

國立交通大學

電控工程研究所

碩士論文

使用粒子濾波器方法設計適用於不定延遲時間與封包遺失
之網路化控制系統之估測器

Observer Design for Networked Control Systems with
Indefinite Delays and Packet Loss Using Particle Filter
Techniques

研究生：林彥良

指導教授：蕭得聖 博士

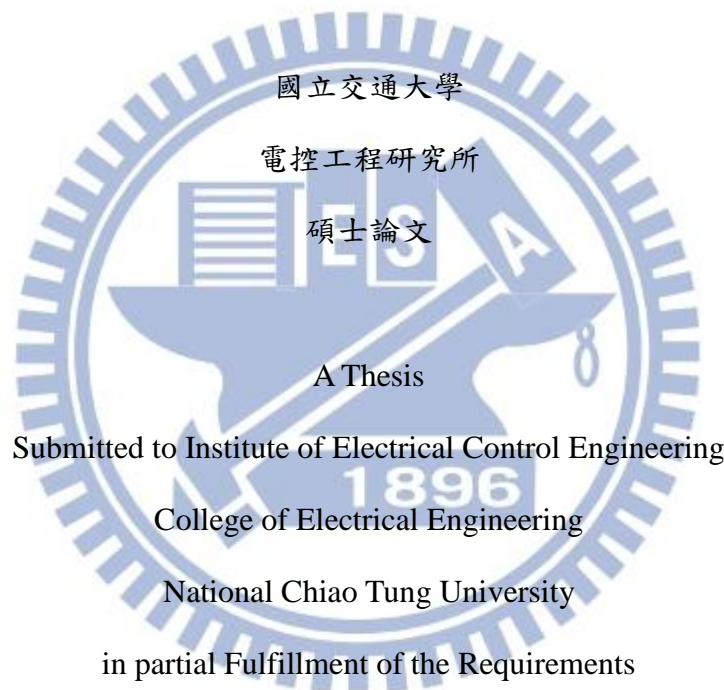
中華民國一百零一年九月

使用粒子濾波器方法設計適用於不定延遲時間與封包遺失
之網路化控制系統之估測器

Observer Design for Networked Control Systems with Indefinite Delays and
Packet Loss Using Particle Filter Techniques

研究生：林彥良
Student : Yan-Liang Lin

指導教授：蕭得聖 博士
Advisor : Dr. Te-Sheng Hsiao



in partial Fulfillment of the Requirements

for the Degree of

Master

In

Electrical Control Engineering

September 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年九月

使用粒子濾波器方法設計適用於不定延遲時間與封包遺失 之網路化控制系統之估測器

學生：林彥良

指導教授：蕭得聖 博士

國立交通大學電控工程研究所

摘要

本研究之目的是為網路化控制系統設計估測器方法，首先針對網路化控制系統當中之網路與受控系統建立一以機率為基礎之模型。其中使用馬可夫鏈隨機變數描述資訊封包在網路中傳遞之情況，用以表現不定之延遲時間與遺失機率。

由於使用機率進行模型化，因而使用粒子濾波器方法為基礎設計對應之估測器，並應用粒子濾波器中常見之改良方法如重新取樣法以及正規化方法以及選擇適當之重要性取樣分布來改善估測效果。同時提出在高階系統與考慮到估測有效視窗長度情況下，對粒子濾波器當中計算權重之公式之近似，用以減少計算需求以及節省記憶空間，同時能取得合理之權重更新趨勢。

模擬與實驗時，選取一加熱系統為處理對象，其具有多觀測端點以及應用情境可遠端之特性，並且能在較少之控制命令下得到多個系統狀態之變化。將此系統所收集到之觀測數據透過網路傳輸並紀錄，而後送至估測器對估測方法作驗證。並且將估測結果與傳統之卡曼濾波器作比較，從而驗證此設計方式之可行性與效果。

