


# 國立交通大學

## 電控工程研究所

### 碩士論文

整合資料預測控制與具有延遲及訊息遺失

之網路伺服馬達控制系統



Integration of the Data-based Predictive Control  
with the Delay and Missing-message Compensation  
on Networked Servo Motor Control Systems

研究生：張志嘉

指導教授：徐保羅 博士

中華民國一百零三年一月

整合資料預測控制與具有延遲及訊息遺失

之網路伺服馬達控制系統

Integration of the Data-based Predictive Control  
with the Delay and Missing-message Compensation  
on Networked Servo Motor Control Systems

研究生：張志嘉

Student : Chih-Chia Chang

指導教授：徐保羅 博士

Advisor : Dr. Pau-Lo Hsu



A Thesis

Submitted to Department of Electrical and Control Engineering  
College of Electrical and Computer Engineering  
National Chiao-Tung University  
in Partial Fulfillment of the Requirements  
for the Degree of Master  
in  
Electrical and Control Engineering  
January 2014  
Hsinchu, Taiwan, Republic of China

中華民國一百零三年一月

# 整合資料預測控制與具有延遲及訊息遺失 之網路伺服馬達控制系統

研究生：張志嘉

指導教授：徐保羅 博士

國立交通大學 電控工程研究所

## 摘要

網路化控制在工業上已經成為一種趨勢，然而，伴隨網路化產生的時間延遲和資料的遺失等問題，輕則降低系統效能，重則使整個系統不穩定。本研究以時間延遲完全補償策略(Perfect Delay Compensation, PDC)處理網路時間延遲的問題。PDC 將延遲的影響簡化為原閉迴路系統串接一時間延遲，避免系統不穩定。針對資料遺失的處理以即時轉移機率估測器和智慧型訊息估測器(Intelligent Message Estimator, IME)作資料遺失的補償。即時轉移機率估測器利用即時短時間內訊息遺失的機率快速地計算出轉移機率，成功且有效率的監控資料遺失的分佈情形，並以估測出的轉移機率來切換不同的補償器來補償不同分佈的資料遺失。為了將 IME 應用於不同的網路控制系統中，提出資料預測控制器(Data-based Predictive Controller, DPC)及時地計算出最符合預期響應之控制訊號，並與即時轉移機率估測器搭配，作為 IME 補償區間切換的指標。最後結合了 DPC、IME 與 PDC，以不受限於單一系統的架構，同時處理了網路資料遺失與時間延遲的對控制系統產生的影響。經模擬及實驗證明本論文所提出的架構，皆能有效且簡單應用在實際的網路控制系統上，達成網路時間延遲及訊息遺失的補償效果，提升網路控制的效能。

# **Integration of the Data-based Predictive Control with the Delay and Missing-message Compensation on Networked Servo Motor Control Systems**

Student: Chih-Chia Chang

Advisor: Dr. Pau-Lo Hsu

Institute of Electrical and Control Engineering

National Chiao Tung University

## **ABSTRACT**

In the network control systems (NCS), both data dropout and time delay greatly affect control performance and stability. In this study, the perfect delay compensation (PDC) is applied to eliminate the network-induced delay effect on the NCS with an AC servo motor, Ethernet, Zigbee, CAN bus. Moreover, applying the least square algorithm and real-time estimation of the transition probability for data dropout, the intelligent message estimator (IME) in a switching mechanism based on the estimated transition probability is developed to reliably estimate the missing message with different dropout distributions. Furthermore, the data-base predictive controller (DPC) is implemented on NCS in this Thesis to predict the best control signals by optimizing system performance. By combining the IME and the predicted signal from DPC, the automatic switching mechanism of IME under different data dropout rates is thus obtained. Finally, the integrated compensator combining all the PDC, IME and DPC leads to satisfactory compensation results to deal with both the time delay and the data dropout in NCS.

## 致謝

首先要感謝我的指導教授徐保羅老師這兩年來對我的指導，讓我在研究上遇到瓶頸與挫折，或是迷失方向時，能夠找到正確的出路，使我能順利的完成碩士學位。同時也要感謝口試委員蕭得聖老師、黃育綸老師、鄭兆凱副總對於本論文的批評指正，以及寶貴的建議，使得本論文更加完整。

亦感謝實驗室的學長、同學與學弟們:博士班賴建良學長、謝鎮州學長、黃煒生學長，平時給我的指導與幫助，以及葉釗甫學長、林軒正學長、蔡政宏學長、廖子期學長、黃思翰學長、郭志瀚學長、林信佑學長、黃琮昇學長給我的寶貴意見及解惑，誌緯、晟傑、振文、昱安、家豪、竣凱在學業上的相互切磋與指教，並且時常分享生活趣事讓實驗室充滿歡笑與活力。特別對林信佑學長、以及誌緯、晟傑、振文在研究與實驗上的大力幫忙，獻上誠摯的感謝。

最後要感謝所有關心我的父母、哥哥、女友以及老朋友們，長久以來在我生活上的支持與鼓勵，使我能全心全意的專注在課業與研究上，願將此榮耀與喜悅和他們一起分享。

# 目錄

摘要.....	i
ABSTRACT.....	ii
目錄.....	iv
圖目錄.....	vii
表目錄.....	i
第一章 緒論.....	1
1.1 研究動機與目的.....	1
1.2 研究背景與發展概況.....	1
1.3 問題描述.....	4
1.4 研究方法.....	4
1.5 論文架構.....	5
第二章 實驗平台與模擬環境.....	6
2.1 CAN 網路協定.....	6
2.2 Zigbee 網路協定.....	7
2.3 乙太網路(Ethernet)協定.....	11
2.3.1 TCP/IP 通訊協定.....	13
2.4 Truetime 模擬環境.....	14
2.4.1 Truetime Kernel 區塊.....	14
2.4.2 Truetime 無線網路區塊.....	15
2.4.3 802.15.4 (ZigBee).....	17
2.4.4 網路控制系統模擬環境介面.....	18
2.5 網路控制系統實驗平台架構.....	19
第三章 時間延遲討論.....	22
3.1 時間延遲的影響.....	22

3.2	時間延遲之量測與產生.....	24
3.2.1	Zigbee 網路的時間延遲.....	24
3.2.2	網路系統時間延遲量測.....	25
3.2.3	固定時間延遲的產生.....	26
3.3	時間延遲補償.....	27
3.3.1	完美延遲補償機制(Perfect Delay Compensation, PDC).....	27
3.3.2	PDC 模擬結果.....	29
3.3.3	PDC 實驗結果.....	31
第四章	資料遺失討論.....	34
4.1	資料遺失介紹.....	34
4.2	即時轉移機率估測器.....	35
4.2.1	資料遺失分布的影響.....	35
4.2.2	Two-state Markov Chain 網路模型.....	37
4.2.3	即時轉移機率估測器(Transition Probability Estimator).....	40
4.3	訊息估測與最小平方估測器.....	42
4.4	智慧型訊息估測器(Intelligent Message Estimator, IME).....	45
4.4.1	四種不同切換策略的 IME 架構.....	47
4.4.2	IME 的模擬結果.....	49
4.4.3	IME 的實驗結果.....	51
4.5	各種資料遺失補償架構結果比較.....	56
第五章	資料預測控制器.....	58
5.1	預測控制器介紹.....	58
5.2	資料預測控制器(Data-based Predictive Controller, DPC).....	60
5.2.1	系統輸入輸出模型推導.....	61
5.2.2	Interaction Matrix 的存在條件.....	65

5.2.3	資料預測控制器公式推導 .....	66
5.2.4	控制器參數設計 .....	68
5.3	資料預測控制器實現.....	69
5.4	DPC 應用於網路資料遺失補償.....	73
5.5	DPC 與 IME 資料遺失補償效能比較 .....	75
5.5.1	不同命令訊號下的資料遺失補償 .....	75
5.5.2	DPC 於網路資料遺失補償的缺點 .....	80
5.6	結合 DPC 與 IME 應用於資料遺失補償 .....	91
第六章	時間延遲與資料遺失綜合討論.....	97
6.1	時間延遲與資料遺失關係.....	97
6.2	不同資料遺失率與時間延遲模擬於網路控制系統中.....	98
6.3	時間延遲與資料遺失補償實驗.....	106
6.3.1	遠端控制實驗 .....	106
6.3.2	Zigbee 干擾實驗.....	107
第七章	結論與未來發展.....	114
7.1	結論.....	114
7.2	未來發展.....	115
參考文獻	.....	116



## 圖目錄

圖 2-1 IEEE 802.15.4 操作頻帶示意圖.....	7
圖 2-2 同步模式 (Beacon Mode) 資料傳送行為.....	8
圖 2-3 非同步模式 (Non-Beacon Mode) 資料傳送行為.....	8
圖 2-4 ZigBee 網路拓撲架構.....	10
圖 2-5 Jennic JN5121 微控制器.....	10
圖 2-6 JN5121 周邊介面.....	11
圖 2-7 Truetime Kernel 以分割片段方式執行使用者定義程式碼.....	15
圖 2-8 Truetime 模擬封包傳輸示意圖.....	15
圖 2-9 Truetime 無線網路區塊對話框.....	16
圖 2-10 Truetime 模擬網路控制系統介面.....	18
圖 2-11 網路控制實驗架構.....	19
圖 2-12 DSP 端實驗平台圖.....	20
圖 2-13 Zigbee 感測器網路示意圖.....	21
圖 2-14 遠端網路控制實驗架構.....	21
圖 3-1 網路控制系統方塊圖.....	22
圖 3-2 時間延遲對系統影響圖(delay time = 20 ms).....	23
圖 3-3 時間延遲對系統影響圖(delay time = 100 ms).....	23
圖 3-4 取樣頻率為 20ms 時 Zigbee 網路控制系統響應圖.....	24
圖 3-5 取樣頻率為 50ms 時 Zigbee 網路控制系統響應圖.....	25
圖 3-6 延遲時間計算.....	26
圖 3-7 回授端加 Buffer 示意圖.....	27
圖 3-8 PDC 應用於網路控制系統.....	28
圖 3-9 未加入 PDC 的系統響應.....	29
圖 3-10 PDC 加入前後的步階響應比較.....	30

圖 3-11 PDC 加入前後的弦波響應比較.....	30
圖 3-12 遠端網路控制系統響應.....	31
圖 3-13 遠端控制的時間延遲.....	32
圖 3-14 加入 PDC 後的遠端控制響應.....	32
圖 4-1 (a)事件觸發(event-trigger)示意圖 (b)時間觸發(time-trigger)示意圖.....	34
圖 4-2 資料遺失示意圖.....	35
圖 4-3 (a)分散式與 (b)集中式的資料遺失 (20% 資料遺失率).....	36
圖 4-4 追跡誤差於(a)分散式 (b)集中式資料遺失 (20% 資料遺失率).....	37
圖 4-5 Two-state Markov chain 網路模型.....	38
圖 4-6 在 $\rho_{D,D} = 0.5$ 的情況下即時計算局部的 transition probability $\hat{\rho}_{D,D}(k)$ .....	42
圖 4-7 不同估測方式的示意圖.....	43
圖 4-8 IME 架構圖.....	47
圖 4-9 IME 應用於網路控制系統補償架構圖.....	48
圖 4-10 網路控制系統模擬之弦波命令.....	49
圖 4-11 one-delay 於 20% 資料遺失率補償之模擬結果.....	49
圖 4-12 IME 於 20% 資料遺失率補償之模擬結果.....	50
圖 4-13 IME_1 與 one-delay 於不同資料遺失率補償之模擬結果比較(a)20% (b)40% (c)60%.....	51
圖 4-14 one-delay 於 20% 資料遺失率補償之實驗結果.....	52
圖 4-15 IME 於 20% 資料遺失率補償之實驗結果.....	53
圖 4-16 One-delay 於 40% 資料遺失率補償之實驗結果.....	53
圖 4-17 IME 於 40% 資料遺失率補償之實驗結果.....	54
圖 4-18 IME_2 於 40% 資料遺失率補償之實驗結果.....	55
圖 4-19 IME_3 於 40% 資料遺失率補償之實驗結果.....	56
圖 5-1 預測控制器設計概念示意圖.....	59

圖 5-2 資料預測控制器示意圖 .....	60
圖 5-3 系統鑑別的輸入(a)與輸出(b) .....	70
圖 5-4 資料預測控制器設計於控制系統響應圖 .....	71
圖 5-5 資料預測控制器設計於不同系統與不同命令之模擬結果 (a)受控體轉移 函數為 $6s + 6$ (b)受控體轉移函數為 $1s^2 + 5s + 4$ .....	72
圖 5-6 資料預測控制器不同 Q 值設計的模擬結果(a)Q=10 (b)Q=1000 (c)Q=1012 .....	73
圖 5-7 DPC 應用於網路控制系統補償策略圖 .....	74
圖 5-8 DPC 應用於封包遺失補償模擬圖(40%資料遺失率) .....	74
圖 5-9 DPC 應用於網路控制系統補償架構圖 .....	75
圖 5-10:不同資料遺失率下兩種補償方式的弦波追跡誤差(a)(b)(c)DPC (d)(e)(f)IME .....	76
圖 5-11 方波輸入訊號 .....	77
圖 5-12 三角波輸入訊號 .....	77
圖 5-13 不同資料遺失率下兩種補償方式的方波追跡誤差(a)(b)(c)DPC (d)(e)(f)IME .....	78
圖 5-14 不同資料遺失率下兩種補償方式的三角波追跡誤差(a)(b)(c)DPC (d)(e)(f)IME .....	79
圖 5-15 DPC 權重函數 Q=108時於不同資料遺失率下的補償響應圖(a)0% (b)30% (c)50% .....	82
圖 5-16 在受控體端加入 20% model 誤差情況下以 IME 補償資料遺失響應圖(a) 無資料遺失 (b)40%資料遺失率.....	83
圖 5-17 在受控體端加入 20% model 誤差情況下以 DPC 補償資料遺失響應圖(a) 無資料遺失 (b)40%資料遺失率.....	84
圖 5-18 在受控體端加入 10 倍增益的情況下以 IME 補償資料遺失響應圖(a)無	

資料遺失 (b)40% 資料遺失率.....	85
圖 5-19 在受控體端加入 10 倍增益的情況下以 DPC 補償資料遺失響應圖(a)無資料遺失 (b)40% 資料遺失率.....	86
圖 5-20 在受控體端加入 0.1 倍增益的情況下以 DPC 補償資料遺失步階響應圖(a)無資料遺失 (b)40% 資料遺失率.....	87
圖 5-21 在受控體端加入 0.1 倍增益的情況下以 DPC 補償資料遺失弦波響應圖(a)無資料遺失 (b)40% 資料遺失率.....	88
圖 5-22 不同時間延遲下的 DPC 補償響應圖(a)無延遲 (b)50ms (c)200ms.....	90
圖 5-23 IME 結合 DPC 應用於網路控制系統補償架構圖 .....	92
圖 5-24 IME_1D+DPC 與 one-delay 於 20% 資料遺失率追跡誤差比較實驗圖 ...	93
圖 5-25 IME_1D+DPC 與 one-delay 於 40% 資料遺失率追跡誤差比較實驗圖 ...	93
圖 5-26 IME+DPC 與 one-delay 於 40% 資料遺失率追跡誤差比較實驗圖 .....	94
圖 5-27 IME+DPC 與 one-delay 於 20% 資料遺失率追跡誤差比較模擬圖 .....	95
圖 5-28 IME+DPC 與 one-delay 於 40% 資料遺失率追跡誤差比較模擬圖 .....	95
圖 5-29 IME+DPC 與 one-delay 於 60% 資料遺失率追跡誤差比較模擬圖 .....	95
圖 6-1 時間延遲對於資料遺失的影響 .....	97
圖 6-2 DPC 與 IME 與 PDC 整合的系統架構圖 .....	98
圖 6-3 無資料遺失與時間延遲的追跡誤差 .....	99
圖 6-4 20% 資料遺失率與 50ms 時間延遲下的追跡誤差圖(a)無補償 (b) IME+DPC+PDC 補償 .....	100
圖 6-5 IME+DPC+ PDC 架構於 20% 資料遺失率與 50ms 時間延遲下的系統響應圖.....	101
圖 6-6 50% 資料遺失率與 50ms 時間延遲下的追跡誤差圖 (a)無補償 (b) IME+DPC+ PDC 補償 .....	102
圖 6-7 IME+DPC+ PDC 架構於 50% 資料遺失率與 50ms 時間延遲下的系統響應	

圖.....	103
圖 6-8 20%資料遺失率與 50ms 時間延遲下的響應圖 (a)無補償 (b)IME+DPC+ PDC 補償.....	104
圖 6-9 50%資料遺失率與 50ms 時間延遲下的響應圖 (a)無補償 (b)IME+DPC+ PDC 補償.....	106
圖 6-10 one-delay+PDC 於資料遺失率 40%之遠端網路控制系統的追跡誤差圖 .....	107
圖 6-11 IME+DPC+ PDC 於資料遺失率 40%的遠端網路控制系統的追跡誤差圖 .....	107
圖 6-12 WiFi 與 Zigbee 共用頻帶示意圖.....	108
圖 6-13 存在 WiFi 干擾的網路控制系統示意圖.....	109
圖 6-14 300kb/s 流量的 WiFi 訊號干擾下網路控制系統之延遲時間.....	110
圖 6-15 以 one-delay 估測器補償干擾因素的追跡誤差圖.....	110
圖 6-16 以 IME+DPC 估測器補償干擾因素的追跡誤差圖.....	111
圖 6-17 以 IME+DPC+ PDC 補償干擾因素的追跡誤差圖.....	111
圖 6-18 於 30ms 取樣週期在 300kb/s 干擾下補償之追跡誤差圖(a)以 one-delay 補 償 (b)以 IME+DPC 架構補償 (c)以 IME+DPC+ PDC 架構補償.....	113

## 表目錄

表 4.1 不同資料遺失補償架構之模擬結果比較.....	56
表 4.2 不同資料遺失補償架構之實驗結果比較.....	57
表 5.1 IME+DPC 與其他資料遺失補償架構之模擬結果比較.....	96
表 5.2 IME+DPC 與其他資料遺失補償架構之實驗結果比較.....	96



# 第一章 緒論

## 1.1 研究動機與目的

近年來將運算、通訊和控制單元，透過網路作整合，以方便系統的建立與管理，進而提高系統的整合度與便利性，乃是控制系統的一大趨勢。也因為網路化的日益普及化，各種應用於不同領域的網路協定也因應而生，常見於工業應用的協定如：WorldFIP、Profibus、SERCOS、CAN 等；其他如 Ethernet、無線網路或是感測器網路 Zigbee 等協定，也廣泛使用於各種應用上，特別是無線網路已是未來網路通訊的發展重點。

利用網路做為運算、通訊和控制單元的媒介，可以在遙遠的距離內控制與監控特定的系統，如果使用無線網路，更可以節省佈線的複雜度與成本，且具有高機動性，所以網路控制系統具有節省配線、系統擴充性增加、減少安裝和維護成本、方便監控等優點。但也有許多所要面臨的問題，首先便是網路傳輸所帶來的影響(time delay)、網路引起的訊息傳遞延遲與訊息遺失(data dropout)、訊息排程、節點間的時脈不同步。這些種種問題，任何一個均可能讓控制效能降低，甚至讓受控體變成不可控的狀況，如何去克服網路傳輸所引起的問題並且發揮遠端網路控制的優點，變成為研究的重點之一。

為了研究這些問題對網路控制系統的影響，本研究以時間觸發(time trigger)作為網路封包傳輸依據，建立以 Zigbee 無線網路為通訊方式的網路控制系統，針對馬達作位置控制實驗，分析探討網路層面對控制系統效能的影響。根據模擬與實驗結果，資料遺失與時間延遲的存在會導致控制效能低落甚至造成系統不穩定，因此本研究特別針對這兩方面提出了的解決辦法，並且結合控制器提出了 model-free 的控制架構，建立一套完善且可靠的網路控制系統。

## 1.2 研究背景與發展概況

隨著網路的蓬勃發展，加上近年來最熱門的領域-無線網路的研究，讓許多

學者與研究人員紛紛投入網路通訊這領域，讓網路概念的延伸與應用處處可見，更可以將網路的概念應用在控制理論上，因此網路控制系統的想法便因應而生，而針對網路控制系統所相關的研究課題與問題也紛紛提出[1]。一般網路控制系統，雖然有網路化的優點，但往往需要犧牲一點控制效能，以下針對一些足以影響控制系統效能的問題，整理後可歸類成下列幾項：

- 網路傳輸所造成的延遲(time delay)

一般的理想控制系統，訊號在傳輸時，其傳輸時所花費的時間極小，通常可以忽略不計，但當控制系統利用網路做為傳輸介面時，由於網路先天上的頻寬限制與網路通訊協定限制下，造成訊號有一段不可忽略的傳輸延遲時間[2,3]。所以在系統的等效模型中，需加入一個時間延遲項，而此時間延遲會造成系統的 phase margin 下降，導致系統不穩定，現在在此方面，已有相關理論解決延遲時間對系統造成的不良影響[4,5]。

- 網路引起的訊息遺失(data dropout)

網路在傳輸的過程中，有可能會發生封包遺失的情形，一般來說，大部分的網路通訊協定都有防止與解決封包遺失的機制，通常是避免資料封包碰撞 (collision) 以及封包遺失後等候節點再一次重傳，資料封包的遺失，會造成控制系統的命令遺失，等候節點再一次重傳的時間會增加延遲時間，有人提出使用泰勒展開法來估測遺失的命令，使得系統不會因為命令遺失而造成系統控制效果不佳[6]。

- 訊息排程(network scheduling)

在網路排程方面，有人提出 adjustable priority scheme 的排程方法[7,8]，藉由累積 CAN 網路上訊息一段時間的傳輸狀況後，逐漸調整訊息優先權，經由此



方法改變網路的排成可以有效的提昇網路傳輸的效率。

- 節點間的時脈不同步(jitter)

在系統取樣週期較小的架構下，時脈漂移造成節點間的時脈不同步，將導致大量的系統資料遺失量，由此可知，時脈同步對於網路系統是不可忽略的因素。關於網路時脈同步的方式，可由系統架構、網路協定等層面解決[9]，以系統架構來說，可利用額外的硬體接線作為同步線路，以此作為節點執行的基準；在網路協定層面，以 CAN bus 為例，透過同步物件的傳送，提供整體網路的基準時脈。這兩類作法各有其優缺點，前者為獨立於原本網路架構之外的作法，可免除網路傳輸的干擾，且經由硬體線路可達到極高的準確度，但正因為是獨立於網路之外，所以必須在系統建構時加以考量，若要應用於已建立的系統之上，就必須重新規劃系統架構；後者則是基於原先的網路協定所衍生的方式，主要優勢在於可經由軟體規範達成而不需改變系統架構，所以，實做上的可行性較高且提高系統維護的便利性，但由於是透過網路封包傳遞的方式，傳輸狀況的影響就無法忽略，傳輸速率、延遲時間和封包碰撞等等因素，都可能影響其效果而必須納入考量，以 CAN bus 為例，由於封包碰撞可由 CAN 的重傳機制避免，而將主要考量放在傳輸延遲上，因此，賦予同步物件極高的網路優先權，降低傳輸的等待時間，便於同步物件順利傳送，以達到較好的同步效果[10]。

- 取樣週期(sampling period)

在數位控制系統下，控制效能會隨著取樣週期變小而變好，但將網路加入控制系統當中，取樣週期則不可以小於某個極限值，如果小於該極限值，反而控制效能會下降[11]，主要原因是因為當取樣週期變小的情況下，所要傳輸的資料量變多，而當網路的頻寬不足以傳輸這些資料量時，便會造成網路塞車、負載量增大，所需要傳輸的資料，必須等待網路允許通過，方可傳輸，造成傳輸時間延遲，甚至資料損失，導致控制效能下降。有人提出利用動態調整取樣頻率的方法

[12,13]，當網路負載大時，取樣週期下降，減少傳輸資料量，有人提出以動態調整取樣頻率為基礎，其適應性控制器的設計[14]。

- QoS(quality of service)[15]

此因素會直接影響到網路控制系統的控制效能，其特性主要受訊息傳輸延遲時間、訊息遺失率、訊息來源的正確性所影響，並非一個可量化的參數。

### 1.3 問題描述

綜合上述各種網路影響控制系統效能的因素，本研究將針對以下三個重點問題作討論

1. 在網路控制系統中，當網路傳輸延遲超過系統取樣週期時，會導致網路資料封包遺失，系統相位落後以及系統不穩定等情形。
2. 資料封包的遺失，造成系統控制命令的遺失，系統沒有收到命令，導致空白取樣，使得控制效能下降。
3. 網路控制系統中控制器的設計，資料遺失與時間延遲的補償方法，一般皆為 model-based 的設計，而無法廣泛應用於不同的網路系中。

### 1.4 研究方法

1. 在處理網路時間延遲的方面，以 perfect delay compensation (PDC)將網路延遲時間對系統的影響化為單純的時間延遲，避免系統因網路延遲影響而導致系統響應發散。
2. 在改善資料封包遺失造成控制命令遺失方面，使用 intelligent message estimator (IME)以及時量測的封包遺失機率配合不同的最小平方估測器補償當前遺失的資料。
3. 對於 IME 可能造成系統不穩定的方面使用 data-based predictive controller (DPC)，透過移動時域控制，即時計算最佳的控制訊號，並以此 控制訊號作為 IME 補償切換機制的指標。

4. 提出 IME+DPC+PDC 的架構，同時解決網路資料遺失與時間延遲對系統產生的影響。

## 1.5 論文架構

本論文共分七章，第一章旨在說明研究動機與目的、研究背景與發展概況回顧、敘述所面臨的相關問題、本論文的研究方法、以及論文架構。第二章先對 CAN、Zigbee 與 Ethernet 等網路通訊協定加以說明，之後介紹擬真的網路模擬環境 Truetime 以及實驗設備平台。第三章介紹時間延遲的影響，然後以完全延遲補償架構(PDC)解決網路延遲對系統的影響。第四章介紹網路資料遺失的問題，資料遺失的分布與網路訊息的狀態，以及訊息估測器的說明而最後以智慧型訊息估測器(IME)解決資料遺失所帶來的問題。第五章介紹資料預測控制器(DPC)，實現 DPC 並將之應用於網路控制系統中，以及結合 IME 達到自動調整資料遺失補償方法。第六章討論網路控制系統中資料遺失與時間延遲同時發生的情形，提出以 DPC 結合 IME 與 PDC 的補償架構，並以實驗和模擬的方式驗證其可行性。第七章則為本論文的結論以及未來的發展方向。

## 第二章 實驗平台與模擬環境

### 2.1 CAN 網路協定

CAN 全名為 Controller Area Network，起源於 1980 年代後期，由德國 Robert Bosch 公司所制定的一種具有高度安全且支援即時分散式控制的通訊協定，最高傳輸速度可達 1M bit per second (bps)。到 1990 年代其通訊協定逐漸成型，相關的 IC 元件紛紛被推出，才被廣泛地應用於工業界。CAN 原先是應用在汽車電子系統，用來連接汽車內防鎖死煞車系統(ABS)或是引擎控制與感測單元等的電子元件，以簡化車內複雜的硬體配線；但由於 CAN 提供可靠快速的資料傳遞，適合用在即時系統(real-time system)且價格低廉，因此，發展至今也被廣泛運用在各類的控制系統上。目前 CAN 已成為國際標準規格 (ISO11898)，CAN 在場域匯流排(field bus)的應用已有 Honeywell 的 SDS 及 Allen-Bradley 的 DeviceNet。

CAN 根據其傳輸協定，具有以下特點：

- 資料訊息具有優先權 (priority)
- 優先權的仲裁 (arbitration) 為非破壞性
- 保證延遲時間
- 彈性的架構
- 採用廣播的方式 (multicast)，並藉由傳輸的訊息做時序同步的動作
- 任一節點皆可主動發出訊息 (multimaster)
- 未傳送成功的訊息會自動重新傳送
- 錯誤檢查

區分暫時性錯誤和永久性錯誤並且自動關閉有問題的節點

## 2.2 Zigbee 網路協定

IEEE 802.15.4/ZigBee 的命名，源自於蜜蜂在發現花粉時，展現如同 Zigzag (Z 字形) 形狀的舞蹈，並與 Bee (蜜蜂)，合成為 ZigBee，形象地表述了 ZigBee 產品是像蜜蜂一樣小而敏捷；是一種相當先進新穎的短距離傳輸技術標準，由家用無線通訊規格 HomeRF 聯盟中所分出來的技術，IEEE 802.15.4 所定義的低傳輸速率之無線個人區域網路，是一種網路結構簡單且低價格的通訊網路，適合在電力供應有限且只需簡單運算之應用的裝置中實現，其主要的特色在於低傳輸速率 (250 Kbps)、短距離傳輸(一般約為 50-100m，依耗電量的不同，可提升至 300m)、可靠的資料傳輸、低耗電量、價格低廉及容易安裝。

IEEE 802.15.4 實體層 (PHY) 採用 Direct Sequence Spread Spectrum (DSSS) 技術，這 DSSS 技術主要的優點在於可以降低類比電路在開發上的複雜度與成本，其傳輸速率介於 250kbps 至 20kbps 之間，看所用頻段而決定，頻段的使用分別是 2.4GHz 的 ISM 頻段、美國開放的 915MHz 頻段，以及歐洲開放的 868MHz 頻段，而不同頻段可使用的通道分別是 16、10、1 個。圖 2-1 為 IEEE 802.15.4 操作頻帶示意圖。

### IEEE 802.15.4 PHY Overview

Frequency	Band	Coverage	Data	# of Channels	Rx Sensitivity	Modulation
2.4GHz	ISM	Worldwide	250kbps	16	-85dBm	O-QPSK
868MHz		Europe	20kbps	1	-92dBm	BPSK
915MHz	ISM	Americas	40kbps	10	-92dBm	BPSK

圖 2-1 IEEE 802.15.4 操作頻帶示意圖

IEEE 802.15.4 的媒體存取層 (MAC)，定義了兩種傳輸協定，一種是同步模式 (Beacon Mode)，另外一種則是非同步模式 (Non-Beacon Mode)。

同步模式則是 IEEE 802.15.4 比較特別的規格，圖 2-2 為同步模式 (Beacon Mode) 資料傳送行為，利用信標的星型拓撲網路中，為於中心的父節點可以和其從屬的多個子節點間進行資料交換，但是子節點和子節點間不能進行資料交換。

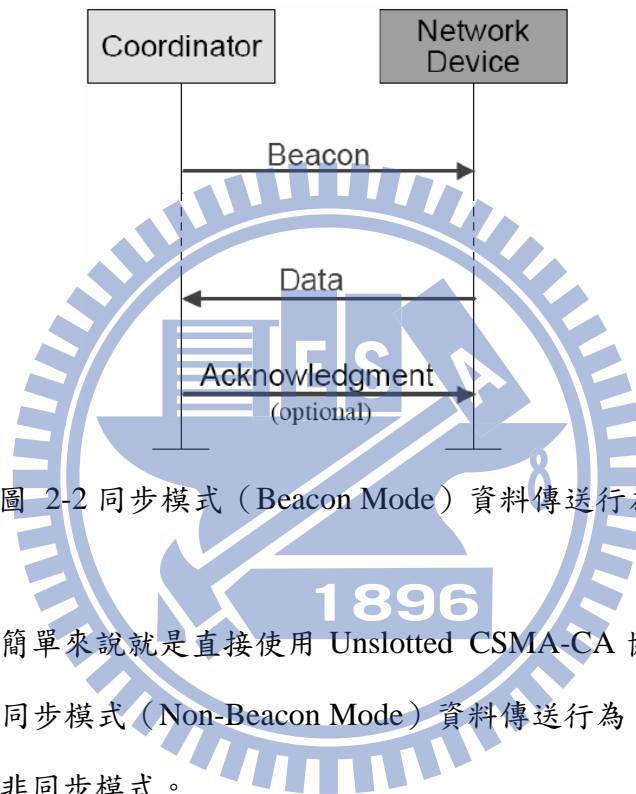


圖 2-2 同步模式 (Beacon Mode) 資料傳送行為

非同步模式簡單來說就是直接使用 Unslotted CSMA-CA 協定，避免傳輸碰撞。圖 2-3 為非同步模式 (Non-Beacon Mode) 資料傳送行為，本論文所使用的資料傳輸方式為非同步模式。

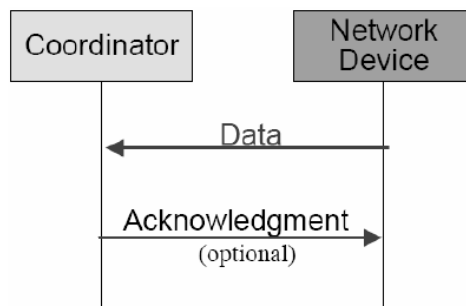


圖 2-3 非同步模式 (Non-Beacon Mode) 資料傳送行為

基於 IEEE 802.15.4 的規格，ZigBee Alliance 定義了網路層 (NWK)、應用層 (AP) 與安全層 (security)。網路層最重要的工作，在於負責網路機制的建立與管理，同時具有自我組態路由路徑 (self configure)，以及自我修復路由路徑 (self healing) 的功能。在網路層中，ZigBee 定義了三種角色，第一個是網路協調者 (WPAN coordinator)，負責網路的建立以及網路位置的分配；第二個是路由器 (router)，主要負責找尋、建立、以及修復資料封包路由路徑，並負責轉送資料封包，同時也具有網路位置的分配的能力；最後一個是末端裝置，只能選擇加入別人已經形成的網路，可以收送資料，但不能幫忙轉送封包。網路功能是 ZigBee 最重要的一個特點，也是與其他 WPAN 標準最不一樣的地方；以規格來說，網路層以 16bits 的資料長度來管理網路位置，理想上在同一個 WPAN，可以存在 65536 個 ZigBee 裝置，彼此透過多重跳點 (multi hop) 的方式，傳遞資訊，圖 2-4 為 ZigBee 網路拓撲架構。為了在省電、複雜度、穩定性與實現難易度等因素取得一個平衡點，網路層的路由演算法，使用了下列三種方式：

1. AODV 演算法：建立隨易網路的拓撲架構，在 ZigBee 規格中稱為 Table Routing。
2. Motorola Cluster-tree 演算法：建立樹狀的拓撲架構，在 ZigBee 規格中稱為 Tree Routing。
3. 利用廣播的方式傳遞資料

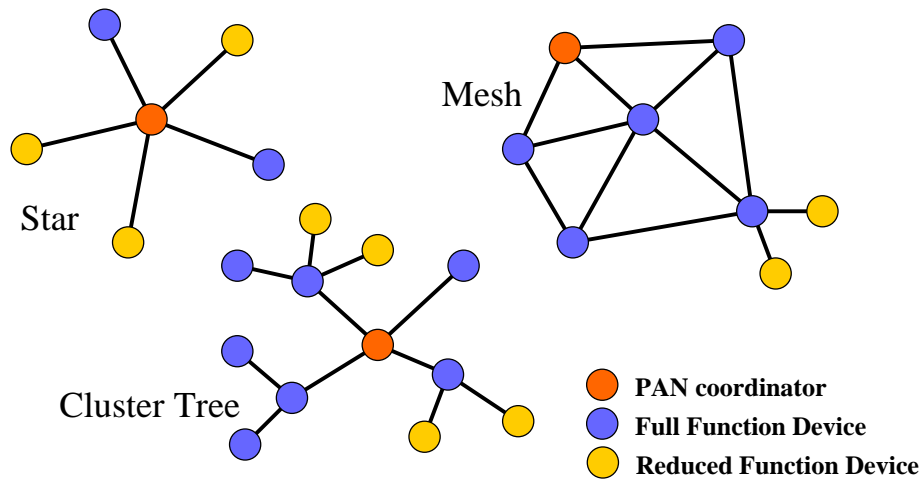


圖 2-4 ZigBee 網路拓撲架構

本論文採用英商 Jennic 公司所研發的 JN5121 wireless microcontroller 做為控制介面，是世界上第一顆整合 RF 與 MCU 之 SOC 架構的低功耗 802.15.4 / ZigBee 晶片，如圖 2-5 所示。以下分別針對微控制器的周邊介面與 ZigBee 無線通訊作介紹。

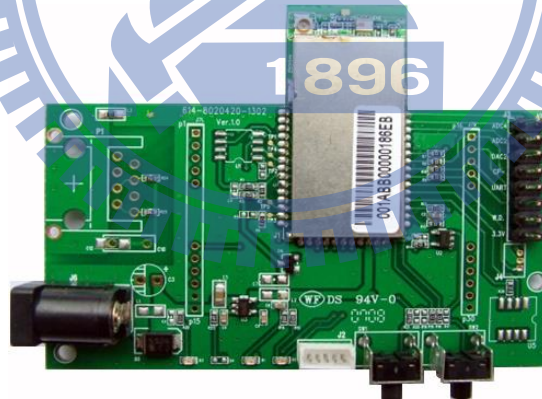


圖 2-5 Jennic JN5121 微控制器

JN5121 微控制器擁有 16 MHz 的處理能力，其核心採用 32 位元定點運算，記憶體容量方面，JN5121 擁有 64K ROM 與 96K RAM，且提供豐富的周邊擴充介面(圖 2-6)，如下所示：

- 兩組 UARTs



- 兩組可程式化 Timer/Counters
- 兩組可程式化 Sleep Timers 與一組 Tick Timer
- 兩組串列介面 (與 SMBus 及 I<sup>2</sup>C 相容)
- Slave SPI port (與 digital I/O 共用)
- 21 個數位 I/O lines (與 UARTs, timers 以及 SPI 共用)
- 四組 12-bit, 100Kbps 類比/數位轉換器
- 兩組 11-bit 數位/類比轉換器
- 可程式化比較器

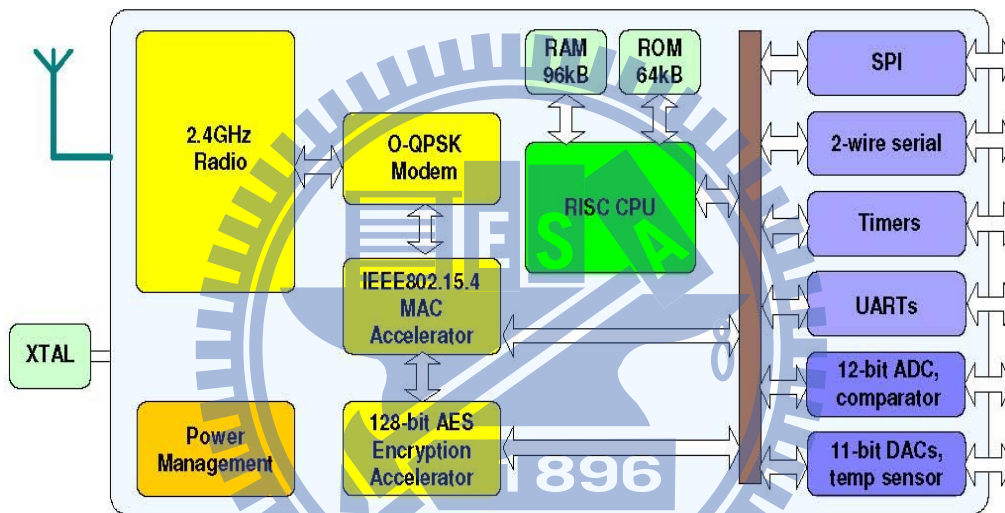


圖 2-6 JN5121 周邊介面

### 2.3 乙太網路(Ethernet)協定

乙太網路(Ethernet)源自於 Xerox 公司的區域網路系統，一開始速度只有 2.49Mbps，僅在 Xerox 公司內部使用，隨後由 DEC、Intel 和 Xerox 三家公司共同參與標準之改進與擴展，並且發表了 Ethernet Version 2(EV2)規格，將網路頻寬提升到 10Mbps，之後由 IEEE 根據 EV2 的內容，在 1983 年通過了 802.3 CSMA/CD 規格，從此成為使用最廣泛的區域網路標準之一。

乙太網路使用 CSMA/CD(載波感應多重存取及衝突偵測)方式來存取介質上

面的資料，在 CSMA/CD 裡，某一站要送資料前必須檢查是否別有別的站正在傳送資料，假如沒有其他站要傳送資料，則這一監聽站就可以將資料傳送到網路上面。在乙太網路上面可能會遇到多站同時要傳遞資料的情況，此情況會造成資料的碰撞。因此所有的站必須持續監聽網路及偵測出任何可能的碰撞。若產生碰撞，所有站放棄所接收到的資料，原先傳送的各站退後等待一段時間再重送。為了降低再次碰撞的可能，每個傳送站各自產生一個亂數來決定要等待多久時間才能傳送。

乙太網路在實際使用上可以為匯流排或是星形方式連接，所有乙太網路的組態在邏輯上是一種匯流排。每個訊框(Frame)可傳送資料到網路線上的各站，但只有被定址到的那個站才能讀到訊框。在實現上目前使用最多的實現方式為 10BASE-T(雙絞線乙太網路)，10BASE-T 是使用星形方式連接各站，速度為 10Mbps，最長傳輸距離為 100 公尺。除了 10BASE-T 之外，還有 10BASE5(粗纜乙太網路)和 10BASE2(細纜乙太網路)，上述兩種均使用匯流排傳輸方式做連接。

目前使用上比較普遍的是快速乙太網路(Fast Ethernet)，快速乙太網路是比乙太網路更新一點的版本，最高速率為 100Mbps 跟乙太網路比較起來，在訊框上沒有做任何改變，唯一改變的是傳輸速度和碰撞區間的大小，傳輸速度提高 10 倍，因此碰撞區間也降低 10 倍。快速乙太網路和 10BASE-T 一樣採用星狀連接方式，其規範定義三種實體介質：100BASE-T4、100BASE-XF 和 100BASE-XF。

至於 Gigabit 網路速度 1000 Mbps，就是 Gigabit Ethernet，只是 Gigabit Ethernet 的網路線就需要更加的精良。當傳輸速度增加時，線材的電磁效應相互干擾會增強，因此在網路線的製作時就得需要特別注意線材的質料以及內部線蕊之間的纏繞情況配置等，以使電子流之間的電磁干擾降到最小，才能使傳輸速度提升到應有的 Gigabit。因此現在高速網路的傳輸介質逐漸的被光纖所取

代。

### 2.3.1 TCP/IP 通訊協定

TCP/IP 主要是用來連結網路上的電腦主機，作為網路傳輸資料的標準協定，是一套完整的通訊協定，他的名稱來自於其中兩個最重要的協定：傳輸控制協定 (transmission control protocol, TCP) 及網際網路協定 (internet protocol, IP)，除了這兩種協定外，TCP/IP 還包含了其他協定，但最重要的就是 TCP 和 IP 兩種協定。

TCP/IP 是一種階層式的協定，這裡所謂的階層式是指每一個較高的協定是由一個或多個較低分層協定所支援。這裡主要介紹網路層的網際網路協定 (IP) 和傳輸層的傳輸控制層協定 (TCP)。

#### (a) 網際網路協定 (IP)

IP 是 TCP/IP 使用的傳輸機制，是一種非可靠性、非預接式的資料封包協定，只提供盡量傳送的服務。所謂盡量傳送是指 IP 沒有提供錯誤檢查或追蹤，IP 假設他的底層是不可靠的而盡力將資料傳送到目的地，但不保證是否傳到目的地。IP 以資料封包方式傳遞，而各個資料封包分別傳送，資料封包在傳輸過程可以經過不同的路徑，可以不按順序到達，也可以被重複，當資料封包達目的時，也沒有方法將資料封包調整回原來的順序。然而 IP 的限制不應被視為缺點，IP 提供骨幹傳輸的功能，且讓使用者自由加入所需要的功能，因而獲得較高的效率。

