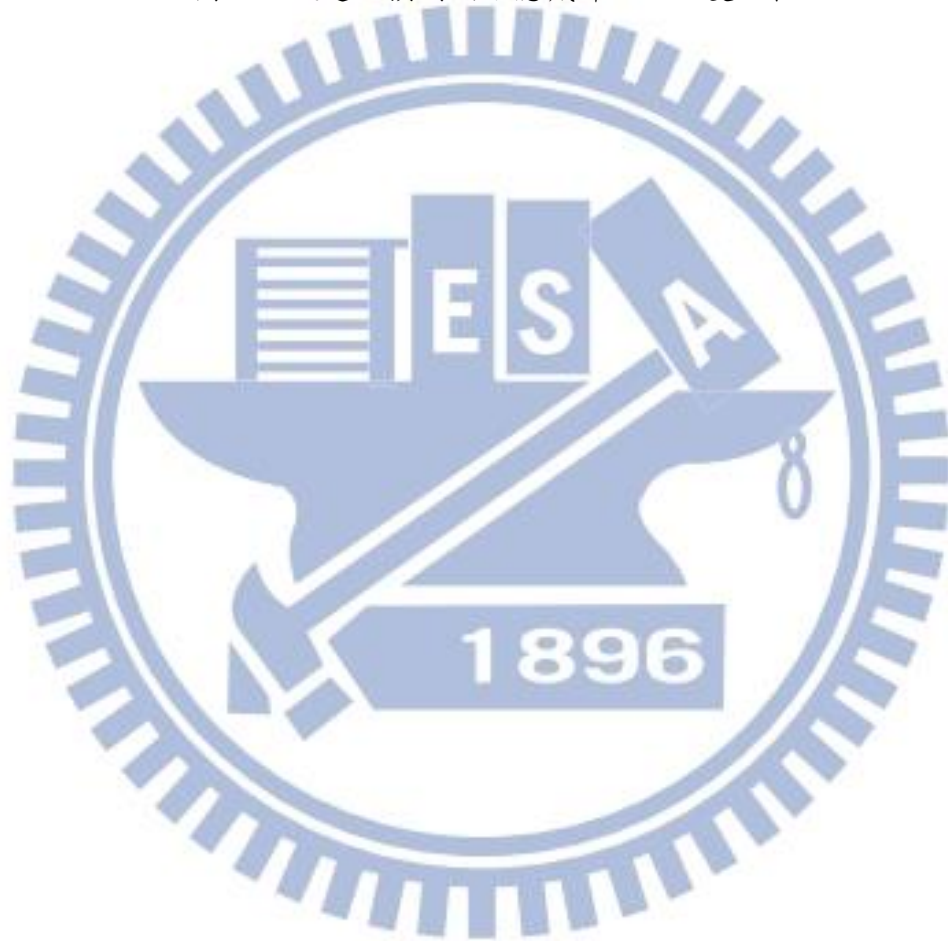


圖 31 風速之預測網路隱藏神經元改變比較



# 第六章 風機迎風轉向控制

## 6-1 風力發電機組技術參數

由文獻[13]中提及的風力發電機組之主要技術參數為參考(如表 9),來作為本研究的研究對象。圖 32 為此風力機台在廠商官方網頁[12]上的示意圖。

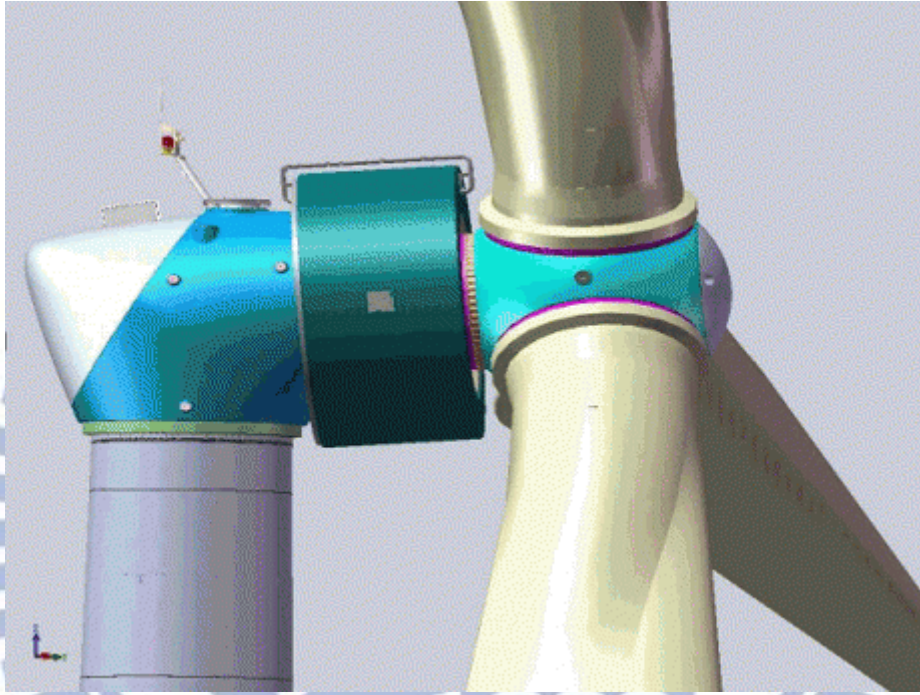


圖 32 XE93-2000 風力發電機組[12]

表 9 XE93-2000 型直驅永磁風力發電機組主要技術參數[12]

名稱	參數	發電機	
額定功率	2000kW	型式	永磁直驅同步發電機
切入風速	3m/s	電壓	660V,50Hz
額定風速	11m/s	重量	
切出風速	25m/s	葉輪	36 噸
極限風速	52.5m/s	機艙	19 噸

控制		發電機	56 噸
槳距系統	主動變槳	其他	
迎風轉向系統	主動迎風 轉向、液 壓制動	機艙與輪殼加葉 片總重量	111000kg
葉輪		機艙半徑	2m
葉片長度	34m	機艙迎風轉向額 定角速度	0.67°/s
直徑	70.65m	總傳動比	13390:1

參考文獻[13]，將表 9 提供的資料做風機扭矩與迎風轉向所需功率之計算。而另一方面，有關迎風轉向系統所使用的規格則以全球迎風轉向電機市占率 29% 的廠商 Bonfiglioli[23]提供的迎風轉向電機資料來做簡化迎風轉向轉移函數估測。

## 6-2 迎風轉向系統轉移函數

先由表 9 資料得電動機的角速度推導此 2MW 風力發電機中迎風轉向電機所需的額定轉速  $\omega_M$  如下：

$$\omega_M = i \cdot \omega_L = 13390 \cdot 0.67 \cdot \frac{\pi}{180} = 156.5(\text{rad/s}) = 1495(\text{rpm}) \quad (6-1)$$