Observer Design for Networked Control Systems with Indefinite Delays and Packet Loss Using Particle Filter Techniques

Student : Yan-Liang Lin

Advisor : Dr. Te-Sheng Hsiao

Institute of Electrical Control Engineering
National Chiao Tung University



Abstract

The purpose of this research is to design a state observer specifically for networked controlled systems. We propose a probabilistic model for the network controlled system which consists of a Markov-chained random variable to represent the transferring status of each data packet. Because of the inherent probabilistic nature of the proposed model, we design the observer by using particle filtering techniques. To improve the performance, both resampling and regularization methods are incorporated into the particle filter, and the importance sampling function is carefully selected. In addition, we simplify the way to update the weightings of the samples in order to reduce the demanding computational power and the need for memory space, while to maintain reasonable state estimation accuracy.

In the simulation and experiment parts, a heating system is chosen. The system has multiple sensing nodes for a remote application, and it can change various state variables with a few control commands. After collecting the temperature data of the system, the data are sent through a real network to the observer for state estimation. The result of the estimation is then compared with the outcome of a traditional Kalman filter in order to verify the feasibility and effectiveness of the proposed method.

致謝

首先感謝我的指導教授蕭得聖博士，老師給予的指導使我了解到做研究所應有的思考方式，以及向他人傳達想法時所應有的表達方向。雖然不敢說能夠達成多少，但是這份智識我會時刻提醒自己要有所把握。

也感謝口試時徐保羅老師與陳永平老師指出我研究說明中遺失的部分，再度告訴我研究方法的重要性，也使得我有機會將這份研究內容做得更加完整。

同時也感謝我們實驗室，特別是李永洲學長對我研究上所給予的建議，對我幫助良多。以及翊熏學長還有柏俊、志偉、昌謀、俊傑同學、學弟仲謙、璟沅、維民，大家在平日的互相鼓勵與幫助，讓我在研究所的這段時間能夠不斷前進。

最後我感謝我的家人，爸爸媽媽對我的支持是我能完成到目前為止的人生成就所不可缺少的。實在很感謝大家能夠給我這個機會讓我有這段體驗與成長，不管未來如何變化，我都會懷念這兩年的時光。



目錄

中文摘要.....	i
英文摘要.....	ii
致謝.....	iii
目錄.....	iv
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	1
1.3 研究貢獻.....	2
1.4 論文架構.....	2
第二章 相關研究.....	4
2.1 以馬可夫鏈隨機過程處理.....	4
2.2 以柏努利隨機過程處理.....	6
2.3 切換系統.....	7
2.4 相關研究之總結.....	10
第三章 系統之假設與模型.....	11
3.1 系統架構與基本假設.....	11
3.1.1 系統架構.....	11
3.1.2 控制命令之假設.....	12
3.1.3 受控系統之假設.....	12
3.1.4 網路之假設.....	13
3.2 系統模型.....	13
3.2.1 受控系統機率模型.....	14
3.2.2 網路模型.....	14
3.2.2.1 網路機率模型.....	14
3.2.2.2 定義收集到之資訊集合.....	16
3.3 貝式網路機率模型.....	17
3.3.1 受控系統之圖形式模型.....	18

3.3.2 將網路模型與受控系統模型結合.....	18
第四章 估測器設計.....	20
4.1 粒子濾波器.....	20
4.1.1 基本原理.....	20
4.1.2 重要性取樣法.....	21
4.1.3 額外之取樣手段.....	23
4.1.3.1 重新取樣.....	23
4.1.3.2 正規化.....	25
4.2 根據網路化控制系統模型設計估測器.....	26
4.2.1 使用網路化控制系統模型公式推導.....	27
4.2.2 網路參數之處理.....	29
4.3 改進估測器方法.....	31
4.3.1 重要性分布的選取.....	31
4.3.2 演算公式的修改.....	34
4.4 最終版本之粒子濾波器.....	38
第五章 數據模擬.....	39
5.1 受控系統之選擇.....	39
5.2 整體系統架構.....	39
5.3 受控系統模型.....	40
5.3.1 暫態熱傳導.....	40
5.3.2 離散化.....	41
5.4 模擬的設定.....	43
5.4.1 加熱系統設定.....	43
5.4.2 網路設定.....	45
5.4.3 模擬之傳輸延遲時間.....	46
5.4.4 有效視窗大小選擇.....	47
5.5 模擬結果.....	48
5.5.1 四個網路情況之結果統計圖.....	50