#### (b) 傳輸控制協定 (TCP)

TCP 提供完整的傳輸服務給所需的應用程式，在應用層與網路層之間提供應用程式和網路運作的中介服務。TCP 在傳輸層中提供了流量控制和錯誤控制的機制，TCP 使用滑動窗口 (sliding window) 的協定來做流量控制，使用回應封包、計時 (time-out)、重送的機制來做錯誤控制。

TCP 為應用程式提供連線的機制，應用程式能夠以資料流方式傳送訊

息給傳輸層，在傳送端的傳輸層負責與接收者建立一個連線後，將資料折成可傳送的單元並給予序號，再將傳送單元一個一個傳送出去。接收端的傳輸層等待這些資料的到來，然後加以檢查，並將沒有錯誤的資料以資料流的方式傳給接收端的應用程式，等全部資料完成後，TCP 要關掉剛剛使用的連線。TCP 是一種連線導向，具可靠的傳輸協定。TCP 把 IP 的服務加入連接導向及可靠性這兩種特點。

## 2.4 Truetime 模擬環境

Truetime 是基於 Matlab/Simulink 環境開發而成的模擬程式，能夠用來設計擁有 real-time kernel、網路傳輸、動態受控體等要素的網路控制系統。Truetime 由六個 Simulink 區塊建構而成，並且定義了一系列的 Matlab 函數可供使用者運用。其中包括了創建任務(task)、中斷處置、流程監視、與計時等設定模擬環境的函數，以及在程式執行時能改變參數、網路資訊傳輸、數位類比轉換等即使處理訊息的函數。

### 2.4.1 Truetime Kernel 區塊

在 Truetime 中最主要的 Simulink 區塊為 Truetime Kernel 區塊。它能夠模擬 CPU 中 real-time kernel 的工作行為。Truetime Kernel 區塊執行使用者定義的任務或是中斷處理，而這些任務或中斷可能是週期性的，也有可能是即時觸發的。例如以時間驅動的週期性網路通訊，或者使用者臨時改變的 I/O 輸入。這些任務的程式以 Matlab 的 m-file 寫成，程式碼會被進一步分解為片段的程式，如圖 2-7 的方式執行。所有的程式被執行後會等待一段時間，這段時間為模擬執行的時間，一直到等待結束，才會繼續進入下一段程式。

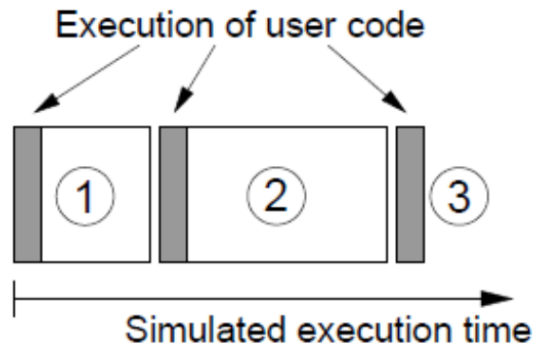


圖 2-7 Truetime Kernel 以分割片段方式執行使用者定義程式碼

## 2.4.2 Truetime 無線網路區塊

Truetime 中的無線網路區塊能夠模擬區域網路中的介質存取和封包傳輸的行為。當網路中有節點想要傳送資料時，觸發信號會被送到 Truetime 無線網路區塊，模擬封包傳送，結束後再傳送觸發信號到對應的 Truetime Kernel 區塊，如圖 2-8 所示。

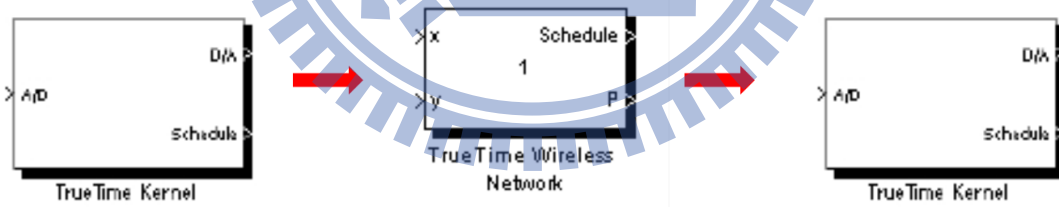


圖 2-8 Truetime 模擬封包傳輸示意圖

傳輸結束後，被發送的訊息會被排入接收端的佇列裡，等待執行。在 Truetime 傳輸模型中，考慮到了無線電訊號衰減的問題，因此可以設定節點的位置，在模擬中呈現距離遠近所造成的影響。初版的 Truetime 實作兩個無線網路協議：802.11b/g (WLAN) 和 802.15.4 (ZigBee)

在無線電訊號的模擬中考量到下面的因素：

- 全向性的天線
- 接收和發送不能同時進行
- 訊號衰減模型為 $\frac{1}{d^a}$ ，其中 d 代表距離(m)、a 代表環境因子
- 來自其他節點的干擾

無線網路的參數透過下圖 2-9 的對話欄設定。

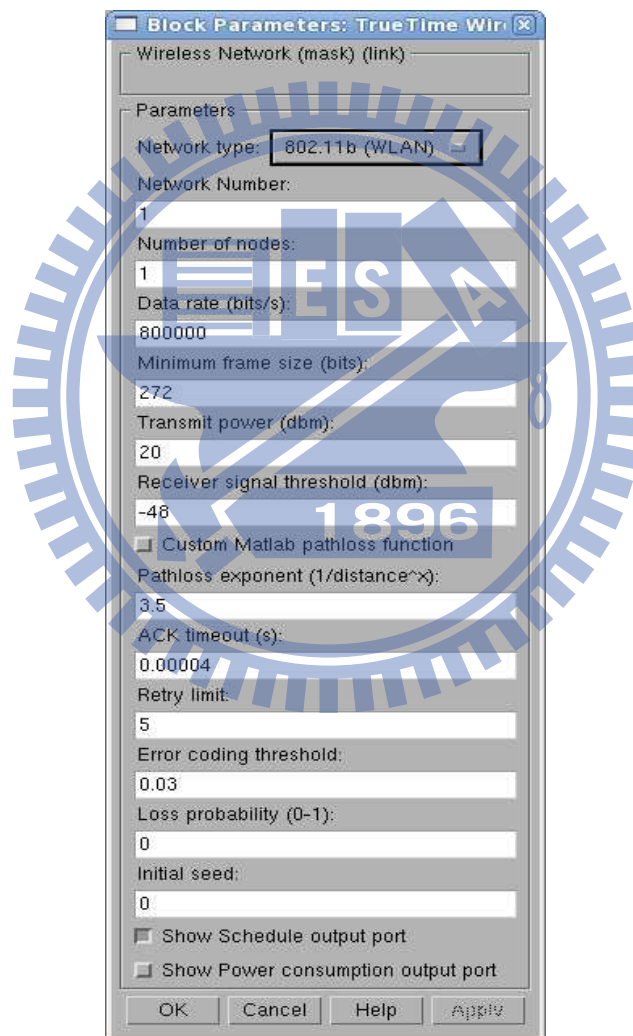


圖 2-9 Truetime 無線網路區塊對話框

### 2.4.3 802.15.4 (ZigBee)

Zigbee 的傳輸設備中包含了感測器和處理器，多個裝置之間能夠形成一個簡單的網路系統。其網路協議設計為低速、低功耗的用途。Zigbee 的介質存取層是基於 802.11b/g 的 CSMA/CA，在 Truetime 中以下面的演算法實現。

1. Initialize:

NB=0

BE=macMinBE

2. Delay for a random number of backoff periods in the interval  $[0, 2BE - 1]$

3. Is the medium idle?

if yes: send

else: goto 4

4. Update the backoff counters:

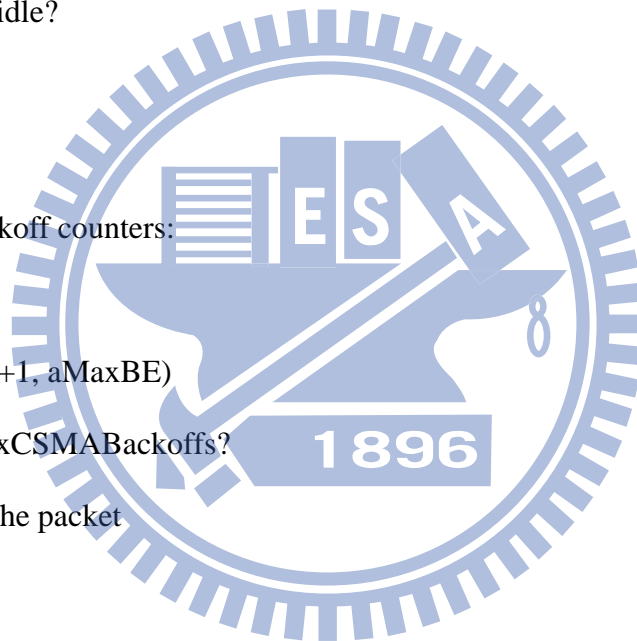
NB=Nb+1

BE=min(BE+1, aMaxBE)

5. Is  $NB > macMaxCSMABackoffs$ ?

if yes: drop the packet

else: goto 2



**NB** Backoff 次數

**BE** Backoff 階數

**macMinBE** CSMA/CA 演算法中 **BE** 的最小值。預設為 3。

在封包傳送的模擬中，若傳送封包的訊號強度大於接收端所能收到最低能量的門檻值，則封包傳送成功，與此同時，Truetime 會考慮目前的背景雜訊計算出訊號雜訊比(SNR)用來求取區塊錯誤率(BLER)。將 BLER 與封包大小納入考量，

可以得知封包裡位元的錯誤率，假如錯誤率低於某個臨界值，則封包的內容能夠被完整地重建。

假使想傳送一筆訊息到接收端節點，而目前有其他的節點在接收端節點的範圍內傳輸資料，且這些訊息的訊號強度(SNR)皆低於我們傳送的訊息，則除了我們送出的封包外這些訊息都視為碰撞。值得注意的是，發送端的節點並不知道它送出封包因為碰撞而遺失了，因此在介質存取層使用 ACK(acknowledge)機制是必須的。而且從發送端的觀點來看，無論訊息是以遭受碰撞的方式遺失或是以其他任何形式的遺失，都沒有差別。因此只要在一定的時間內(ACK timeout)沒有收到 ACK，則訊息會重新傳送，一直到重新傳送的次數達到上限(Retry limit)。

#### 2.4.4 網路控制系統模擬環境介面

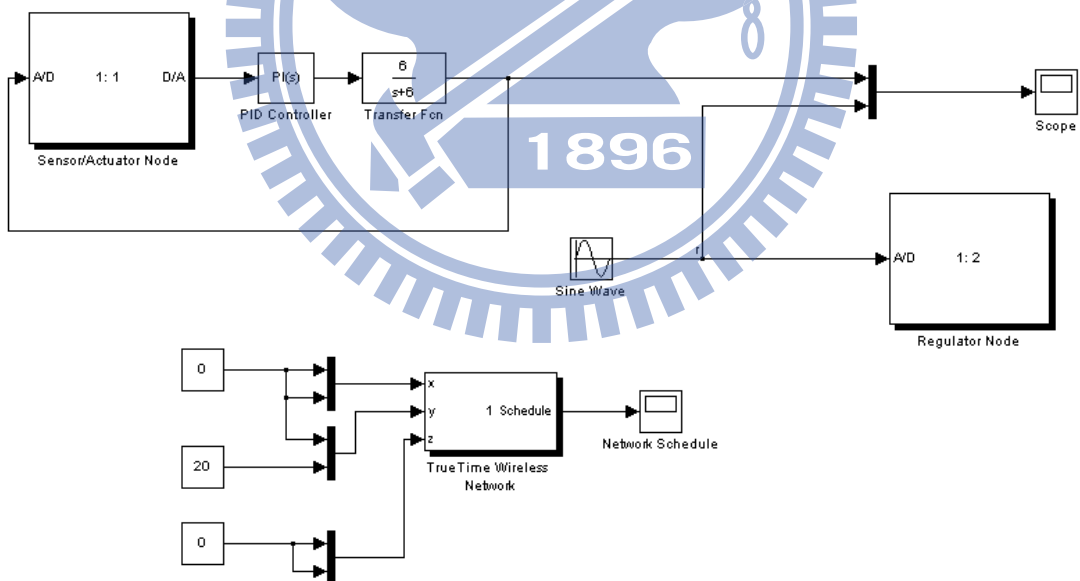


圖 2-10 Truetime 模擬網路控制系統介面

圖 2-10 為在 Truetime 環境下開發的網路控制系統模擬介面，regulator node 透過 Truetime wireless network 將網路訊息傳送給 sensor/actuator node，



sensor/actuator node 接收到資料後將命令透過控制器送入受控體，同時 sensor/actuator node 抓取受控體的響應再經由 Truetime wireless network 傳回給 regulator node。其中受控體轉移函數如式子(2.1)所示，PI 控制器的  $K_p = 4.9074$ ， $K_i = 28.9857$ ，取樣週期 50ms。

$$G(s) = \frac{6}{s+6} \quad (2.1)$$

## 2.5 網路控制系統實驗平台架構

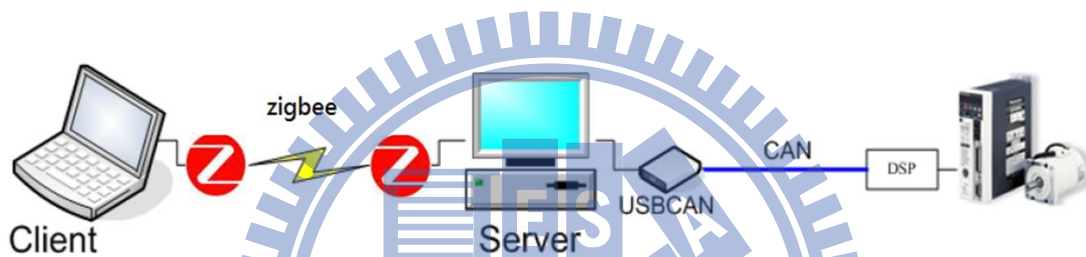


圖 2-11 網路控制實驗架構

圖 2-11 為本研究實驗的系統架構圖，此架構由三個部份所組成，第一個部份為 TI 的 F2812 DSP 和 Panasonic AC 伺服馬達所組成，馬達工作模式調整為速度模式，第二部份為一台電腦和 USB CAN 所組成的 Server 端，Server 端是連結 CAN 網路和 Zigbee 網路的溝通橋樑，具有閘道器的功用，並且具有簡單的管理功能，第三部份為負責網路控制的電腦。DSP 跟 Server 端的 USBCAN 之間的聯繫是透過 CAN 來連結，Server 端和負責遠端控制的 Client 端中間透過 Zigbee 網路來連結，整個控制迴路為位置迴路控制，在 DSP 端 PI 控制器被寫入 TI TMS320F2812 DSP 裡，其中  $K_p = 0.0002$ ， $K_i = 0.0000001$ ，馬達轉移函數如式子(2.2)，取樣時間為 50ms。

$$G(z) = \frac{5.277z}{4} \frac{14.89788z}{3} \frac{1}{z^2 - 0.848051 - 0.205173} \quad (2.2)$$

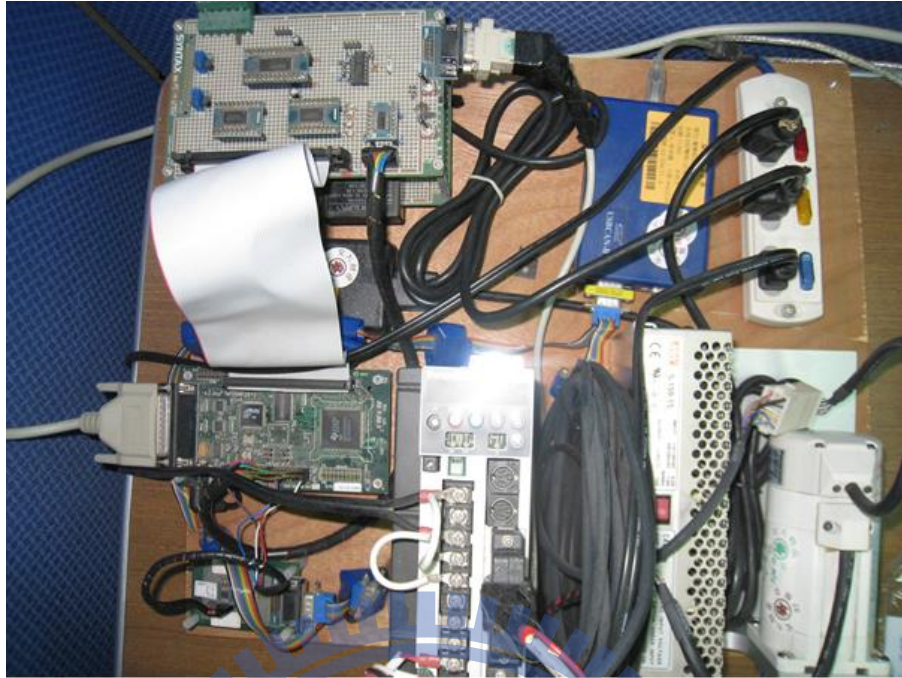


圖 2-12DSP 端實驗平台圖

當有資料要從 DSP 傳送到 Client 的控制端時，首先須要將訊息種類(Type)及傳送資料(Data)的訊息格式設定好，設定好訊息格式之後將他放到 CAN 封包的 Data 部份透過 CAN 網路傳到 Server。當 Server 接收到 CAN 送過來的封包之後，將 CAN 封包的 Data 部分解析出來，然後透過 Zigbee 以 Zigbee 網路傳送到 Client 端。Client 端收到由 Zigbee 網路送過來的封包後，分析資料格式，確認訊息的種類及訊息的數值，即完成 DSP 傳送資料到 Client 端的工作。

若要將資料由 Client 控制端傳送到 DSP，先將要傳送的訊息種類(Type)和傳送資料(Data)訊息格式設定好，放到 Zigbee 封包中，以 Zigbee 網路傳送給 Server，Server 將封包解析過後取出使用者定義的部分放入 CAN 的封包裡，以 CAN 網路將封包傳送到 DSP，即完成從 Client 端將控制命令傳送訊息到 DSP 的動作。

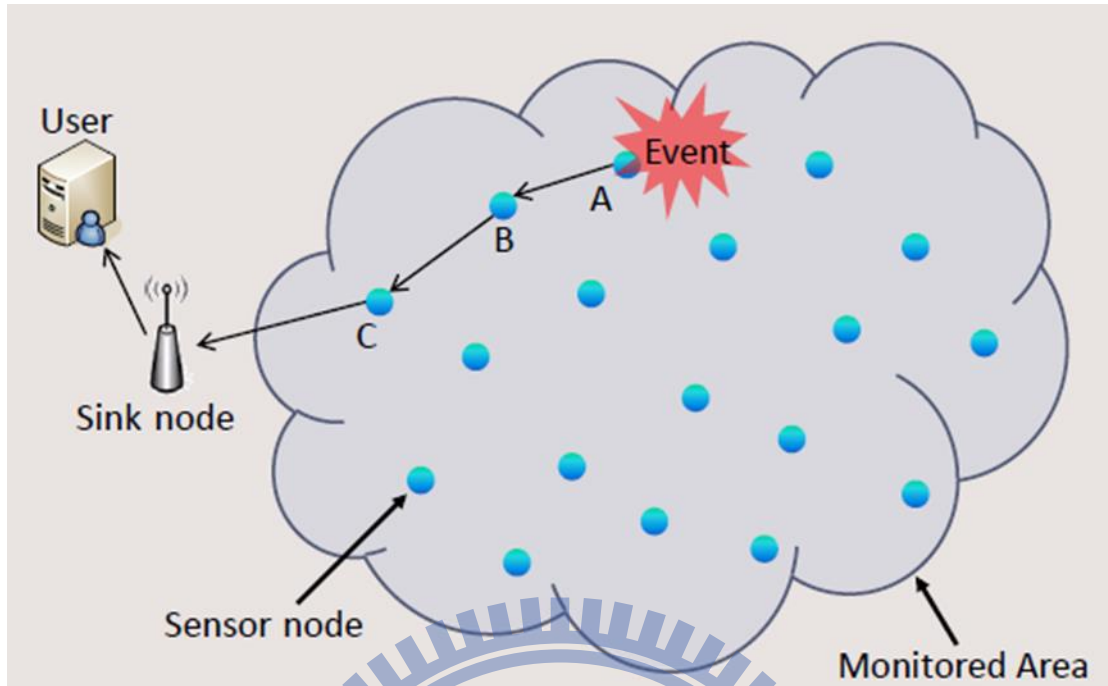


圖 2-13 Zigbee 感測器網路示意圖

對於無線感測器網路而言，當監視區域裡的 sensor 有事件發生時，訊息常常是通過數個感測器網路接續傳遞，將訊息匯集於 sink node，然後再通過一般的有線或無線網路將資料傳送給使用者端，如圖 2-13 所示。本論文為了模擬此情境，在原本的實驗架構中網路傳輸的部分加入乙太網路。使原本 Client-Server-DSP 的訊息傳遞途徑，變為 Remote-Client-Server-DSP，其中 Remote 和 Client 之間使用乙太網路傳輸，Client 和 Server 之間使用 Zigbee 網路傳輸，如圖 2-14 所示。而其中 Zigbee 網路與乙太網路訊息交換的方式與上面提到的 Zigbee 網路和 CANBUS 訊息交換方式相同，因此不再多加贅述。



圖 2-14 遠端網路控制實驗架構

## 第三章 時間延遲討論

### 3.1 時間延遲的影響

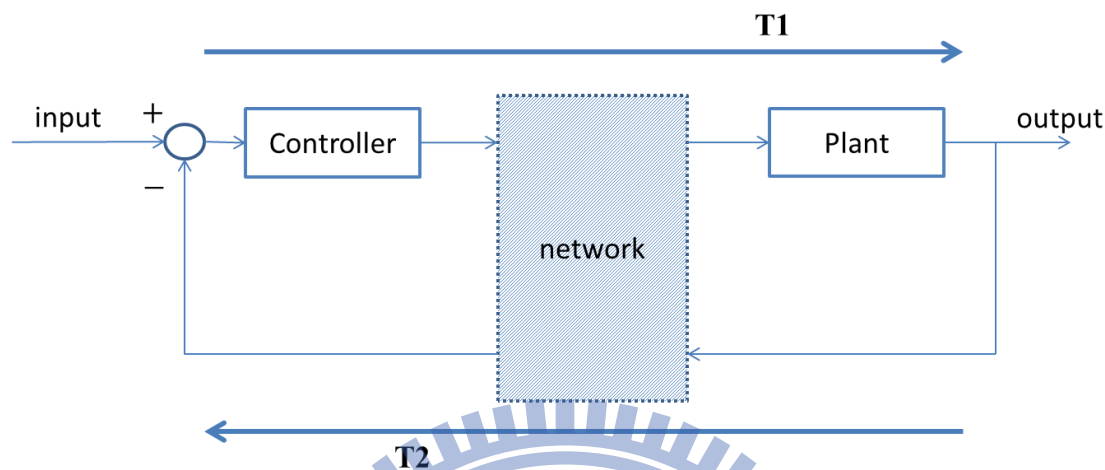


圖 3-1 網路控制系統方塊圖

圖 3-1 為一般網路控制系統方塊圖，在控制端與受控端之間的命令和回授訊號是透過網路來傳輸，而由網路來當作傳輸媒介就會產生延遲時間(delay time)。網路系統所產生的時間延遲分成兩類，一種為命令時間延遲(command time delay)是由控制端將命令傳到受控端所產生的延遲時間，另外一種則是回授時間延遲(feedback time delay)，回授時間延遲是指受控端將感測器的資料傳回控制端所產生的時間延遲。

延遲時間若小於系統的取樣週期，則我們可以將延遲時間忽略掉，但是當延遲時間大於取樣週期時，延遲時間對整個控制系統的影響，輕則會降低系統的效能，嚴重則會造成整個控制系統的不穩定。基於式子(2.1)的網路控制系統，圖 3-2 和圖 3-3 分別為模擬的網路控制系統在取樣週期為 20ms 時的響應圖，這兩張圖中虛線部份是不經過網路控制的系統響應圖，而實線部份則是經過網路控制的響應圖，圖 3-2 的延遲時間為 20 ms，此時響應圖與未受網路控制做比較，可以看出因延遲時間造成的系統震盪，若再將時間延遲加大到 100 ms，整個網路控制

系統已經發散不受控制。

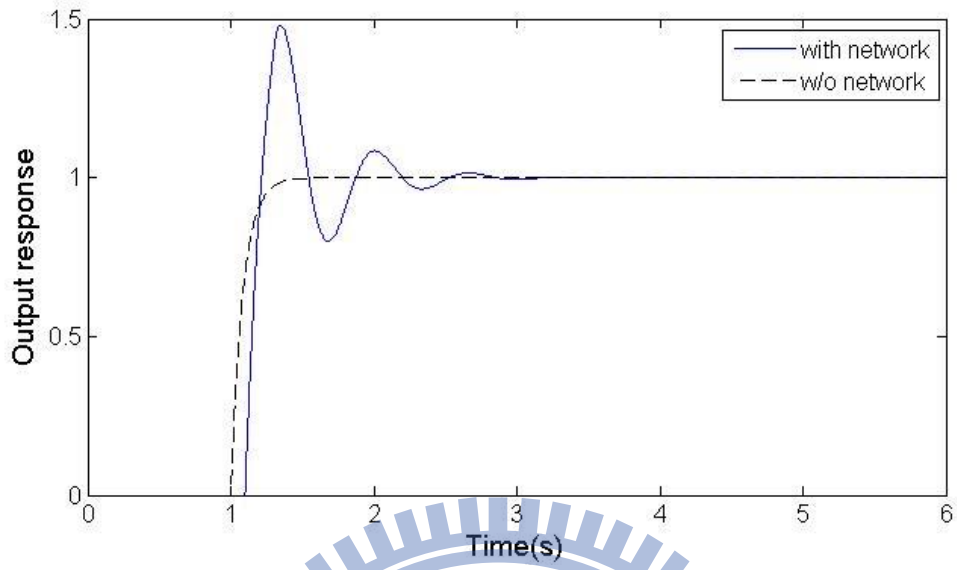


圖 3-2 時間延遲對系統影響圖(delay time = 20 ms)

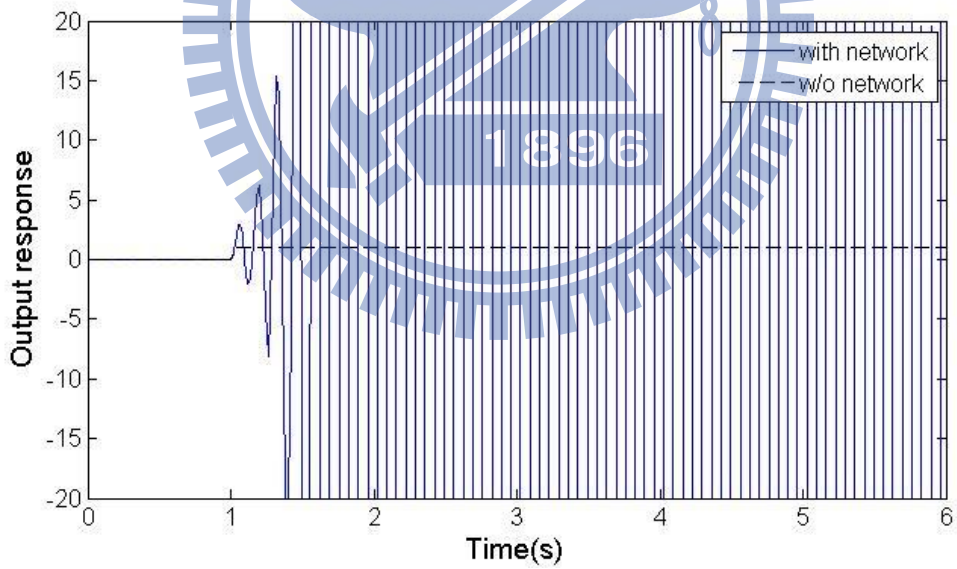


圖 3-3 時間延遲對系統影響圖(delay time = 100 ms)

## 3.2 時間延遲之量測與產生

### 3.2.1 Zigbee 網路的時間延遲

以圖 2-11 的方式實現以 Zigbee 與 CAN 網路傳輸的網路控制系統，當取樣頻率為 20ms 時，實驗結果如圖 3-4 所示，原本的控制效能因為 Zigbee 網路的存在而受到了影響。

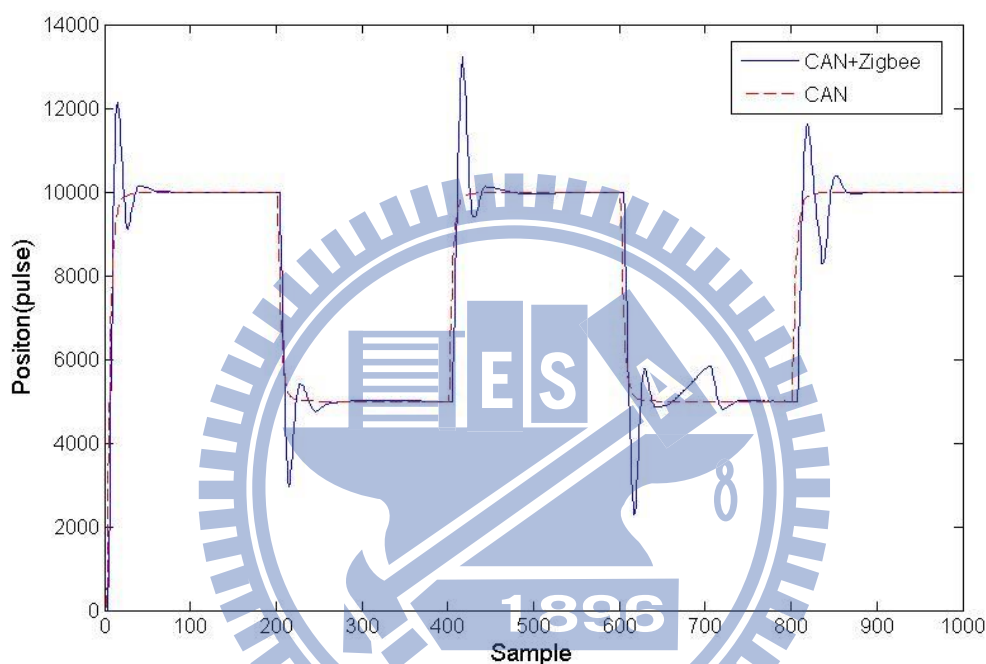


圖 3-4 取樣頻率為 20ms 時 Zigbee 網路控制系統響應圖

為了測量 Zigbee 網路的延遲時間，在這邊以兩個 Zigbee 裝置互相傳送封包，並記錄時間。實驗方式為，第一個 Zigbee 傳送封包給第二個 Zigbee，當第二個 Zigbee 收到封包後立即回傳，第一個 Zigbee 記下總經過時間。實驗 200 次紀錄結果後計算其平均時間為 34.1687(ms)，也就是說 Zigbee 網路每傳送一次封包平均需要花費 34 毫秒的時間。

因為 Zigbee 網路的傳輸時間為 34 毫秒，大於系統的取樣週期因而導致系統響應變差，因此更改系統取樣週期則可以改善此現象。若將系統取樣週期更改為 50ms，此時可得到結果如圖 3-5 所示，將 20ms 取樣週期時過大的振幅以及震盪

現象消除了。

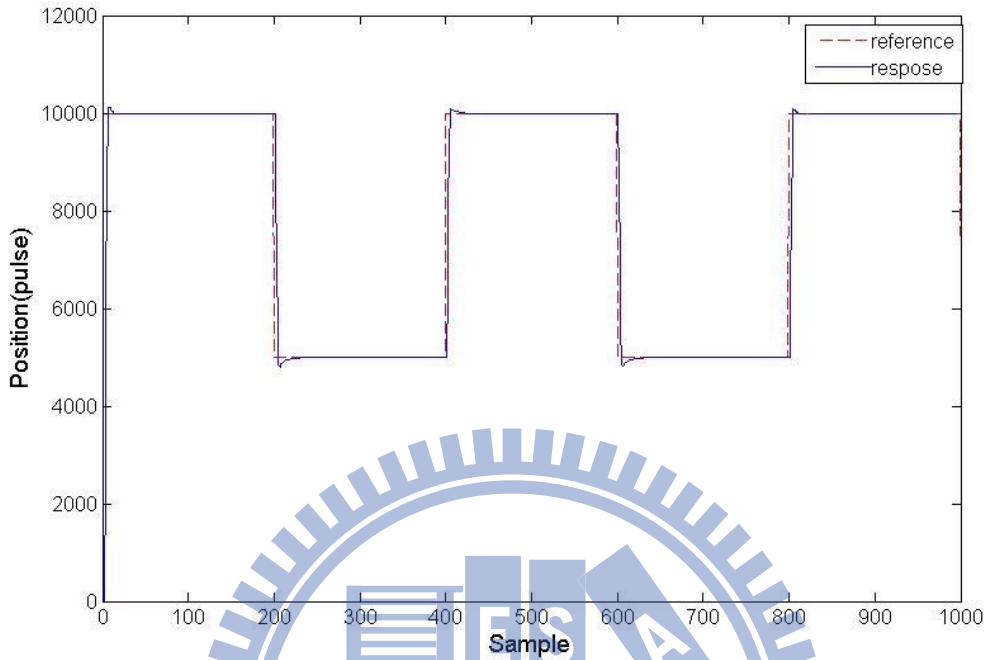


圖 3-5 取樣頻率為 50ms 時 Zigbee 網路控制系統響應圖

### 3.2.2 網路系統時間延遲量測

為了探討時間延遲對於網路控制系統的影響，必須先取得整個網路的延遲時間，此網路延遲時間是指圖 3-1 中  $T_1$  與  $T_2$  時間的總和， $T_1$  為命令延遲時間， $T_2$  為回授延遲時間。其中  $T_1$  和  $T_2$  延遲時間為每個節點的應用層到另外一邊應用層加起來的總和，而不是單純指硬體層對硬體層之間的傳輸時間，如圖 3-6 所示。

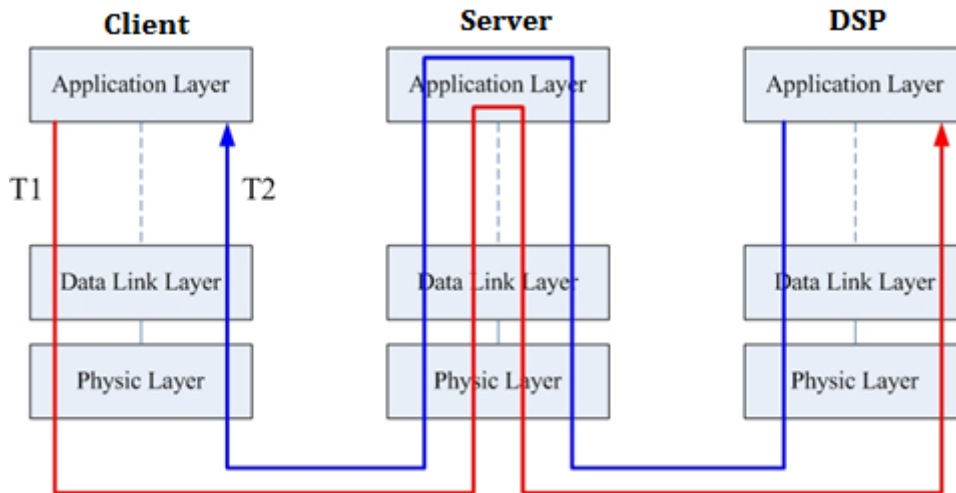


圖 3-6 延遲時間計算

時間延遲的量測方法是由 DSP 端每一取樣週期傳送一筆帶有 Index 的資料封包經由 Server 傳送到 Client 端，這裡 Index 指的是該封包的編號，封包傳送出去前會記錄封包發送的起始時間，當 Client 端接收封包之後，立刻將封包原封不動傳回 DSP，DSP 接到封包後記錄到達時間，並且根據封包上面的 Index 找出發送時的起始時間，將到達時間跟起始時間相減即可求得延遲時間。

### 3.2.3 固定時間延遲的產生

若想分析較長的時間延遲會對系統造成什麼影響，或想測試設計的控制架構在長時間延遲情況下的效能，一個方法是將 DSP 端與 Client 端分別放置在相隔遙遠的兩地，將網路傳輸的實體線路拉長，中間可能經過多個 Hub，使網路實際傳輸的時間拉大。另外一個方式可以在命令端或是回授端建立一個 Buffer 如圖 3-7 所示，當有回授訊號要傳回 Remote 端時，回授訊號會先進入 Buffer，然後每隔一次取樣週期位移一個位置，直到離開 Buffer 之後 Remote 端的控制器才會取得回授訊號，如此每筆回授訊號依序進入 Buffer 再出來，每筆訊息的延遲時間會等量的增加，增加後的延遲時間和 Buffer 大小關係如下：



$$T_{d\_new} = (Size_{Buffer} - 1) \cdot T_s + T_d \quad (3-1)$$

其中  $T_{d\_new}$  為延長過後的延遲時間， $Size_{Buffer}$  為 Buffer 大小， $T_s$  為系統取樣週期， $T_d$  為原本網路的延遲時間。此方法不只可以造成較長的時間延遲，同時也能產生固定的時間延遲，這對於分析時間延遲對於系統架構的引響，便利性相當大。

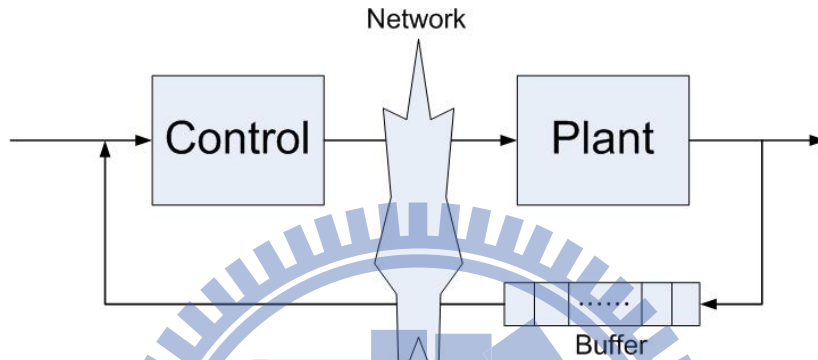


圖 3-7 回授端加 Buffer 示意圖

### 3.3 時間延遲補償

#### 3.3.1 完美延遲補償機制(Perfect Delay Compensation, PDC)

大部分的網路控制系統在設計時，都假設網路延遲時間為一固定項，然而，這種設計方法在現實情況下是不切實際的，因為網路產生的延遲會隨著時間有著巨大的變動，而結果往往會造成系統相位的落差以及不穩定。為了解決這問題，本研究使用了時間延遲完全補償機制 perfect delay compensation (PDC)來處理網路的時間延遲。

圖 3-8 為 PDC 應用於一般網路控制系統的架構圖，以下將推導 PDC 架構的數學模型。

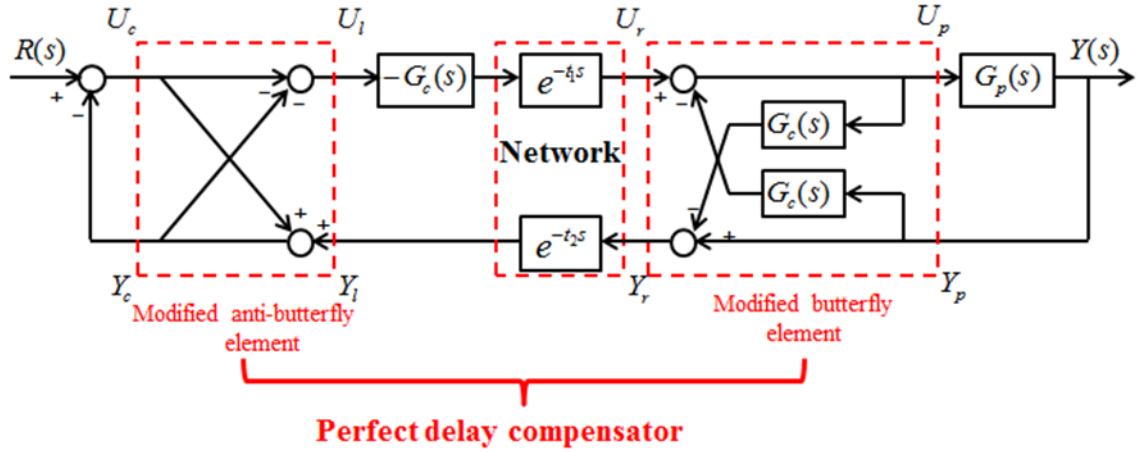


圖 3-8 PDC 應用於網路控制系統

首先，從圖 3-8 得知，modified butter element 的輸出結果可以被表示成

$$U_p(s) = U_r(s) - G_c(s)Y_p(s); Y_r(s) = -G_c(s)U_p(s) + Y_p(s) \quad (3-1)$$

接著， $U_r$  到  $Y_r$  的轉移函數可以表示成為

$$\frac{Y_r}{U_r} = \frac{-G_c(s)U_p(s) + Y_p(s)}{U_p(s) + G_c(s)Y_p(s)} = \frac{G_p(s) - G_c(s)}{1 + G_c(s)G_p(s)} \quad (3-2)$$

因為  $\frac{Y_p(s)}{U_p(s)} = G_p(s)$ ，故左側  $U_c$  到  $Y_c$  的轉移函數可表示為

$$\frac{Y_c(s)}{U_c(s)} = \frac{(1 + G_c(s)G_p(s)) + G_c(s)(G_p(s) - G_c(s)) \cdot e^{-(t_1+t_2)s}}{(1 + G_c(s)G_p(s)) - G_c(s)(G_p(s) - G_c(s)) \cdot e^{-(t_1+t_2)s}} \quad (3-3)$$

而系統的開迴路轉移函數可被表示為

$$\frac{Y_c(s)}{R(s)} = \frac{(1 + G_c(s)G_p(s)) - G_c(s)(G_p(s) - G_c(s)) \cdot e^{-(t_1+t_2)s}}{2 \cdot (1 + G_c(s)G_p(s))} \quad (3-4)$$

網路控制系統的閉迴路轉移函數則為

$$G_{NCS}(s) = \frac{Y(s)}{R(s)} = \frac{G_c(s)G_p(s) \cdot e^{-t_1 s}}{1 + G_c(s)G_p(s)} = G_o(s) \cdot e^{-t_1 s} \quad (3-5)$$

由(3-5)式得知，最後的結果原本無網路的閉迴路轉移函數與 $t_1$ 的延遲項相乘， $G_c$ 為原本系統的控制器可以直接使用，不需要因為加入網路的關係而另外作修。而回授延遲時間 $t_2$ 不影響輸出響應。

### 3.3.2 PDC 模擬結果

本節中將使用 Truetime 模擬環境建構出的網路控制系統來印證時間延遲補償架構 PDC 的推導結果。在 Truetime 中建立如圖 2-11 的 Zigbee 網路控制系統，受控體轉移函數為 $\frac{s}{s+6}$ ，取樣週期為 50ms，時間延遲為 250ms。如圖 3-9 所示，在未加入 PDC 架構前系統響應會呈現震盪的現象。加入 PDC 架構後從模擬結果可以看出原本震盪的現象已經消失，而系統響應變成單純的將控制時間延後，延後的長度即為網路延遲時間的大小，如圖 3-10 與圖 3-11 所示。