$i$  為總傳動比、 $\omega_L$  為機艙迎風轉向之額定角速度。

經由上式推導，可知此發力發電機可選用額定轉速  $n_M$  為 1500rpm 的電動機作為迎風轉向電機。而機艙扭矩及機艙轉動一圈的時間分別推導如(6-2)、(6-3)。

$$T_{LF} = F \times R = m \cdot \omega_L \cdot R \cdot R = 111000 \cdot 0.67 \cdot 2 \cdot 2 = 199311.6 (\text{N} \cdot \text{m}) \quad (6-2)$$

$m$  為風機機艙與輪殼加葉片總重量、 $R$  為機艙半徑。

$$T = \frac{i}{n_M} = \frac{13390}{1500} = 8.9(\text{min}) \quad (6-3)$$

假設迎風轉向電機在峰值轉矩下以最高轉速不斷地驅動負載，則迎風轉向電機功率可以下式做大略估算

$$P_M \approx (1.5 \sim 2.5) \frac{T_{LF} \cdot \omega_L}{\eta} \quad (6-4)$$

$\eta$  為傳動效率。(6-4)中之常數 1.5~2.5 為經驗係數，在此  $P_M$  選擇以係數 2.5 做計算。

$$p \approx \frac{2.5 T_{LF} \omega_L}{\eta} = \frac{2.5 \cdot 199311.6 \cdot 0.67 \cdot \frac{\pi}{180}}{0.8} = 7279.7 (\text{Watt}) \quad (6-5)$$

由以上可知[13]的風力發電機轉向時迎風轉向電動機所需功率約為 7.3kW 且風機轉向角速度為 0.67°/s [13]。而風機的迎風轉向方程式則由文獻[14]提及如下式

$$I \ddot{\phi} = T - M_{f0} - \Delta M_f (\dot{\theta}_1 - \dot{\theta}_2) \quad (6-6)$$

其中  $M_{f0}$  為靜摩擦力矩，取靜摩擦係數  $\mu = 0.05$ ，滑動面的平均半徑為  $R = 2 \text{ m}$ ，則機艙重量產生的靜摩擦力矩如下

$$M = \mu WR = 0.05 \cdot 111000 \cdot 9.81 \cdot 2 = 108891 (\text{N} \cdot \text{m}) \quad (6-7)$$

而  $\Delta M_f(\cdot)$  函數值如圖 33 所示



圖 33  $\Delta M_f(\cdot)$  函數曲線

由於  $\Delta M_f(\cdot)$  非線性函數無法確立，故改以另一方式建立轉移函數。由[15]提供的



角位移對電壓之迎風轉向轉移函數算法如下

$$\frac{\theta(s)}{V(s)} = \frac{K_m}{s(T_m s + 1)} \quad (6-8)$$

其中  $K_m = K_{te} / (B_L + B_M)$ 、 $T_m = J_i / (B_L + B_M)$

$K_{te}$ ：電動機力矩常數， $B_L$ ：總摩擦係數， $B_M$ ：電動機力矩常數與電動勢常數相乘之值， $J_i$ ：總負載慣量。

由[16]、[17]所得資料帶入求得轉移函數如下

$$\frac{\theta(s)}{V(s)} = \frac{2.5 \times 10^{-3}}{s(508.33s + 1)} \quad (6-9)$$

$\theta$ ：角位移

$V$ ：電壓。

因此得其迎風轉向動態響應式為

$$\dot{\theta} = 0.67 \cdot \left( 1 - e^{-\frac{t}{508.33}} \right) \quad (6-10)$$

得時間常數 Time constant  $T = 508.33(s)$ ，取其積分後得角位移  $\theta$  解為

$$\theta = \theta_0 + 0.67 \cdot \left( t + T \cdot e^{-\frac{t}{T}} - T \right) \quad (6-11)$$

而如第二章第二節所提，剎車時，施力放大倍率與液壓控制系統成正比。故剎車時之轉移函數的 Time constant  $T_s$  可推導為

$$T_s = \frac{24}{160} \cdot T \quad (6-12)$$

經由上述推導，可見重達 111 噸之大型風機所需的迎風轉向系統響應安定時間相當長，若想將面向控制如預期，響應時間是重要考量因素。

### 6-3 風機面向角度控制方式

考量預測風力過程及搜尋最佳面向角度過程均需時間去做計算，故控制法則以預測十分鐘後之風速風向為基礎，設風機面向與預測之十分鐘後風向角度差為

$\alpha$ ，如下式

$$\alpha = \theta_{10\text{分鐘後之風向角}} - \theta_{\text{風機面向角}} \quad (6-13)$$

當  $\alpha$  與風機轉向慣性同向時，轉向控制為加速後剎車；反向時，系統需先剎車後逆向加速至適當位置時剎車。以下列出同向與反向的加速、剎車動態方程式。

1. 同向：

(a) 加速

$$0.67 \cdot \left( 1 - e^{-\frac{t_0}{T}} \right) = \dot{\theta}_{\text{慣性轉速}} \Rightarrow t_0 = -T \cdot \ln \left( 1 - \frac{\dot{\theta}_{\text{慣性轉速}}}{0.67} \right) \quad (6-14)$$

$$\dot{\theta}(t) = \pm 0.67 \cdot \left( 1 - e^{-\frac{-(t+t_0)}{T}} \right) \quad (6-15)$$

$$\theta(t) = \int \pm 0.67 \left( 1 - e^{-\frac{-(t+t_0)}{T}} \right) dt = \pm 0.67 \cdot \left( t + T \cdot e^{-\frac{-(t+t_0)}{T}} - T \cdot e^{-\frac{-t_0}{T}} \right) \quad (6-16)$$

(b) 剎車

$$\dot{\theta}(t) = \dot{\theta}_{\text{剎車前的速度}} \mp 0.67 \cdot \left( e^{\frac{t}{T_s}} - 1 \right) \quad (6-17)$$

$$\theta(t) = \int_0^t \dot{\theta}_{\text{剎車前的速度}} \mp 0.67 \cdot \left( e^{\frac{t}{T_s}} - 1 \right) dt = \dot{\theta}_{\text{剎車前的速度}} \cdot t \mp 0.67 \cdot \left( T_s \cdot e^{\frac{t}{T_s}} - t - T_s \right) \quad (6-18)$$