5.5.2 統計結果說明.....	52
第六章 實驗驗證.....	53
6.1 實驗平台.....	53
6.2 平台中各裝置.....	54
6.2.1 發熱器.....	54
6.2.2 被加熱銅板.....	55
6.2.3 熱電耦.....	55
6.2.4 熱電耦用放大與校正晶片.....	56
6.2.5 抗交疊濾波器(anti-aliasing filter).....	57
6.2.6 F2812 數位訊號處理器.....	57
6.3 網路傳輸紀錄.....	58
6.4 加熱方程式之改動.....	59
6.5 實驗進行之設定.....	60
6.5.1 加熱系統設定.....	60
6.5.2 加熱命令.....	61
6.5.3 網路情況.....	62
6.6 實驗結果.....	64
6.6.1 三個網路情況之結果統計圖.....	66
6.6.2 結果說明.....	68
第七章 結論與未來工作.....	69
7.1 結論.....	69
7.2 未來工作.....	70
參考文獻.....	71

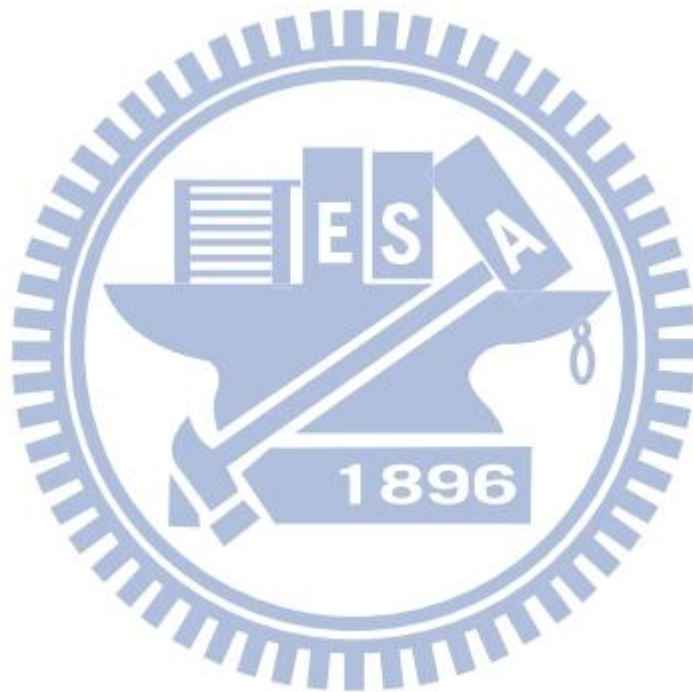
圖目錄

圖 2-1 J. Nilsson 提出方法處理之系統架構.....	4
圖 2-2 [11]中所使用之系統架構.....	7
圖 3-1 系統架構圖	12
圖 3-2 封包傳輸狀態之狀態機.....	15
圖 3-3 受控系統之圖形化模型.....	18
圖 3-4 整體系統之圖形化模型.....	19
圖 4-1 重新取樣示意圖.....	24
圖 4-2 正規化示意圖	26
圖 4-3 重要性分布選取示意圖.....	31
圖 4-4 不同網路情況下之機率密度函數比較圖.....	35
圖 5-1 模擬用系統架構.....	40
圖 5-2 有限差分法示意圖.....	41
圖 5-3 模擬用加熱系統平面圖.....	44
圖 5-4 I 與 III 號熱源之輸入.....	44
圖 5-5 II 號熱源之輸入.....	44
圖 5-6 網路情況 π_2 產生之資訊延遲時間.....	46
圖 5-7 網路情況 π_3 產生之資訊延遲時間.....	47
圖 5-8 網路情況 π_4 產生之資訊延遲時間	47
圖 5-9 不同有效視窗大小結果.....	48
圖 5-10 網路情況沒有延遲(π_1)之估測目標溫度	49
圖 5-11 網路情況平均延遲 4 時間(π_3)之估測目標溫度.....	50
圖 5-12 網路情況 π_1 沒有延遲.....	51
圖 5-13 網路情況 π_2 平均延遲兩個時間單位	51
圖 5-14 網路情況 π_3 平均延遲四個時間單位.....	51

圖 5-15 網路情況 π_4 平均延遲六個時間單位	52