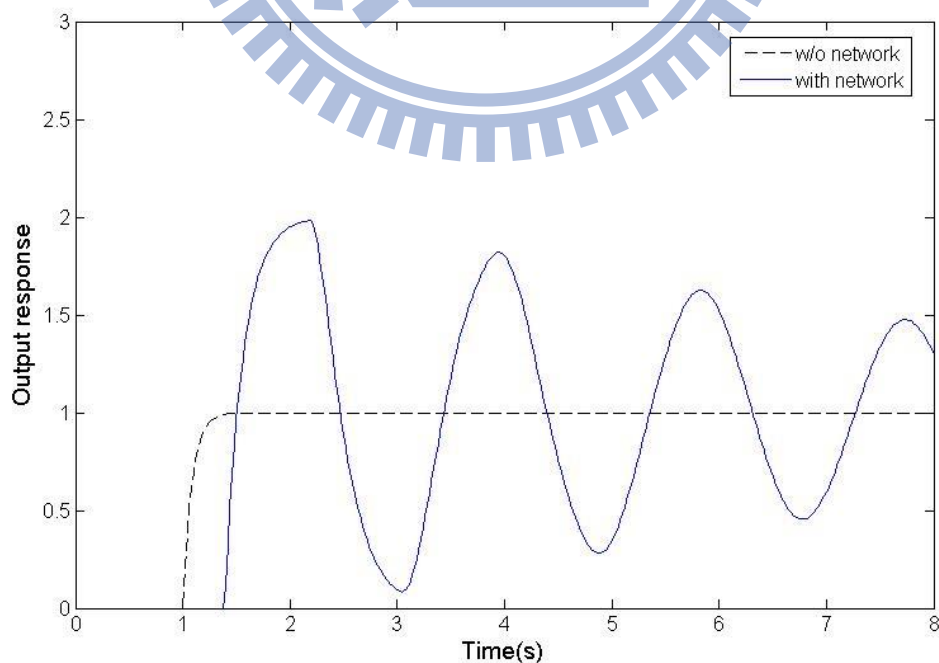


圖 3-9 未加入 PDC 的系統響應

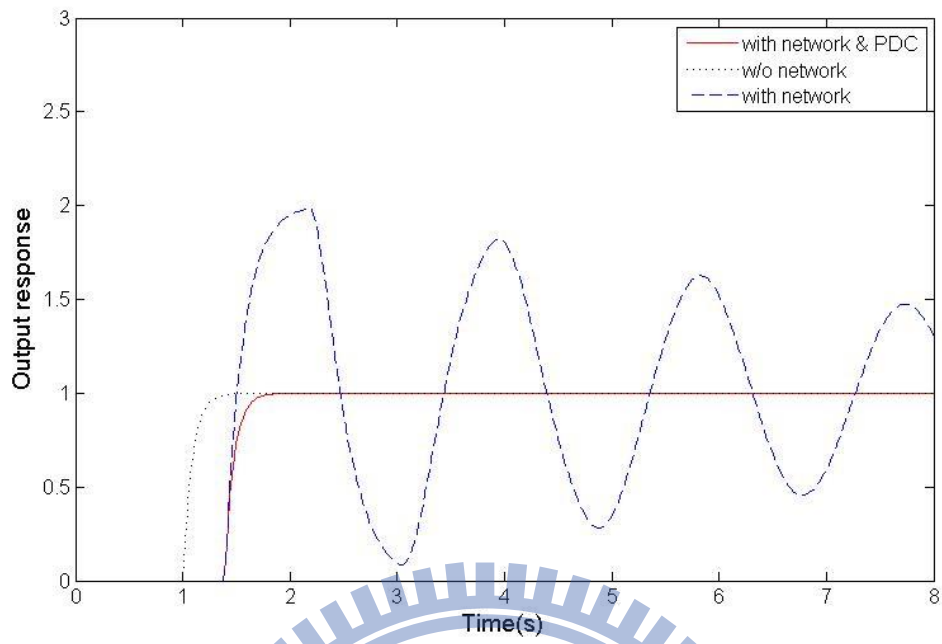


圖 3-10 PDC 加入前後的步階響應比較

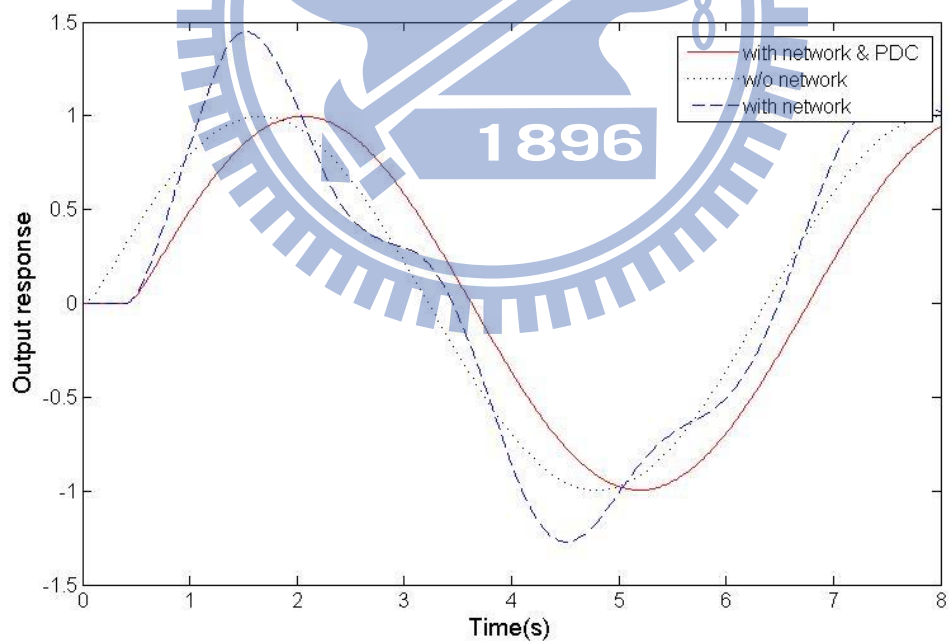


圖 3-11 PDC 加入前後的弦波響應比較

### 3.3.3 PDC 實驗結果

將 PDC 架構實現於如圖 2-14 所示的遠端網路控制系統中，在原本的 Zigbee 網路中加入 Ethernet。remote 節點在 5Km 外的距離透過 Ethernet 與 client 節點溝通並且下達命令，client 節點再藉由 Zigbee 網路將資料傳給 DSP 端的 AC 伺服馬達。因為經過了乙太網路，無可避免的時間延遲將會降低系統的穩定度與可靠度，因此 PDC 架構會被實現且應用於此遠端網路控系統中。PI 控制器被寫入 TI TMS320F2812 DSP 裡，其中  $K_p = 0.0002$ ， $K_i = 0.0000001$ ，取樣時間為 50ms。

圖 3-13 為量測到的網路時間延遲，而基於式子(2.2)的伺服馬達網路控制系統之實驗結果如圖 3-12 所示，網路控制系統的響應若是沒有 PDC 架構補償時會產生劇烈的震盪。但是經過 PDC 架構補償後系統的控制效能可以得到相當理想的改善，如圖 3-14。

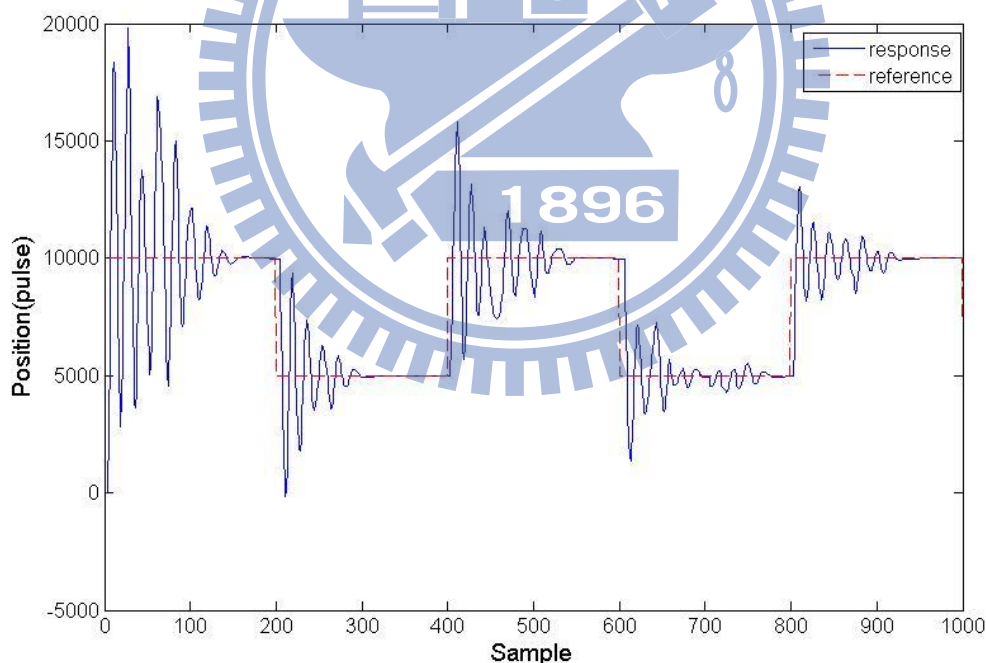


圖 3-12 遠端網路控制系統響應

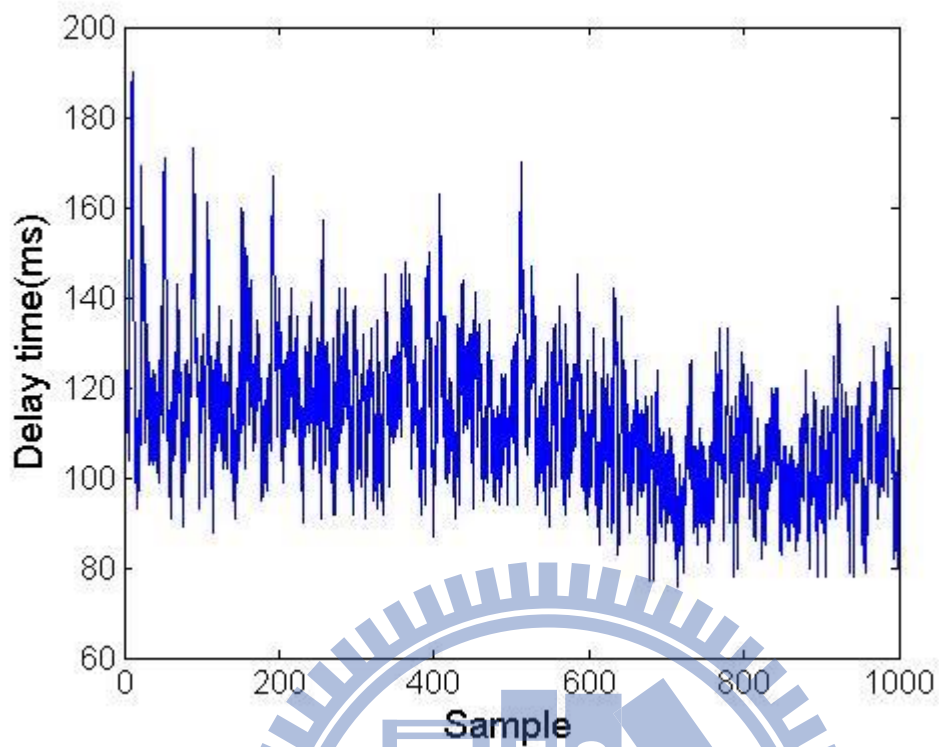


圖 3-13 遠端控制的時間延遲

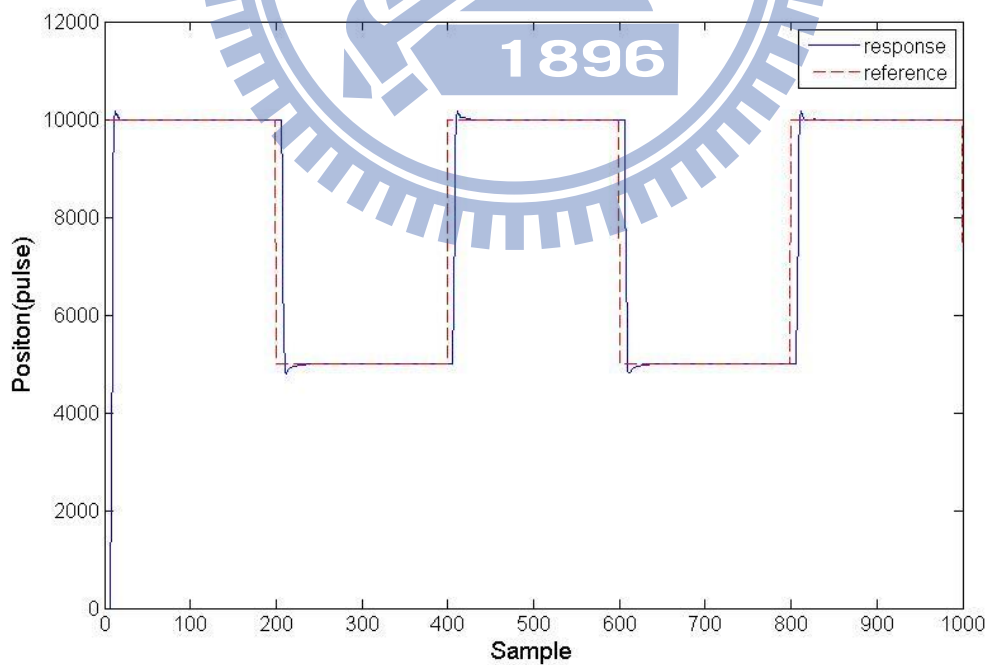


圖 3-14 加入 PDC 後的遠端控制響應

不管在實驗中或模擬中，使用 PDC 價購時不需要鑑別系統模型，甚至不需要知道網路延遲時間，就能夠將網路延遲時間對系統的影響轉變為單純的時間延遲，相當的有效且方便。在本章裡無論是分析上或是實作中都證明了 PDC 架構能成功有效的補償未知且變動的時間延遲，即使是對於遠端控制而言，透過 PDC 的補償，結果就如同直接在旁邊控制馬達般，將時間延遲的影響減至最低。



## 第四章 資料遺失討論

### 4.1 資料遺失介紹

一般的網路控制系統分為事件觸發(event-trigger)與時間觸發(time-trigger)兩種模式。事件觸發(event-trigger)為根據數據封包的資料是否傳達做為系統觸發的依據，在網路系統中，每個封包抵達的時間可能不盡相同，因此控制訊號的下達時間並不是固定的，如圖 4-1(a)所示。事件觸發的優點是架構簡單、無須短時間內多次傳送封包，能夠避免網路壅塞，此模式多數用在低精確度控制的領域上，而其缺點為無法及時回授感測器資料，也無法及時下達控制命令，若應用於精密控制的領域上時會有取樣點數不足，誤差過大，控制效能低落的問題。時間觸發(time-trigger)為本研究採用觸發模式，控制命令與回授命令會在每一固定的取樣週期內傳送，如圖 4-1(b)所示，相較於事件觸發，具有高取樣頻率且閉迴路即時控制的優點。

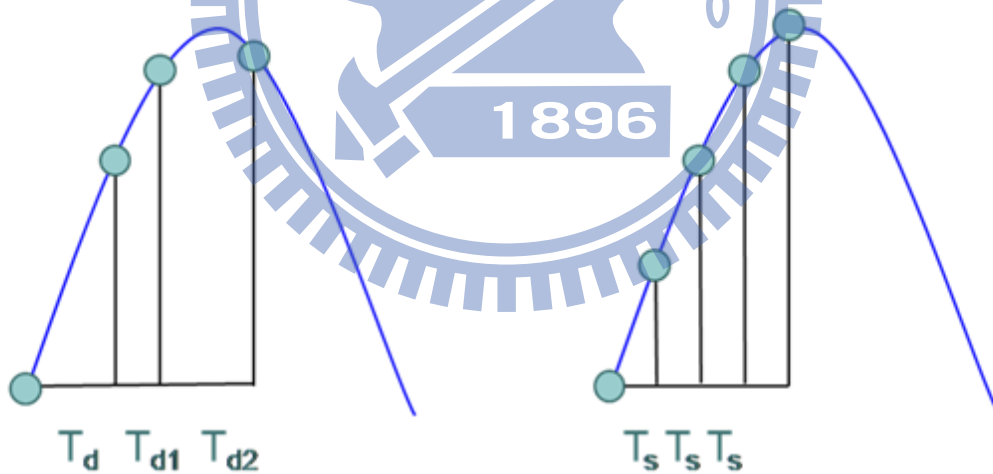


圖 4-1 (a)事件觸發(event-trigger)示意圖 (b)時間觸發(time-trigger)示意圖

當命令訊號或回授訊號的傳輸無法滿足規定的時限時，則將此現象定義為資料遺失(data dropout)，如圖 4-2 所示。對於即時網路控制系統而言，若網路產生的延遲時間大於系統的取樣週期時，就會發生資料遺失。除此之外，單就網路



傳輸的部分來看，外界環境的干擾，封包之間彼此互相的碰撞等各種因素都會造成資料遺失。因此，資料遺失在講求高速高精度的網路控制系統中是不可避免的，同時也是設計網路控制系統的重要議題之一。

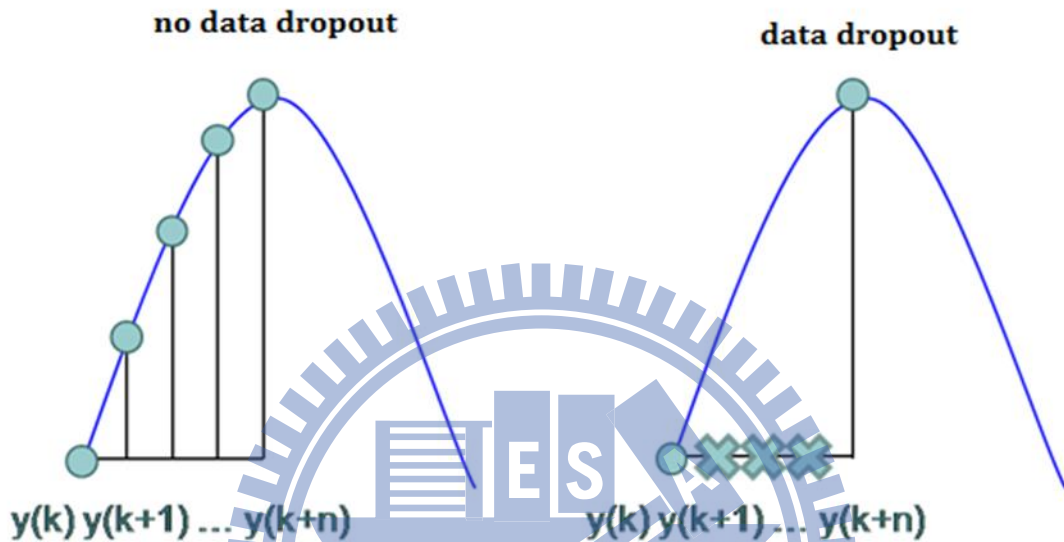
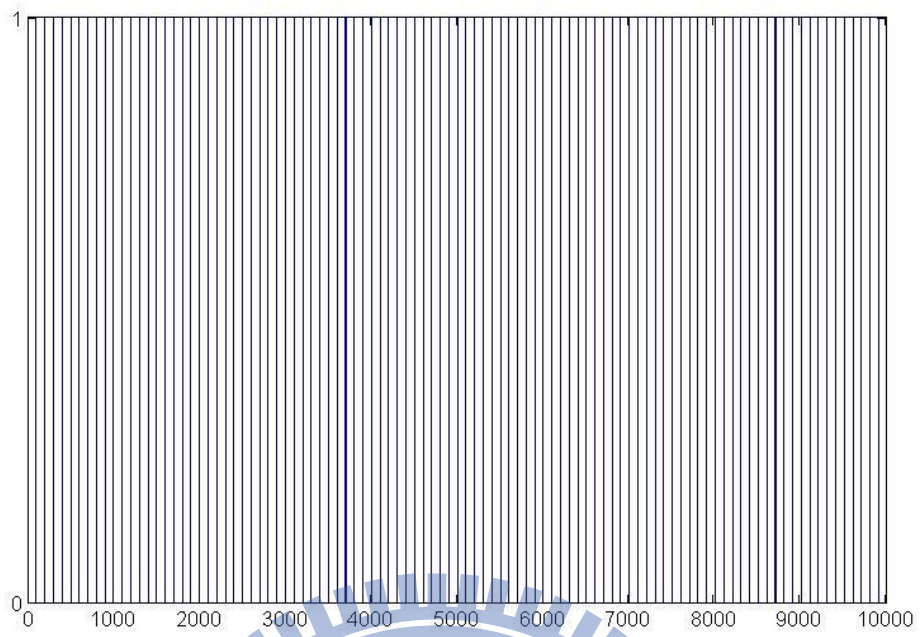


圖 4-2 資料遺失示意圖

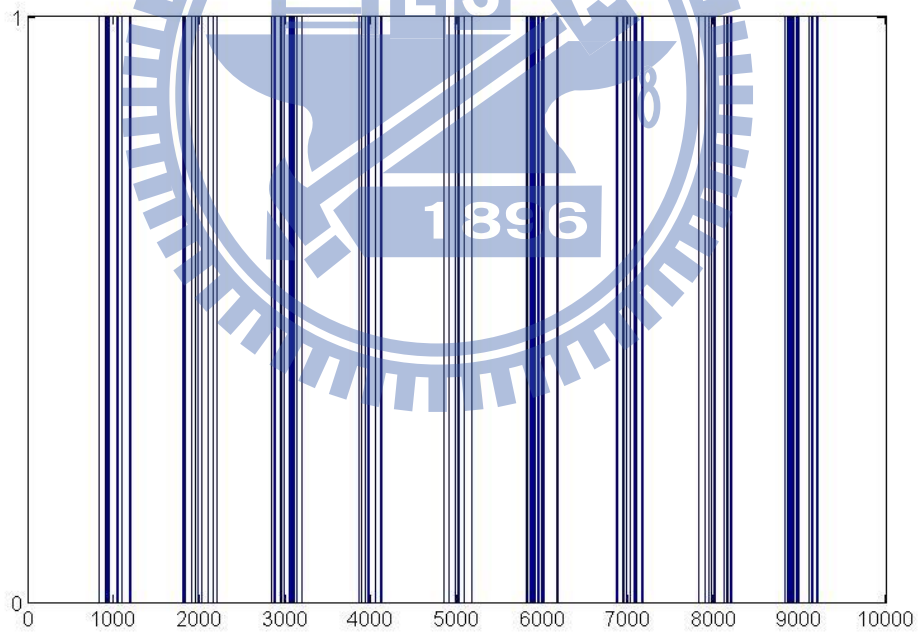
## 4.2 即時轉移機率估測器

### 4.2.1 資料遺失分布的影響

對於控制系統而言，比起平均分布的資料遺失，連續性的資料遺失會造成更為嚴重的控制誤差。圖 4-3(a)和(b)為兩個同為 20% 資料遺失率應用於網路控制系統上，模擬結果顯示，當資料遺失集中於某一部分的時候，控制效果會明顯的下降，如圖 4-4 所示。這結果說明除了資料遺失本身以外，資料遺失的分布情形同時會對控制系統造成很大的影響。



(a)



(b)

圖 4-3 (a)分散式與 (b)集中式的資料遺失 (20%資料遺失率)

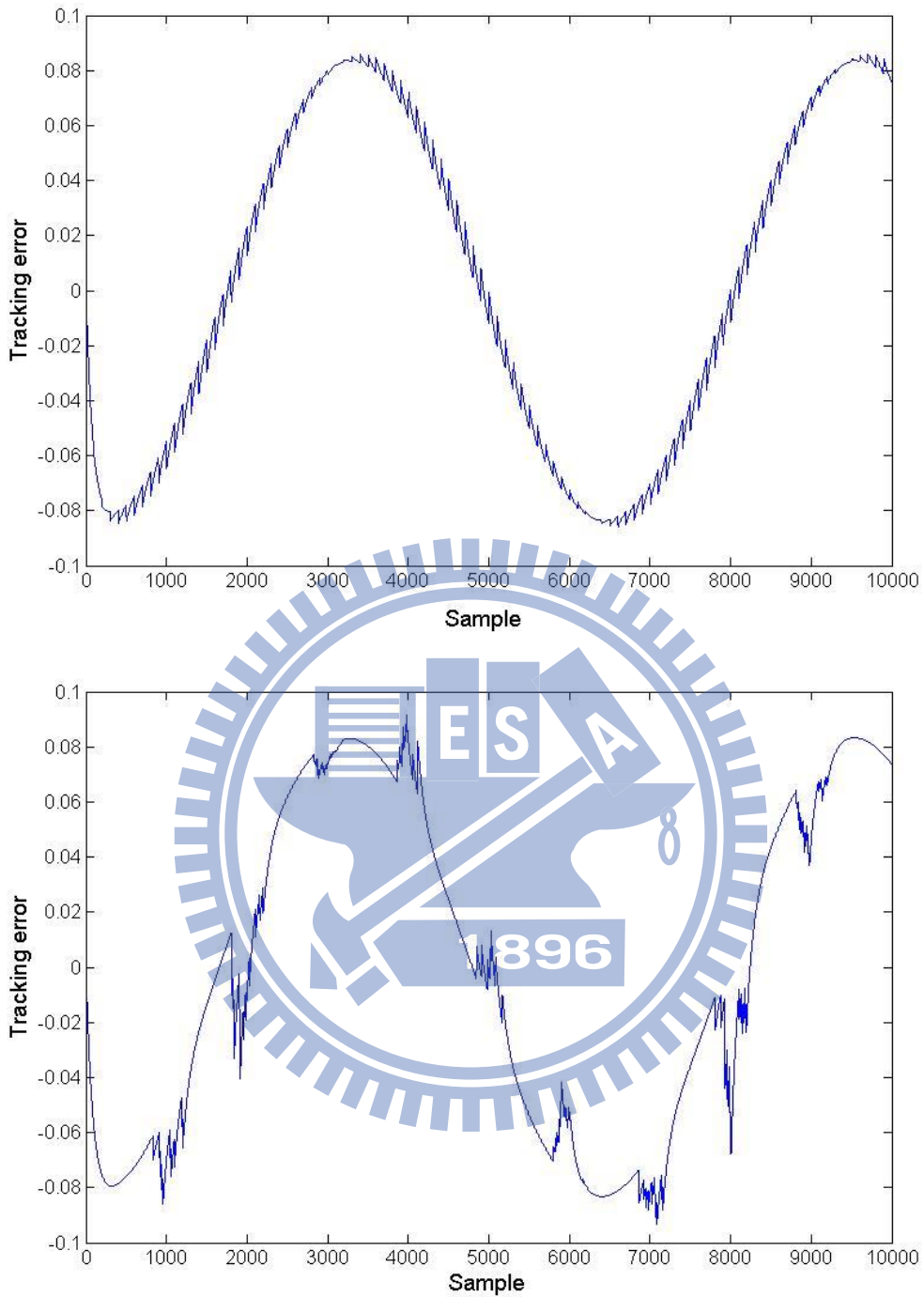


圖 4-4 追蹤誤差於(a)分散式 (b)集中式資料遺失 (20%資料遺失率)

#### 4.2.2 Two-state Markov Chain 網路模型

在網路控制系統傳輸封包的過程中，資料遺失可能為均勻分布或是集中於某些時間點，而由上一小節的討論可知，集中式的資料遺失對於控制系統操作的影

響特別劇烈。為了分析集中式的資料遺失，在這裡使用了 two-state Markov chain (Gilbert, 1960; Wang and Moayeri, 1995)。在 two-state Markov chain 中，系統未來的狀態只與現在的狀態有關，與過去任何時間點的狀態皆無關，換句話說，系統是無記憶性的(memoryless)。此模型中使用了兩個參數描述資料遺失的情況， $D$  表示資料遺失的狀態， $R$  表示資料未遺失的狀態。因此，在每一個取樣時間點資料為遺失或是未遺失的機率  $\rho_{i,j}$  可以寫為

$$\rho_{ij} = \Pr[\alpha(k+1) = j | \alpha(k) = i] \text{ for } i, j \in \{R, D\} \quad (4.1)$$

其中  $\rho_{D,R}$  為從  $D$  狀態轉換為  $R$  狀態的機率，而  $\rho_{R,D}$  為從狀態  $R$  轉變為狀態  $D$  的機率。所有的狀態轉換的機率可以圖 4-5 表示或者以式子(3.2)的 Markov transition matrix 表示。

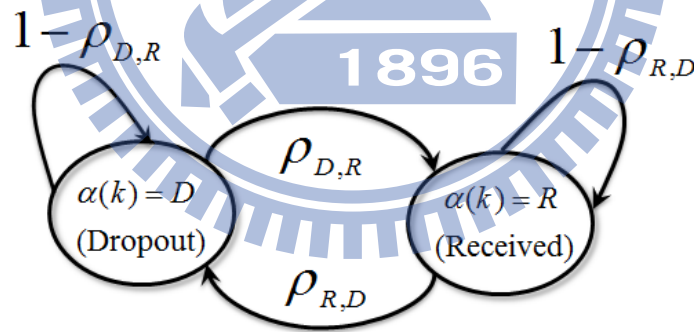


圖 4-5 Two-state Markov chain 網路模型

$$P = [\rho_{ij}] = \begin{bmatrix} \rho_{D,D} & \rho_{D,R} \\ \rho_{R,D} & \rho_{R,R} \end{bmatrix} = \begin{bmatrix} 1 - \rho_{D,R} & \rho_{D,R} \\ \rho_{R,D} & 1 - \rho_{R,D} \end{bmatrix} \quad (4)$$

Markov transition matrix  $P$  表示所有一階的狀態轉換，而  $P^m$  則表示所有未來  $m$

階的狀態轉換  $P^m = [\rho_{i,j}]$ ，其中

$$\rho_{ij} = \Pr[\alpha + (k-m)\alpha | j = k] \quad \text{if } 0, i, j \in \{RD\} \quad (4)$$

在本研究中，m 的值为 1。

Markov transition matrix 也提供了系統平均行為的訊息，如果轉換矩陣裡所有元素的值都是正的，則當  $k \rightarrow \infty$  時，存在唯一的向量  $\mu$  會使得  $(P^T)^k \mu_0 \rightarrow \mu$ 。向量  $\mu$  在這裡同時代表 eigenvalue 為 1 的 eigenvector 以及所有元素值總合為 1 的機率向量。向量  $\mu$  對於 Markov chain 的網路模型架構而言，描述了網路資料平均的接收和遺漏 (Leon, 1998)。舉個例子，當網路資料的傳輸情形為  $(\rho_{D,R}, \rho_{R,D}) = (0.7, 0.2)$ ，其對應的轉移矩陣為

$$P = \begin{bmatrix} 0.3 & 0.7 \\ 0.2 & 0.8 \end{bmatrix}$$

其中

$$\text{eig}(P^T) = 0.1, 1$$

$$\text{eigenvectors} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.222 \\ 0.778 \end{bmatrix}$$

對應於 eigenvalue 為 1 的 eigenvector 表示了在那時的情況下平均而言系統在狀態 R 佔據了 77.8% 的時間，而狀態 D 佔據了 22.2% 的時間。對於不同的網路情形以及不同的網路參數產生的影響，Two-state Markov 模型的描述是非常便利且利於圖形視覺化的。

為了更進一步的探討網路參數的物理含意，可以將網路資料遺失情況表示為網路參數的函數[Kawka 2006]。連續 D 狀態的期望值與連續 R 狀態的期望值可以寫成下式

$$\begin{aligned} E(\text{cons } D \text{ states}) &= \frac{1}{\rho_{D,R}} = \frac{1}{1 - \rho_{D,D}} \\ E(\text{cons } R \text{ states}) &= \frac{1}{\rho_{R,D}} = \frac{1}{1 - \rho_{R,R}} \end{aligned} \quad (44)$$

### 4.2.3 即時轉移機率估測器(Transition Probability Estimator)

因為 two-state Markov chain 網路模型能夠清楚的表示網路參數的特性，轉移機率(transition probability)的測量變得非常重要且有價值。然而，轉移機率由不可預測的封包碰撞、網路流量負載和系統取樣頻率等種種因素形成的，因此測量轉移機率相當的耗時且困難。先前的研究中可以成功測量轉移機率並建構 two-state Markov chain 網路模型的成果都必須耗費大量的時間和計算複雜度(Bertuccelli and How, 2008; Azimi et al., 2005; Azimi et al., 2003)，而這在實際的應用中有著相當大的限制。

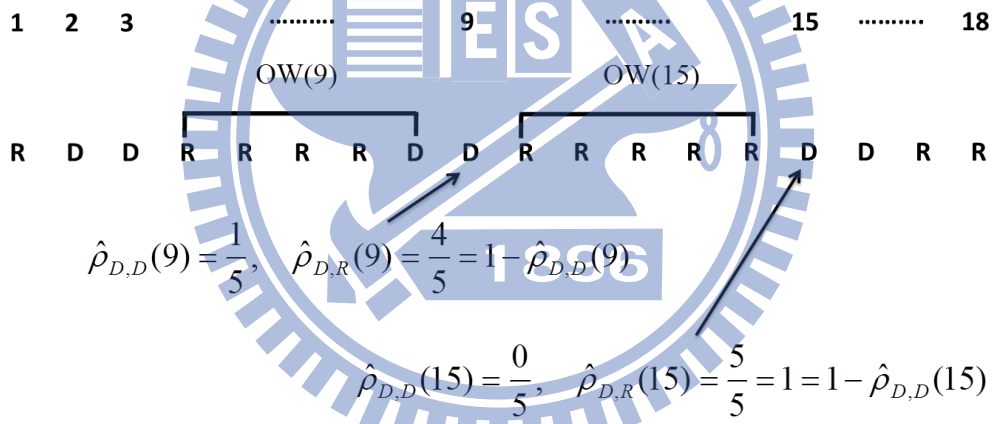
網路控制系統需要快速且精確的控制，然而傳統的轉移機率著眼的目標為網路整體的訊號，並不適合應用於即時的網路控制系統。因此在這邊提出即時的轉移機率估測方式，對於控制系統估測區域性的即時轉移機率。

在即時的即時轉移機率估測器中，有兩點與傳統的轉移機率估測器不同。第一為採用觀測框(observation window)，觀測框的定義為當前時間點有限長度的前幾筆資料。舉個例子，如果觀測框的長度為五，則在時間點  $k=10$  時， $OW(10)=[\alpha(5) \alpha(6) \alpha(7) \alpha(8) \alpha(9)]$ ，其中  $\alpha(k)$  為網路訊號。藉由觀測框的運用，網路的特性可以被快速的分析。第二點不同的地方為重新定義  $\hat{\rho}_{ij}(k)$

$$\hat{\rho}_{i,j}(k) = \Pr[\Pr(OW(k) = j) | \alpha(k) = i] \text{ for } i, j \in \{R, D\} \quad (4.5)$$

在這個定義下，因為所有先前的網路訊號皆已知，因此可以有效的估測區域性的轉移機率， $\hat{\rho}_{i,j}(k)$ 。除此之外，如果觀測窗長度夠長的話，個別 $\hat{\rho}_{i,j}(k)$ 的平均更可以有效的代表整體的轉移機率， $\rho_{i,j}$ 。

將提出的局部轉移機率即時估測器分別應用於兩個有相同轉移機率， $\rho_{D,D} = 0.5$ 的信號上，如圖 4-6 (a)和圖 4-6 (b)所示。雖然這兩個信號的 $\rho_{D,D}$ 相同，但是以局部的觀點來看，它們資料遺失的位置卻是完全不同。對於這兩個信號在第 15 個時間點估測的結果分別為 $\hat{\rho}_{D,D}(15) = 0$ 、 $\hat{\rho}_{D,D}(15) = 60\%$ 。因此，局部的轉移機率， $\hat{\rho}_{i,j}(k)$ 比起傳統的 $\rho_{i,j}$ 更為精確的指示出網路控制系統中資料遺失的分布情形。



$$\text{Total transition probability, } \rho_{D,D} = \frac{3 \text{ (numbers of D} \rightarrow \text{D)}}{6 \text{ (numbers of D)}} = 0.5$$

(a)

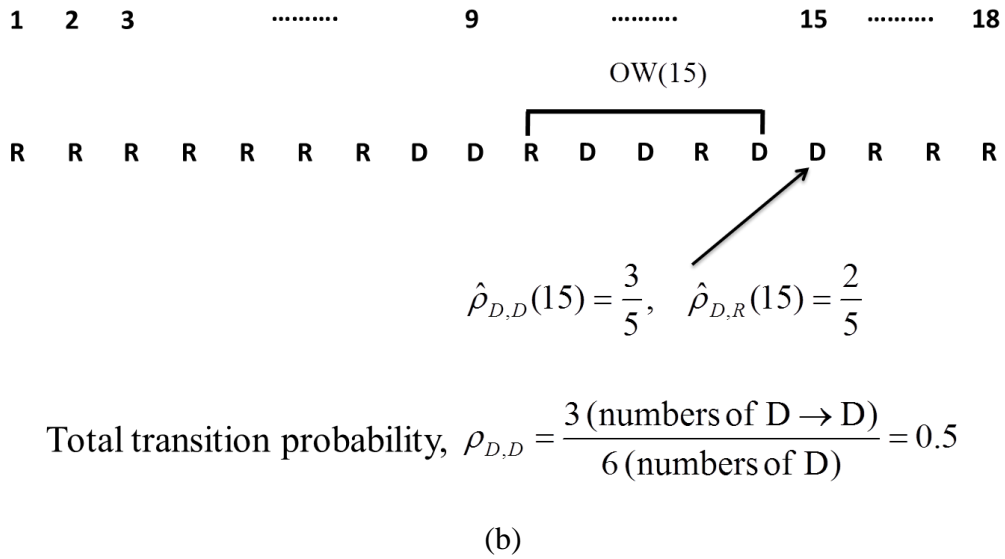


圖 4-6 在  $\rho_{D,D} = 0.5$  的情況下即時計算局部的 transition probability  $\hat{\rho}_{D,D}(k)$

### 4.3 訊息估測與最小平方估測器

本研究是採用時間觸發作為封包傳輸的基準，對於網路控制系統而言，受控端與控制端會因為網路的因素而不能即時接收資料，導致無法即時作出控制而使輸出誤差上升。對於無法及時接收的命令，解決之辦法為：

1. 直接沿用前一筆資料(One-delay)
2. 使用估測器估測當下的數值

以估測法則來預估未能即時接收的資料是一種較為精確的方法，特別是針對突然的少筆資料遺失，但在某些狀況，One-delay 仍有其好處，在下面會加以說明。



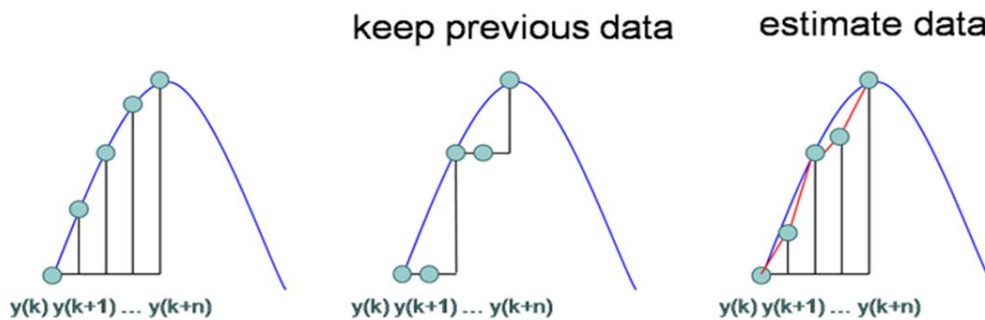


圖 4-7 不同估測方式的示意圖

圖 4-7 為不同估測方式的示意圖，若每一筆資料均可以透過網路即時地傳送，則不需要估測，如同圖(a)所示。若沿用前一筆透過網路傳送的資料，是最為簡單的方法來估測未能及時接收的資料，但從圖(b)可看出，會有估測誤差過大的問題，此估測誤差會連帶影響輸出誤差，導致在網路延遲時間過長的情況下，會有控制效能降低、甚至不穩定的問題產生。倘若利用估測法則來設計估測器，如：泰勒估測(Taylor estimate)、最小平方估測(Least-square estimate)等，如圖(c)所示，會降低估測誤差，得到較好的控制效能。

最小平方法是以前項式曲線的方式來估測，假設有  $M$  個已知數值，欲得到第  $M+1$  個的數值，藉由前  $M$  筆已知數值與  $N$  次方多項式曲線來做逼近，該  $N$  次方多項式曲線與  $M$  個已知數值的平方距離為最小，則該  $N$  次方多項式曲線為其他未知數值所估測出的集合，此為最小平方法的原理。以下為最小平方法的推導過程：

對於一般的時間序列  $x[0], x[1], \dots, x[M]$ ，可以多項式適當的描述如下：

$$x[k] = c_0 + c_1 k + c_2 k^2 + \dots + c_N k^N \quad (4.6)$$

因此

$$\begin{aligned}
x[1] &= c_0 + c_1 + c_2 + \cdots + c_N \\
x[2] &= c_0 + c_1 \cdot 2 + c_2 \cdot 2^2 + \cdots + c_N \cdot 2^N \\
&\vdots \\
x[M] &= c_0 + c_1 \cdot M + c_2 \cdot M^2 + \cdots + c_N \cdot M^N
\end{aligned} \tag{4.7}$$

將 Eq. (4.7) 重整如下

$$\begin{bmatrix} x[1] \\ x[2] \\ \vdots \\ x[M] \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 2^0 & 2 & \cdots & 2^N \\ \vdots & \vdots & & \vdots \\ M^0 & M & \cdots & M^N \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_N \end{bmatrix} \equiv x = A \cdot c \tag{4.8}$$

$$c = (A^T A)^{-1} A^T x \tag{4.9}$$

因此，當前遺失的資料值可以被估測為

$$\begin{aligned}
x[M+1] &= c_0 + c_1 (M+1) + c_2 (M+1)^2 + \cdots + c_N (M+1)^N \\
&= \begin{bmatrix} (M+1)^0 & (M+1)^1 & \cdots & (M+1)^N \end{bmatrix} \cdot c \\
&= \begin{bmatrix} (M+1)^0 & (M+1)^1 & \cdots & (M+1)^N \end{bmatrix} \cdot (A^T A)^{-1} A^T x \\
&\equiv LSE(M, N) \cdot x
\end{aligned} \tag{4.10}$$

其中  $M$  代表欲計算的資料數目， $N$  代表欲逼近的多項式階數。

為了達到網路控制系統的即時線上估測，上述參數必須事先決定。然而，最小平方估測器的階數和資料量長度必須根據實際情況決定。舉個例子，NURBS signal 可以被最小平方方法近似為三階多項式(Sorenson, 1970)。因此，觀測框的長度可以選擇五個資料量的長度來估測NURBS的曲線。為了避免費時的線上運算，所有的 LSE 公式會被事先計算，以下即為三個為了實際應用而事先計算的

$LSE(M, N)$ :

- $LSE(5,3) = 3.2z^{-1} - 2.8z^{-2} - 0.8z^{-3} + 2.2z^{-4} - 0.8z^{-5}$  (4.11)

- $LSE(3,2) = 3z^{-1} - 3z^{-2} + z^{-3}$  (4.12)

- $LSE(2,1) = 2z^{-1} - z^{-2}$  (4.13)

#### 4.4 智慧型訊息估測器(Intelligent Message Estimator, IME)

為了測量在網路傳輸中遺失的訊息，one-delay 估測器簡單的採用最後收到的訊息作為遺失訊息的補償，最小平方估測器從過去收到的資料估測現在的遺失訊號，但若是過去的訊號也遺失的話，則從估測器獲得的訊息也會變得不可靠。在本研究中，以智慧型訊息估測器(Intelligent Message Estimator, IME)基於不同階數的最小平方估測器配合及時量測的 transition probability ( $\rho_{D,D}$ )估測遺失的資料。當訊息嚴重遺失時，因為前幾筆的資料已經不再可靠，此時 one-delay 估測器會被納入 IME 估測器中。