2. 反向：

(a) 剎車

$$\dot{\theta}(t) = \dot{\theta}_{\text{慣性轉速}} \pm 0.67 \cdot \left( e^{\frac{t}{T_s}} - 1 \right) \quad (6-19)$$

$$\theta(t) = \int_0^t \dot{\theta}_{\text{慣性轉速}} \pm 0.67 \cdot \left( e^{\frac{t}{T_s}} - 1 \right) dt = \dot{\theta}_{\text{慣性轉速}} \cdot t \pm 0.67 \cdot \left( T_s \cdot e^{\frac{t}{T_s}} - t - T_s \right) \quad (6-20)$$

(b) 加速

$$\dot{\theta}(t) = \pm 0.67 \cdot \left( 1 - e^{-\frac{t}{T}} \right) \quad (6-21)$$

$$\theta(t) = \int \pm 0.67 \left( 1 - e^{-\frac{t}{T}} \right) dt = \pm 0.67 \cdot \left( t + T \cdot e^{-\frac{t}{T}} - T \right) \quad (6-22)$$

(c) 剎車

$$\dot{\theta}(t) = \dot{\theta}_{\text{第二次剎車前的速度}} \mp 0.67 \cdot \left( e^{\frac{t}{T_s}} - 1 \right) \quad (6-23)$$

$$\theta(t) = \int_0^t \dot{\theta}_{\text{第二次剎車前的速度}} \mp 0.67 \cdot \left( e^{\frac{t}{T_s}} - 1 \right) dt = \dot{\theta}_{\text{第二次剎車前的速度}} t \mp 0.67 \cdot \left( T_s \cdot e^{\frac{t}{T_s}} - t - T_s \right) \quad (6-24)$$

以上為在各種情況下的角速度及角位移動態方程式。 $\theta$  為風機面向之角度， $T$ 、 $T_s$  分別為加速時間與剎車時間之時間常數。令控制前的風機面向角為  $\theta_0$ ，且同向時加速的時間為  $t_v$ 、剎車的時間為  $t_s$ ；反向時第一次剎車時間為  $t_{s1}$ 、加速時間為  $t_v$ 、第二次剎車時間為  $t_{s2}$ 。假設風機電能擷取公式中參數  $C_p = 40\%$ 、 $A = \frac{1}{4} \times \pi \times 70.65^2 = 3918.27(m^2)$ 、 $\rho = 1.18(kg/m^3)$ ，並依情況選取上述的動態方程式代入至風機電能擷取公式  $E = \int \frac{\rho A C_p}{2} v^3 \cos(\alpha - \theta) dt$  以及轉向耗能公式  $W = \int m r R \cdot \dot{\theta} dt$  後可計算出最後的淨獲能：

$$\text{同向時， } E = E_{t_v} + E_{t_s} \quad 、 \quad W = W_{t_v} + W_{t_s}$$

$$\text{逆向時， } E = E_{t_{s1}} + E_{t_v} + E_{t_{s2}} \quad 、 \quad W = W_{t_{s1}} + W_{t_v} + W_{t_{s2}}$$

最終取出構成  $E - W$  的最大值的加速時間及剎車時間，即可得到最適合的轉向角度，並以此作為轉向指令來控制風機。上述流程圖示如下：

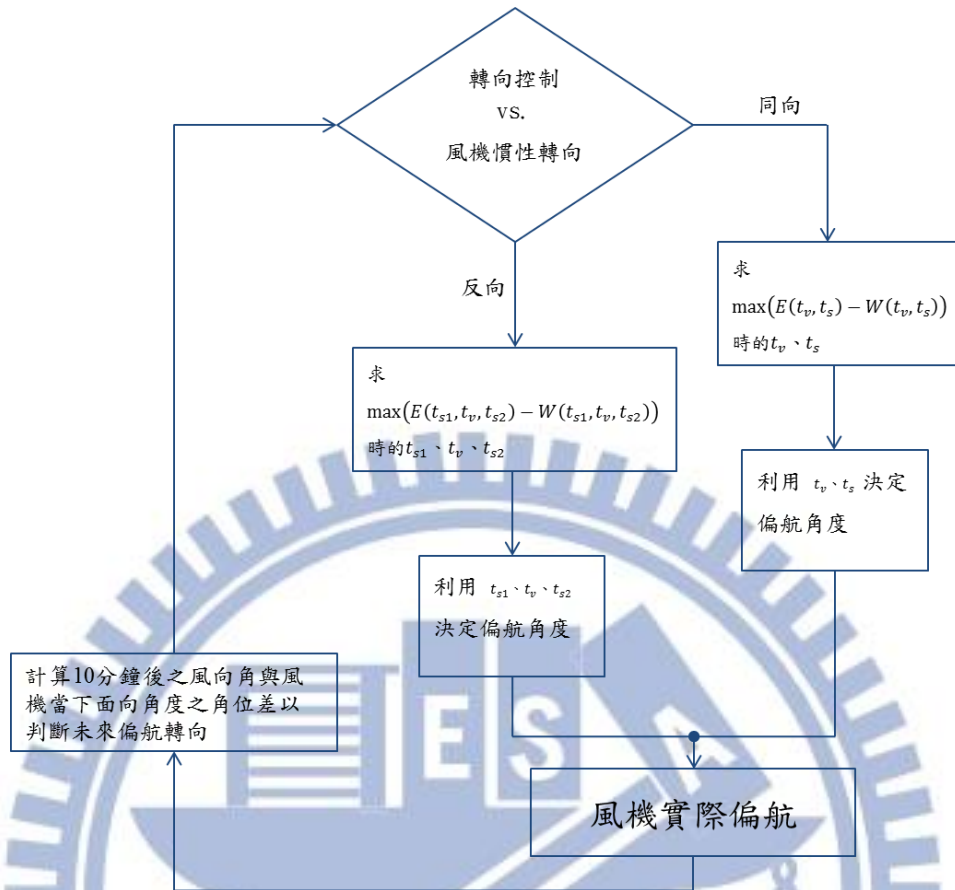


圖 34 決定迎風轉向角度之流程圖

本研究利用 Matlab 內建用於最佳化之函數 `fmincon`，在參數間的非線性及線性限制條件下，找出性能曲面之最佳解。同向時，線性限制條件為  $t_v + t_s \leq 300$

(s)，非線性限制條件為  $0.67 \left( 1 - e^{-\frac{-(t_v + t_s)}{T}} \right) - 0.67 \left( e^{\frac{t_s}{T_s}} - 1 \right) \geq 0$ ；反向時，線性限制條

件為  $t_{s1} + t_v + t_{s2} \leq 300$ (s)、非線性限制條件為  $0.67 \left( 1 - e^{-\frac{t_v}{T}} \right) - 0.67 \left( e^{\frac{t_{s2}}{T_s}} - 1 \right) \geq 0$