圖 6-1 實驗之系統架構	53
圖 6-2 實驗平台之實物	54
圖 6-3 使用之可控溫焊槍與其陶瓷發熱體	54
圖 6-4 熱電耦裝置	55
圖 6-5 單組 AD8494 晶片之應用電路	57
圖 6-6 抗交疊濾波器電路圖	57
圖 6-7 實驗之網路紀錄方式示意圖	59
圖 6-8 實驗之加熱系統配置	61
圖 6-9 加熱點 II 之輸入溫度	62
圖 6-10 加熱點 II 之輸入溫度	62
圖 6-11 網路低負載時之延遲時間單位	63
圖 6-12 網路中-高負載之延遲時間單位	63
圖 6-13 網路高負載之延遲時間單位	64
圖 6-14 低負載網路之單次估測結果	65
圖 6-15 網路高負載之單次估測結果	66
圖 6-16 低負載網路之估測結果	67
圖 6-17 中-高負載網路之估測結果	67
圖 6-18 高負載網路之估測結果	67

表目錄

表 4-1 某一時刻以原分布與近似分布計算之有效點數占總點數百分比.....	36
表 6-1 金屬銅之物理特性.....	55
表 6-2 AD8494 特性.....	56
表 6-3 實際銅板之參數.....	61
表 6-4 三種網路情況之平均延遲.....	64



符號表

- S : 網路封包傳輸參數
- π_R : 網路封包傳輸參數轉變為接收機率
- π_D : 網路封包傳輸參數維持延遲機率
- π_L : 網路封包傳輸參數轉變遺失機率
- Z : 透過網路傳輸後收取之資料
- M : 有效視窗大小
- N_s : 粒子濾波器取樣數目
- w : 權重
- \tilde{w} : 一般化後之權重
- $N_{threshold}$: 重新取樣有效點數閾值
- T : 端點溫度
- α : 熱傳導率



第一章 緒論

1.1 研究背景與動機

在一般的即時控制系統中，當感知器之數量較少、容易連接時，通常直接將感知器之輸出端以直接連接的方式連至估測/控制端。以此方式所獲得之觀測資訊幾乎是即時並且連續完好。

但隨著感知器數量的增加、或者系統本身與估測/控制端之距離的增加，為了周邊連接線路的減少與統整、為了將資訊送至估測/控制端，而有以網路作為資訊傳輸介面的方式。而此種在系統當中包含網路作為傳輸介面的控制系統，便稱為網路化控制系統。以網路作為傳輸介面後，原先繁雜的連接線路以網路線取代，後續之整合與調整都變得容易；而難以達成的多端點與遠端監控也可以藉由網路輕易達到。

而在控制系統中，對回授之資訊永遠是希望越接近真實、誤差越少越好，但是在網路化控制系統中，由於使用網路作為資訊的交換介面，因此資訊的獲得便會受到網路傳輸的影響。如由網路延遲與資料封包的遺失，則估測端便會因此接收到較舊之資訊或者沒有收到任何資訊，因此，在估測器當中所使用的演算法，便需要有不同的設計。