- $LSE(5,3)$  用於低度資料遺失的情況下

網路傳輸的過程中，在長度為五個資料點的觀測框裡所有的資料皆有被接收到，或是最多只有一個資料遺失。 $LSE(5,3)$  在此時被用來以前五個資料點估測三階曲線命令，當然其中可能包含了先前被估測的資料點。三階的  $LSE(5,3)$  可以考慮到速度、加速度、甚至是加速度的改變來適當的估測運動軌跡，其公式參數來自於式子(4.11)。

- $LSE(3,2)$  用於中度資料遺失的情況下

在中度資料遺失的情形下，資料遺失率介於  $0.2 < \hat{P}_{D,D}(k) \leq 0.4$  之間時，在觀測框中只有三個可靠的資料值能用來估測現在遺失的資料。此時選用  $LSE(3,2)$  以前三個資料點估測二階的曲線軌跡，同時考慮到了速度和加速度的因素，如式子(4.12)。

- $LSE(2,1)$  用於高度資料遺失的情況下

在這個情形下，資料遺失率介於  $0.4 < \hat{P}_{D,D}(k) \leq 0.6$  之間，選用  $LSE(2,1)$  考慮到速度的因素以前兩筆資料點的訊息來估測曲線軌跡，如式子(4.13)。

- One-delay 估測器用於嚴重資料遺失的情況下

此時網路的通訊呈現著非常嚴重的資料遺失，資料遺失率介於  $0.6 < \hat{P}_{D,D}(k) \leq 1$ 。最小平方法的估測結果已經不再可靠，選用 one-delay 估測器直接採用前一點的資料值作為估測結果

$$1\text{-delay estimator} = z^{-1} \quad (4.14)$$

將資料遺失補償方法統整結果如(4.15)式子所示，補償架構如圖 4-8，此即為 IME 補償架構，基於及時量測到的  $\hat{P}_{D,D}$  切換到不同的估測器，補償網路控制系統中的資料遺失，由於應用了 transition probability 的概念，因此不管是集中式的資料遺失或是分散式的資料遺失都能得到適當的補償。

$$\begin{cases} 0 \leq \hat{P}_{D,D}(K) \leq 0.2, & LSE(5,3) \text{ is adopted} \\ 0.2 < \hat{P}_{D,D}(K) \leq 0.4, & LSE(3,2) \text{ is adopted} \\ 0.4 < \hat{P}_{D,D}(K) \leq 0.6, & LSE(2,1) \text{ is adopted} \\ 0.6 < \hat{P}_{D,D}(K) \leq 1, & 1\text{-delay estimator is adopted} \end{cases} \quad (4.15)$$

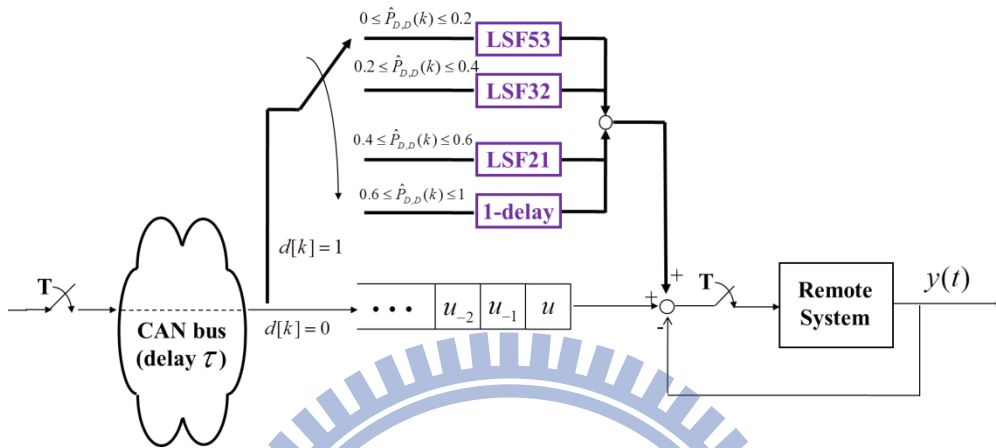


圖 4-8 IME 架構圖

#### 4.4.1 四種不同切換策略的 IME 架構

除了上面提到的基於(4.15)式補償資料遺失的 IME 架構外，這邊再針對切換區間的修改，得到另外三種切換策略的 IME 補償架構，並分別命名為 IME\_1，IME\_2，IME\_3。其中 IME\_1 為將 LSE(3,2)補償方式去除，同樣的資料遺失區間改以 LSE(2,1)補償，如(4.16)式所示

$$\begin{cases} 0 < \hat{P}_{D,D}(K) \leq 0.2, & LSE(5,3) \text{ is adopted} \\ 0.2 < \hat{P}_{D,D}(K) \leq 0.6, & LSE(2,1) \text{ is adopted} \\ 0.6 < \hat{P}_{D,D}(K) \leq 1, & 1\text{-delay estimator is adopted} \end{cases} \quad (4.16)$$

IME\_2 為將 LSE(5,3)的補償區間以 LSE(3,2)取代，如(4.17)式所示

$$\begin{cases} 0 < \hat{P}_{D,D}(K) \leq 0.4, & LSE(3,2) \text{ is adopted} \\ 0.4 < \hat{P}_{D,D}(K) \leq 0.6, & LSE(2,1) \text{ is adopted} \\ 0.6 < \hat{P}_{D,D}(K) \leq 1, & 1\text{-delay estimator is adopted} \end{cases} \quad (4.17)$$

IME<sub>3</sub> 為將 LSE(5,3)的補償區間以 LSE(3,2)取代且將 LSE 的補償區間擴大，如 (4.18)式所示

$$\begin{cases} 0 < \hat{P}_{D,D}(K) \leq 0.2, & LSE(3,2) \text{ is adopted} \\ 0.2 < \hat{P}_{D,D}(K) \leq 0.6, & LSE(2,1) \text{ is adopted} \\ 0.6 < \hat{P}_{D,D}(K) \leq 1, & 1\text{-delay estimator is adopted} \end{cases} \quad (4.18)$$

這四種 IME 架構將以下圖圖 4-9 的方式實現於網路控制系統中，並於不同的資料遺失率下在模擬及實驗中分析比較。

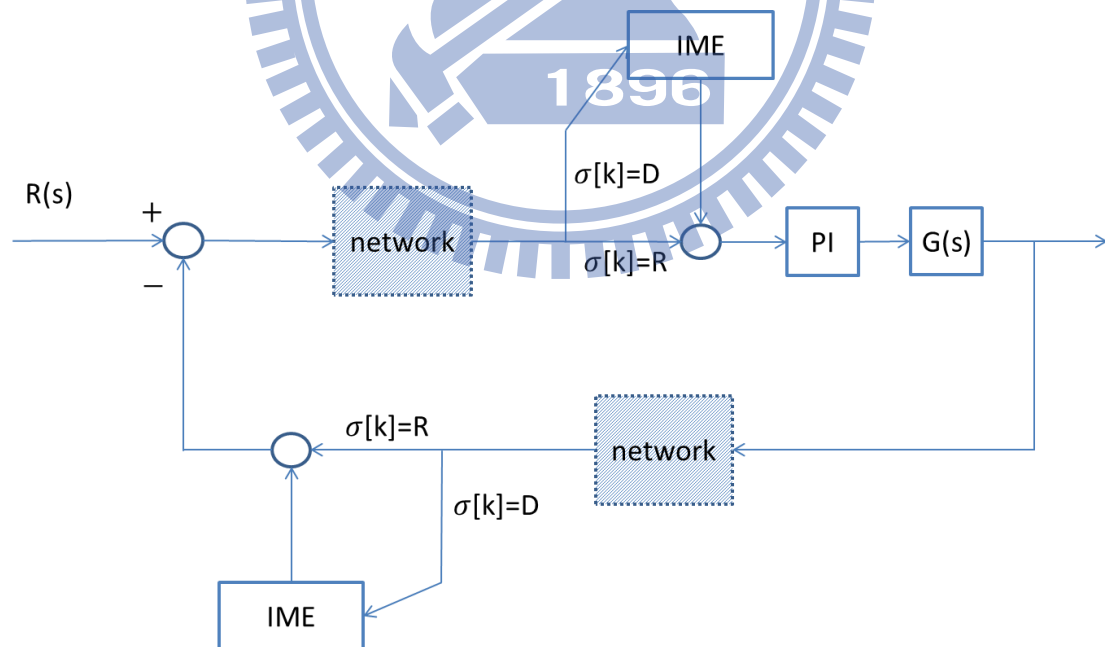


圖 4-9 IME 應用於網路控制系統補償架構圖

## 4.4.2 IME 的模擬結果

以 Truetime 模擬環境建構 Zigbee 網路的網路控制系統，在網路傳輸中制定不同的資料遺失率，然後實現 IME 架構於此系統中與 one-delay 估測器的補償結果互相比較，命令如圖 4-10 所示，受控體轉移函數為  $\frac{6}{s+6}$ ，取樣頻率為 10ms，模擬結果顯示如下。

如圖 4-11 與圖 4-12，在 20% 資料遺失率時 one-delay 估測器可以降低資料遺失的影響，然而，IME 補償架構卻會使系統響應發散。

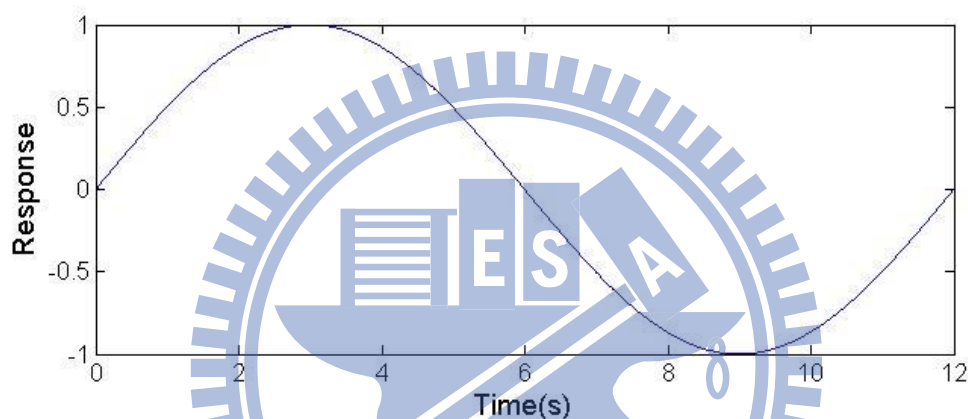


圖 4-10 網路控制系統模擬之弦波命令

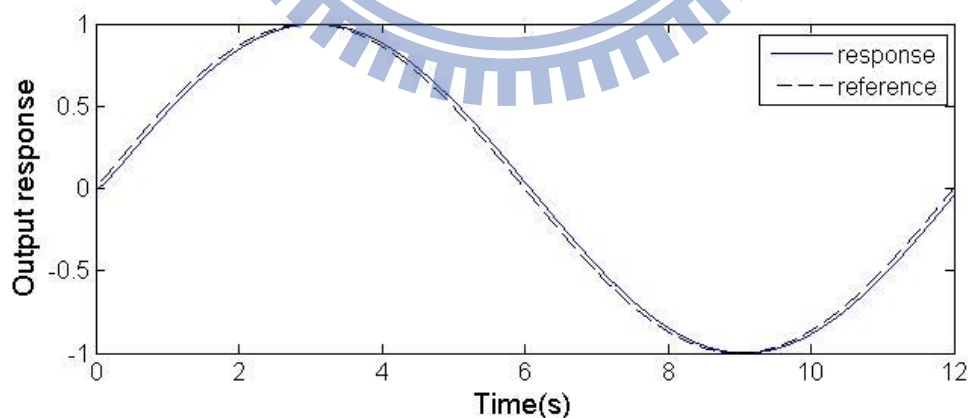


圖 4-11 one-delay 於 20% 資料遺失率補償之模擬結果

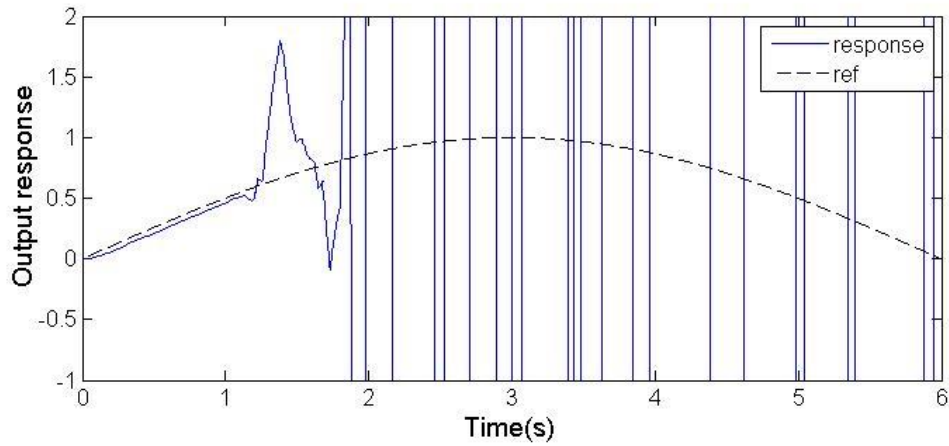
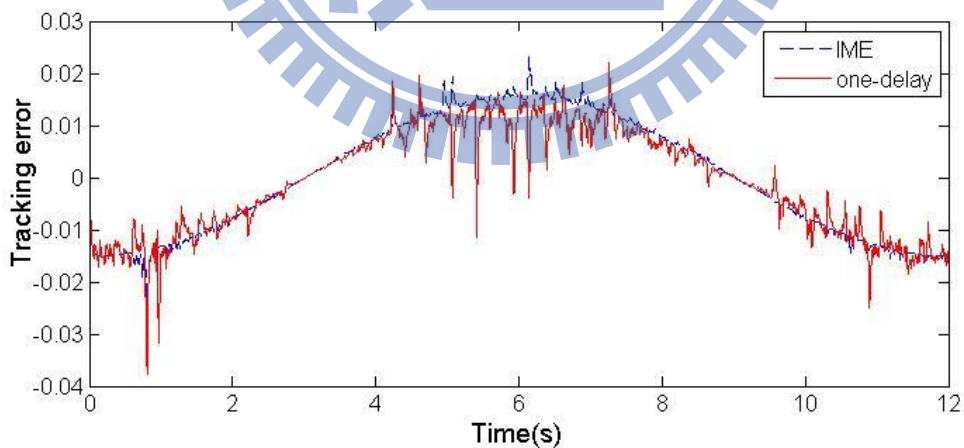


圖 4-12 IME 於 20% 資料遺失率補償之模擬結果

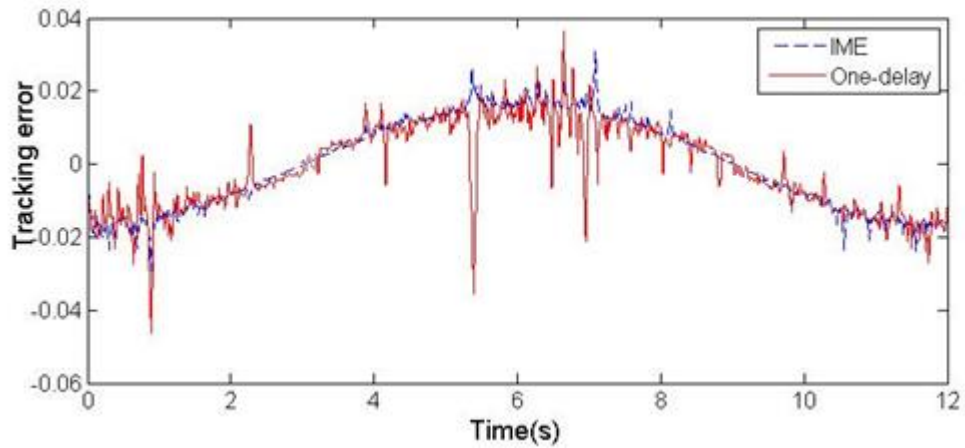
產生此模擬結果的原因為 IME 架構使用到的前幾筆資料有很大的機會也是透過補償得來的結果，若是先前補償的值與原先期望的結果已經有了不小的差異，再將此值列入當前補償的運算時會再將兩者間的誤差擴大。也就是說，經過不斷錯誤的補償，使得響應與原本命令的誤差越來越大，最終導致系統發散，而這在高階的 LSE 中特別明顯。

將補償架構替換成 IME\_1 並重新模擬，可得結果如圖 4-13 所示。



(a)





(b)

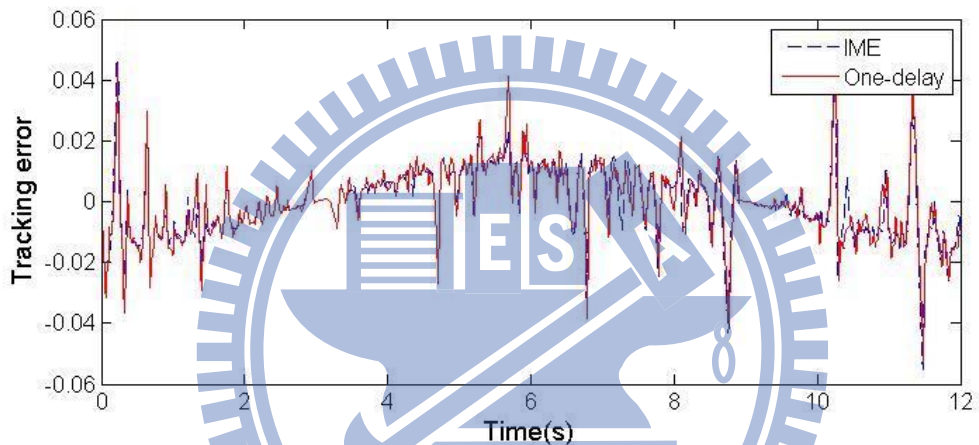


圖 4-13 IME\_1 與 one-delay 於不同資料遺失率補償之模擬結果比較(a)20%

(b)40% (c)60%

模擬結果顯示，IME\_1 架構在資料遺失率 60% 時，與 one-delay 估測器的補償效果相差不多，但是在資料遺失率 20% 與資料遺失率 40% 時能夠比 one-delay 估測器有著更好的補償效果，這是因為 IME\_1 架構在資料遺失率 60% 時幾乎都是切換到 one-delay 補償，然而在低資料遺失率時，IME\_1 能透過不同的 LSE 補償達到比 one-delay 估測器更好的效果。

#### 4.4.3 IME 的實驗結果

圖 4-14 與圖 4-15 分別為以 one-delay 估測器和 IME 補償架構應用於圖 2-11 網

路控制系統平台的實驗結果。由實驗結果圖可以看出，在 20% 資料遺失率的情況下，IME 補償架構的補償能力比 one-delay 估測器更為優秀。

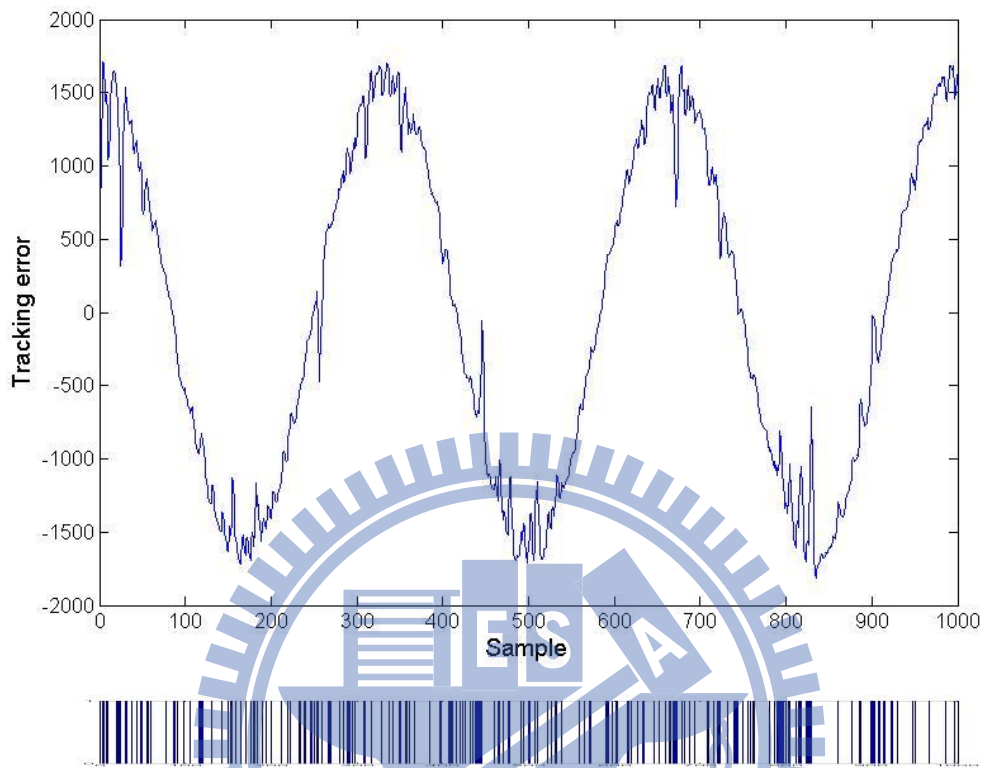


圖 4-14 one-delay 於 20% 資料遺失率補償之實驗結果

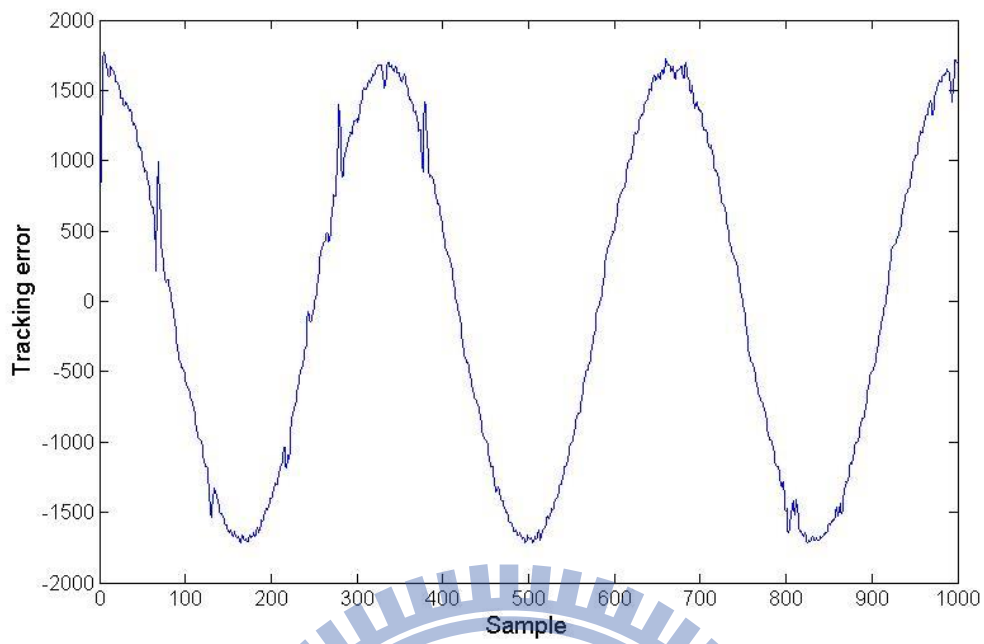


圖 4-15 IME 於 20% 資料遺失率補償之實驗結果

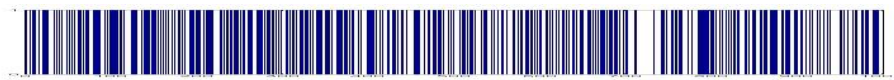
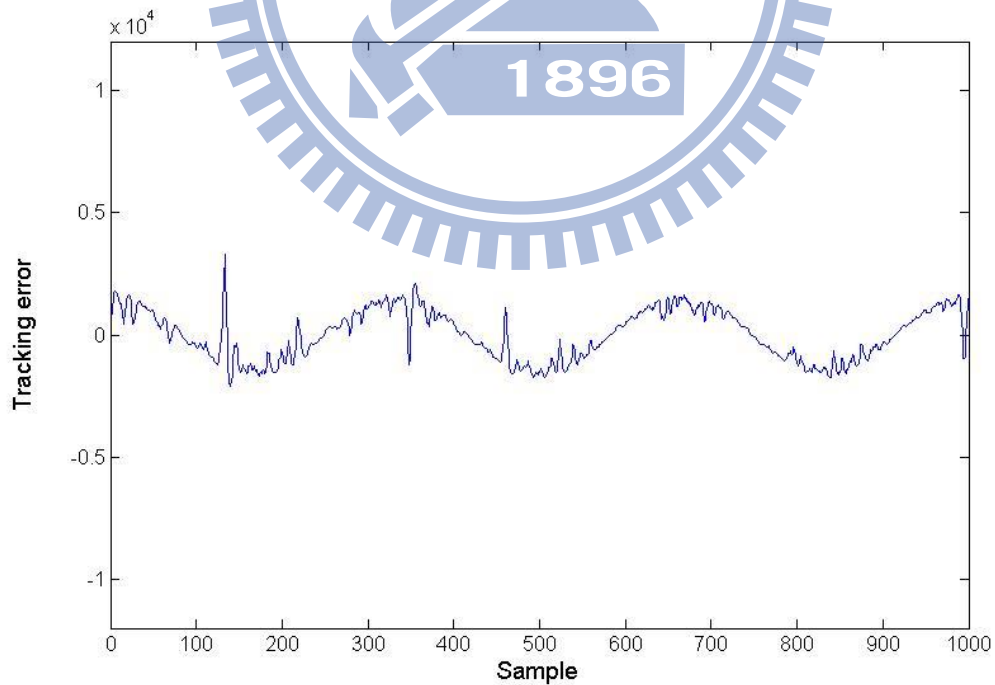


圖 4-16 One-delay 於 40% 資料遺失率補償之實驗結果

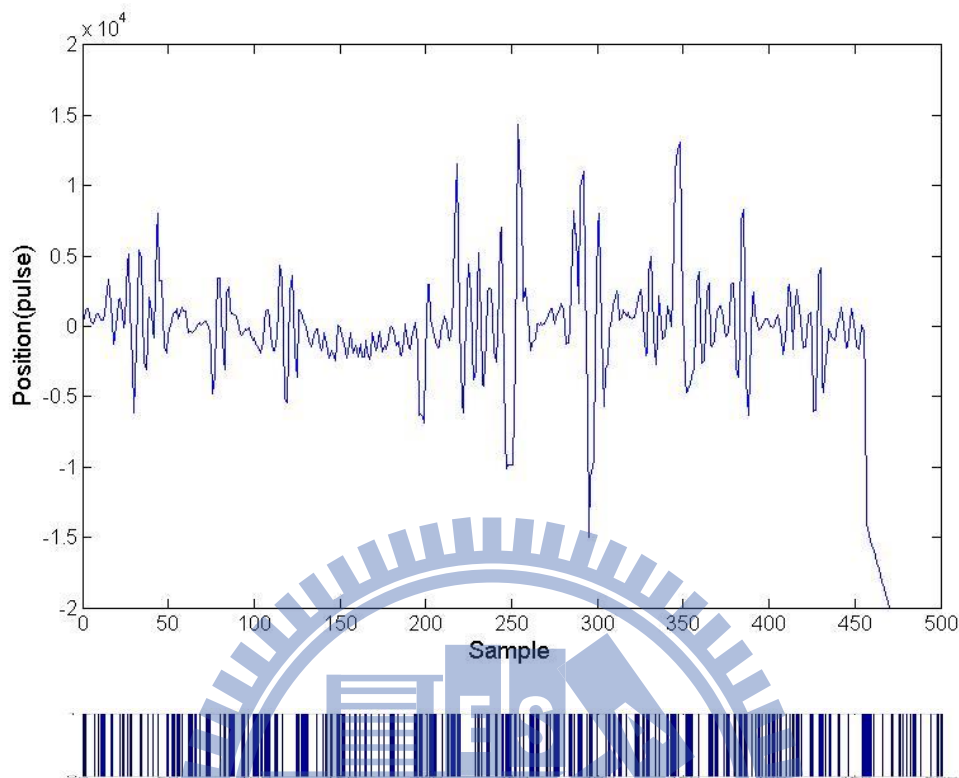


圖 4-17 IME 於 40% 資料遺失率補償之實驗結果

圖 4-16 與圖 4-17 分別為以 one-delay 估測器和 IME 補償架構應用於 40% 資料遺失率的網路控制系統中。與模擬的情形相似，IME 補償架構最終會導致系統發散。

由上一小節的模擬結果可知，IME 架構的補償策略並不完全通用於任何系統，因此為了在使用 IME 架構的同時能保持系統響應的穩定，必須對於 IME 的切換準則進行修正，而修正的原則為避免使用會導致系統發散的高階 LSE 做補償，也就是說，去除掉會導致系統發散的 LSE 並且針對各個不同的系統使用不同的 IME 補償架構。

將補償架構替換成 IME\_2 並重新實驗，可得實驗結果如圖 4-18 所示。

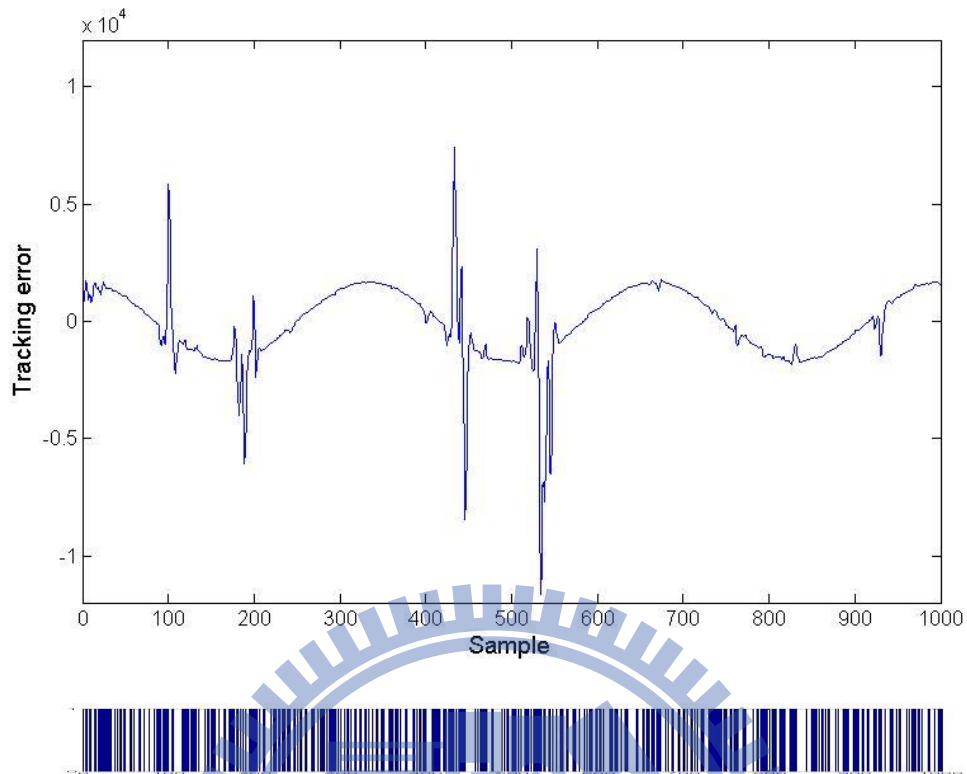


圖 4-18 IME\_2 於 40% 資料遺失率補償之實驗結果

雖然將補償區間更改後的 IME\_2 補償架構不會導致系統響應發散，但是從實驗結果圖中可以明顯的看到某些地方的誤差值有著劇烈的變化，補償效果甚至不如 one-delay 估測器。為了使響應的誤差減小，將再次更改 IME 補償架構，改以(4.18)式的 IME\_3 進行資料遺失補償，而實驗結果如圖 4-19，此時以 IME\_3 作為資料遺失補償架構的系統，得到了最佳的補償效果，將 40% 資料遺失的影響降至最低。

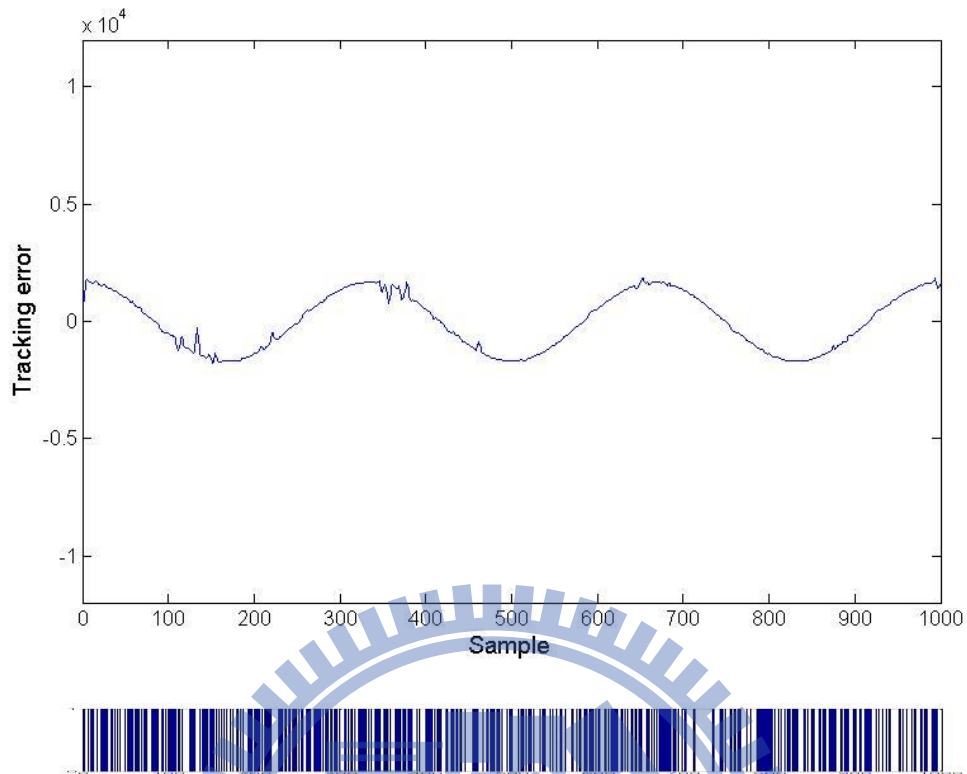


圖 4-19 IME 3 於 40% 資料遺失率補償之實驗結果

#### 4.5 各種資料遺失補償架構結果比較

本節中將針對上面提出的各種資料遺失補償架構進行追跡的方均根誤差 (root mean square error) 比較，將結果整理於表 4.1 與表 4.2。

表 4.1 不同資料遺失補償架構之模擬結果比較

	20% 資料遺失率	40% 資料遺失率	60% 資料遺失率
one-delay	0.0104	0.0128	0.0154
IME	unstable	-----	-----
IME_1	0.0082	0.0105	0.0147

表 4.2 不同資料遺失補償架構之實驗結果比較

	20% 資料遺失率	40% 資料遺失率
one-delay	1214.6	1814.6
IME	1099.7	unstable
IME_2	-----	2438.5
IME_3	-----	1587.1

透過表 4.1 以及表 4.2，整體來說，在高資料遺失率時方均根誤差比在低資料遺失率時大，且 IME 資料遺失補償架構在 20% 及 40% 資料遺失率時能夠獲得比 one-delay 補償低的方均根誤差值。若 IME 的補償導致系統不穩定時，則可以透過適當更改的切換策略穩定補償系統，在模擬中，將 IME 替換成 IME\_1，而在實驗中，將 IME 替換成 IME\_2 及 IME\_3。



## 第五章 資料預測控制器

### 5.1 預測控制器介紹

預測控制器之主要設計，其概念為首先將預定期望之系統響應與未來預測訊號長度，透過預測未來響應、最佳化指標與實際回授修正策略，即時計算出最符合預期響應的控制訊號。在此以一簡單的例子作預測控制器的比喻：

1. 我們可以想像有一工作團隊，目的為設計並實現出一個遠端操作控制系統，而我們的工作團隊的工作方法如下：在目標達成之前，在每個小時的開始時擬訂接下來四個小時的計畫藍圖，並執行第一個小時的工作內容。
2. 而訂定計畫的標準在於我們必須了解即將面臨何種問題，例如系統模型分析、設計、模擬與實現，這些都可以看成在訂定計畫時的操縱變數；另外，我們也必須將限制條件考慮進來，例如工作團隊對於系統模型的了解程度、設計的能力，與實現的能力，以上種種資訊在訂定計畫時皆為極重要的因素。
3. 掌握一切所需資訊後，工作團隊開始訂定接下來每個小時要做的工作，在考慮本身與系統條件限制的情況下，設計出最好的方法達成目標。計畫完成後即開始實行。
4. 第一個小時的工作結束後，工作團隊會審視這個小時的實際工作進度，由於計畫與實現往往存在著無法避免的誤差，種種因素猶如系統模型掌握不正確、工作成員的能力不足或低估，都會影響整體成果，所以在實作後蒐集這些資訊，加入接下來四個小時的計畫考慮因素中，重新設計後四個小時中每個小時的工作內容，如何以最大的效率完成計畫。接著進行下個小時的工作內容，一個小時後檢視成果並且設計接下來四個小時的策略，如此重複，直到目標達成為止。



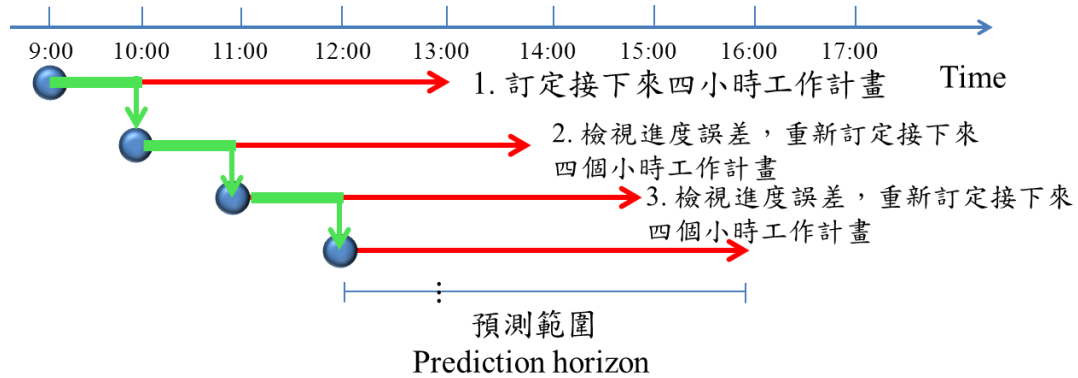


圖 5-1 預測控制器設計概念示意圖

在整個計畫流程中有五項關鍵變數：

1. 預測範圍(prediction horizon)：決定將計畫未來幾個小時內的工作內容，在上述的例子中預測範圍為四小時。
2. 移動式預測窗口(moving horizon window)：由於在每個小時( $t_i$ )都會重新計畫接下來四個小時( $T_p=4$ )的工作內容，如此的動作猶如一個移動的窗口(從  $t_i$  到  $t_i + T_p$ )
3. 移動時域控制(receding horizon control)：雖然每次都會將四個小時的工作計畫訂定，但是工作團隊只會執行第一個小時的計畫內容，而其他三個小時的計畫不會接下去執行，此方法稱為移動時域控制。
4. 每個時間點的狀態(status)：在每個開始計畫的時間點都會有根據上個小時的工作成果獲得的資訊，這些資訊決定著接下來的設計限制與結果，所以工作團隊必須根據每個時間點的狀態決定接下來的工作計畫。

設計出最佳工作計畫的準則：為了設計出最有效率的工作計畫，工作團隊必須根據目標訂定出評斷成果優劣的準則，此準則即為預期達到的目標與實際實行後結果之差距，而工作計畫的設計目標就是將此差距降為最低。

## 5.2 資料預測控制器(Data-based Predictive Controller, DPC)

本節將介紹一個由輸入資料與輸出資料設計而得的預測控制器(Data-based predictive controller, DPC)，與一般的控制器設計不同，此方法不需要詳細了解系統本身的資訊，而是僅以數值處理的方式直接設計有效的控制器，如圖 5-2 所示。

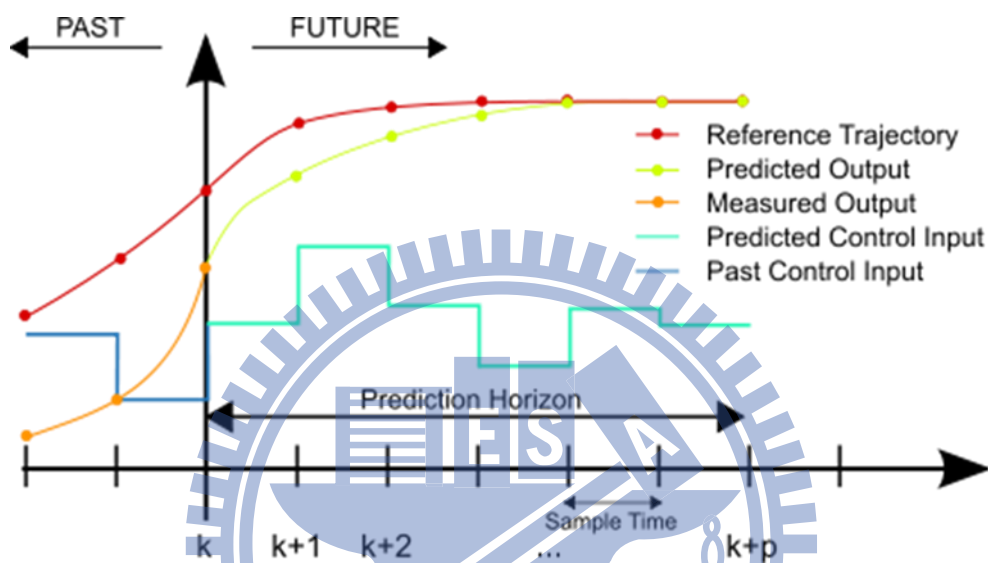


圖 5-2 資料預測控制器示意圖

假設欲控制的未知系統可以標準的  $n$  階  $r$  輸入、 $m$  輸出的離散時間系統表示

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_d v(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (5-1)$$

式子(5-1)中  $v(k)$  是  $d \times 1$  的未知週期性干擾。將受干擾的輸入與輸出資料假設為

$$\{u\} = \{u(0), u(1), u(2) \dots\} \quad \{y\} = \{y(0), y(1), y(2) \dots\} \quad (5-2)$$

目的為設計輸出回授控制器

$$u(k) = \sum_{i=1}^p Gu(k-i) + \sum_{i=1}^p Hy(k-i) \quad (5-3)$$

使下式的cost function最小化

$$J(k) = \sum_{i=0}^{p-1} y^T(k+q+i)Qy(k+q+i) + \sum_{i=0}^{p+q-1} u^T(k+i)Ru(k+i) \quad (5-4)$$

式子(5-4)中，控制器階數 $p$ 代表預測範圍(prediction horizon)，也就是說如果現在的時間指標為 $k$ 的話，則預測的輸出範圍會從 $y(k+q)$ 到 $y(k+q+p-1)$ 。必須注意的是，在之後的推導中，除了系統可能的階數上限和干擾的頻率數上限外，控制器的設計過程中都沒有假設任何與系統有關的訊息。

### 5.2.1 系統輸入輸出模型推導

資料預測控制器設計的第一步為推導未來多步的輸入輸出模型。為了簡化標誌，對於任何的 $s$ 和 $k$ ，“super-vector” $z_s(k)$ 被定義為 $sg \times 1$ 的向量，其內容為 $g \times 1$ 的 $z$ 向量從時間 $k$ 延伸到未來的時間 $k+s-1$ 。

$$z_s(k) \equiv \begin{bmatrix} z(k) \\ z(k+1) \\ \vdots \\ z(k+s-1) \end{bmatrix}_{sg \times 1} \quad (5-5)$$