且滿足  $\left| \dot{\theta}_{\text{慣性轉速}} \right| - 0.67 \cdot \left( e^{\frac{t_{s1}}{T_s}} - 1 \right) = 0$ 。下圖為擷取某一時間點的性能函數曲面，因

Matlab 內建 `fmincon` 以解出性能函數最低點為目的，故作計算時，實際上是求  $-(E - W)$  之最小值。

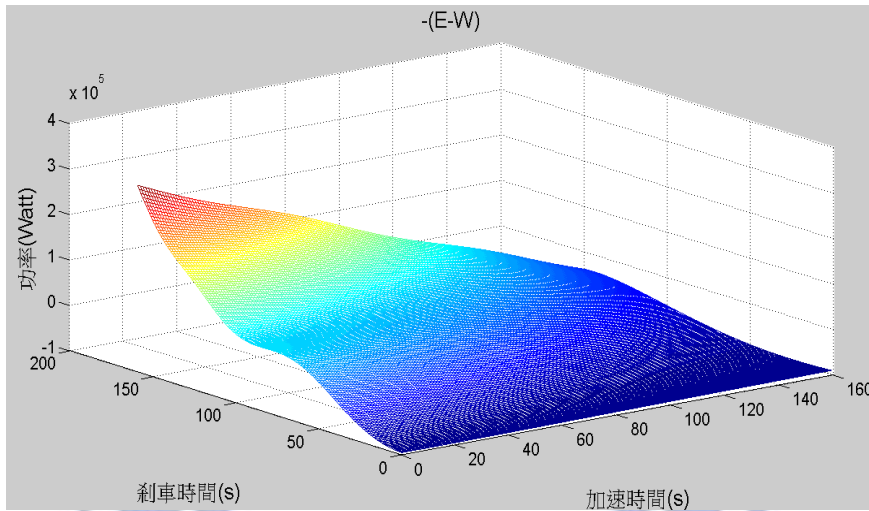


圖 35 同向時，以  $t_v$ 、 $t_s$  構成的  $W-E$  性能函數曲面

#### 6-4 風機實際反應之模擬

從美國風帆網站 Wind Power Weather Center [25] 中，擷取一段每五分鐘一筆的風力資料做預測並控制。而預測結合迎風轉向控制的步驟如下：

1. 利用 1250 筆訓練資料、15 個輸入的五分鐘平均風向、風速資料架構類神經網路。
2. 以此類神經網路預測十分鐘後之平均風向、風速。
3. 依 6-3 節講述的方法判斷風機最佳迎風轉向方向。
4. 將 Matlab 的 Simulink 與 Guide 結合，透過人機介面顯示風機控制及風力實際變化以及獲能與耗能走勢。

經由上述步驟後，將實際風向與利用預測之風速、風向來做迎風轉向控制的角度相比較如圖 36。而預測之風向角度與控制後面向之角度比較如圖 37，控制後角度因轉向需時間故比預測出迎風轉向角度之時間點晚五分鐘才能完成迎風轉向。實際的角度與經最佳化後決定控制的角度差繪製如圖 38。圖 39 與圖 40 為風機應用控制轉向後之每五分鐘內平均獲能功率及平均耗能功率走勢圖。