1.2 研究目的

在網路化控制系統中，受到以網路為傳輸介面的影響，估測器端接收到的資料有延遲與遺失的情況。本研究提出一個描述網路延遲與封包遺失的機率化模型，並根據此模

型設計一基於粒子濾波器(particle filter)的狀態估測器，以減少資料延遲與遺失所帶來的估測誤差。

為了驗證所提出的網路模型與狀態估測器之效果，本論文將所提出的方法應用於一加熱系統之溫度估測。該加熱系統有感測資訊量大，且有遠端應用之特性。藉由模擬與實作，並與傳統卡曼濾波器做比較，凸顯本方法對網路化控制系統狀態估測之效果。

1.3 研究貢獻

本論文針對網路化之控制系統設計了一網路模型與對應之估測方法。而此網路模型無針對任一網路協定做描述，可適用於已知之多數網路應用情況。而提出之估測法為粒子濾波器之應用，針對網路模型做適當調整，以及對實際運算情況設計對應的相關處理手段。

同時，使用一多觀測點之加熱系統以及實際採集之網路資訊做驗證。此驗證提供了方法應用的可能情境以及運作方式。

1.4 論文架構

第一章是為緒論，此章節闡述研究之背景與目的，以及本研究之貢獻。

第二章接著顯示與網路化控制系統相關之研究，注重於估測以及網路之描述。

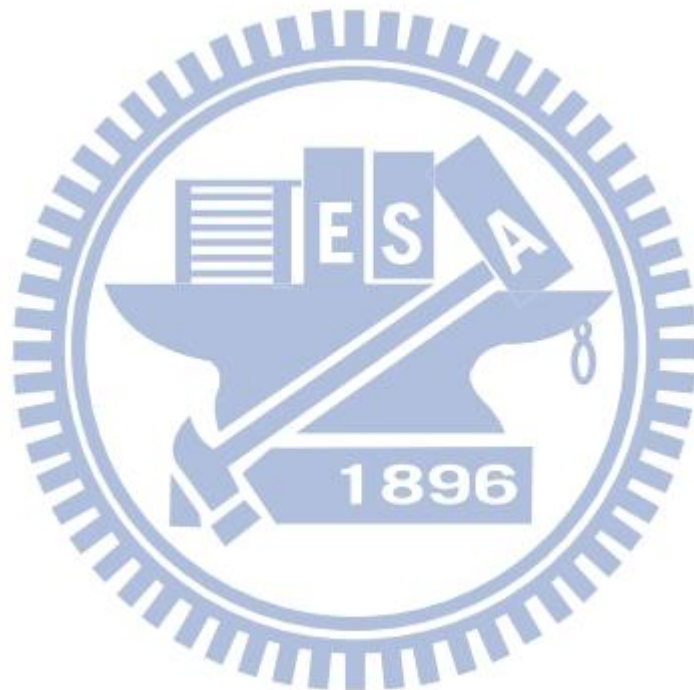
第三章說明本研究對網路化控制系統當中出現之各個系統方塊之假設以及其對應之模型推導。

第四章說明本研究所使用之估測器方法的基礎，粒子濾波器的原理與運作方式說明，並且將前面所定義與推導之各個變數公式應用到估測器的設計當中。同時並討論此估測法的改進手法。

第五章將設計一適合網路化控制系統的模擬情境，使用軟體模擬的方式來測試所設計之估測器之效果。

第六章將製作一硬體平台，將前面所提之模擬情境實現，利用此一平台說明整體系統的可能應用情境，並且在實際採集數據後用以驗證估測法的效果。

第七章為結論與未來工作，討論本研究所提之成果以及可能之改良與發展方向。



第二章 相關研究

網路化控制系統是一個逐漸發展中的課題[1]，其中針對以網路為傳輸介面所引起的延遲與遺失資訊現象，有許多不同的模型化方式及應對的方法。在本章節中，將舉出其不同的網路模型化方式以及估測策略，說明其利用上之缺失與可能改進之處。