使用 super-vectors 表達輸入與輸出，式子(5-4)可以改寫為

$$J(k) = [y_p(k+q)]^T Q y_p(k+q) + [u_{p+q}(k)]^T R u_{p+q}(k) \quad (5-6)$$

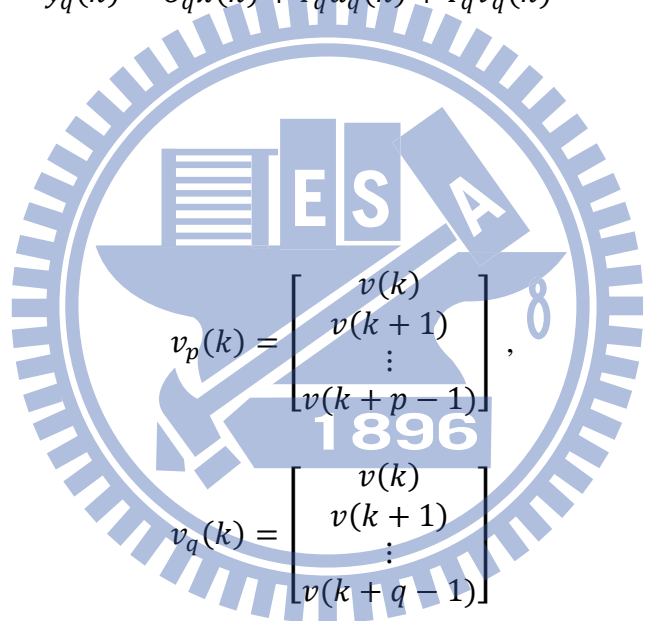
其中Q和R為

$$\begin{aligned} Q &= \text{diag}[Q_0, Q_1, \dots, Q_{p-1}] \\ R &= \text{diag}[R_0, R_1, \dots, R_{p+q-1}] \end{aligned} \quad (5-7)$$

將(5-1)式第二個方程式反覆帶入第一個方程式可得

$$\begin{aligned} x(k+p) &= A^p x(k) + E_p u_p(k) + \bar{E}_p v_p(k) \\ y_q(k) &= O_q x(k) + T_q u_q(k) + \bar{T}_q v_q(k) \end{aligned} \quad (5-8)$$

其中



$$\begin{aligned} v_p(k) &= \begin{bmatrix} v(k) \\ v(k+1) \\ \vdots \\ v(k+p-1) \end{bmatrix}, \\ v_q(k) &= \begin{bmatrix} v(k) \\ v(k+1) \\ \vdots \\ v(k+q-1) \end{bmatrix} \end{aligned} \quad (5-9)$$

$$\begin{aligned} E_p &= [A^{p-1}B, \dots, AB, B], \\ \bar{E}_p &= [A^{p-1}B_d, \dots, AB_d, B_d] \end{aligned} \quad (5-10)$$

$$O_q = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{q-1} \end{bmatrix} \quad (5-11)$$

$$T_q = \begin{bmatrix} D & & & \\ CB & D & & \\ \vdots & \ddots & \ddots & \\ CA^{q-2}B & \dots & CB & D \end{bmatrix},$$

$$\bar{T}_q = \begin{bmatrix} 0 & & & \\ CB_d & 0 & & \\ \vdots & \ddots & \ddots & \\ CA^{q-2}B_d & \dots & CB_d & 0 \end{bmatrix} \quad (5-12)$$

將(5-8)第一個方程式前面乘上 $O_q$ 可得

$$O_q x(k+p) = O_q A^p x(k) + O_q E_p u_p(k) + O_q \bar{E}_p v_p(k) \quad (5-13)$$

(5-13)式等式右手邊包含了系統的狀態，以及受到干擾的輸入，而這兩者皆為未知。此設計希望找到一個機制，能夠將這兩項所造成的影響移到方程式的係數上，使它們不會明確地出現在方程式的項目中。根據Phan et al. (1998)，假設存在 $m_q \times m_q$ 的 interaction matrix  $M$ ，則將乘積 $My_q(k)$ 加入式子(5-13)的右手邊，可得到

$$\begin{aligned} O_q x(k+p) &= O_q A^p x(k) + O_q E_p u_p(k) + O_q \bar{E}_p v_p(k) \\ &\quad + M[O_q x(k) + T_q u_q(k) + \bar{T}_q v_q(k)] - My_q(k) \\ &= (O_q A^p + MO_q)x(k) + O_q E_p u_p(k) \\ &\quad + MT_q u_q(k) + O_q \bar{E}_p v_p(k) + M\bar{T}_q v_q(k) - My_q(k) \quad (5-14) \\ &= (O_q A^p + MO_q)x(k) + (O_q E_p + [MT_q, 0_{qm \times (p-q)r}])u_p(k) \\ &\quad + (O_q \bar{E}_p + [M\bar{T}_q, 0_{qm \times (p-q)d}])v_p(k) - My_q(k) \end{aligned}$$

在式子(5-14)中，假設 $p \geq q$ ，並且加入額外的零向量 $[MT_q, 0_{qm \times (p-q)r}]$ ，

$[M\bar{T}_q, 0_{qm \times (p-q)d}]$  藉此將  $u_p(k)$  和  $v_p(k)$  的因素去除。將(5-14)最後的式子代入(5-8)的第二個式子產生輸出未來  $q$  步的表示式

$$\begin{aligned}
 y_q(k) &= (O_q A^p + M O_q) x(k-p) + (O_q E_p + [M T_q, 0_{qm \times (p-q)r}]) u_p(k-p) \\
 &\quad + (O_q \bar{E}_p + [M \bar{T}_q, 0_{qm \times (p-q)d}]) v_p(k-p) - M y_q(k-p) \\
 &\quad + T_q u_q(k) + \bar{T}_q v_q(k)
 \end{aligned} \tag{5-15}$$

上式雖然已經是設計的目標，但仍必須將未知狀態  $x(k-p)$  和未知干擾去除。因此假設存在  $M$  使得

$$O_q A^p + M O_q = 0 \tag{5-16}$$

$$(O_q \bar{E}_p + [M \bar{T}_q, 0_{qm \times (p-q)d}]) v_p(k-p) + \bar{T}_q v_q(k) = 0, \forall k \geq p \tag{5-17}$$

若  $M$  存在，那麼式子(5-15)會變為

$$y_q(k) = (O_q E_p + [M T_q, 0_{qm \times (p-q)r}]) u_p(k-p) + T_q u_q(k) - M y_q(k-p) \tag{5-18}$$

$$y_q(k+p) = (O_q E_p + [M T_q, 0_{qm \times (p-q)r}]) u_p(k) + T_q u_q(k+p) - M y_q(k) \tag{5-19}$$

接下來將(5-19)式右手邊調整為只需要歷史的輸出訊息，令

$$q = p^* \quad p - q = q^* \quad k + q = k^*$$

$$k = k^* - p^* \quad p = q^* + p^*$$

將變數變換帶入(5-19)式

$$y_p(k^* + q^*) = (O_q^* E_{q^*+p^*} + [MT_{p^*}, 0_{p^*m \cdot q^*r}])u_{q^*+p^*}(k^* - p^*) + T_{p^*}u_{p^*}(k^* + q^*) - My_{p^*}(k^* - p^*) \quad (5-20)$$

在分別將 $k^*$ 、 $p^*$ 、 $q^*$ 以 $k$ 、 $p$ 、 $q$ 取代

$$y_p(k + q) = (O_q E_{p+q} + [MT_p, 0_{pm \cdot qr}])u_{p+q}(k - p) + T_p u_p(k + q) - My_p(k - p) \quad (5-21)$$

(5-21)即為期望的未來 $p$ 步輸入輸出模型。

## 5.2.2 Interaction Matrix 的存在條件

如上面推導的結果所示，假設interaction matrix的存在使得系統模型能夠只用輸出的資料來呈現，因此interaction matrix存在相當重要。此小節必須驗證式子(5-16)和(5-17)中 $M$ 存在的條件。

將式子(5-17)的第一項係數分解

$$(O_q \bar{E}_p + [M\bar{T}_q, 0_{qm \times (p-q)d}]) = [O_q A^{p-q} \bar{E}_p + M\bar{T}_q, O_q \bar{E}_{p-q}] \quad (5-22)$$

則(5-17)式可改寫為

$$[O_q A^{p-q} \bar{E}_p + M\bar{T}_q, O_q \bar{E}_{p-q}, \bar{T}_q] D_t = 0 \quad (5-23)$$

其中 $D_t$ 為

$$D_t = [v_{p+q}(0), v_{p+q}(1), v_{p+q}(2), \dots] \quad (5-24)$$

因為干擾被定義為週期性， $D_t$ 的rank被限制為相異干擾頻率的數量。若相異的干擾頻率數量是 $f$ ，則 $D_t$ 的rank是 $2f+1$ 。其中1代表可能存在的常數偏差(Darling and Phan, 2003)。如果 $D$ 代表 $D_t$ 中 $2f+1$ 線性獨立的行向量所構成的矩陣，則 $D_1$ 、 $D_2$ 、 $D_3$ 為 $D$ 矩陣列向量的三個部分且分別對應到 $O_q A^{p-q} \bar{E}_p + M \bar{T}_q$ 、 $O_q \bar{E}_{p-q}$ 和 $\bar{T}_q$ ，如此可以將(5-23)改寫為

$$M[O_q, \bar{E}_q D_1] = -[O_q A^p, O_q A^{p-q} \bar{E}_p D_1, O_q \bar{E}_{p-q} D_2, + \bar{T}_q D_3] \quad (5-25)$$

(5-25)式為一組線性方程式，因此，只要 $[O_q, \bar{E}_q D_1]$ 為滿秩矩陣就能夠保證interaction matrix  $M$ 的存在。(5-25)式為 $qm \times (n + 2f + 1)$ 的方程式組，若要確保為滿秩矩陣，必須選擇 $q$ 使得 $qm \geq n + 2f + 1$ 。

在先前的討論中，使用了變數變換 $q = p^*$ ，而我們知道 $p^*$ 及等於 $p$ ，因此構成(5-21)式的條件為

$$pm \geq n + 2f + 1 \quad (5-26)$$

在控制器設計的過程中，控制器階數 $p$ 的假設必須滿足(5-26)式。

### 5.2.3 資料預測控制器公式推導

系統的輸出入模型與 cost function 在上面的小節裡皆已推導完成。控制器設計的最後步驟，只剩下將(5-21)式代入(5-6)式中，並令 $u_{p+q}(k)$ 最小化。但在這之前，必須先將(5-21)式重新整理，使 $u_{p+q}(k)$ 明確的顯現在方程式中。首先



$$O_p E_{p+q} = [O_p A^q E_p, O_p E_q] \quad (5-27)$$

(5-21)式右手邊第一項可改寫為

$$\begin{aligned} & (O_q E_{p+q} + [MT_p, 0_{pm \cdot qr}])u_{p+q}(k-p) \\ &= [O_p A^q E_p + MT_p, O_p E_q] \begin{bmatrix} u_p(k-p) \\ u_q(k) \end{bmatrix} \\ &= (O_p A^q E_p + MT_p)u_p(k-p) + O_p E_q u_q(k) \end{aligned} \quad (5-28)$$

如此可將(5-21)式改寫成

$$\begin{aligned} y_p(k+q) &= (O_p A^q E_p + MT_p)u_p(k-p) \\ &\quad + O_p E_q u_q(k) + T_p u_p(k+q) - M y_p(k-p) \\ &= (O_p A^q E_p + MT_p)u_p(k-p) \\ &\quad + [O_p E_q, T_p]u_{p+q}(k) - M y_p(k-p) \end{aligned} \quad (5-29)$$

令變數

$$\begin{aligned} P_1 &= O_p A^q E_p + MT_p \\ P_2 &= M \\ W &= [O_p E_q, T_p] \end{aligned} \quad (5-30)$$

則(5-29)式可以簡化為

$$y_p(k+q) = P_1 u_p(k-p) - P_2 y_p(k-p) + W u_{p+q}(k) \quad (5-31)$$

將(5-31)式代入(5-6)式並將 $u_{p+q}(k)$ 最小化可得

$$u_{p+q}(k) = A_1 u_p(k-p) - A_2 y_p(k-p) \quad (5-32)$$

其中

$$A_1 = -BP_1, \quad A_2 = -BP_2, \quad B = (R + W^T Q W)^+ W^T Q \quad (5-33)$$

(5-32)式定義了輸出回授形式的預測控制器

$$u(k) = \sum_{i=1}^p G_r u(k-i) + \sum_{i=1}^p H_r y(k-i) \quad (5-34)$$

其中控制器參數 $[G_p \ G_{p-1} \ \cdots \ G_1]$ 和 $[H_p \ H_{p-1} \ \cdots \ H_1]$ 分別為 $A_1$ 和 $A_2$ 前 $r$ 列。

## 5.2.4 控制器參數設計

若系統的參數與干擾的性質已知時，可以從(5-30)式和(5-33)式設計最佳化的預測控制器。但是即使這些資訊都未知，仍然可以直接從輸出入的資料來設計控制器，而這即是資料預測控制器(data-based predictive controller, DPC)呈現出的最大特點之一。

為了得知參數 $A_1$ 、 $A_2$ 和 $B$ ，必須先獲得 $P_1$ 、 $P_2$ 和 $W$ 的值，這可以藉由改寫(5-31)式得到

$$y_p(k+q) = [P_1 \ P_2 \ W] \begin{bmatrix} u_p(k-p) \\ -y_p(k-p) \\ u_{p+q}(k) \end{bmatrix} \quad (5-35)$$

考慮到對於任何  $k$  值(5-35)式都必須成立，因此當  $k \geq p$  時，

$$Y = [y_p(p+q) \ y_p(p+q+1) \ \cdots \ y_p(k+q+l-1)] \quad (5-36)$$

$$V = \begin{bmatrix} u_p(0) & u_p(1) & \cdots & u_p(l-1) \\ -y_p(0) & -y_p(1) & \cdots & -y_p(l-1) \\ u_{p+q}(p) & u_{p+q}(p+1) & \cdots & u_{p+q}(p+l-1) \end{bmatrix} \quad (5-37)$$

其中  $l$  為可用的輸出入資料點長度，則未知參數  $P_1$ 、 $P_2$ 、 $W$  可由下式計算出

$$[P_1 \ P_2 \ W] = YV^+ = YV^T(VV^T)^+ \quad (5-38)$$

(5-38)式給出了  $P_1$ 、 $P_2$  和  $W$  的最小平方近似解，將此結果代入(5-33)式即能夠得到控制器的參數  $A_1$ 、 $A_2$  與  $B$ 。

### 5.3 資料預測控制器實現

在設計資料預測控制器時，必須知道參數  $A_1$ 、 $A_2$  與  $B$  的值，也就是必須先得到  $Y$  和  $V$  兩個矩陣的值，因此在系統啟動前先收集足夠多的輸入與輸出資料來滿足  $Y$  和  $V$  矩陣。以下為資料預測控制器的設計步驟

1. 紀錄系統的輸入與輸出資訊滿足  $V$  和  $Y$  矩陣所需的資料
2. 初始化控制器參數  $p$ 、 $q$ 、 $l$ 、 $Q$ 、 $R$  ( $p$  為 horizon length、 $q$  為 prediction horizon、 $l$  為資料點個數、 $Q$  和  $R$  為權重函數)
3. 經由  $V$ 、 $Y$  矩陣計算  $P_1$ 、 $P_2$  和  $W$
4. 經由  $P_1$ 、 $P_2$  和  $W$  計算  $A_1$ 、 $A_2$  與  $B$
5. 將  $A_1$  與  $A_2$  代入(5-32)式，即可求得預測的控制命令

本研究使用的馬達輸入電壓範圍為 $\pm 10V$ ，工作模式為速度模式，馬達轉一圈輸出 1000 個 pulse。在控制系統啟動前輸入長達 10 秒的 5V 振幅 PRBS 訊號，輸出馬達位移角度，取樣時間為 50ms。圖 5-3(a)、(b)分別為系統的輸入與輸出訊號，將輸入訊號與輸出訊號記錄下來，令 $p = 10$ 、 $q = 1$ ，並且調整 Q 和 R 值一直到輸出響應達到可接受的程度。資料預測控制器設計完成的控制結果如圖 5-4 所示。

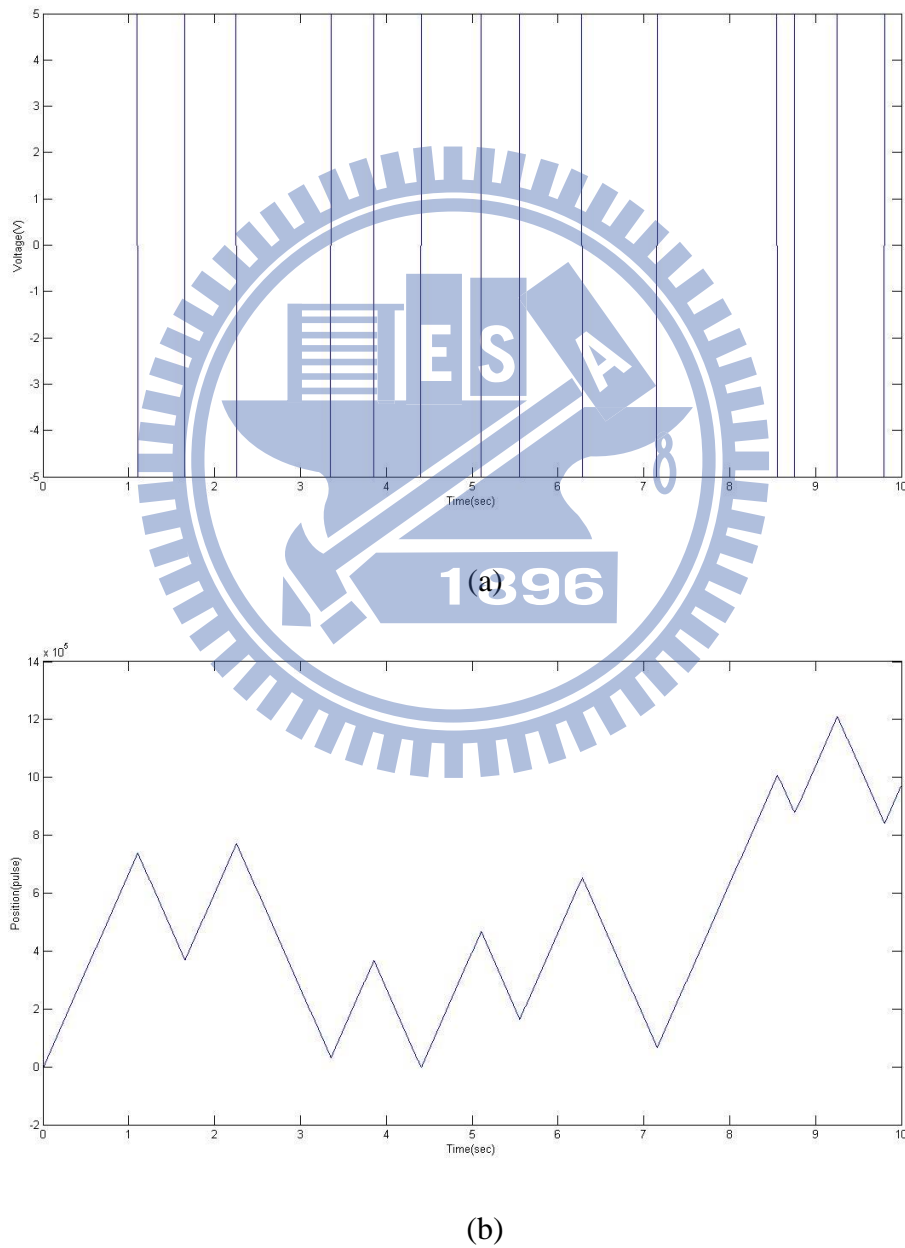


圖 5-3 系統鑑別的輸入(a)與輸出(b)

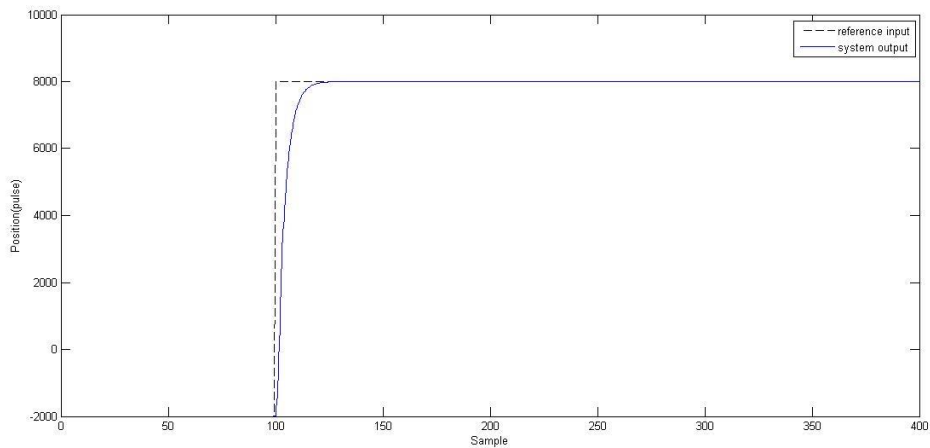


圖 5-4 資料預測控制器設計於控制系統響應圖

以同樣的設計方式，將資料預測控制器實現於 Truetime 建立的網路控制模擬環境中，圖 5-5(a)和圖 5-5(b)分別為使用兩個不同的受控體轉移函數以及不同的輸入命令所得到的結果，圖 5-6 為受控體轉移函數為 $\frac{6}{s+6}$ 時不同 Q 值設計的模擬結果。不管在實驗中或是模擬中，設計資料預測控制器時都不需事先知道系統的任何資訊，只憑著系統輸入和輸出的訊息以及命令訊號就能達到令人滿意的響應，這就是資料預測控制器強大的地方。

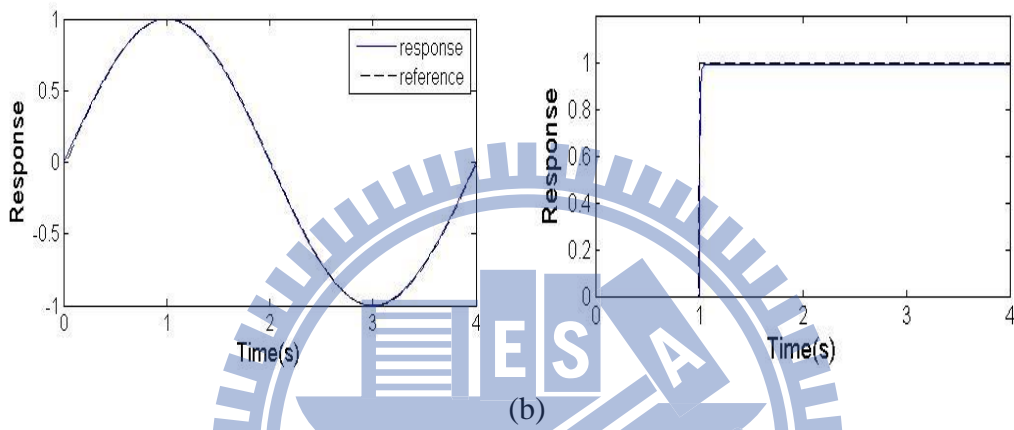
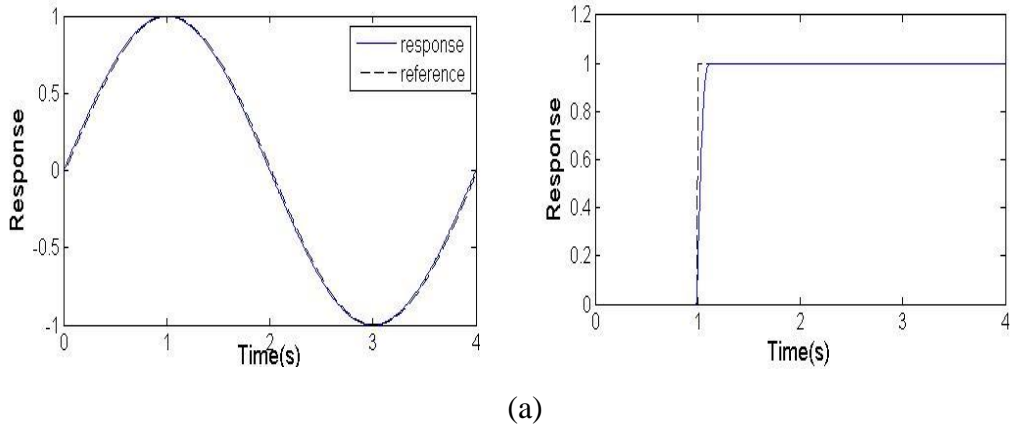
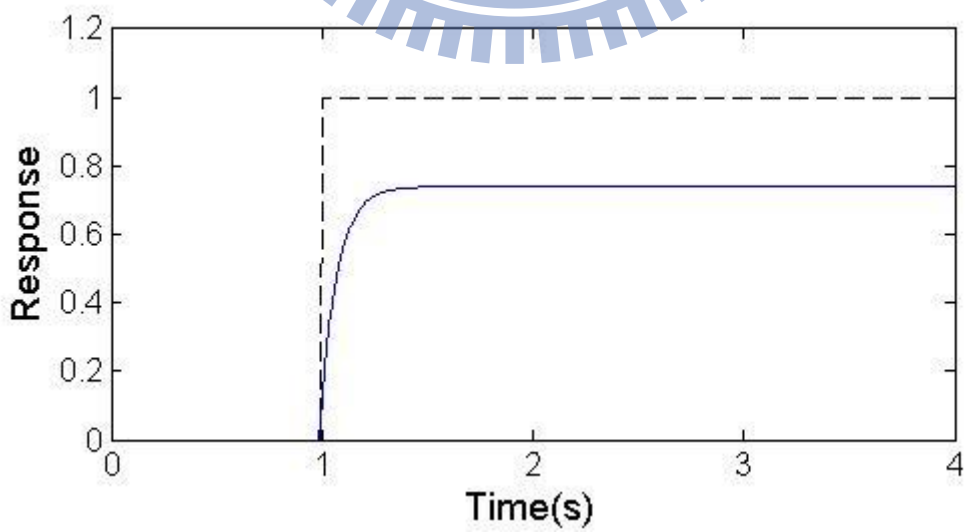


圖 5-5 資料預測控制器設計於不同系統與不同命令之模擬結果 (a)受控體轉移

函數為  $\frac{6}{s+6}$  (b)受控體轉移函數為  $\frac{1}{s^2+5s+4}$



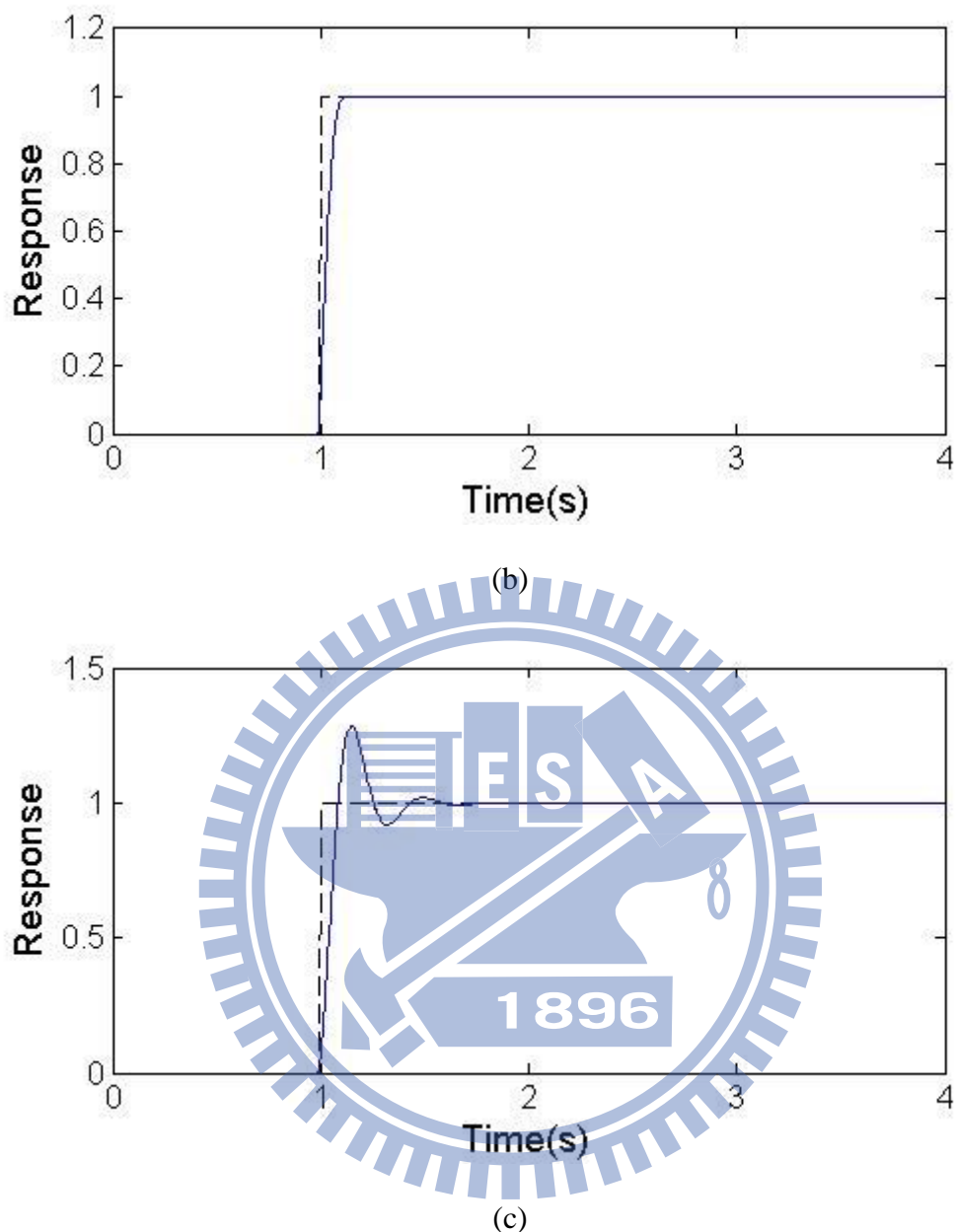


圖 5-6 資料預測控制器不同 Q 值設計的模擬結果(a)Q=10 (b)Q=1000 (c)Q=10<sup>12</sup>

#### 5.4 DPC 應用於網路資料遺失補償

資料預測控制器能夠預測未來多步時間點的控制命令，將資料預測控制器應用於網路控制系統的補償中，主要的概念為傳送的資料封包內不只包含現在的控制命令，同時也包含了預測出的未來控制命令。因此，在受控端設立一個 buffer，每次傳送封包時，未來的控制命令會隨著傳送並儲存在受控端的 buffer 中，而一

但有封包遺失的情形發生，受控端可以即時存取 buffer 中最新一筆的控制命令作為封包遺失的補償，如圖 5-7 所示。其中 buffer 的設定為，每當有新的封包抵達時 buffer 的內容會被覆蓋以確保 buffer 的內容是一直是最新的，受控端會依序使用 buffer 內的資料，而此策略可以補償的連續資料遺失值即為 buffer 的長度，因此 buffer 長度的設定可以合理的選擇為預測命令的長度(prediction horizon)。此設計的中心思想為，如果可以使用最新接收到資料，若無法取得及時的命令則使用 buffer 內的預測命令資料。圖 5-8 為將 DPC 網路補償架構實現於 Truetime 模擬環境中建構的網路控制系統模擬結果圖，此時的資料遺失率為 40%，由結果可以看出以 DPC 控制處理資料遺失補償的效果相當理想。

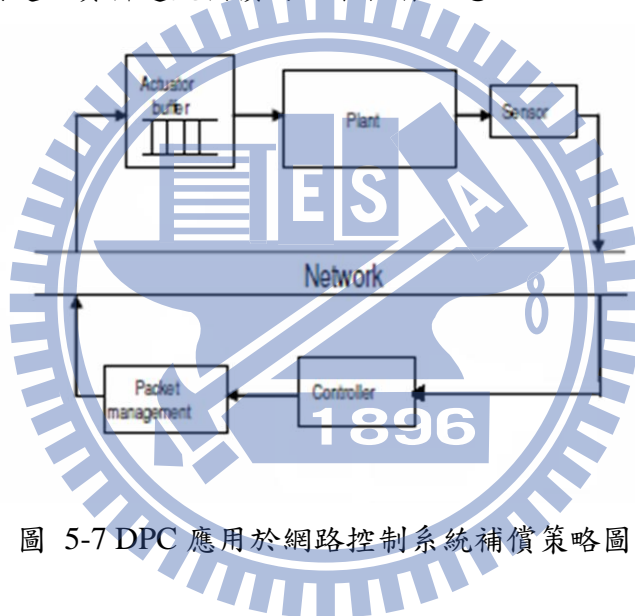


圖 5-7 DPC 應用於網路控制系統補償策略圖

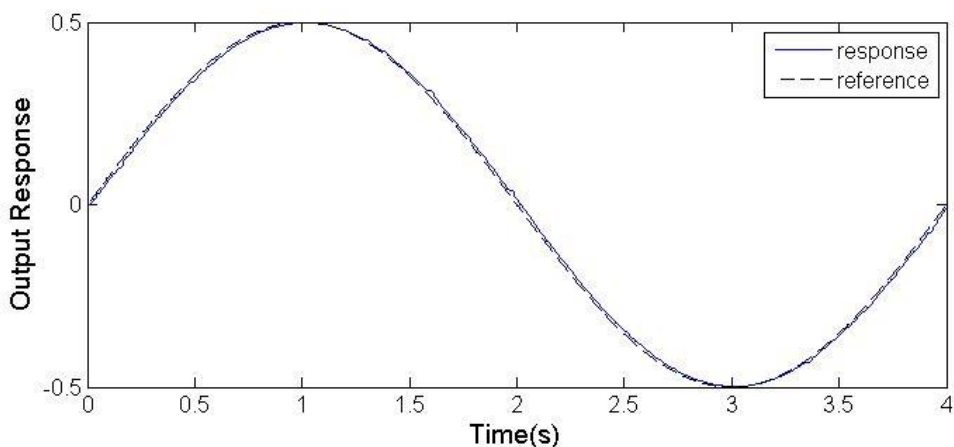


圖 5-8 DPC 應用於封包遺失補償模擬圖(40%資料遺失率)



## 5.5 DPC 與 IME 資料遺失補償效能比較

### 5.5.1 不同命令訊號下的資料遺失補償

在此節中將分析資料預測控制器(DPC)和智慧型訊息估測器(IME)兩者運用於網路封包遺失時的補償效果。模擬環境為 TrueTime 系統，使用的受控體轉移函數為 $\frac{1}{s^2+5s+4}$ ，取樣頻率為 10ms，IME 資料遺失補償和 DPC 資料遺失補償的實現方式分別如圖 4-9 與圖 5-9 所示。圖 5-10 為兩種補償方式在不同資料遺失率下的弦波追跡誤差，由圖可以看出，DPC 與 IME 在低資料遺失率時補償效果相近，但是隨著資料遺失率的上升，IME 的補償能力會下降，而 DPC 依然能夠維持不錯的補償效能。

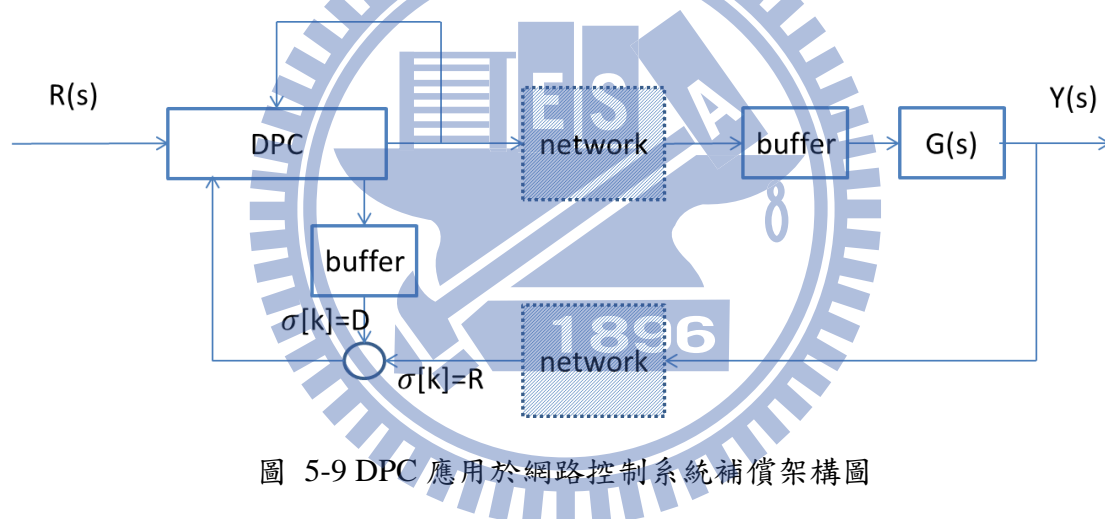
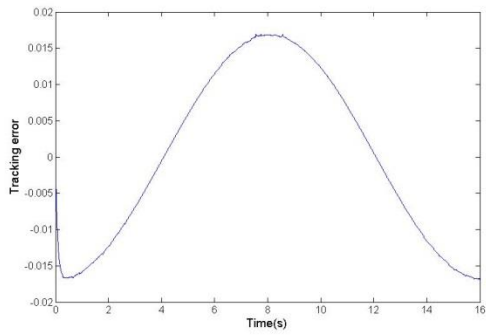
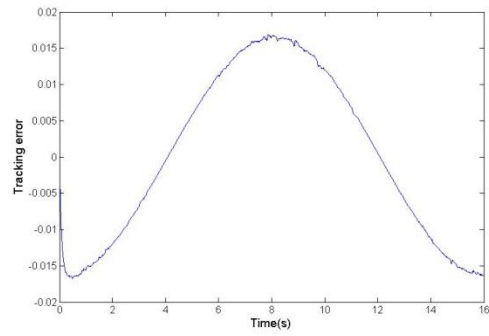


圖 5-9 DPC 應用於網路控制系統補償架構圖

$\rho = 10\%$

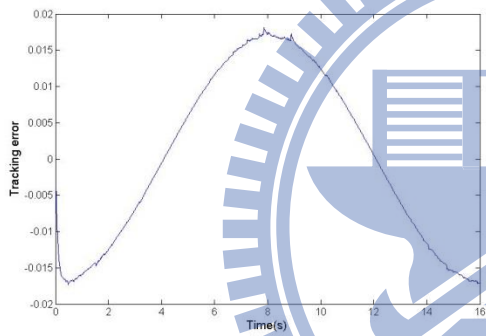


(a)

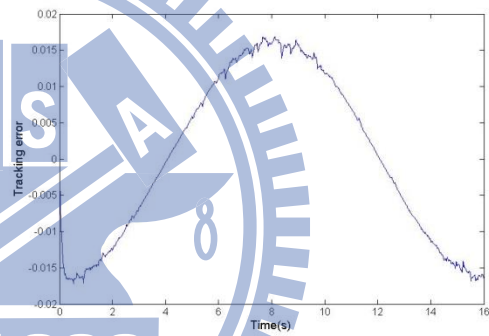


(d)

$\rho = 30\%$

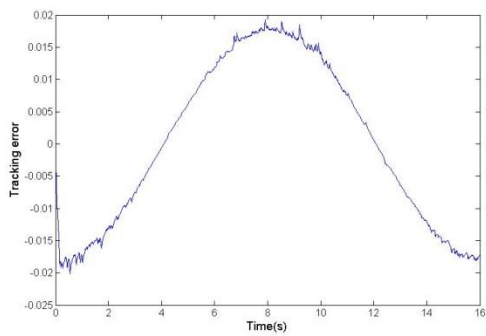


(b)

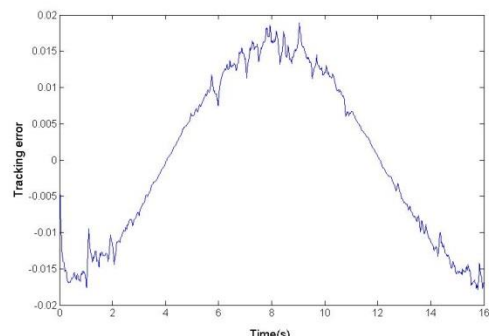


(e)

$\rho = 50\%$



(c)



(f)

圖 5-10:不同資料遺失率下兩種補償方式的弦波追蹤誤差(a)(b)(c)DPC

(d)(e)(f)IME

為了確保分析的結果不只限於單一情形而是能廣泛的符合不同的系統，除了以上述的弦波命令測試外，另外將分別以圖 5-11 的方波訊號和圖 5-12 的三角波訊號兩種輸入訊號測試 DPC 與 IME 補償方式於網路資料遺失補償時的效能。圖 5-13 為輸入方波訊號的測試結果，如圖所示，在低資料遺失率時兩種補償方式都能達到令人滿意的結果，而當資料遺失率上升時，IME 補償方式在命令瞬間變化的區間有著不小的誤差變動，並且有若干的延遲產生。DPC 補償方式在資料遺失率 30% 時幾乎不受影響，而在資料遺失率 50% 時某些地方有著少數的延遲，但大部分能維持完好的波形。圖 5-14 為輸入三角波訊號的測試結果，模擬結果與弦波測試命令相似，在低資料遺失率時，以 DPC 作為補償方式的補償結果差異不大，在資料遺失率 50% 時才能明顯的看到誤差變動。IME 補償方式在資料遺失率大於 30% 時就有明顯的誤差變動，而且相比於 DPC 的誤差變動也來的更大。

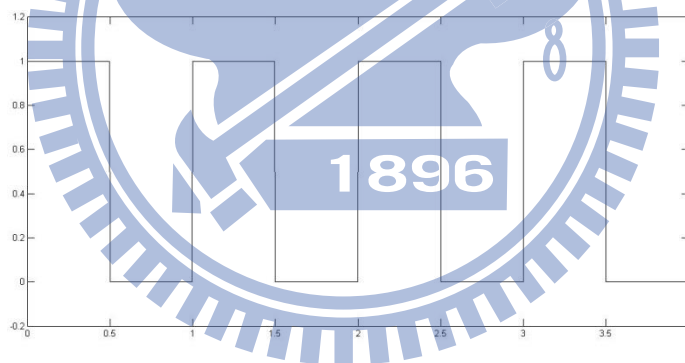


圖 5-11 方波輸入訊號

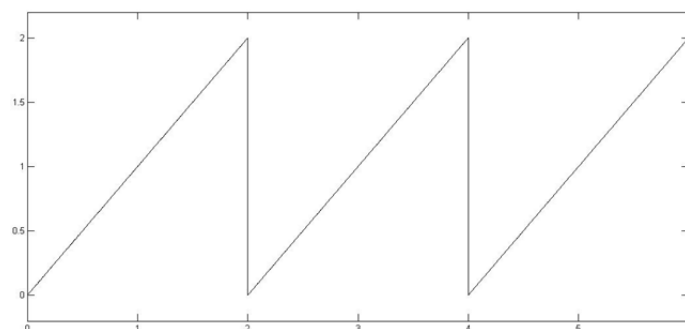
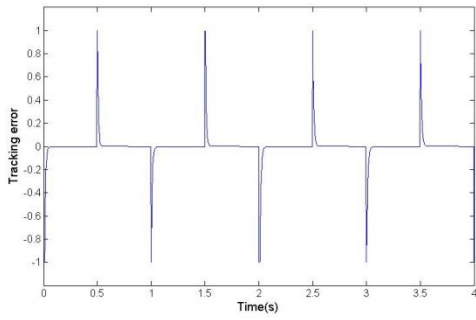
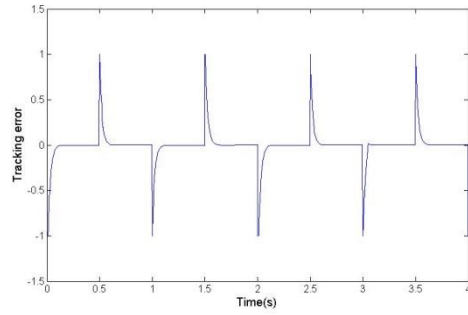