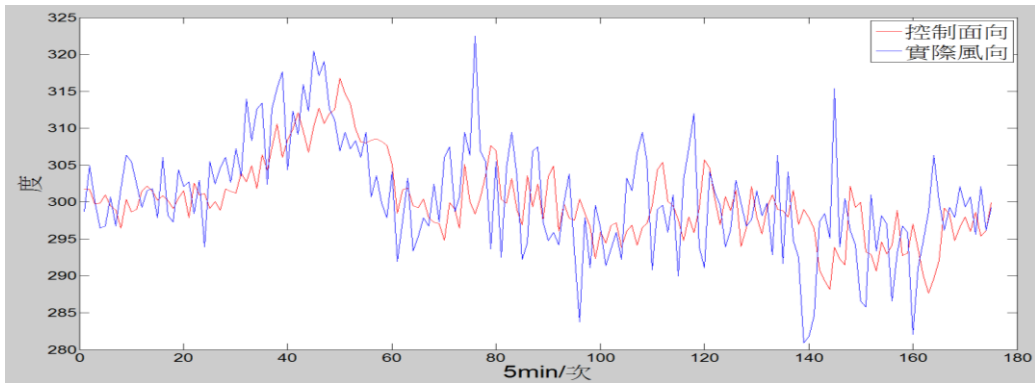


圖 36 實際風向與能量最佳化之轉向控制比較圖

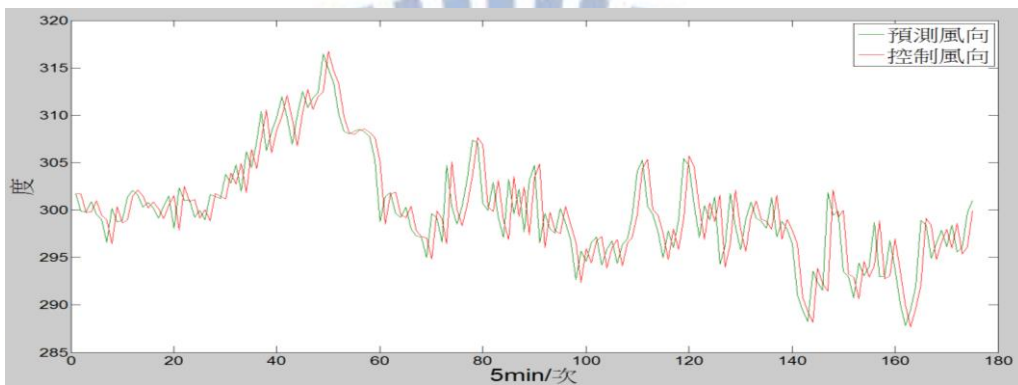


圖 37 預測風向與能量最佳化之轉向控制比較圖

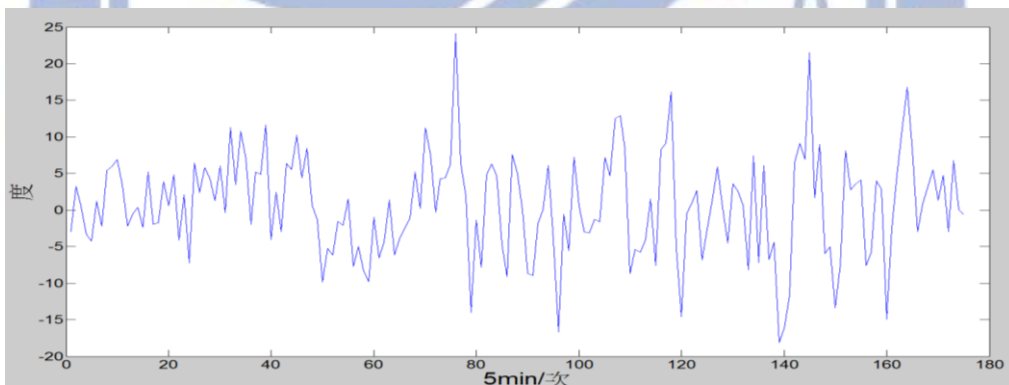


圖 38 控制與實際的相差角度

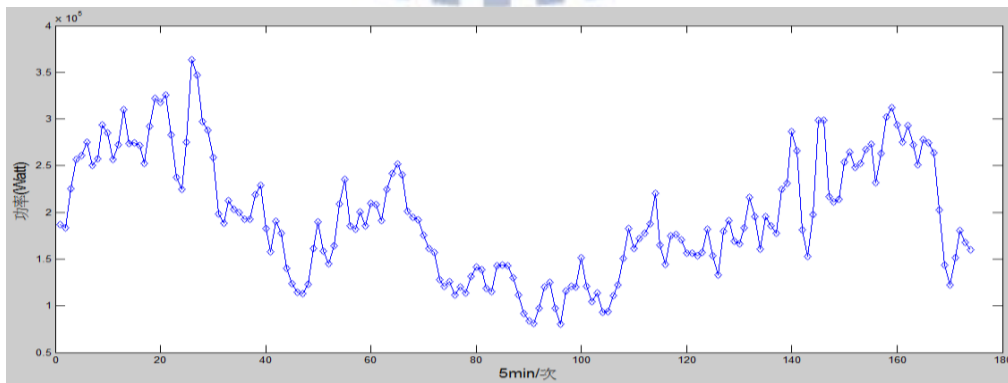


圖 39 風機轉向五分鐘平均獲得功率

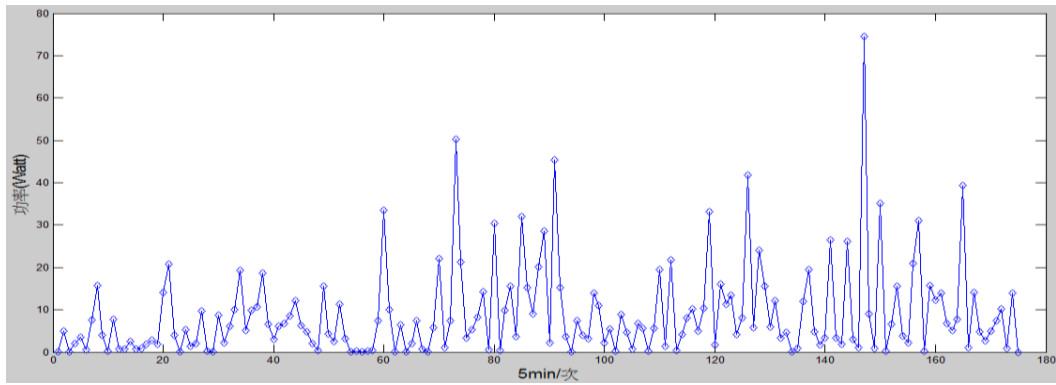


圖 40 風機轉向五分鐘平均消耗功率

除此之外，本研究為瞭解風機在未經過預測下，直接隨每五分鐘得到的風向訊號迎風轉向可得的能量，因此以風能獲能公式及風機轉向耗能公式搭配風機的動態方程式模擬風機隨過去五分鐘風向轉向的功率走勢。其風機迎風轉向如圖 41 所示，其風機測得風向後隨即改變迎風轉向角度追隨得到的風向訊號。拿此未經預測風力風向控制的迎風轉向走勢與經預測後控制的迎風轉向走勢圖比較呈現於圖 42。最後將兩者每五分鐘平均的耗能功率、獲能功率、總淨獲能功率比較圖示於圖 43 至圖 45。而圖 46 則是兩者間每五分鐘總淨獲能功率之差異曲線。