圖 5-12 三角波輸入訊號

$\rho = 10\%$

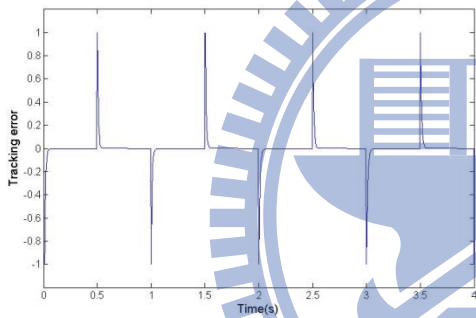


(a)

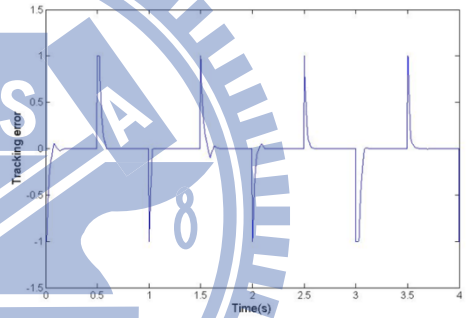


(d)

$\rho = 30\%$

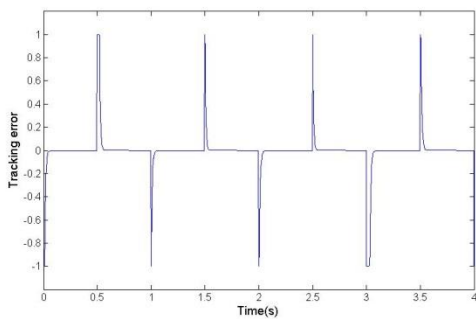


(b)

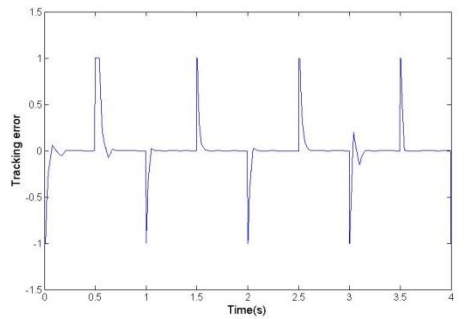


(e)

$\rho = 50\%$



(c)

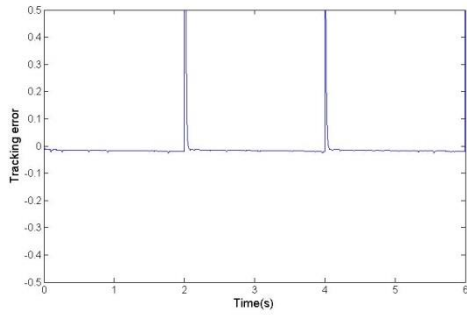


(f)

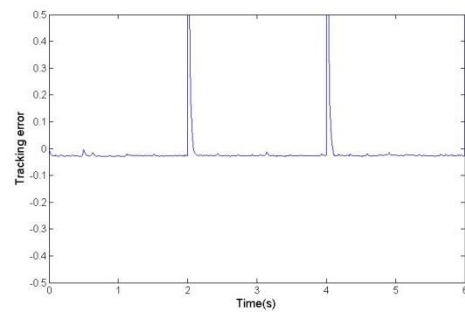
圖 5-13 不同資料遺失率下兩種補償方式的方波追跡誤差(a)(b)(c)DPC

(d)(e)(f)IME

$\rho = 10\%$

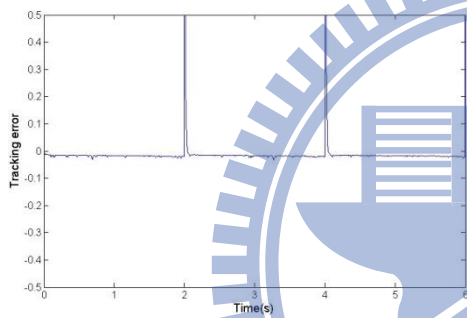


(a)

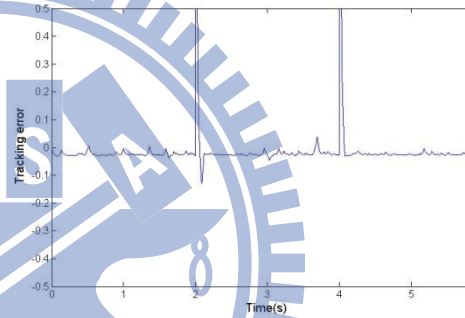


(d)

$\rho = 30\%$

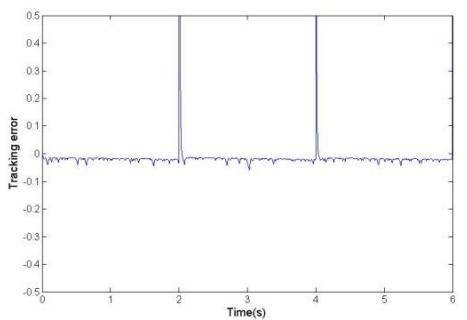


(b)

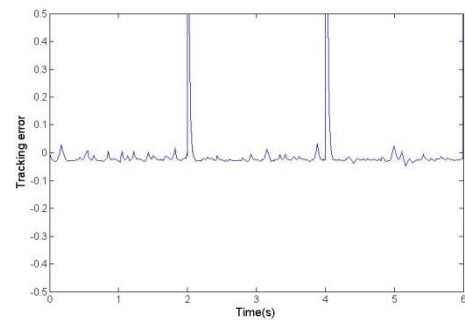


(e)

$\rho = 50\%$



(c)



(f)

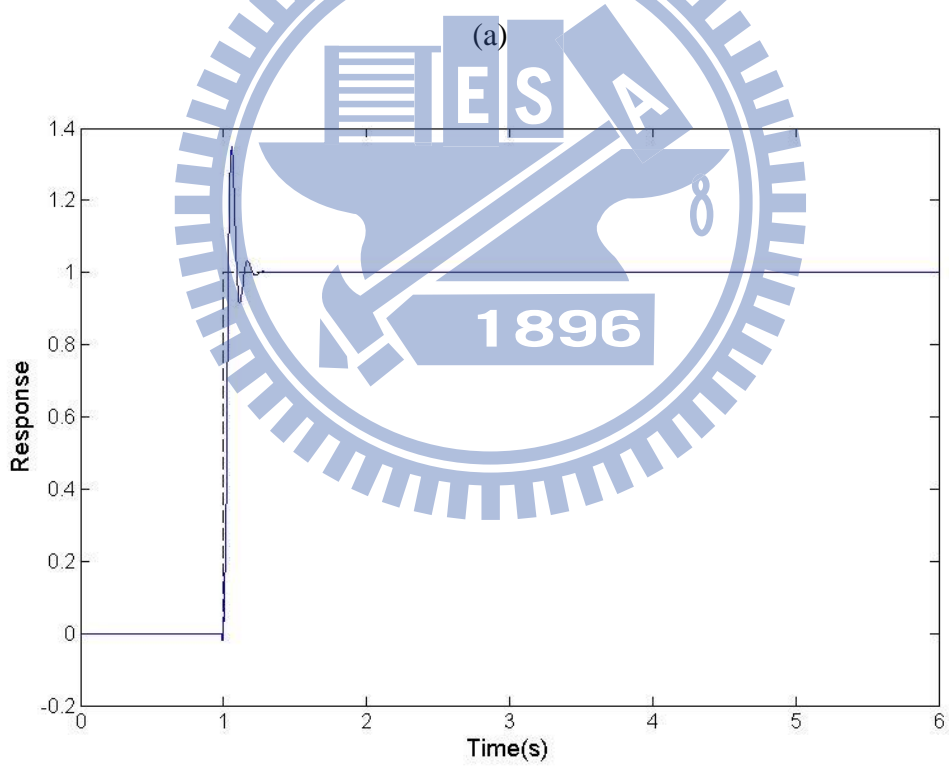
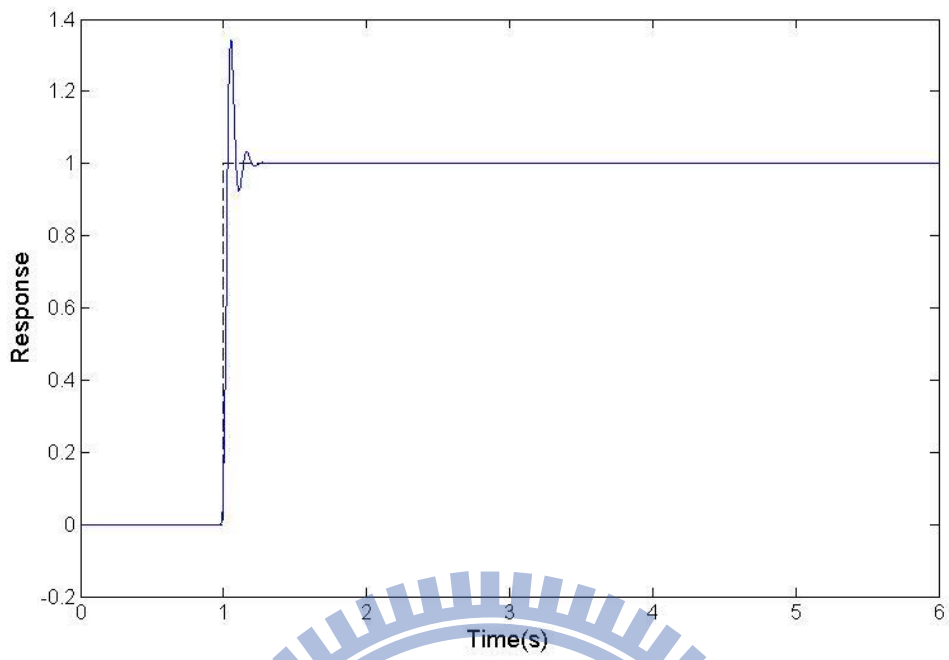
圖 5-14 不同資料遺失率下兩種補償方式的三角波追跡誤差(a)(b)(c)DPC

(d)(e)(f)IME

## 5.5.2 DPC 於網路資料遺失補償的缺點

從上一小節的觀察可知，以 DPC 作為網路訊息遺失的補償方式能夠達到比 IME 估測器更好的補償結果。然而，DPC 控制器的架構是建立在系統鑑別上的，因此若是系統鑑別的階段出了差錯，導致鑑別出來的系統與原系統有某種程度上的差異，則 DPC 的控制能力會下降，而以 DPC 作為網路補償的架構也會受到影響。圖 5-15(a)為以 DPC 作為控制器的系統步階響應圖，與先前討論不同的地方在於 DPC 控制器的參數  $Q$  並沒有經過適當的調整，此時權重函數  $Q$  值設為  $10^8$  而正常設計下的  $Q$  值為  $10^6$ ，控制器設計不夠妥善，由模擬結果可以看出步階響應的峰值過大。圖 5-15(b)為 30% 資料遺失時以此 DPC 補償的模擬結果，如圖所示，雖然 DPC 抑制了訊息遺失的影響，但因為控制器設計不良，補償後的結果即是沒有訊息遺失的控制結果。當資料遺失增大為 50% 時，結果如圖 5-15(c)所示，與上一小節討論中方波訊號的測試結果相比較，DPC 補償的效能明顯下降，控制誤差增大。

從上述的結果可知，DPC 控制器的設計會巨大的影響系統響應，若是設計時出了差錯，以 DPC 補償網路信息遺失的架構也會連帶受到影響，不只控制不良，在有資料遺失的情形下甚至會使整體響應變得更差。



(b)

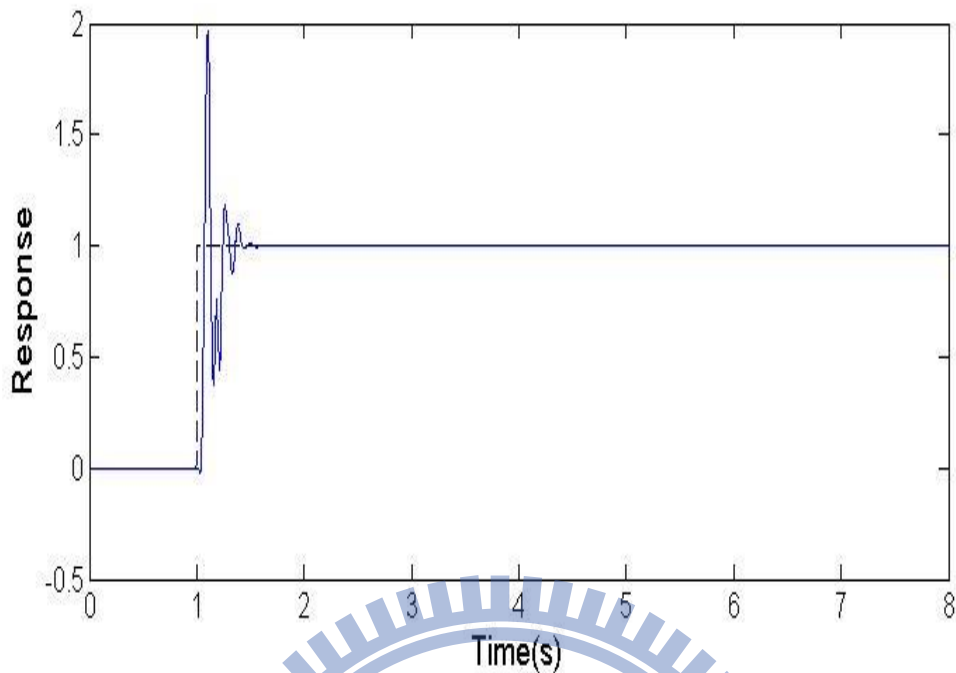
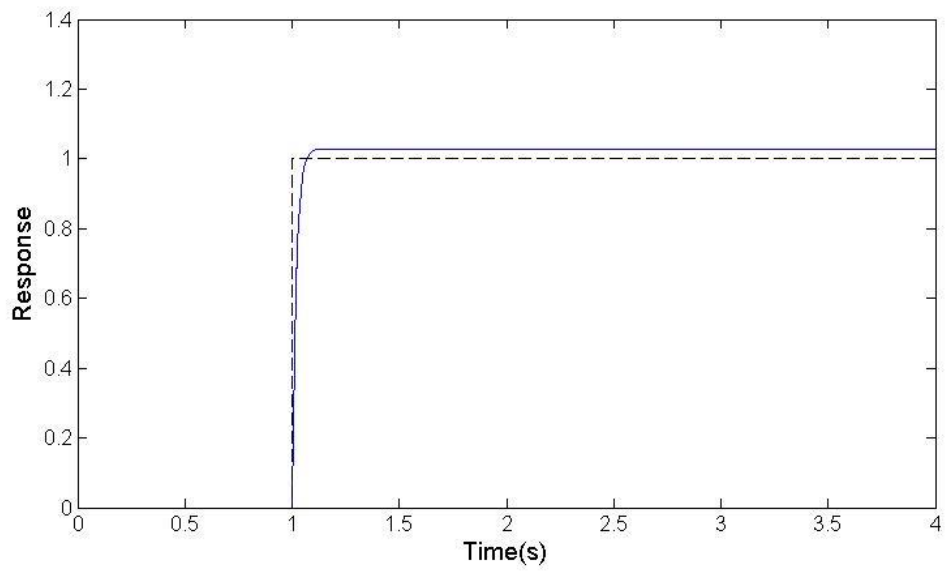


圖 5-15 DPC 權重函數  $Q=10^8$  時於不同資料遺失率下的補償響應圖(a)0% (b)30%

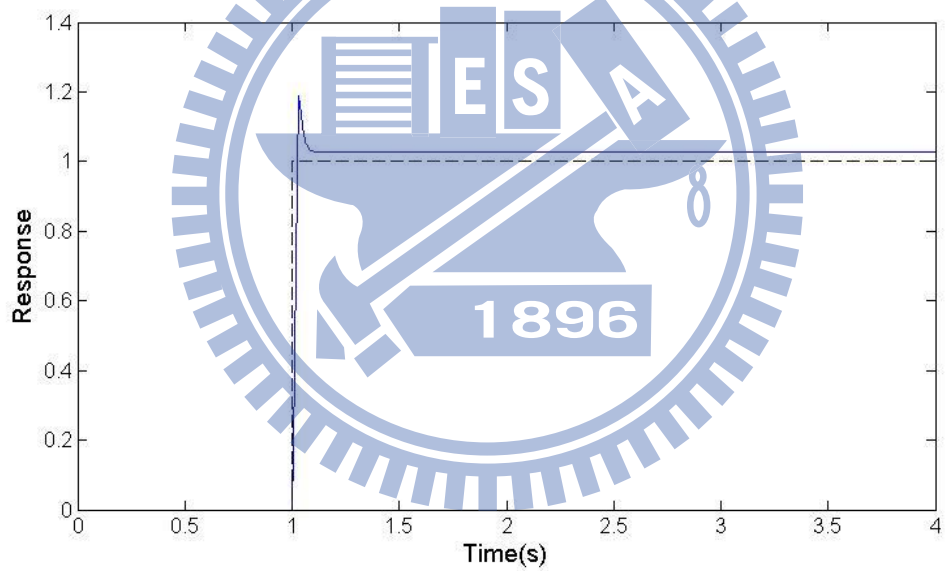
(c)50%

圖 5-16 和圖 5-17 為在有 20% model 誤差時分別以 IME 和 DPC 補償資料遺失的響應圖。因為控制器設計不變但是 model 改變而導致響應的終值與命令稍有落差，而在有資料遺失發生時，IME 與 DPC 補償能力與無 model 誤差時相似，DPC 的補償能力略勝過 IME 的補償能力。圖 5-18 和圖 5-19 為在受控體端加入 10 倍權重時分別以 IME 和 DPC 補償資料遺失的響應圖，此時 model 的誤差大於 20% 的情況。IME 控制系統與 DPC 控制系統相比暫態的震盪持續較久，但是與命令的終值落差較小，而當有資料遺失發生時，IME 補償的系統響應會無法收斂，DPC 補償系統的響應則會沿著終值持續震盪。圖 5-20 和圖 5-21 為在受控體端加入 0.1 倍增益時以 DPC 補償資料遺失的響應圖。此時與加入 10 倍增益的情況類似，響應與命令已經有所落差，而在 40% 資料遺失率下，欲補償的響應則延著錯誤控制的基準線震盪。





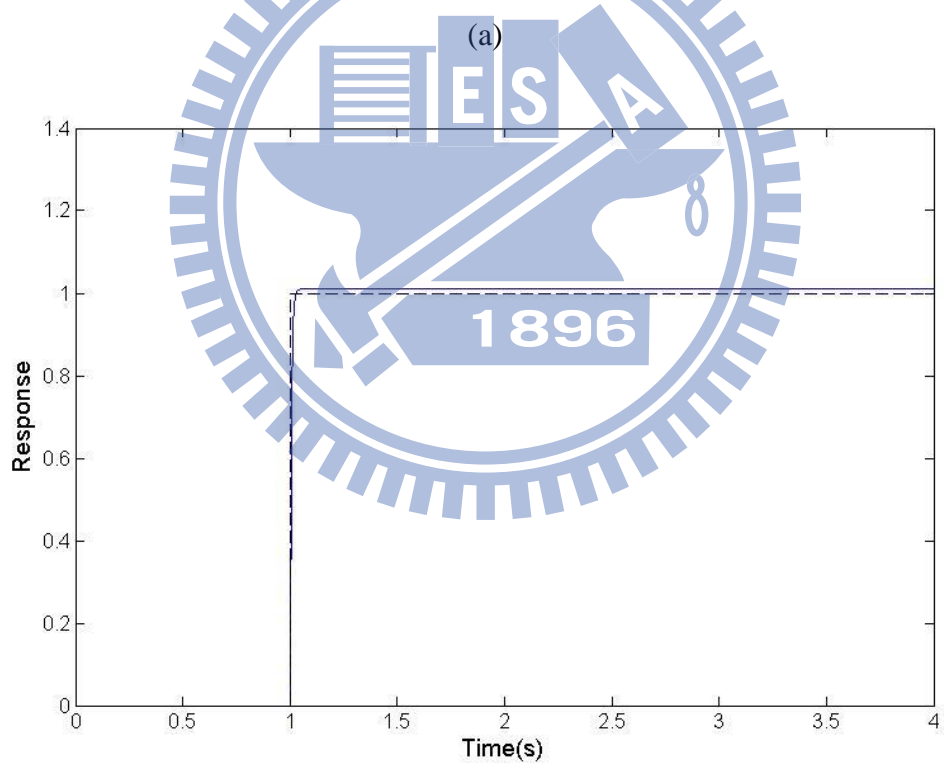
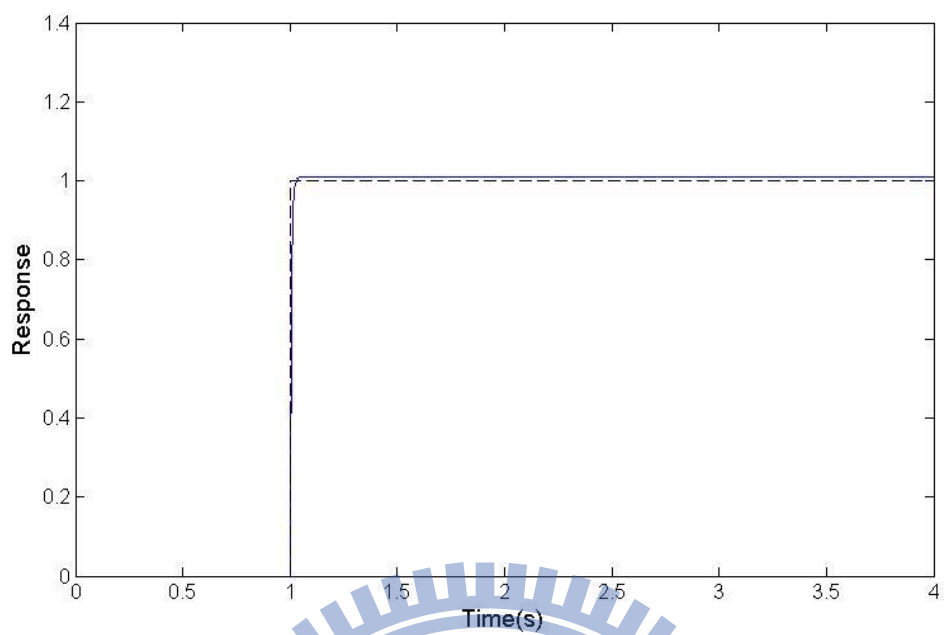
(a)



(b)

圖 5-16 在受控體端加入 20% model 誤差情況下以 IME 補償資料遺失響應圖(a)

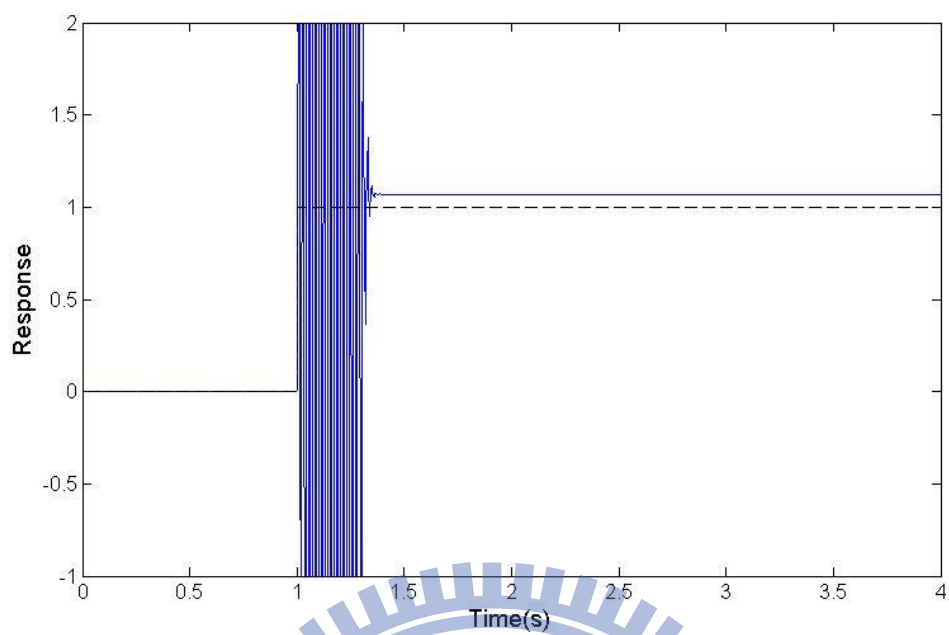
無資料遺失 (b)40%資料遺失率



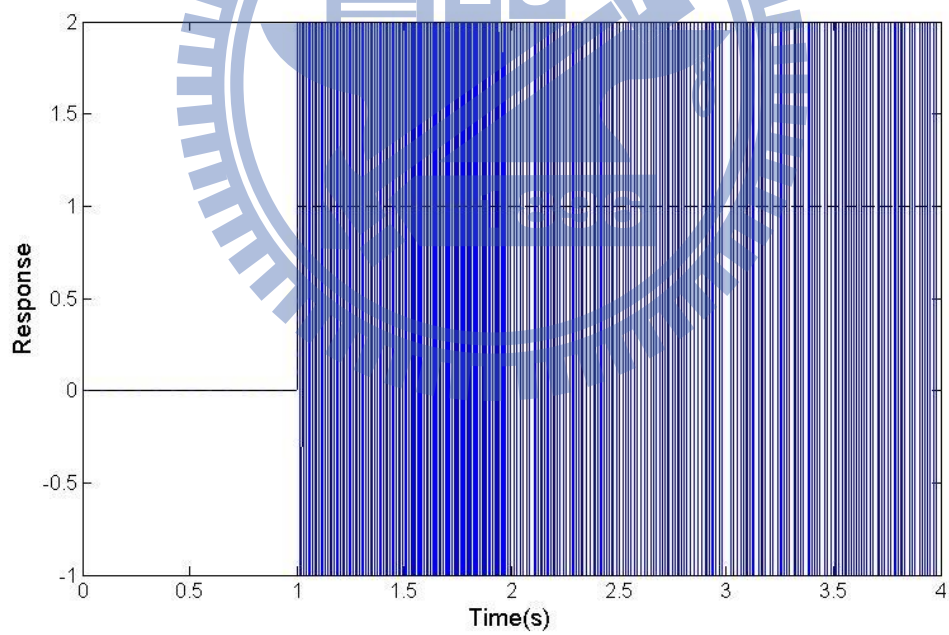
(b)

圖 5-17 在受控體端加入 20% model 誤差情況下以 DPC 補償資料遺失響應圖(a)

無資料遺失 (b)40%資料遺失率

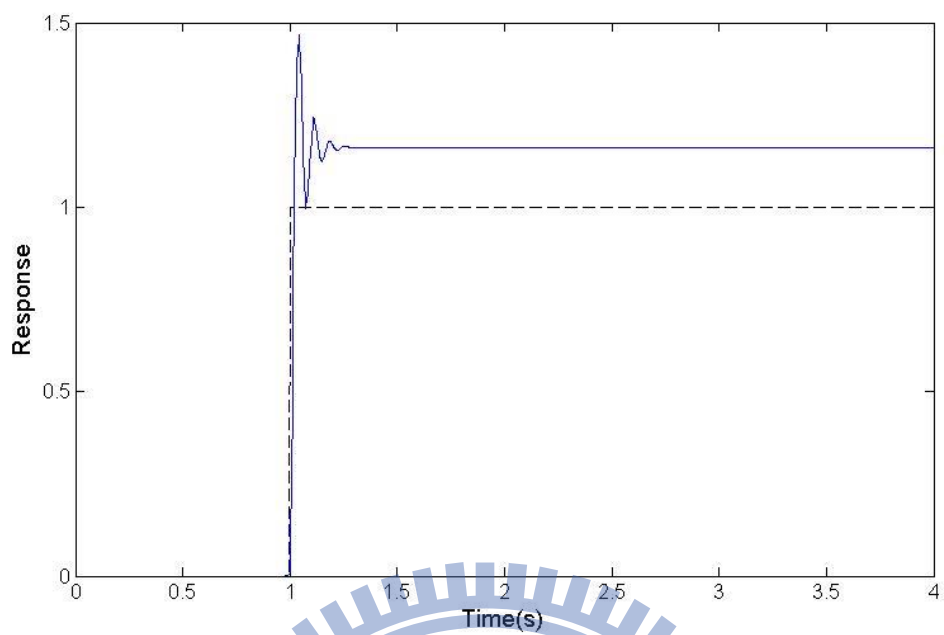


(a)

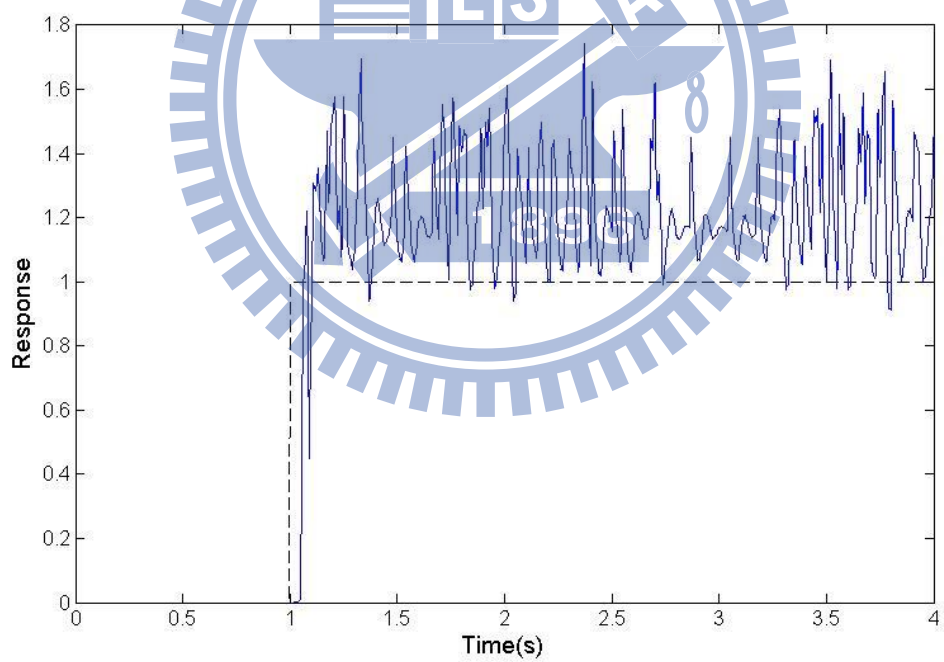


(b)

圖 5-18 在受控體端加入 10 倍增益的情況下以 IME 補償資料遺失響應圖(a)無資料遺失 (b)40%資料遺失率

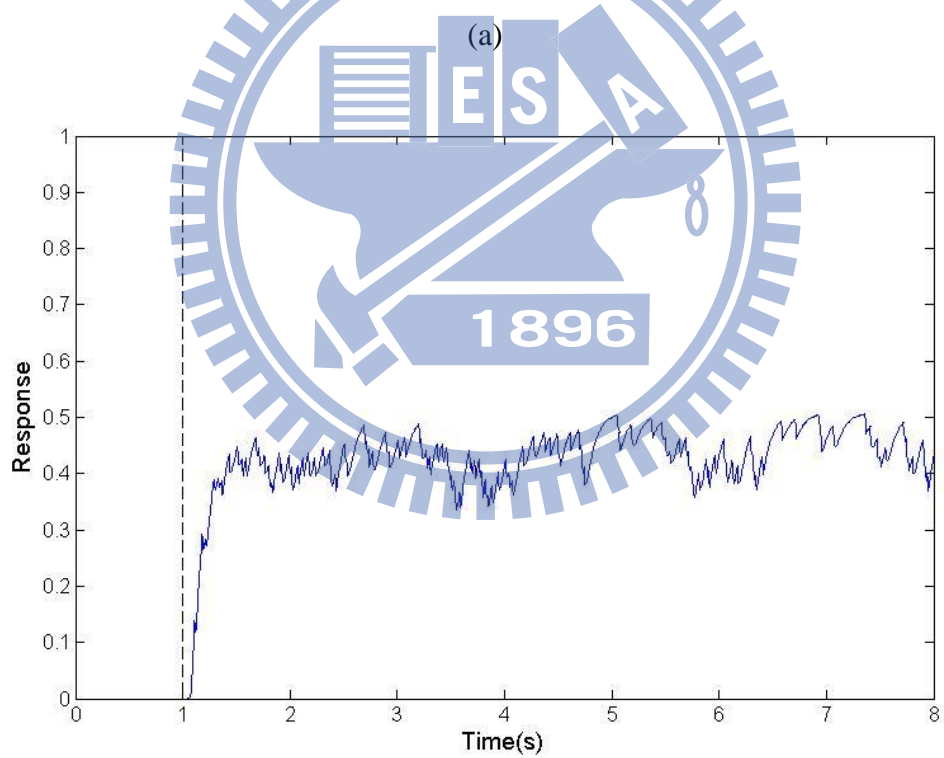
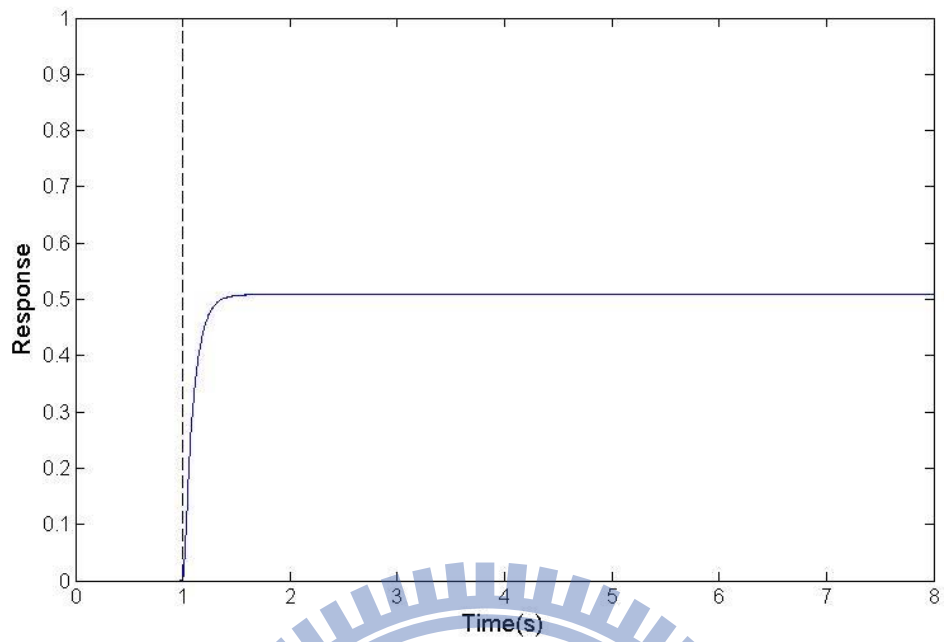


(a)



(b)

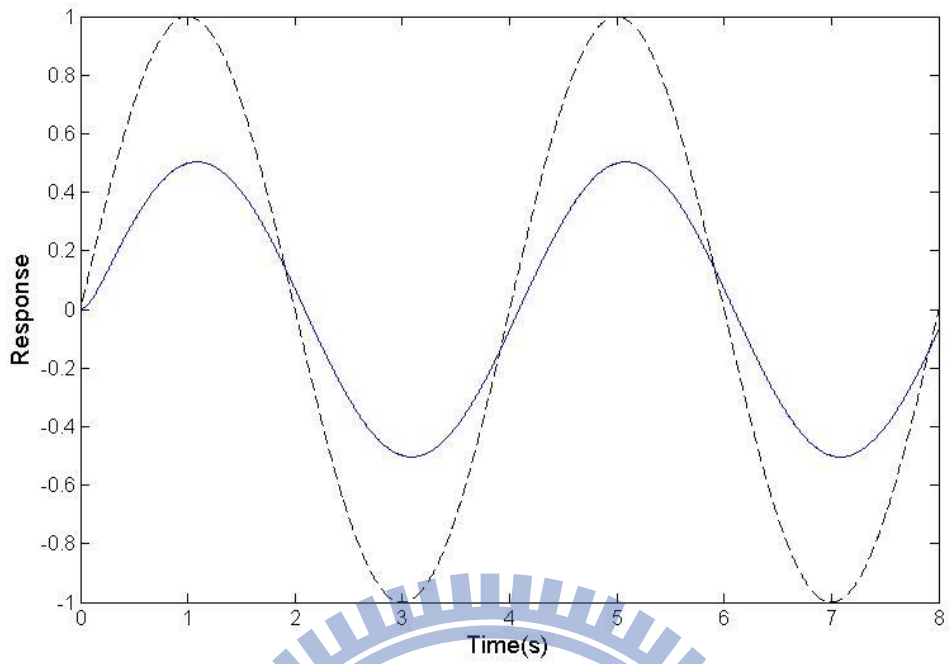
圖 5-19 在受控體端加入 10 倍增益的情況下以 DPC 補償資料遺失響應圖(a)無資料遺失 (b)40%資料遺失率



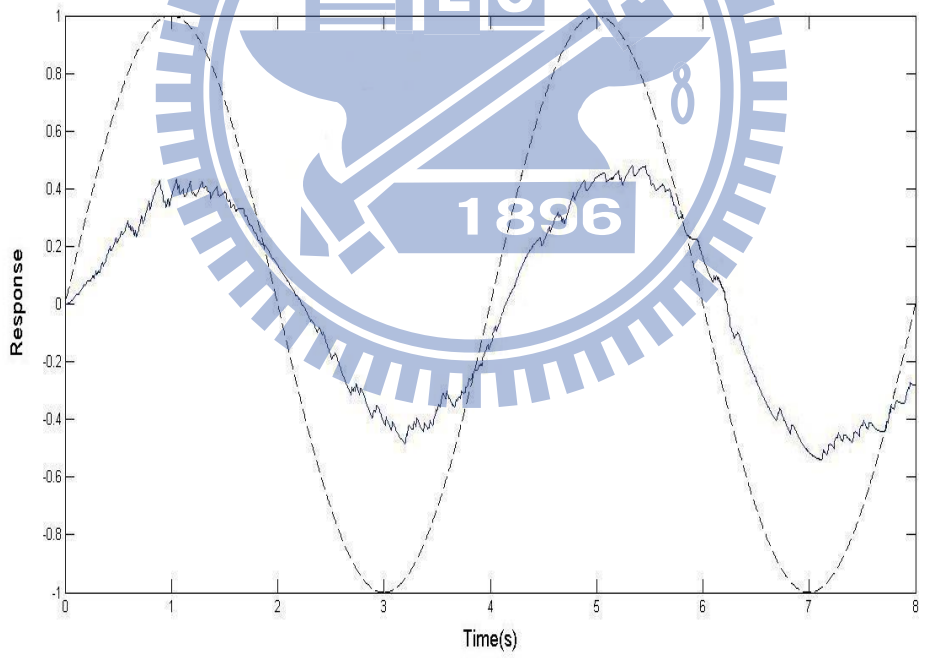
(b)

圖 5-20 在受控體端加入 0.1 倍增益的情況下以 DPC 補償資料遺失步階響應圖(a)

無資料遺失 (b)40%資料遺失率



(a)

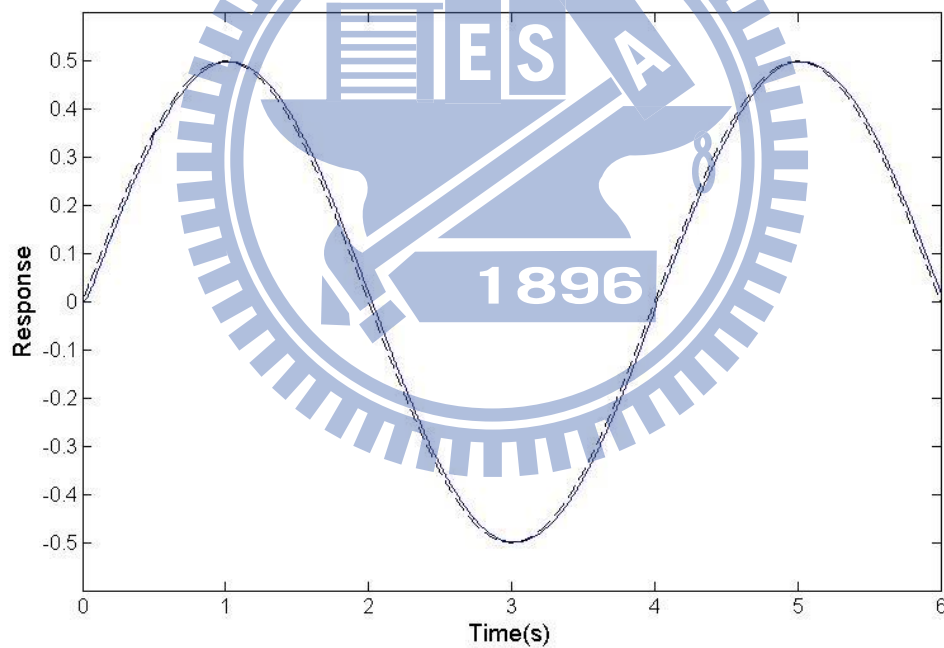


(b)

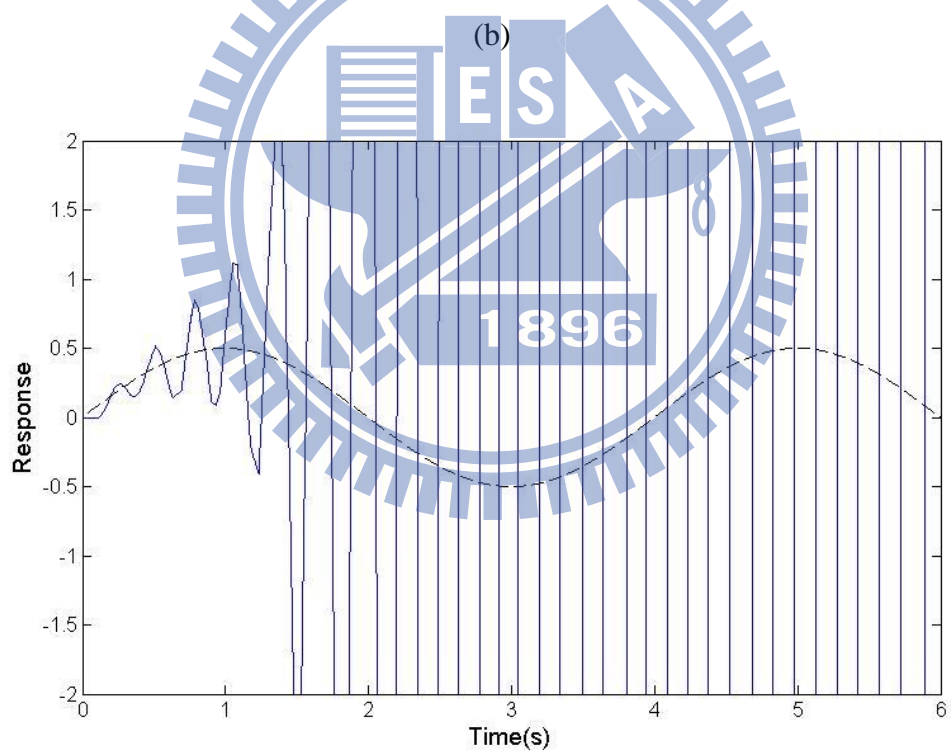
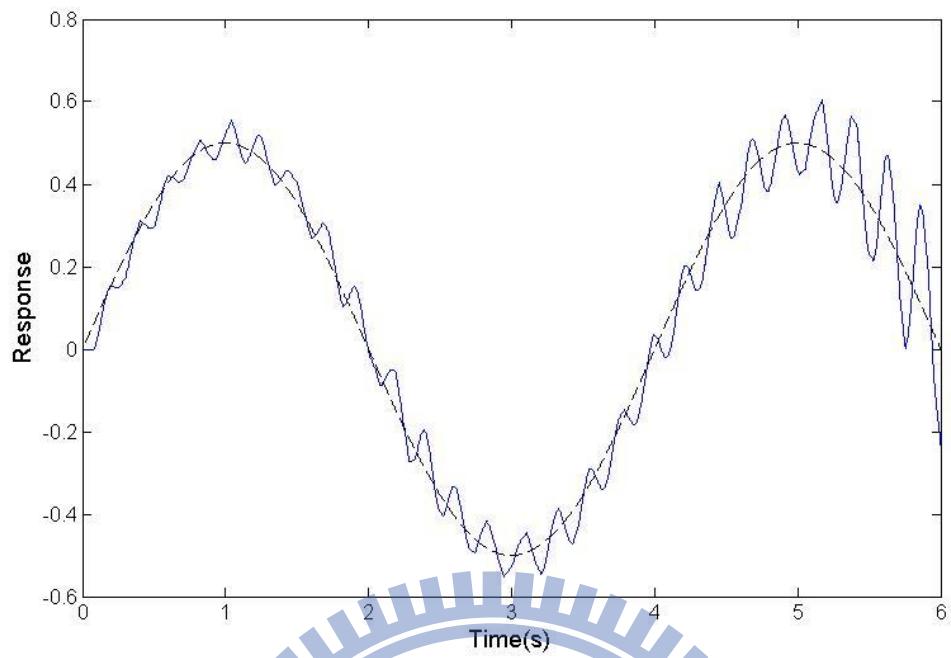
圖 5-21 在受控體端加入 0.1 倍增益的情況下以 DPC 補償資料遺失弦波響應圖(a)

無資料遺失 (b)40%資料遺失率

網路系統中不只有資料遺失，同時也存在著網路時間延遲，而對於網路時間延遲而言，若是只有短暫且少數的延遲，則能夠透過訊息估測的方式補償，若是長時間的大量延遲，則就必須透過特殊的方式補償，如第三章所提到的 PDC 時間延遲補償架構。但是因為 PDC 架構的特殊形式，DPC 控制器無法建構於 PDC 架構上，因此，若是網路控制系統中同時存在著時間延遲與資料遺失的情形，無法以 PDC 結合 DPC 的方式善，也就是說，在這邊測試 DPC 補償架構對於網路時間延遲影響的改善程度是相當重要的。圖 5-22 即為 DPC 網路補償架構在有網路時間延遲時的模擬結果，從圖中很明顯可以得知，當網路時間延遲增大時，僅以 DPC 網路補償架構補償的網路控制系統會發生震盪的現象，更嚴重時會導致系統不穩定。



(a)



(c)

圖 5-22 不同時間延遲下的 DPC 補償響應圖(a)無延遲 (b)50ms (c)200ms

從本小節的討論中，可以得知一般的情況下 DPC 應用於網路資料遺失補償



的效果更優於 IME 的補償效果，但是 DPC 本身也存在著不能忽視的缺陷。因為 DPC 本身建構於系統鑑別的基礎上，因此鑑別的過程若是有缺失存在，則得到的結果也會有缺失，以及當網路有時間延遲時，DPC 無法應付長時間的時間延遲，而且 DPC 的設計需要將命令納入考量，因此也無法結合 PDC 架構解決時間延遲的問題。

## 5.6 結合 DPC 與 IME 應用於資料遺失補償

由前一章 IME 補償架構的實驗與模擬結果可知，高階的 LSE 是造成系統響應發散的主因，雖然選用低階的 LSE 補償可以避免系統不穩定的情形發生，但是 LSE 的階數越低，會越接近 one-delay 估測器，補償的效果也會越差。因此為了將 IME 補償架構應用於未知系統上且保持穩定，必須適當的選擇 LSE 的補償階數。

本研究針對 IME 架構的切換機制做調整，除了以  $\rho_{D,D}$  作為切換基準外，再加入輸入與輸出的資訊作為判斷的指標，而此時資料預測控制器恰能很好的提供系統的輸出入資訊，因此在此提出結合 IME 與 DPC 的補償架構。假設系統的命令為有界的且命令曲線為圓滑變化，也就是說命令不會有突然的劇烈變化，而這在運動控制系統中是很合理的，若命令滿足上述所說，則可以對命令設定一個適當的上限值，假使在某時間點 IME 估測出的數值超出此上限，則改以 one-delay 估測器補償該筆遺失的資料，此方法能夠確保系統不會因為錯誤的補償而導致響應發散。而其中上限的選擇以 DPC 預測出的控制命令作為標準，如(5-39)式所示，若當下的條件滿足(5-39)則 IME 會切換為 one-delay 補償器。以此種策略進行補償的架構命名為 IME\_1D+DPC，而結合方式如圖 5-23 的架構圖所示。

$$u_{IME}(k) - u(k-1) > a(u_{DPC}(k) - u(k-1)) \quad (5-39)$$

其中 $u_{IME}(k)$ 為 IME 估測出的命令值， $u_{DPC}(k)$ 為 DPC 估測出的命令值， $u(k-1)$ 為上一點的命令值， $a$ 為權重值，本研究中  $a$  值設定為 3。

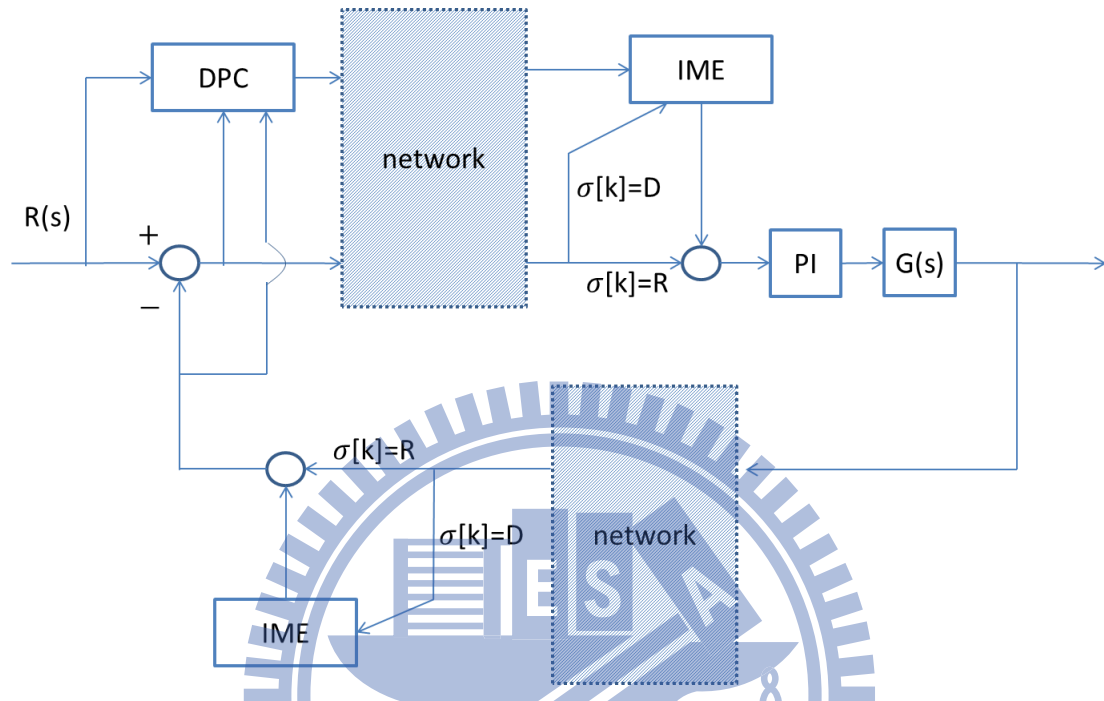


圖 5-23 IME 結合 DPC 應用於網路控制系統補償架構圖

圖 5-24 為 20%資料遺失率的實驗比較圖，由結果可以看出 IME\_1D+DPC 補償架構的追跡誤差比 one-delay 估測器補償架構的追跡誤差圓滑，補償效果較為傑出。圖 5-25 為 40%資料遺失率的實驗比較圖，此時 IME\_1D+DPC 補償架構在某些地方的誤差值有著劇烈的變化，這是因為錯誤補償產生的結果，但因為設定了上限值，因此在這些地方的錯誤補償不會造成系統響應發散。

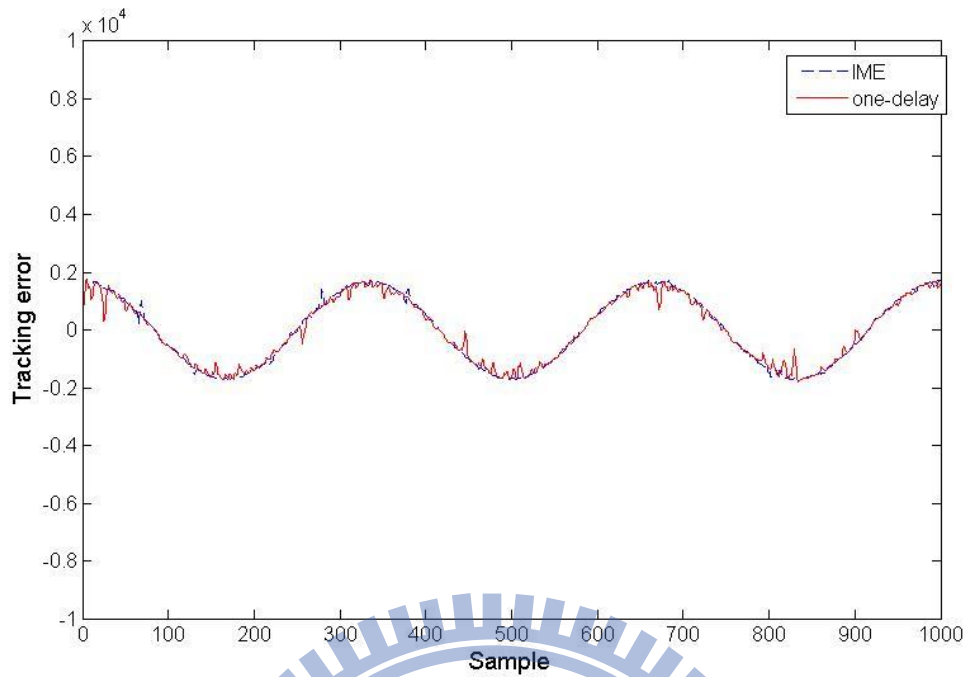


圖 5-24 IME\_1D+DPC 與 one-delay 於 20% 資料遺失率追跡誤差比較實驗圖

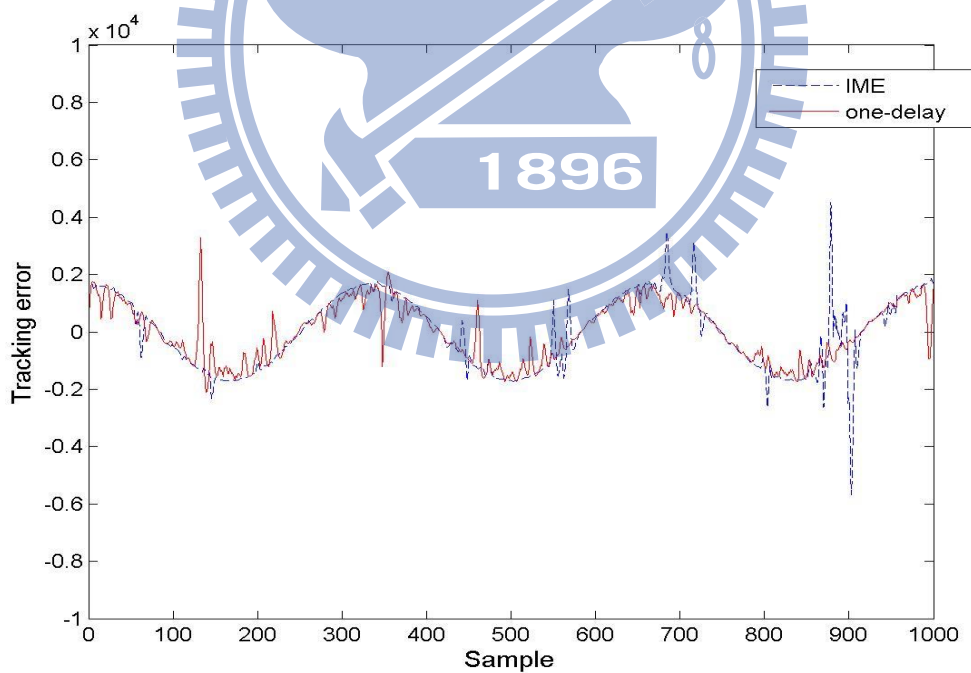


圖 5-25 IME\_1D+DPC 與 one-delay 於 40% 資料遺失率追跡誤差比較實驗圖

除了單純設定防止系統響應發散的上限值外，為了達到更好的控制效果，將

更進一步的調整 IME 切換的策略。其中的概念為自動切換 IME 補償區間，若條件滿足式子 5-39，則將 IME 補償法則中的 LSE 降一階並重新運算直到條件不滿足為止，而以此種策略進行補償的架構命名為 IME+DPC。

此時實驗結果如圖 5-26 顯示，IME+DPC 的補償架構明顯的降低了資料遺失的影響，除了不會使系統響應發散外，IME+DPC 的補償能力此時也比 one-delay 估測器來的優秀。

將 IME+DPC 結合的補償方式應用於 Truetime 模擬環境中，網路以及控制參數如前一章所述，模擬結果如圖 5-27 與圖 5-28。由圖可以看出，不管在資料遺失率 20%或是資料遺失率 40%時，提出的 IME+DPC 的架構都能達到理想的補償效果。

最後將 IME+DPC 與其它補償架構的比較結果整理於表 5.1 與表 5.2。

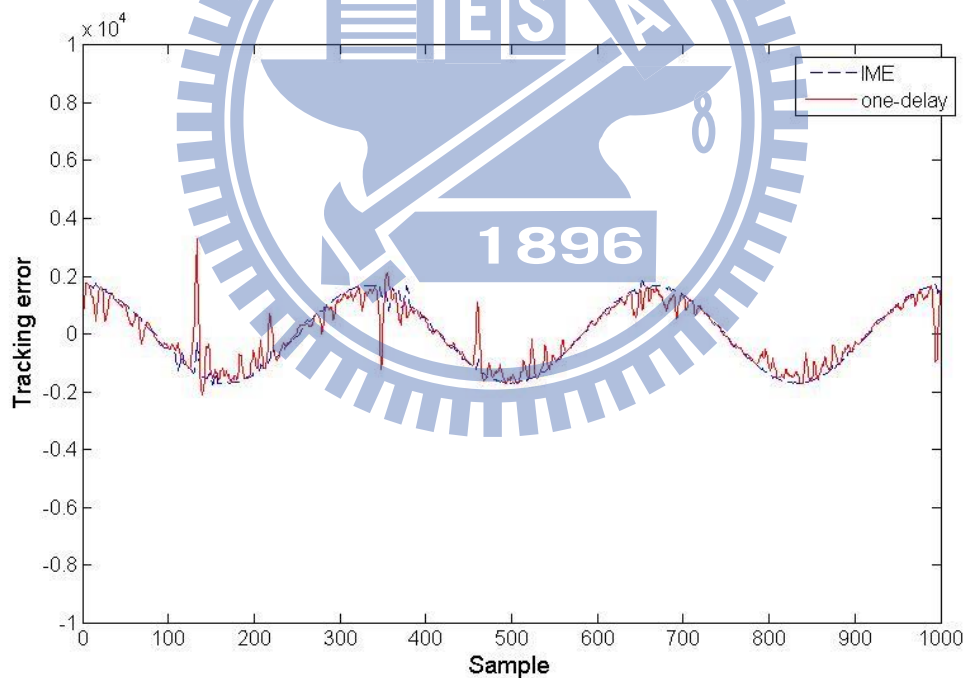


圖 5-26 IME+DPC 與 one-delay 於 40% 資料遺失率追蹤誤差比較實驗圖

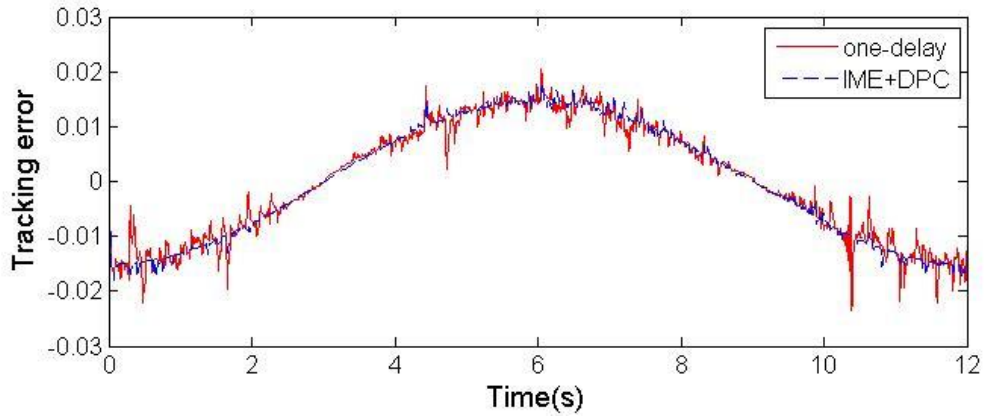


圖 5-27 IME+DPC 與 one-delay 於 20% 資料遺失率追跡誤差比較模擬圖

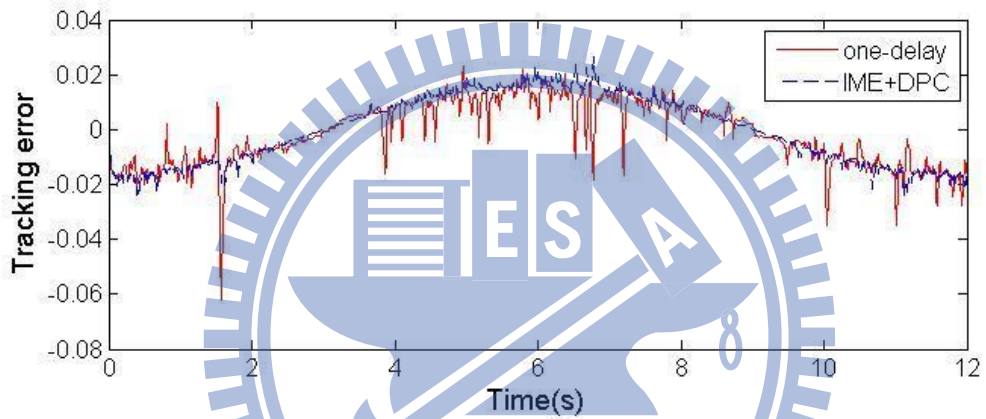


圖 5-28 IME+DPC 與 one-delay 於 40% 資料遺失率追跡誤差比較模擬圖

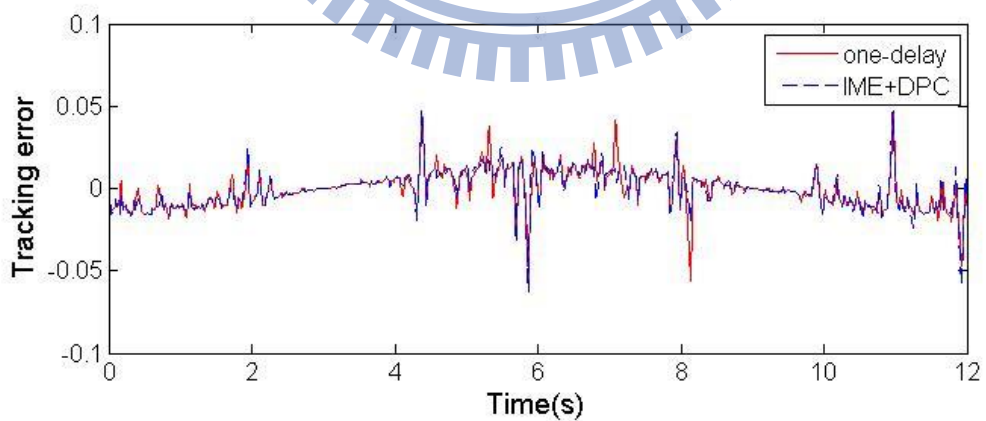


圖 5-29 IME+DPC 與 one-delay 於 60% 資料遺失率追跡誤差比較模擬圖

表 5.1 IME+DPC 與其他資料遺失補償架構之模擬結果比較

	20% 資料遺失率	40% 資料遺失率	60% 資料遺失率
one-delay	0.0104	0.0128	0.0154
IME	unstable	-----	-----
IME_1	0.0082	0.0105	0.0147
IME+DPC	0.0088	0.0097	0.0149

表 5.2 IME+DPC 與其他資料遺失補償架構之實驗結果比較

	20% 資料遺失率	40% 資料遺失率
one-delay	1214.6	1814.6
IME	1099.7	unstable
IME_2	-----	2438.5
IME_3	-----	1587.1
IME+DPC	-----	1622.1

## 第六章 時間延遲與資料遺失綜合討論

### 6.1 時間延遲與資料遺失關係

當網路時間延遲增加時，會發生連續性的資料遺失，而當網路時間延遲降低時，在相同的取樣周期內會收到多餘一筆的資料，但此時只有最後收到的資料會被採用，這叫做資料覆蓋(Data rejection)，如圖 6-1。由此可見，網路時間延遲為資料遺失產生的因素，當網路的時間延遲大於系統的取樣頻率時，資料遺失就會發生。實際上，一般的網路協定通常為了避免資料遺失而設定了重傳的機制，雖然這種機制能確保資料的傳輸但於此同時卻會提升網路延遲的時間。在時間觸發的網路控制系統中，此時上升的網路延遲就造成系統大量的資料遺失，換句話說，資料遺失會導致時間延遲而時間延遲也會導致資料遺失，因此，時間延遲與資料遺失可以說是一體兩面，對於網路控制系統而言，兩者同時的討論是必須且相當重要的。

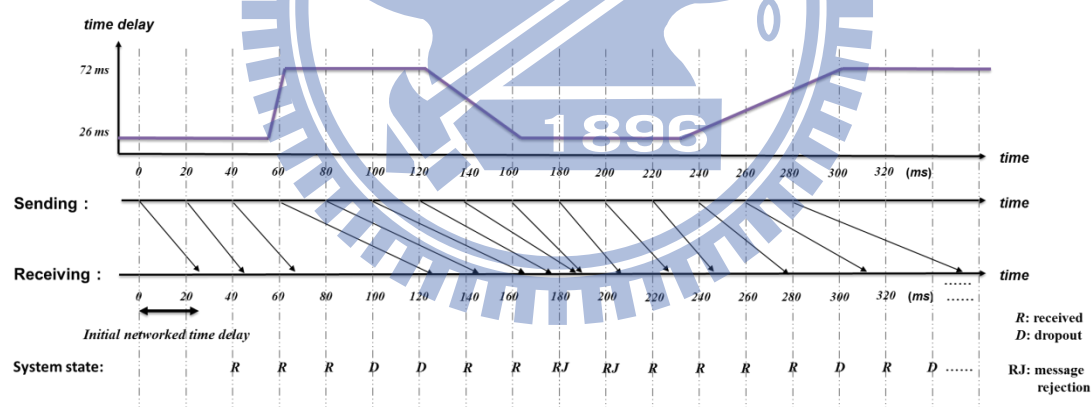


圖 6-1 時間延遲對於資料遺失的影響

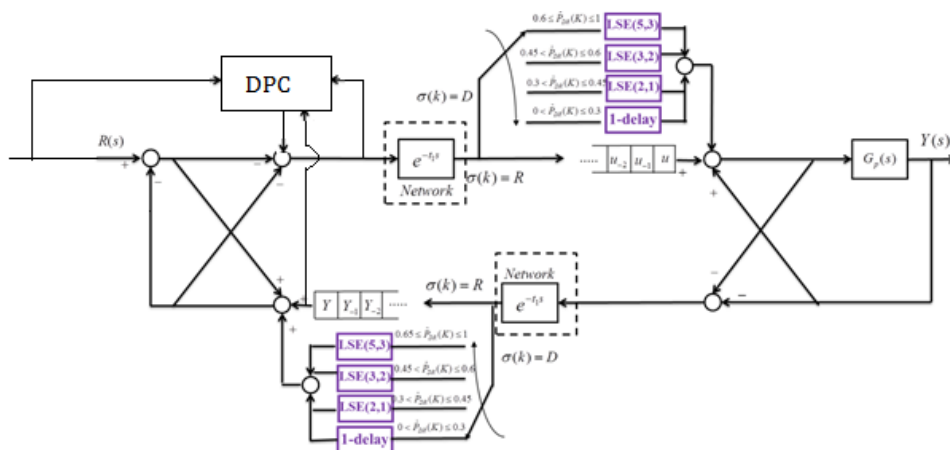


圖 6-2 DPC 與 IME 與 PDC 整合的系統架構圖

## 6.2 不同資料遺失率與時間延遲模擬於網路控制系統中

在實際的環境中，時間延遲與資料遺失對於使用網路控制的系統來說是難以避免的存在，它們可能會同時發生，也可能會隨著時間或外在環境產生變動，時間延遲與資料遺失的特性幾乎是無法預測的。因此為了分析網路控制系統在真實環境中的操作情形，本節將會討論不同大小的延遲時間與資料遺失率對網路控制系統產生的影響，在 Truetime 環境中分別以未補償的系統架構和 DPC+IME+PDC 的系統架構兩種方式進行模擬，受控體轉移函數為  $\frac{1}{s^2+5s+4}$ ，取樣週期為 50ms，使用的網路傳輸協定為 Zigbee 網路。

以下將討論五種不同的網路情形來驗證 IME+DPC+PDC 結合的補償架構的效果相較於未補償的系統。因為未補償系統仍舊屬於時間觸發的系統，因此對於遺失的數據依然會使用 one-delay 估測器補償。

Ideal Case: 無時間延遲與資料遺失

Case A: 低資料遺失(20%)與低時間延遲(50ms)

Case B: 高資料遺失(50%)與低時間延遲(50ms)



Case C:低資料遺失(20%)與高時間延遲(200ms)

Case D:高資料遺失(50%)與高時間延遲(200ms)

**Ideal Case:** 此為理想狀況，在控制系統中沒有因為網路造成的延遲也沒有資料遺失發生。弦波的追跡誤差如圖 6-3 所示，將會以此為基準與其他狀況下的系統響應比較。

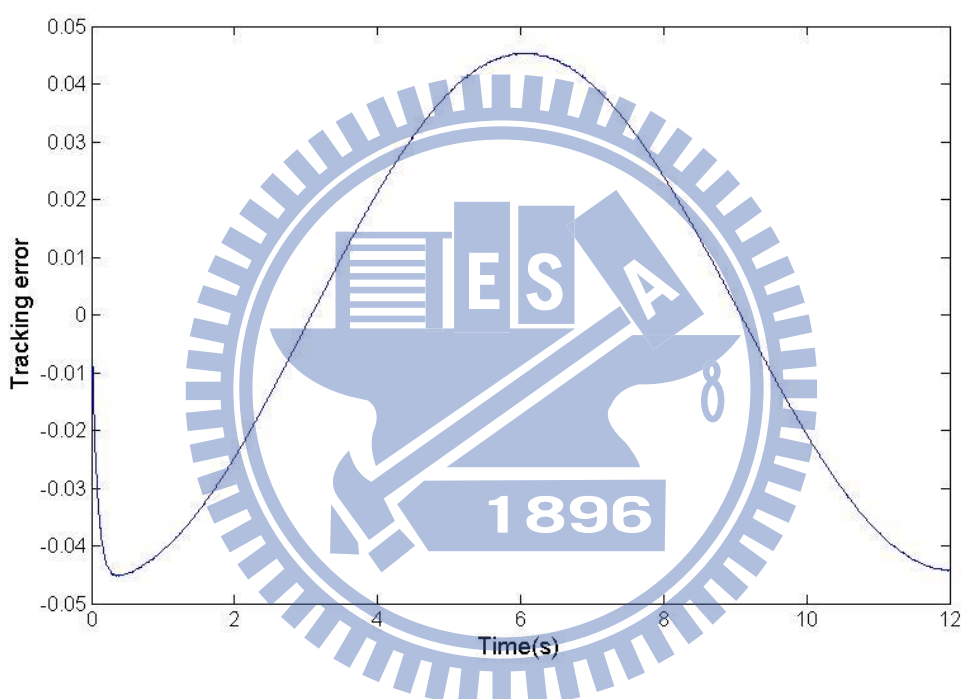
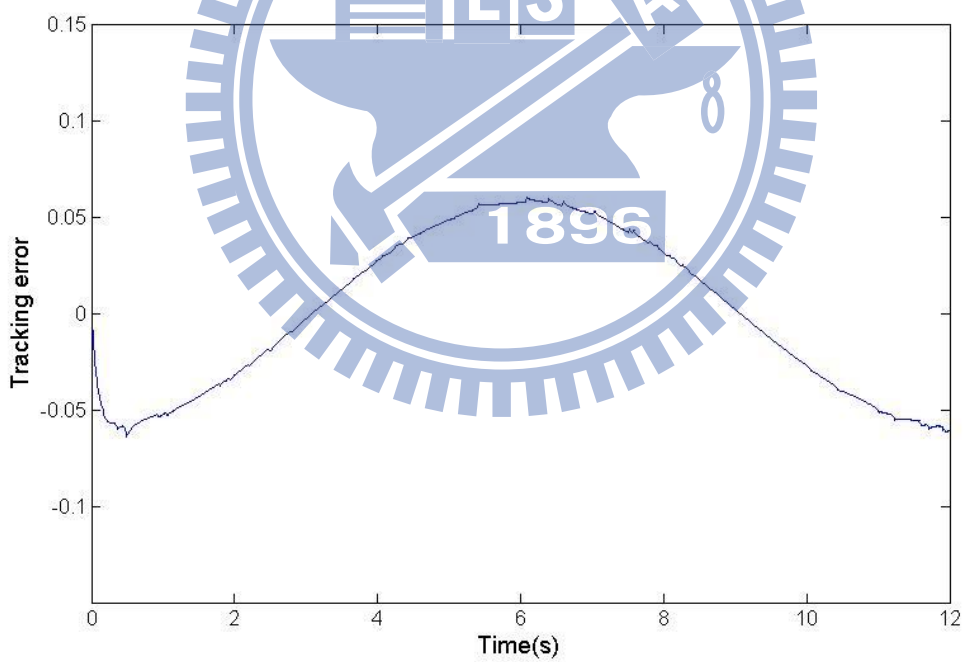
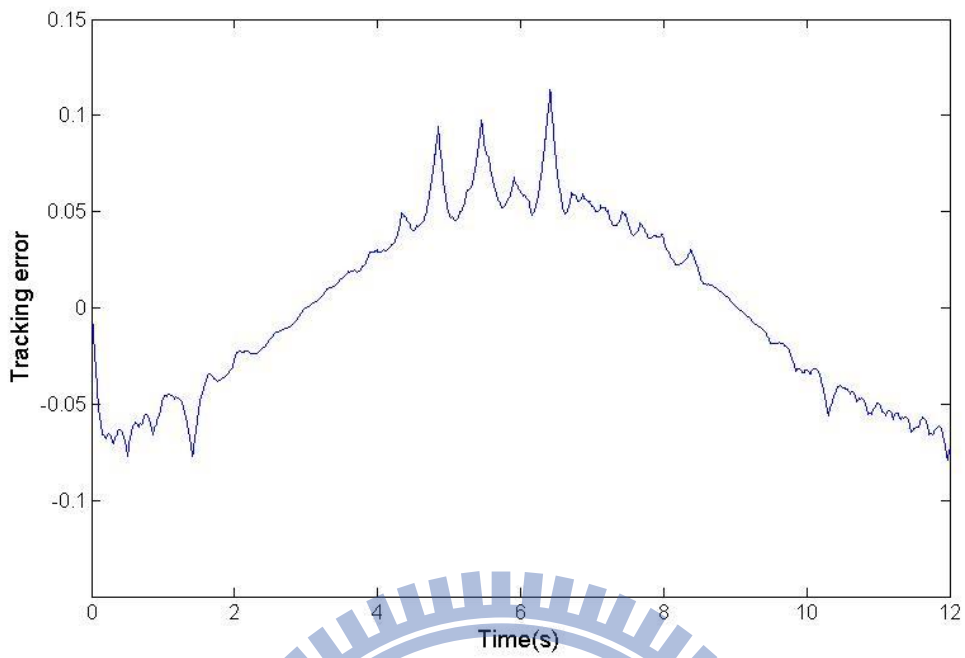


圖 6-3 無資料遺失與時間延遲的追跡誤差

**Case A:** 在網路控制系統中導入 20%的資料遺失與 50ms 的時間延遲。未補償的結果顯示如圖 6-4(a)，使用 IME+DPC+PDC 架構補償的結果如圖 6-4(b)所示。由模擬結果顯示出 IME+DPC+PDC 的架構非常有效的消除了資料遺失與時間延遲所產生的影響。圖 6-5 為整合補償架構的系統響應與命令比較圖，由此圖可以看出，PDC 將時間延遲的影響化為單純的響應位移。



(b)

圖 6-4 20%資料遺失率與 50ms 時間延遲下的追跡誤差圖(a)無補償 (b)

IME+DPC+PDC 補償

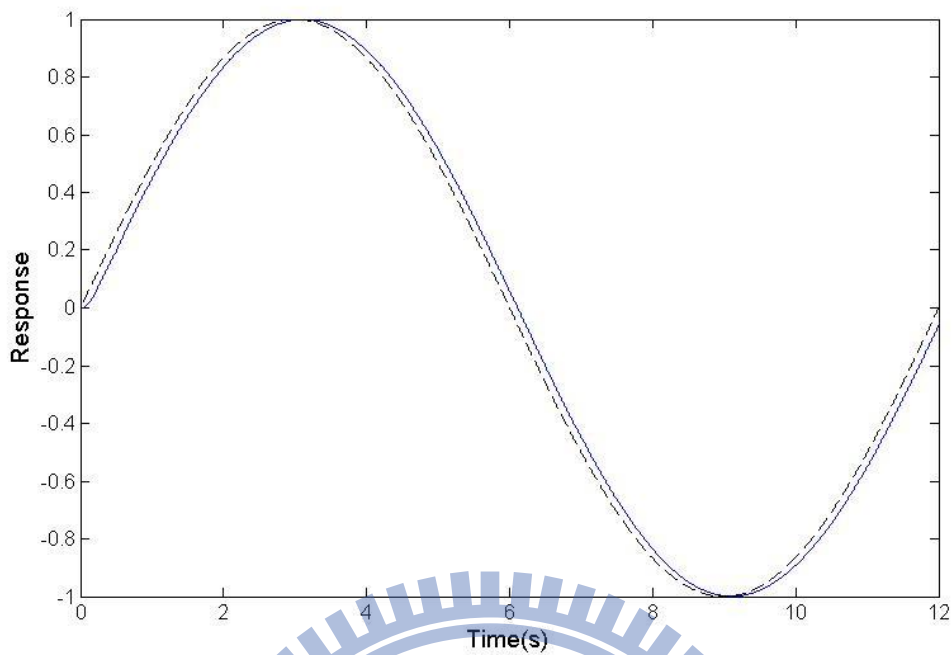
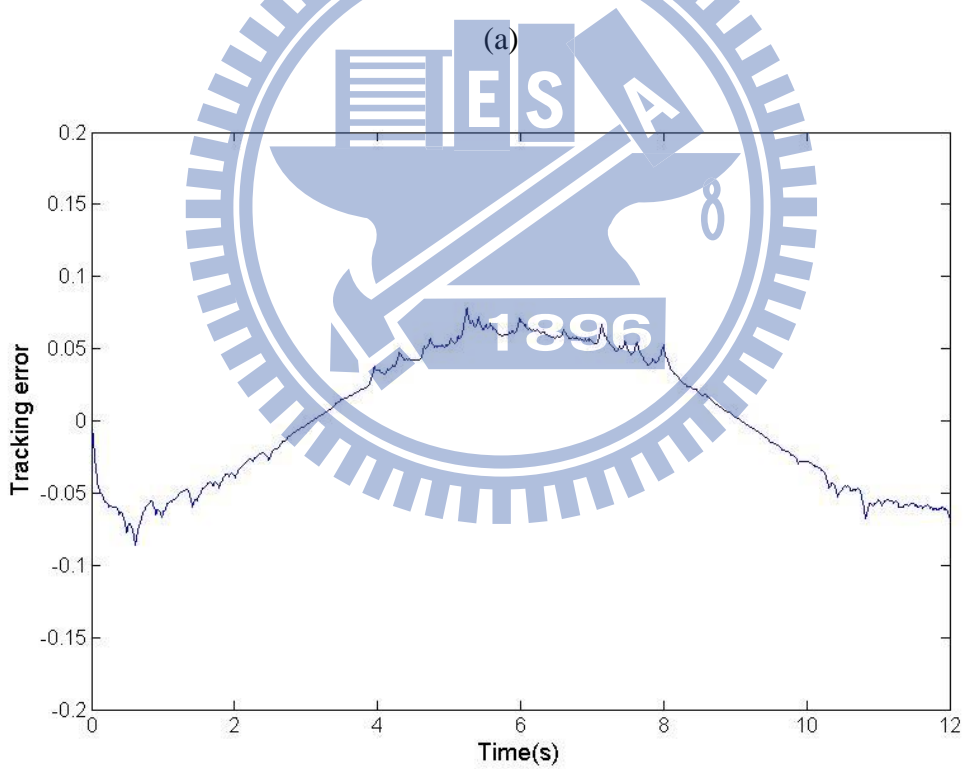
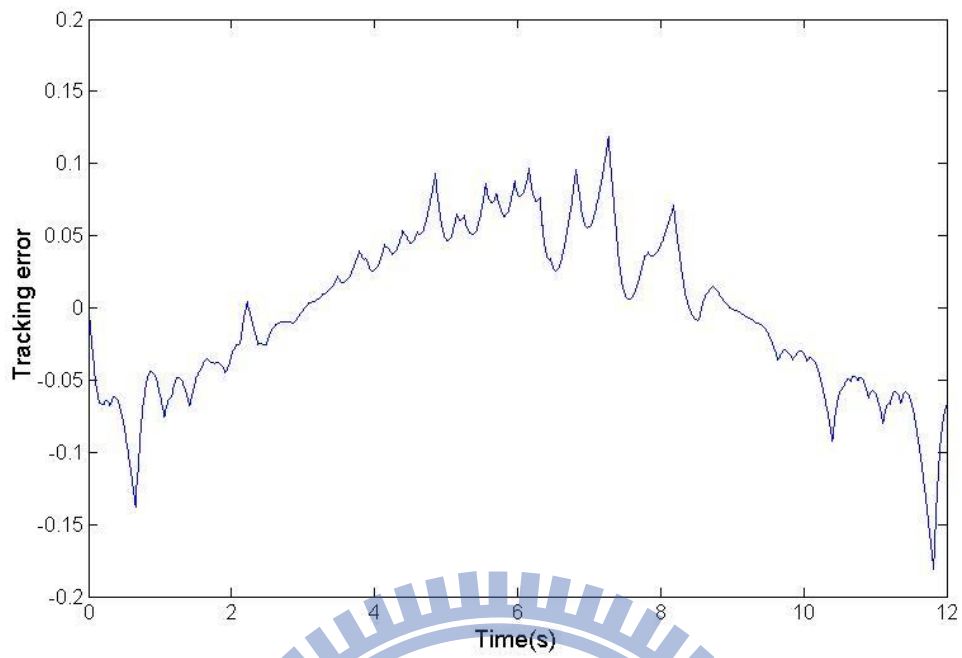


圖 6-5 IME+DPC+ PDC 架構於 20% 資料遺失率與 50ms 時間延遲下的系統響應

**Case B:** 50% 的資料遺失率與 50ms 的時間延遲被導入網路控制系統中。未補償結果如圖 6-6(a)所示，使用 IME+DPC+ PDC 架構補償的結果如圖 6-6(b)所示。模擬結果顯示提出的架構能有效地降低資料遺失與時間延遲產生的影響，即使在高資料遺失率的情況下，能仍達到很好的補償效果，而圖 6-7 為整合補償架構的系統響應與命令比較圖。



(b)

圖 6-6 50%資料遺失率與 50ms 時間延遲下的追跡誤差圖 (a)無補償 (b)