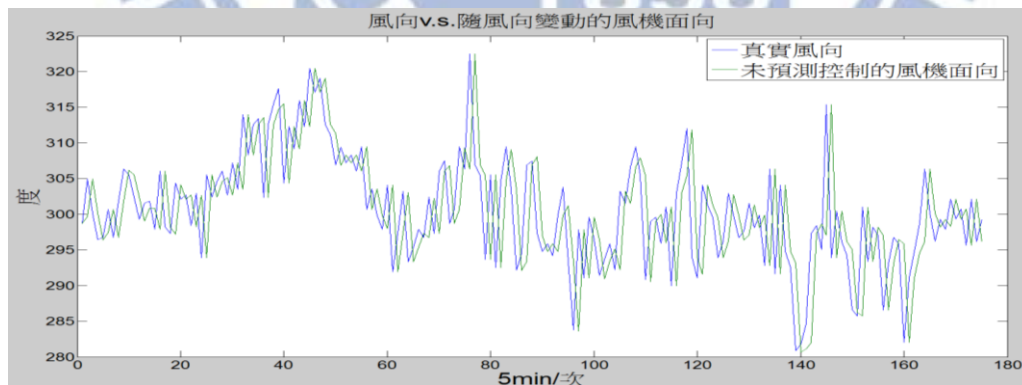


圖 41 隨風向迎風轉向的風機面向與真實風向比較圖

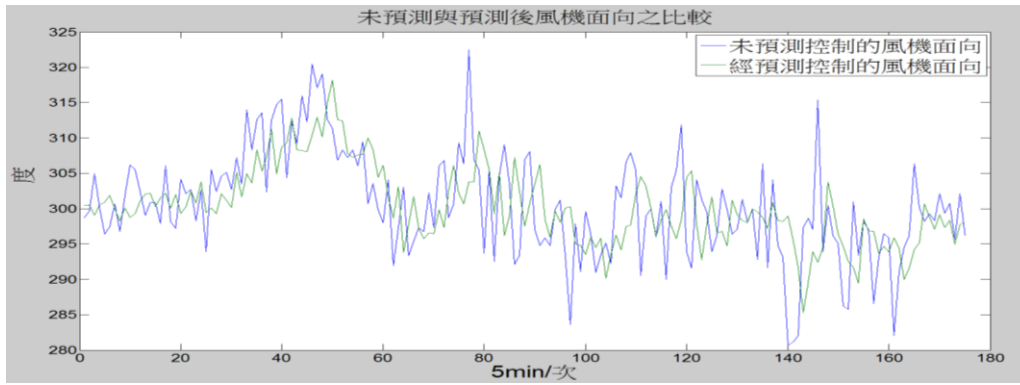


圖 42 隨風向迎風轉向與經預測控制迎風轉向之風機面向比較圖

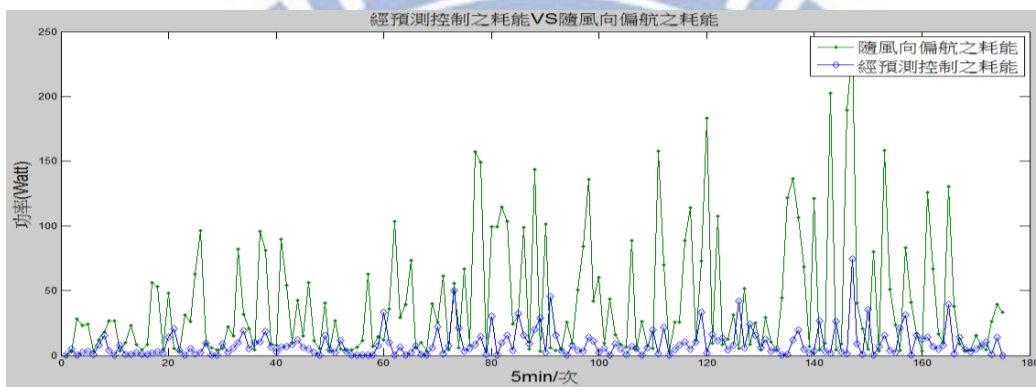


圖 43 隨風向迎風轉向與經預測控制迎風轉向五分鐘平均耗能功率比較圖

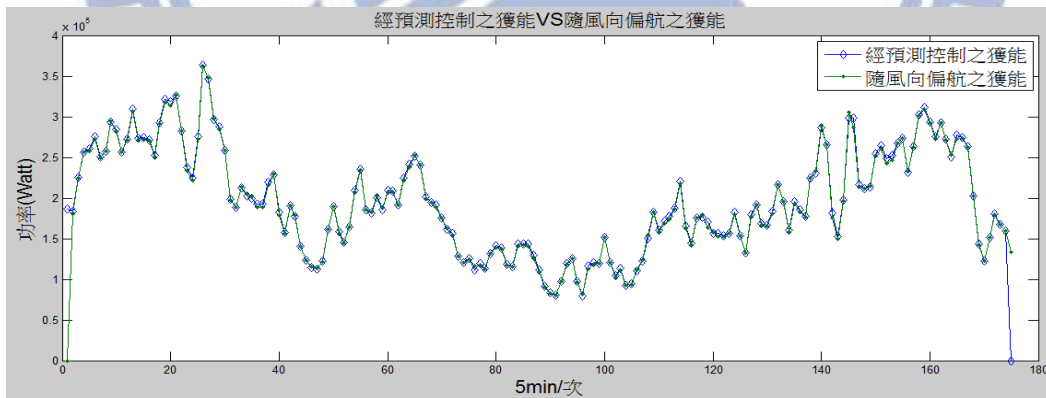


圖 44 隨風向迎風轉向與經預測控制迎風轉向五分鐘平均獲能功率比較圖



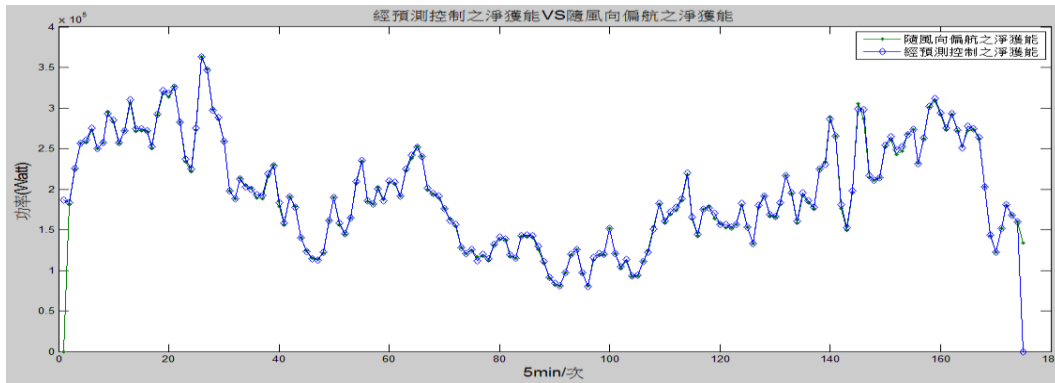


圖 45 隨風向迎風轉向與經預測控制迎風轉向之五分鐘平均淨獲能功率比較

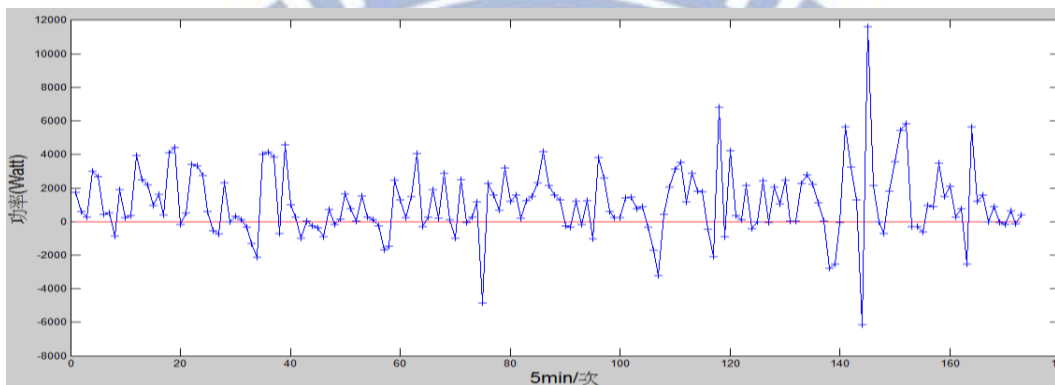


圖 46 隨風向迎風轉向與經預測控制迎風轉向之五分鐘平均淨獲能功率差異

經計算後得平均下來以預測做控制可獲得之淨獲能比隨風向轉動可得之淨獲能多出約 1100W 的功率，也就是每小時多出約 1.1kWh 的能量。而此結論之風場資料來自風帆網站，本文用以模擬之平均風力大小約為 5.924m/s，約為滿載發電額定風速 11m/s 之 0.538 倍，而風能正比於風速三次方，故若能改為使用真正的大型風場資料做預測控制，必定能使省下的功率大幅提升。且經預測後控制的风機所需轉動的角度比隨風向轉的风機角度擺盪幅度小很多，從機構耗損的觀點，可以幫助增長迎風轉向系統壽命。

本研究在最後利用 Matlab 之 Simulink 工具與 Guide 工具結合繪製風機轉向的人機介面顯示視窗以方便觀察控制過程，視窗如圖 47 所示。左邊三個動態走勢圖每五分鐘更新一次，左上圖為控制風機後實際五分鐘內平均獲得之能量功率；

左中圖為經控制後的風機轉向所需五分鐘內平均消耗之能量功率，而視窗左下圖則為獲能與耗能相抵之後的淨能量功率走勢。視窗右方的動態圖為從空中俯視風機轉動之二維圖，紅色箭頭為實際風向而長度則代表風速。

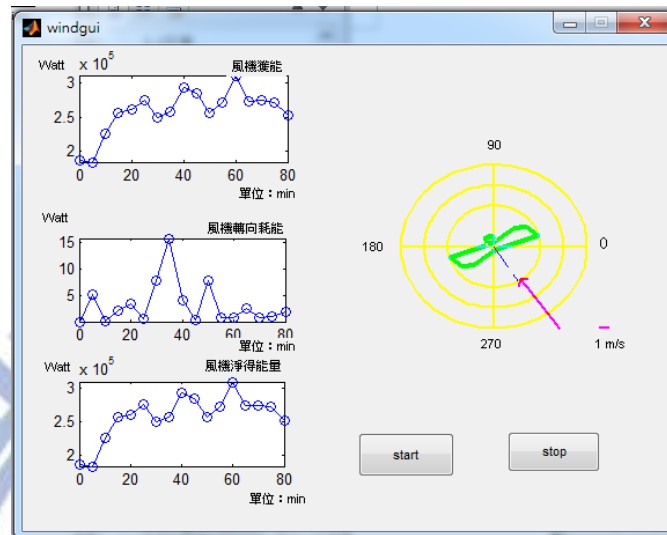


圖 47 顯示風機面向與實際風向及功能獲取的人機介面

# 第七章 結論與未來展望

## 7-1 結論

本研究針對風場變化對大型風機轉向機制影響，利用預測之風速風向找出淨獲能最大的迎風轉向方向。研究內容可分預測及迎風轉向兩部分。

預測方面：本研究以美國風帆網站的風速風向資料套用進 Matlab 的 nntool toolbox 架構各種類神經網路預測短期風力，將風力資料分為訓練組與驗證組後經由各種前饋倒傳遞演算法(如梯度下降倒傳遞、共軛梯度倒傳遞...等)的類神經網路做預測並比較，最後選擇以前饋倒傳遞中之 Levenberg-Marquardt 演算法類神經網路來做預測。隨後比較在不同訓練筆數下所得到的預測誤差，而選擇了 1250 的訓練筆數。確定筆數後，調整輸入端個數去比較最適合的輸入層元素數量，而決定以 15 個輸入構成此類神經網路的輸入層。最後固定此輸入個數並在輸入總數開根號的附近找尋最佳的隱藏層神經元個數，得解為 4。結論用來預測本研究資料的類神經網路架構以 Levenberg-Marquardt method、1250 訓練筆數、15 個輸入層元素、4 個隱藏層神經元來預測未來十分鐘的風速、風向資料。

迎風轉向方面：經由從文獻中得到的風機迎風轉向系統參數來推導出 2MW 的風機迎風轉向之轉移函數及動態方程式，而從此轉移函數可之風機迎風轉向所需的響應安定時間極久，不僅加速，剎車也亦如此。利用加速時間及煞車時間兩項變數依造不同的狀況帶入至風機迎風轉向的動態方程式後，再套入轉向獲能與轉向耗能的公式去做積分，將其繪圖並找出加速時間及煞車時間的最佳解，而後將此時間變數解代回風機迎風轉向動態方程式得角速度及角位移的控制指令。

本文在最後以預測加迎風轉向所得之能量與風機隨風向轉變而得之能量比較，得到本文提出之研究方法在美國風帆網站的風速下平均每小時可多出約 1.1kWh 的能量。最後以 Matlab 的 guide toolbox 及 simulink 結合製出人機介面，將每五分鐘的平均獲能、耗能與淨獲能以及風機迎風轉向的動畫繪製於圖 47。

## 7-2 未來展望

1. 由於風帆網站的風力資料與大型風場的類型不同，風向、風速多變，故在預測上很難達到理想，本論文僅能以大量比較取出最低誤差值來決定網路的架構及其使用的演算法。未來若能有大型風機風場的短期資料，則對於預測的準度相信會有一定的提升。另一方面，大型風機之運轉參數不易取得，若能有更詳盡的風機規格，則必能更精確地做迎風轉向控制。
2. 目前本研究已完成用類神經網路做風力風向預測及轉向取最佳化面向角度的部分。而以下為本文認為未來可繼續在風力預測及風機迎風轉向上努力的地方：
  - (a) 修正類神經網路架構使其更準確。由文獻中所述，尚有其餘演算法可供預測風力資料使用，如 SVM、徑向基類神經網路、基因演算法等，可以多加比較找出最適合的預測方法。
  - (b) 設計當預測誤差過大導致風機迎風轉向失誤時的應變措施。預測總會有誤差，當預測出現誤差時，該如何做出相關應變措施，如調整葉片角度或即時變更迎風轉向指令或是參考[29]的爬山法等。
  - (c) 在控制程式中寫入當預測到突然特大強風時事先採取 90 度側風保護風機。若預測到會有突如其來超過風機負載的特大強風時，應事先採取保護機制如 90 度側風或控制葉面角度減緩力矩。
  - (d) 在控制程式中寫入當風機電纜轉超過三圈時的解纜機制。當風機持續順時針或逆時針迎風轉向時，應設計讓風機無條件轉回解纜。