IME+DPC+ PDC 補償

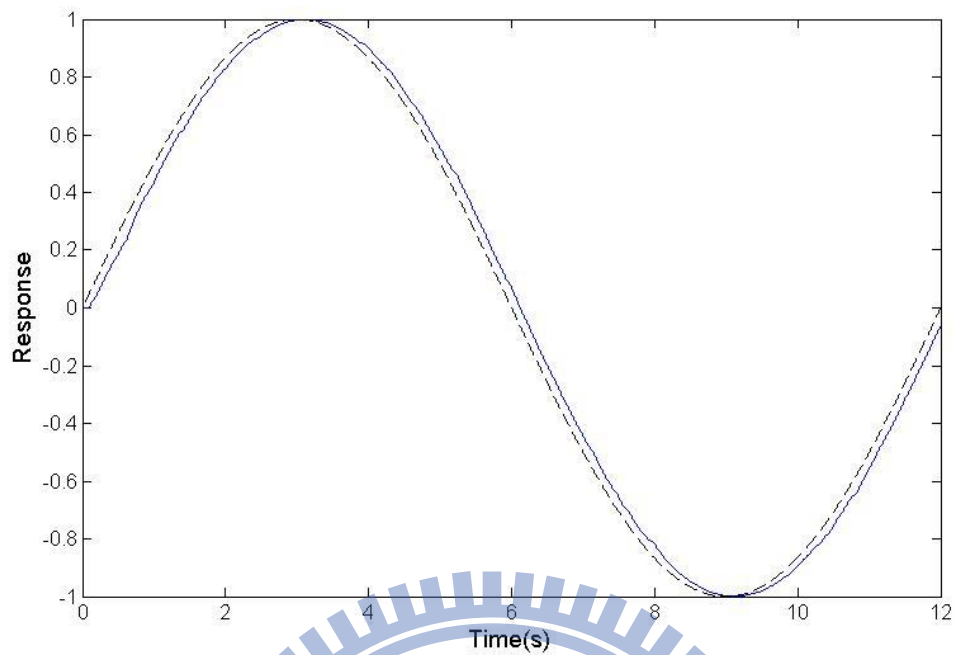
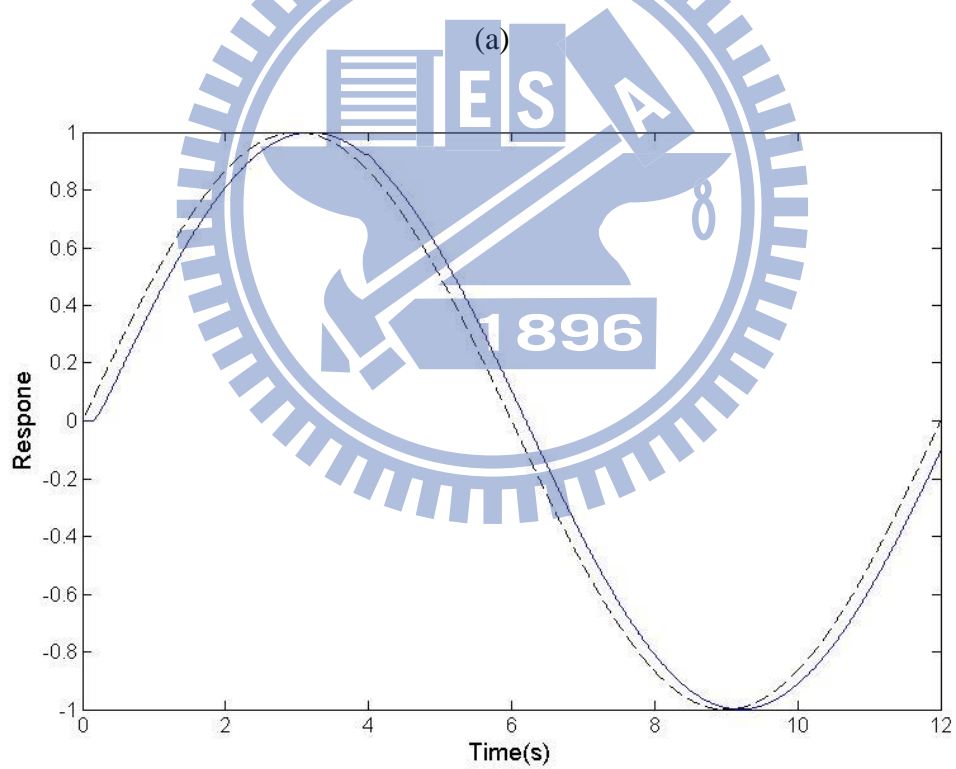
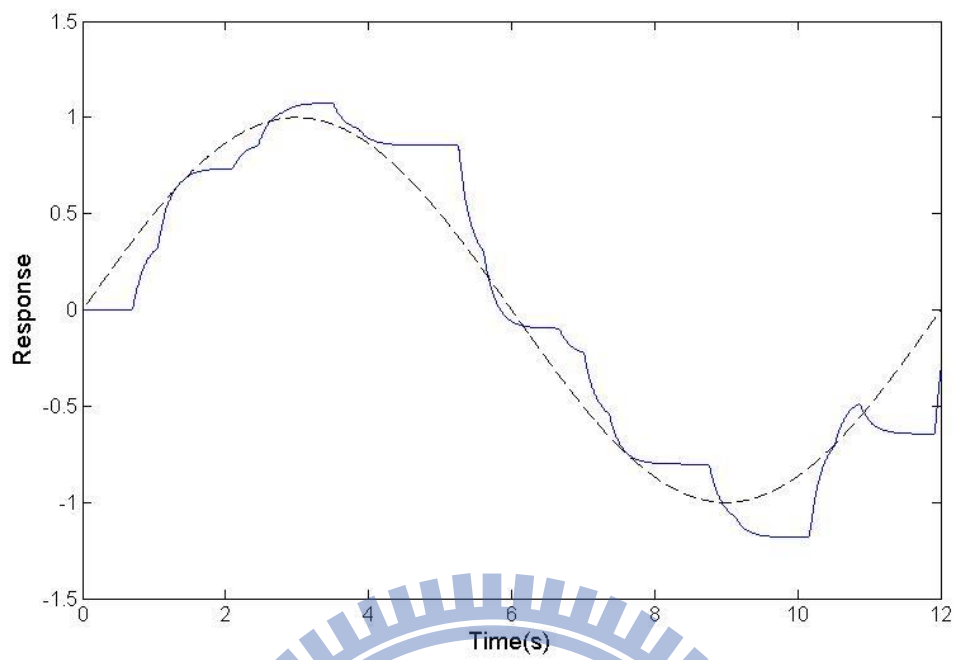


圖 6-7 IME+DPC+PDC 架構於 50% 資料遺失率與 50ms 時間延遲下的系統響應

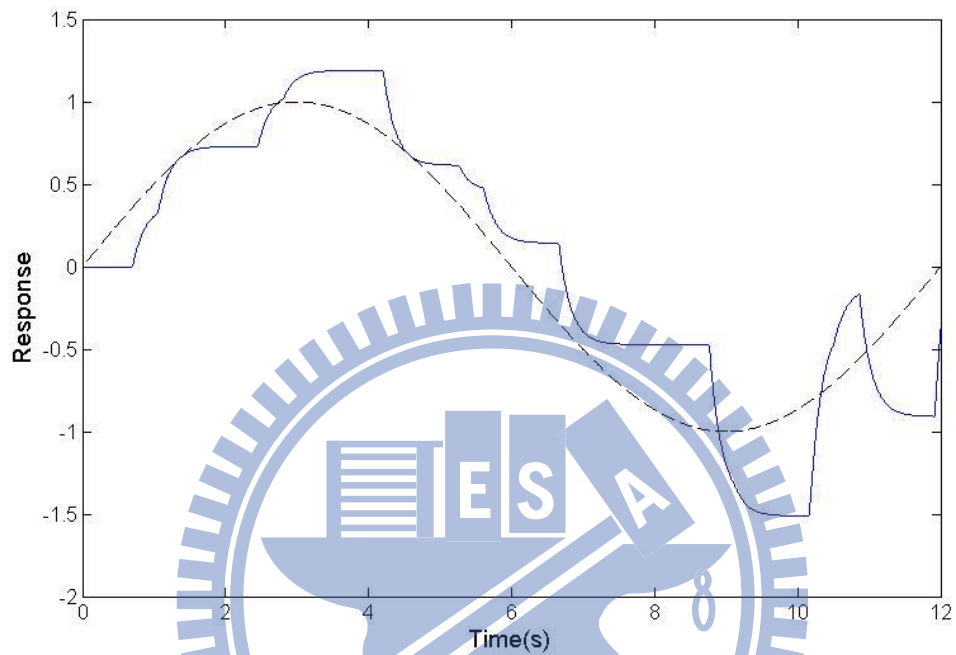
**Case C:** 低資料遺失(20%)與高時間延遲(200ms)被導入網路控制系統中。此時時間延遲對於未補償系統的影響甚大，誤差的比較已經意義不大，因此直接以系統響應呈現模擬結果。圖 6-8(a)為未補償系統的響應圖，由圖可以看出未補償系統的控制效能已經下降很多，但是提出的 IME+DPC+PDC 的架構卻依然能夠消除資料遺失與時間延遲的所帶來的影響，如圖 6-8(b)。



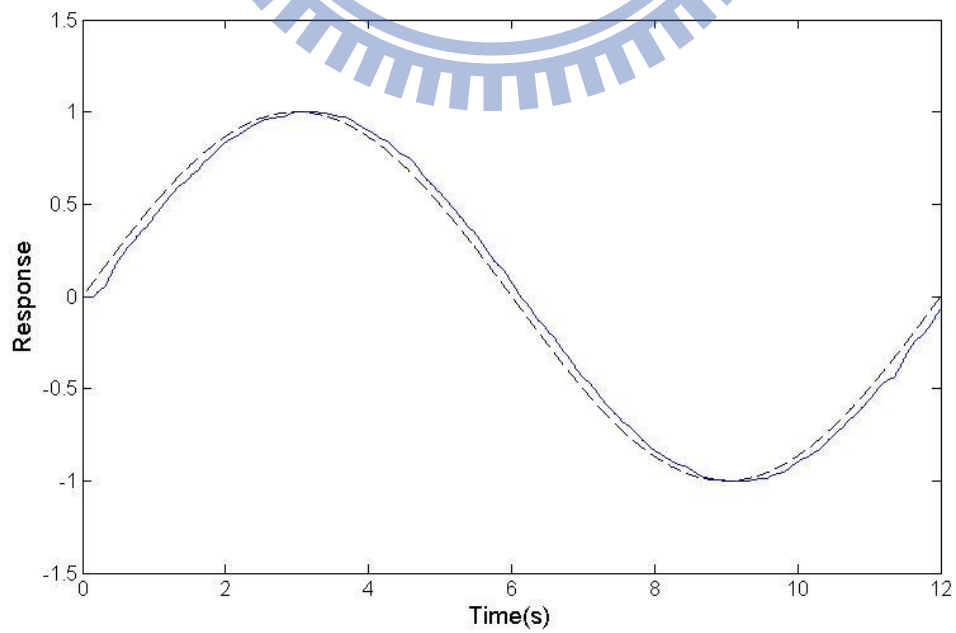
(b)

圖 6-8 20%資料遺失率與 50ms 時間延遲下的響應圖 (a)無補償 (b)IME+DPC+  
PDC 補償

**Case D:** 高資料遺失(50%)與高時間延遲(200ms)被導入網路控制系統中，模擬結果顯示未補償架構有著相當糟糕的響應，如圖 6-9(a)所示。然而，提出的IME+DPC+ PDC 整合架構即使在這種惡劣的環境下依然能達到不錯的補償效果，如圖 6-9(b)。



(a)



(b)

圖 6-9 50% 資料遺失率與 50ms 時間延遲下的響應圖 (a)無補償 (b)IME+DPC+  
PDC 補償

## 6.3 時間延遲與資料遺失補償實驗

### 6.3.1 遠端控制實驗

本小節將 DPC 結合 IME 與 PDC 的整合架構實現於圖 2-14 的遠端網路控制系統中，以實際的網路平台驗證此架構的可行性。網路傳輸部分為 5Km 外的 Remote 端點透過 Ethernet 連結本地端的 Client 再將資料以 Zigbee 網路傳給 AC 伺服馬達，而資料遺失部分為在封包傳送的途中給予 40% 的資料遺失機率。與此實驗中，首先測試了 one-delay 估測器與 IME+DPC 估測器兩者應用在遠端網路控制實驗平台的結果，然而若不加上 PDC 的延遲補償架構，實驗結果兩者的系統響應皆會發散，因此更改以 one-delay 估測器加上 PDC 的架構以及 DPC 結合 IME 與 PDC 的整合架構兩者進行實驗，得到的實驗結果如圖所示，圖 6-10 為 one-delay 估測器加上 PDC 架構的響應追跡誤差圖，而圖 6-11 為以 IME+DPC+PDC 整合架構的響應追跡誤差圖。由結果可以看出，提出的架構能夠有效的降低時間延遲與資料遺失所帶來的影響，即使是對處在遠方的控制系統依然能達到妥善的控制響應，而對於資料遺失補償的部分，IME+DPC 的補償效能明顯的能夠將追跡誤差減少。



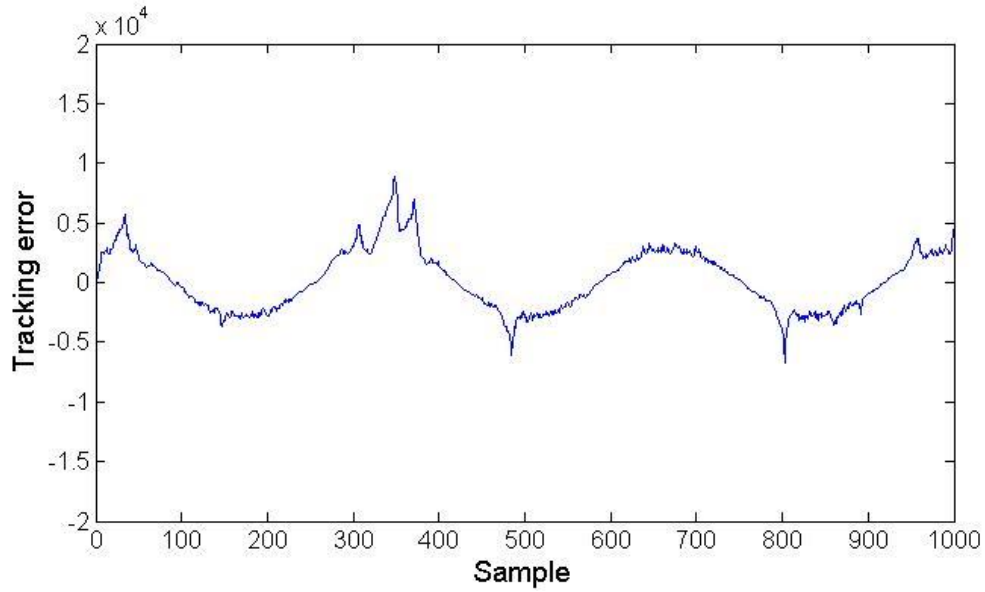


圖 6-10 one-delay+PDC 於資料遺失率 40%之遠端網路控制系統的追跡誤差圖

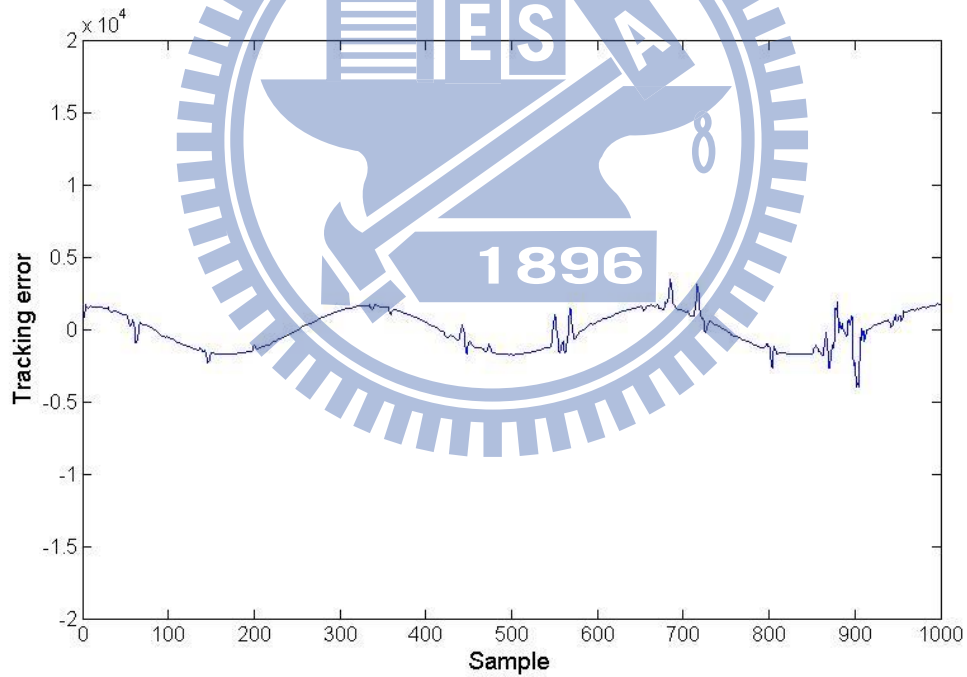


圖 6-11 IME+DPC+ PDC 於資料遺失率 40%的遠端網路控制系統的追跡誤差圖

### 6.3.2 Zigbee 干擾實驗

在本研究中使用 Zigbee 網路作為封包傳輸的主要媒介，Zigbee 網路屬於無

線網路的一種，而無線網路的特性即為容易受到外界環境的干擾，即使是相同的實驗，在白天或晚上，溫度高低或是濕度的不同，不同時間點做都會得到不同的結果。雖然如此，以上的因素對於無線網路傳輸來說並不會造成不可挽回的影響，然而，若是在網路傳輸的途中受到干擾的影響，則會因為干擾的影響使得封包碰撞遺失，或是造成封包內容錯誤，若是更嚴重的情形，該環境下有長期的干擾存在，則可能使得網路傳輸變得困難。

因為 WiFi 網路與 Zigbee 網路使用相同的頻帶傳輸，如圖 6-12，因此當 WiFi 訊號存在時，Zigbee 網路的傳輸品質會受到影響，對於 Zigbee 網路而言，此時的 WiFi 訊號即為外在干擾。

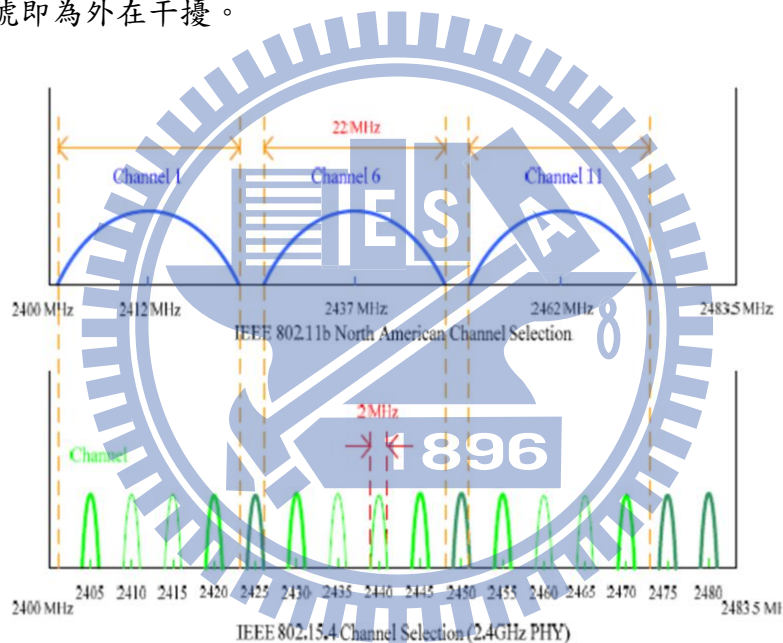


圖 6-12 WiFi 與 Zigbee 共用頻帶示意圖

為了探討當外在干擾存在時，使用 Zigbee 設備作為網路傳輸方式的控制系統效能會受到什麼樣的影響，此時在原本的網路控制系統中加入傳輸流量為 300kb/s 的 WiFi 訊號對網路控制系統進行干擾，實驗架構如圖 6-13 所示。實驗首先以 one-delay 估測器補償控制系統，結果如圖 6-15 所示，在 300kb/s 流量的 WiFi 訊號下，僅以 one-delay 估測器補償的系統響應有著巨大的追跡誤差。接著以 IME+DPC 的架構進行網路控制資料遺失的補償，得到的結果僅稍微比

one-delay 估測器架構好一些。最後以 IME+DPC+PDC 架構進行補償的系統，除了零星部分有較大的誤差外，幾乎將外在干擾產生的影響降至最小。進一步地，將取樣週期更改為 30ms 重新實驗流程，可得圖 6-18 的結果，僅以 one-delay 和 IME+DPC 這兩種資料遺失補償的架構進行補償最終都會導致系統響應發散，而加入 PDC 後以 PDC 結合 DCP 和 IME 的架構進行補償，結果如圖 6-18(c)所示，能夠將干擾的影響大幅降低。

最後將實驗在 WiFi 訊號流量加大的情形下，網路控制系統的效能。以 WiFi 訊號流量超過 1M/s 的外在干擾情形下，進行網路控制的實驗。然而，在此條件下因為 Zigbee 搜尋不到可以使用的頻道，Zigbee 網路無法建立，因此兩種補償策略皆無法控制受控端馬達。

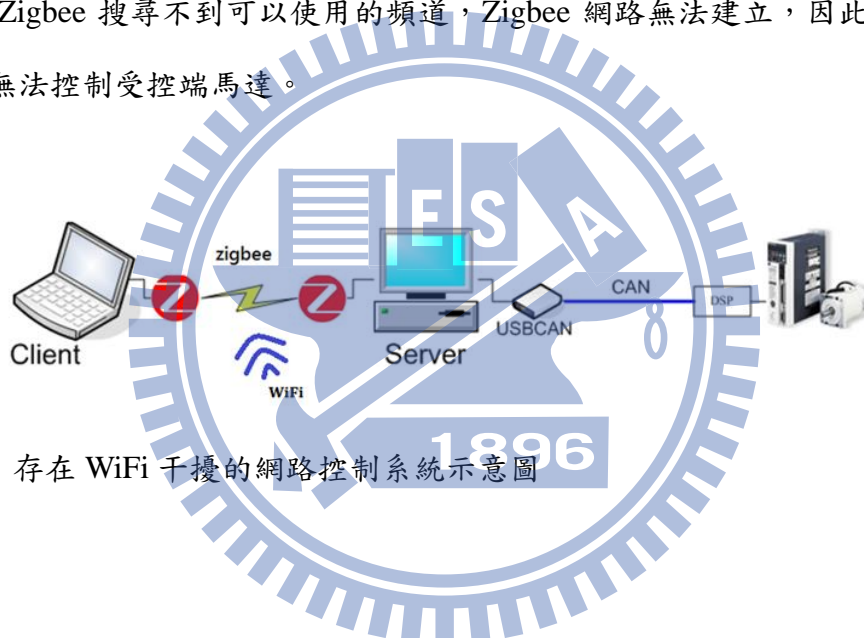


圖 6-13 存在 WiFi 干擾的網路控制系統示意圖

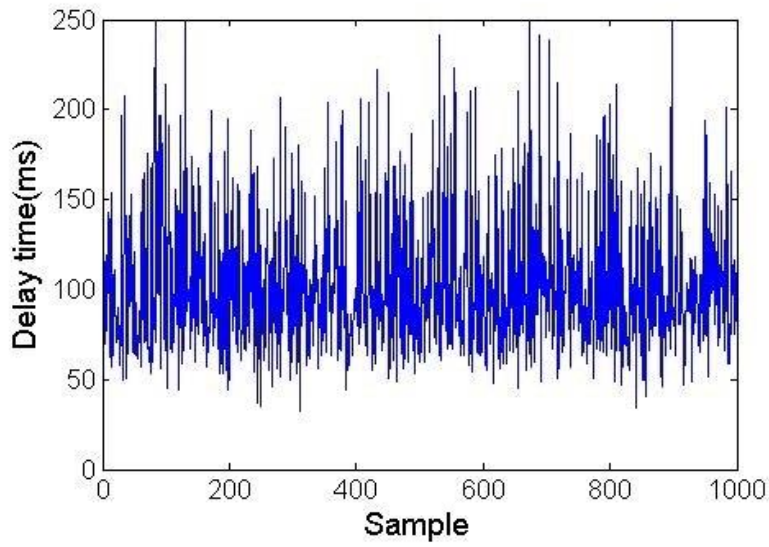


圖 6-14 300kb/s 流量的 WiFi 訊號干擾下網路控制系統之延遲時間

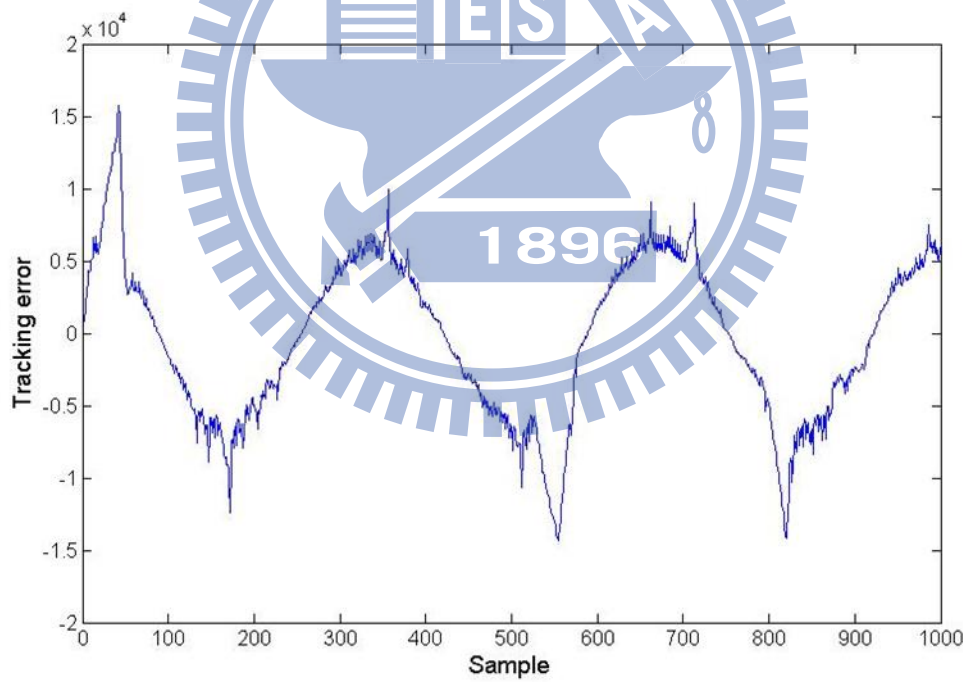


圖 6-15 以 one-delay 估測器補償干擾因素的追跡誤差圖

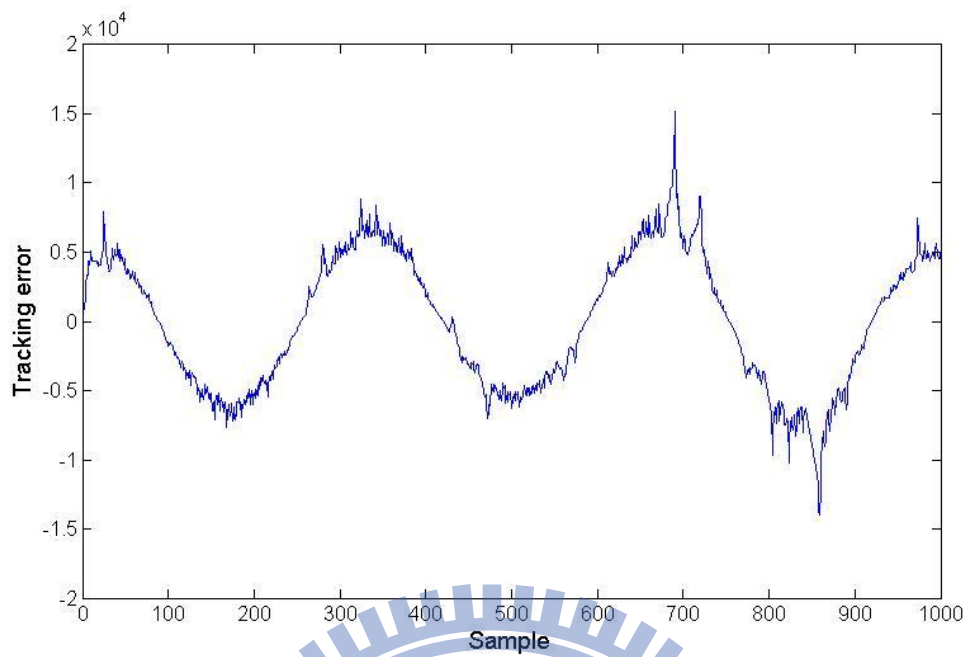


圖 6-16 以 IME+DPC 估測器補償干擾因素的追跡誤差圖

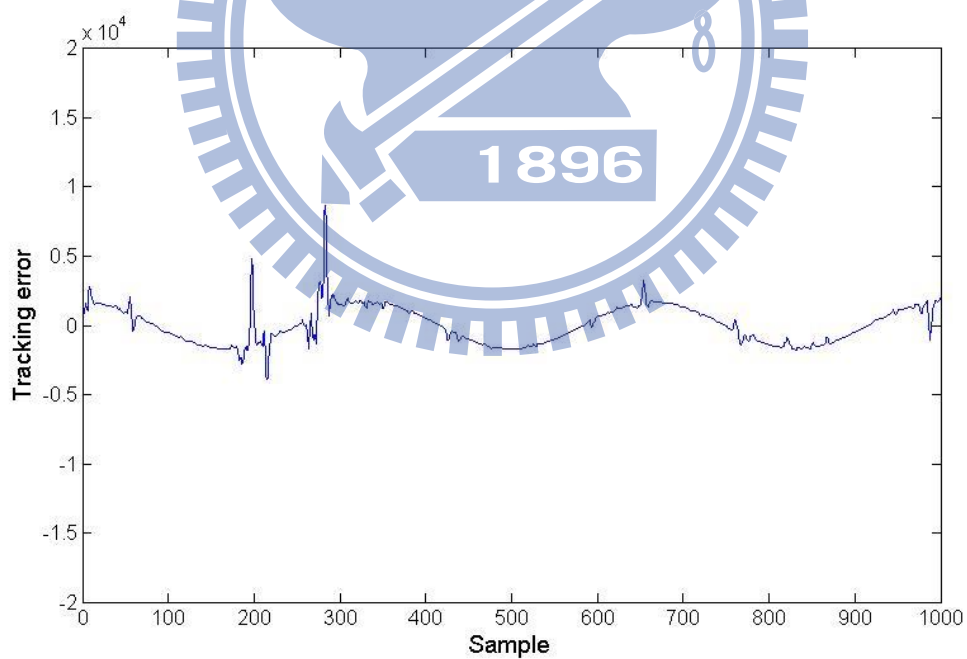
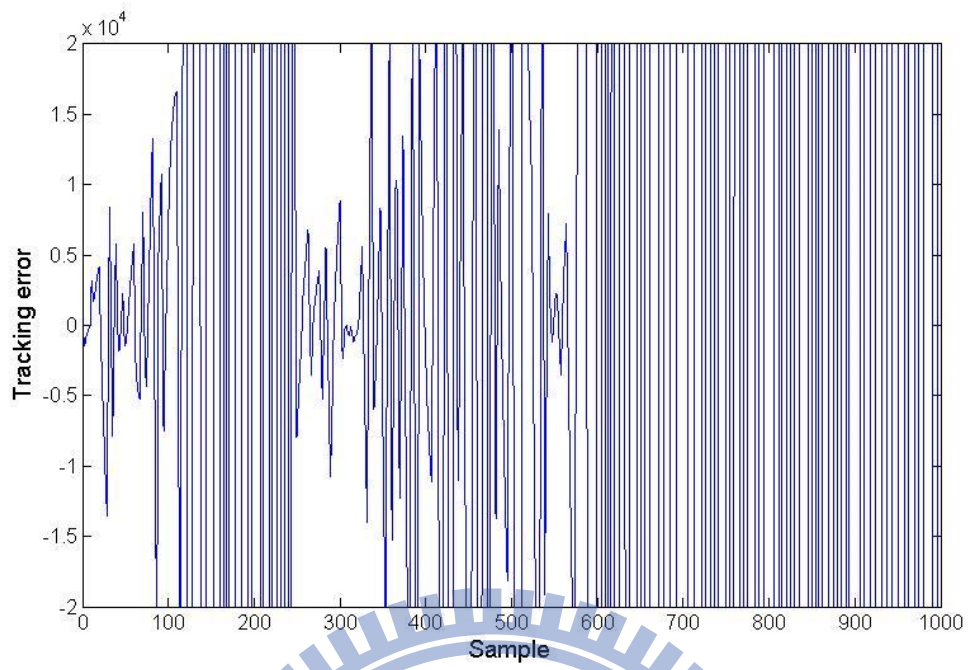
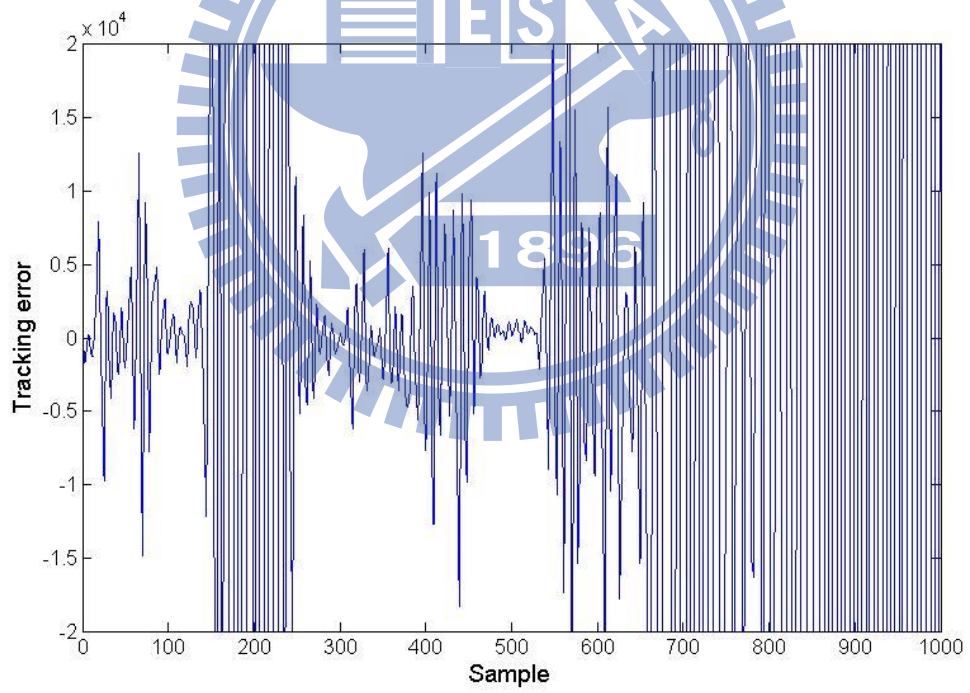


圖 6-17 以 IME+DPC+PDC 補償干擾因素的追跡誤差圖



(a)



(b)

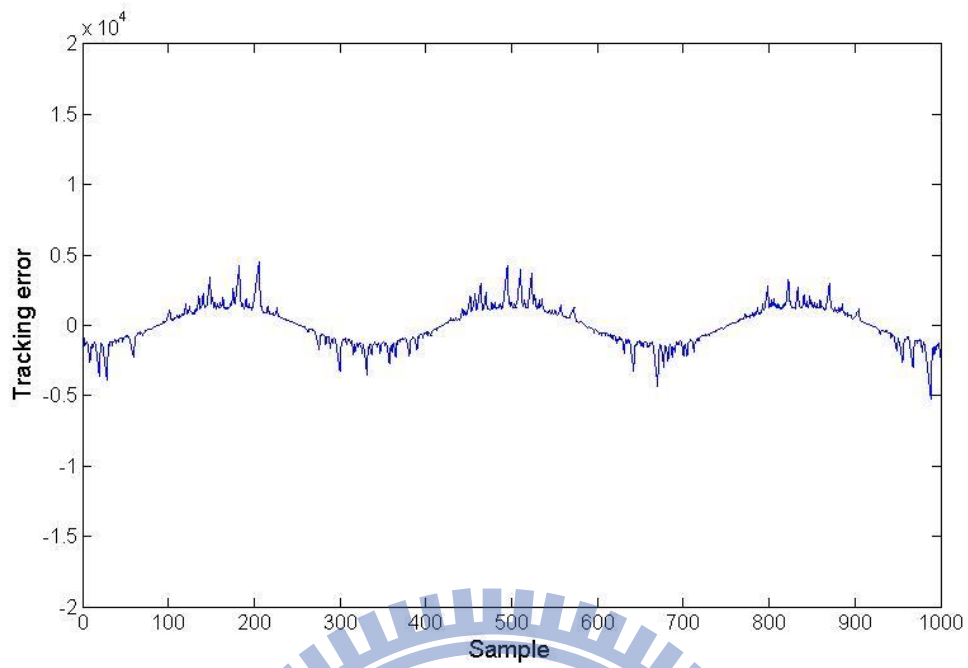


圖 6-18 於 30ms 取樣週期在 300kb/s 干擾下補償之追蹤誤差圖(a)以 one-delay 補償 (b)以 IME+DPC 架構補償 (c)以 IME+DPC+PDC 架構補償

## 第七章 結論與未來發展

### 7.1 結論

本研究針對以 Zigbee 網路為傳輸媒介的網路控制系統進行探討，其中討論了 Zigbee 本身的傳輸時間，網路系統的延遲時間等問題，接著對於資料遺失的分布以及資料遺失率對系統的影響作探討，之後提出了資料預測控制器並將其加入網路補償的架構中，在各種條件下，透過一系列的實驗與模擬，一步步的處理時間延遲和資料遺失對網路控制系統造成的問題，以下為歸納出的三點結論：

1. 提出不同的資料遺失補償架構：

對於網路資料遺失的補償，首先使用了 one-delay，IME 等架構來進行補償，但是 one-delay 在低度資料遺失時的補償效果不如 IME 而 IME 架構的切換區間應用於不同系統時需要調整，否則有機會使得系統響應不穩定，因此提出了改變切換區間的 IME\_1，IME\_2，IME\_3 等補償策略，經過模擬以及實驗的結果證明，對於不同的系統選用不同補償區間的 IME 能達到比 one-delay 和原本 IME 更好的補償效果。

2. 分析 DPC 及 IME 補償網路訊息的優劣：

使用了三種不同的控制命令來比較 DPC 與 IME 網路資料遺失的補償能力，模擬結果顯示在 DPC 控制器設計妥善沒有模式誤差的情況下，DPC 應用於網路資料遺失補償的能力會比 IME 來的優秀。然而當 DPC 存在設計問題或是有模式誤差時，本身控制能力會下降，此時以 DPC 應用在網路資料遺失補償的能力也會下降。IME 本身為 model-free 的架構，能夠因此避免模式誤差的問題，且當網路有時間延遲發生時，IME 能夠結合 PDC 達到網路時間延遲的補償架構。

3. 提出能廣泛適用於各種系統的網路時間延遲及資料遺失補償架構：



對於一般的網路控制系統而言，大部分的補償系統都是針對 model-based 的設計。本論文提出了 IME+DPC 的網路資料遺失補償架構，能夠以資料遺失轉移機率及系統輸出入資訊得到自動切換補償區間的 IME 設計，能夠針對不同的資料遺失進行區間的切換達到完善的補償效果，然後加入 PDC 補償網路時間延遲的問題，得到 IME+DPC+PDC 的網路資料遺失及時間延遲補償架構，此架構能夠廣泛適用於各種不同的網路控制系統中，同時解決網路資料遺失與時間延遲對網路控制系統所產生的影響。

## 7.2 未來發展

本論文中使用的網路為 Zigbee 網路，為了達到更好的控制效能，未來可將此網路換成速度更快的網路，如此一來即能夠提高取樣頻率，使控制效果更好。若是同樣使用 Zigbee 網路，則可以探討 Zigbee 感測器互相傳輸的方面，建構多點的感測器網路，使控制系統延伸，擴大網路。而對於預測控制器而言，未來可將預測控制器以即時的方式實現，或者對應於不同情況自動建構出不同參數的預測控制器。在網路補償方面，可以建立一網路監控系統，根據不同的網路情形，切換不同的取樣頻率或補償策略。

## 參考文獻

- [1] J. Y. Yu, S. M. Yu, and H. Q. Wang, "Survey on the Performance Analysis of Networked Control Systems," *IEEE International Conference on Systems, Man and Cybernetics*, Volume 6, pp.5068–5073, 10-13 Oct. 2004.
- [2] P. L. Tang, and C. W. de Silva, "Compensation for Transmission Delays in an Ethernet-Based Control Network Using Variable-Horizon Predictive Control," *IEEE Transaction on Control System Technology*, Volume 14, No.4, pp.707-718, Jul. 2006.
- [3] J. Nilson, B. Bermhardsson, and Wittermark, "Stochastic Analysis of Control of Real Time Systems with Random Time Delays," *Automatica*, Volume 34, Issue 1, pp.57-64, B.1998.
- [4] 鄭景文, "動態網路控制系統之時間延遲分析," 國立交通大學, 碩士論文, 中華民國 95 年
- [5] P. Sourdille and A. O'wyer, "A New Modified Smith Predictor Design", *Proceedings of the 1st international symposium on Information and communication technologies*, Vol. 49, pp. 385-390, Sep. 2003.
- [6] C. C. Hsieh, P. L. Hsu and B. C. Wang, "The Motion Message Estimator in Real-Time Network Control Systems", *Conference of the IEEE Industrial Electronics Society*, Nov. 2006.
- [7] S. H. Hong, "Scheduling Algorithm of Data Sampling Times in the Integrated Communication and Control Systems," *IEEE Transactions on Control Systems Technology*, Volume 3, Issue 2, pp.225-231, Jun 1995.
- [8] 薛伊婷, "以硬體實現主動可調式CAN 網路排程系統", 國立交通大學, 碩士論文, 中華民國 93 年7 月.
- [9] P. Marti, G. Fohler, K. Ramamritham, and J. M. Fuertes, "Jitter Compensation for Real-Time Control Systems," *22nd IEEE Real-Time Systems Symposium*, London,

UK, December, 2001.

[10] 鄧元銘, “CAN 網路同步應用協定之設計與實現”, 國立交通大學, 碩士論文, 中華民國 96 年 7 月.

[11] F. L. Lian, ” Analysis, Design, Modeling, and Control of Networked Control Systems.” Ph.D thesis, University of Michigan, May 2001.

[12] Z. Li and M. Y. Chow, “Adaptive Multiple Sampling Rate Scheduling of Real-time Networked Supervisory Control System – Part I”, *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference*, pp. 4604 – 4609, Nov. 2006.

[13] Z. Li and M. Y. Chow, “Adaptive Multiple Sampling Rate Scheduling of Real-time Networked Supervisory Control System – Part II”, *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference*, pp. 4615 – 4620, Nov. 2006.

[14] A. Antunes, P. Pedreiras, L. Almeida, and A. Mota, “Dynamic Rate and Control Adaptation in Networked Control Systems”, *5th IEEE International Conference on Industrial Informatics*, Volume 2, pp.841-846, Jun. 2007.

[15] X. D. Ren, S. B. Li, Z. Wang, M. Z. Yuan, and Y. X. Sun, “A QoS Management Scheme for Paralleled Networked Control Systems with CAN Bus”, *IEEE Industrial Electronics, IECON 2003 - 29nd Annual Conference*, Volume 1, pp.842-847, 2-6 Nov. 2003.

[16] “CAN Specification Version 2.0”, *BOSCH*, 1991.

[17] C. C. Hsieh and P. L. Hsu, “Design the Remote Control System With the Time-Delay Estimator and the Adaptive Smith Predictor,” *Industrial Electronics, IEEE Transactions on*, vol. 6, p. 73 - 80, 2010.

[18] 張哲維, “無線網路控制系統之訊息估測與網路流量監控”, 國立交通大學, 碩士論文, 中華民國 98 年 7 月.

[19] 林軒正, “異質性雙軸網路系統的同步控制策略”, 國立交通大學, 碩士論文,

中華民國 101 年 7 月.

[20] R. S. Darling, “Data-Based Predictive Control”, Ph.D thesis, Dartmouth College, May 2006.

[21] J. S. Barlow, “Data-based Predictive Control with Multirate Predictive Step,” *American Control Conference*, Baltimore, 2010, pp. 5513-5519.

[22] J. S. Barlow, and M. Q. Phan, “Direct Data-Based Model Predictive Controller Design,” *Proceedings of the AIAA Conference on Guidance, Navigation, and Control*, Keystone, CO, August 2006.

[23] Y. Wang, Z. Ji, “Data-based Predictive Control for Networked Control Systems,” *Proceedings of the 10th World Congress Conference on Intelligent Control and Automation*, Beijing, 2012, pp. 2302-2305.

[24] ZigBee Alliance, ZigBee Specification: ZigBee Document 053474r17 2008.

[25] S. Farahani, *ZigBee Wireless Networks and Transceivers* Newton, MA: Newnes, 2008.

[26] P. Yi, A. Iwayemi, and C. Zhou, “Frequency agility in a ZigBee network for smart grid application,” *Proc. Innovative Smart Grid Technol. (ISGT)*, 2010, pp. 1–6.

[27] 宋春光, “ZigBee無線感測器網路之傳輸分析與資料遺失補償”, 國立交通大學, 碩士論文, 中華民國 101 年 2 月.

[28] C. C. Hsieh and P. L. Hsu, “Networked Control Systems Design with the Intelligent Message Estimator for Data Dropout Compensation”, National Chiao Tung University, Doctoral Thesis, 2012.

[29] 廖子期, “遠端操作系統之網路延遲訊號估測與影像輔助機制設計”, 國立交通大學, 碩士論文, 中華民國 100 年 11 月

[30] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Årzén, “How does control timing affect performance?,” *IEEE Control Systems Magazine*, vol. 23, no. 3, pp. 16–30, June 2003.

[31] M. D. Biasi, C. Snickars, K. Landernäs, A. J. Isaksson, “Simulation of Process Control with WirelessHART Networks Subject to Packet Losses,” *IEEE Conference on Automation Science and Engineering*, Washington DC, Aug. 2008, pp. 548-553.