## 第八章 參考文獻

1. 朱翊誌，「含風力發電為電網之孤島運轉策略」，國立中山大學電機工程學系，碩士論文，中華民國 98 年 6 月。
2. Andrew Kusiak, Zijun Zhang, “Short-Horizon Prediction of Wind Power\_ A Data-Driven Approach” IEEE TRANSACTIONS ON ENERGY CONVERSION, VOL. 25, NO. 4, December 2010
3. C. Pérez-Llera, M.C. Fernández-Baizán, J.L. Feito , V. González del Valle, “Local Short-Term Prediction of Wind Speed A Neural Network Analysis” TECHNIQUES AND METHODOLOGIES, iEMSs 2002 sessions (part two)
4. S. Salcedo Sanz, A'. Pe'rez-Bellido, E. Ortiz-Garc, A. Portilla-Figueras, L. Prieto, D. Paredes, F. Correoso “Short-term Wind Speed Prediction by Hybridizing Global and Mesoscale Forecasting Models with Artificial Neural Networks” Hybrid Intelligent Systems, Eighth International Conference on 2008. HIS '08.
5. Felix A.Farret, Luciano L.Pfitscher, Daniel P.Bernardon , “Sensorless active yaw control for wind turbines” , Industrial Electronics Society, The 27th Annual Conference of the IEEE 2001. IECON '01.
6. Feifei Bu, Wenxin Huang, Yuwen Hu, Yunqing Xu, Kai Shi, Qianshuang Wang , “Study and Implementation of A Control Algorithm for Wind Turbine Yaw Control System”, World Non-Grid-Connected Wind Power and Energy Conference, WNWEC 2009
7. Andrew Kusiak, Haiyang Zheng, Zhe Song ,“Power optimization of wind turbines with data mining and evolutionary computation”, Renewable Energy, 2010
8. Andrew Kusiak, Haiyang Zheng, Zhe Song, “ Short-Term Prediction of Wind Farm Power: A Data Mining Approach” IEEE TRANSACTIONS ON ENERGY

CONVERSION, VOL. 24, NO. 1, MARCH 2009.

9. Mrs. Patil Sangita B & Technology Prof. Mrs. Surekha. R. Deshmukh. "Use of Support Vector Machine for Wind Speed Prediction", Power and Energy Systems (ICPS), International Conference on 2011.
10. 羅華強，類神經網路-MATLAB 的應用，三版，高立圖書有限公司，中華民國一百年十二月。
11. 台北市校園數位氣象網，  
<http://weather.tp.edu.tw/Scripts/active.php?TitleImg=title01>。
12. 湘電風能(福建)有限公司，  
<http://www.xemc-windfj.com/product/product.php?lang=cn&itemid=108>
13. 韓兵，「直驅永磁風力發電機組偏航系統的設計」，中國湖南大學，碩士學位論文，中華民國一百年五月十一日。
14. 廖明夫，黃巍，董禮，「風力機偏航引起的失穩振動」，西北工業大學旋轉機械與風能裝置測控研究所太陽能學報，第04期，2009年。
15. Kung Chris Wu, Rony K. Joseph, Nagendra K. Thupili, "Evaluation of Classical and Fuzzy Logic Controllers for Wind Turbine Yaw Control", Aerospace Control Systems, the First IEEE Regional Conference on 1993.
16. 義大利迎風轉向電機廠商Bonfiglioli, <http://www.bonfiglioli.com/en/corporate>
17. T. Harris, J.H. Rumbarger, C.P. Butterfield, "Wind Turbine Design Guideline DG03: Yaw and Pitch Rolling Bearing Life", Technical Report NREL/TP-500-42362, December 2009
18. 吳繼平，「應用類神經網路及基因演算法預測風速與風力發電量」，中原大學電機工程學系，碩士學位論文，中華民國九十六年七月。
19. 張家豪，「風力發電機動態行為之研究」，國立台灣科技大學電機工程系，碩士學位論文，中華民國九十四年七月七日。
20. 陳清嚴，「風力發電機功率輸出模式之建立與應用」，國立台灣科技大學電機工程系，碩士學位論文。中華民國九十七年一月十七日。
21. 周家得，「以支向機為基礎並結合特徵擷取之語音辨識系統」，國立台灣科

技大學資訊工程系，碩士學位論文，中華民國九十五年一月十一日。

22. George Sideratos , Nikos D. Hatziargyriou, “An Advanced Statistical Method for Wind Power Forecasting”, IEEE TRANSACTIONS ON POWER SYSTEMS, VOL. 22, NO. 1, February 2007
23. Colin Marson, Lisergy Consulting, “Wind Turbine Pitch and Yaw Drive Manufacturers” , <http://www.geartechnology.com>, January/February 2010
24. 永傳能源股份有限公司，福海離岸風力發電計畫  
<http://www.taiwangenerations.com/project.php>
25. Wind Power Weather Center , <http://www.windpowerweather.com/>
26. L.M. Saini , M.K. Soni, “Artificial neural network based peak load forecasting using Levenberg-Marquardt and quasi-Newton Method”, Generation, Transmission and Distribution, IEE Proceedings- (Volume:149 , Issue: 5 )
27. Martin T. Hagan , Mohammad B. Menhaj , “Training Feedforward Networks with the Marquardt Algorithm”, IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 5, NO. 6, November 1994.
28. J. P. S. Catalão, H. M. I. Pousinho, V. M. F. Mendes , “An Artificial Neural Network Approach for Short-Term Wind Power Forecasting in Portugal” Intelligent System Applications to Power Systems, ISAP '09. 15th International Conference on 2009.
29. WU Xin, LIU Yanping, TENG Wei, “Modified Hill Climbing Method for Active Yaw Control In Wind turbine” Control Conference (CCC), Chinese , March 2012.
30. 「呼呼海風賺綠金」，台灣光華雜誌，  
[http://www.sinorama.com.tw/show\\_issue.php?id=201160006022c.txt&cur\\_page=2&table=1&distype=&h1=&h2=&search=&height=&type=&scope=&order=&keyword=&lstPage=&num=&year=2011&month=06](http://www.sinorama.com.tw/show_issue.php?id=201160006022c.txt&cur_page=2&table=1&distype=&h1=&h2=&search=&height=&type=&scope=&order=&keyword=&lstPage=&num=&year=2011&month=06)。

31. Pérez-Llera, C., Fernández-Baizan, M. C., Feito, J. L., & Gonzalez, V. "Local short-term prediction of wind speed: a neural network analysis." In Proceedings of the iEMSs Vol. 2, pp. 124-129 ,June 2002.
32. Han-Soon Choi, Jeong-Gi Kim, Jang-Hwan Cho, Yoon-su Nam, "Active Yaw Control of MW class Wind Turbine", Control Automation and Systems (ICCAS), International Conference on 2010.
33. 中華太陽能聯誼會，<http://www.solar-i.com/wi.htm>。
34. 張德豐，MATLAB 神經網路應用設計，二版，機械工程出版社，北京，2011年12月。
35. 黃靖哲，「全橋半控型整流器及阻抗源變流器整合之風力發電系統」國立交通大學機械系，碩士學位論文，中華民國一百零一年三月。
36. 曾炫錡，「順滑模態觀測速度控制應用於無轉速感測之直驅式可變速風力發電機」，國立交通大學機械系，碩士學位論文，中華民國一百零一年六月。
37. 江國維，「基於倒階法則之積分型順滑模態控制應用於可變速風力發電機」，國立交通大學機械系，碩士學位論文，中華民國一百零一年六月。