2.1 以馬可夫鏈隨機過程處理

在 1998 年，Johan Nilsson 提出將網路延遲時間以馬可夫鏈的方式模型化[2]，在那之後亦有人利用類似作法發表了相關研究[3][4]。

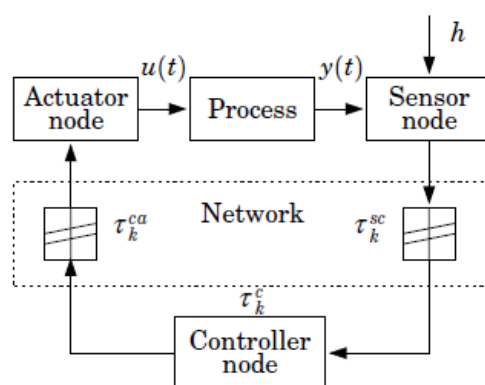


圖 2-1 J. Nilsson 提出方法處理之系統架構

在此作法中，感測器與控制器之間是以網路傳遞資訊，有延遲時間 τ_{sc} ，以及控制器到制動器之間亦使用網路傳輸，因此有 τ_{ca} 的延遲時間，再加上控制命令計算時間 τ_k^c ，使得整個控制命令的延遲時間達到 $\tau_k = \tau_k^{sc} + \tau_k^c + \tau_k^{ca}$ 。

$$(2.1)$$

同時，假設系統可以在接收到控制命令後回傳確切之 τ_{k-1}^{ca} 資訊，並且此資訊抵達時間不會晚於下個週期內，

$$\tau_{k-1}^{sc} + \tau_{k-1}^{ca} + \tau_{k-1}^{ca-b} < \tau_k^{sc} + h \quad (2.2)$$

其中 τ_{k-1}^{ca-b} 為回傳所需時間， h 為一個取樣週期。

在確認網路延遲的約略變化範圍後，將延遲時間由最小至最大切割成 s 個小單位時間， $r_k \in \{1, \dots, s\}$ 。於是 τ_k 為一個馬可夫鏈隨機變數，如同 r_k ，並且其狀態轉移矩陣為

$$Q = \{q_{ij}\}, \quad q_{ij} = p\{r_{k+1} = j | r_k = i\} \quad (2.3)$$

於是網路傳輸資訊延遲時間以此方式令其成為馬可夫鏈隨機變數，在得到上一個時刻之延遲時間後，得到當前延遲時間長短的分布機率，依此將此系統表示如下：

$$x_{k+1} = \Phi x_k + \Gamma_0(\tau_k)u_k + \Gamma_1(\tau_k)u_{k-1} + v_k \quad (2.4)$$

其中

$$\begin{aligned} \Phi &= e^{Ah} \\ \Gamma_0(\tau_k^{sc}, \tau_k^{ca}) &= \int_0^{h-\tau_k^{sc}-\tau_k^{ca}} e^{As} ds B \\ \Gamma_1(\tau_k^{sc}, \tau_k^{ca}) &= \int_{h-\tau_k^{sc}-\tau_k^{ca}}^h e^{As} ds B. \end{aligned}$$

v_k : 系統雜訊

接著此篇研究提出一搭配之 LQG 控制器，並在最後附上穩定性之證明推導。

此篇研究所假設之網路延遲時間被切割成一個離散的時間單位，並且將其變化描述成馬可夫鏈的形式。並且在估算系統狀態時，考慮到前一個延遲時間的長短，來推斷將來控制命令的延遲時間，如此之設定使得其效能相當依賴其對網路情況之掌握情形。

同時，此篇研究對於延遲時間長短亦有限制，目標的處理對象為延遲時間小於一個取樣週期、觀測資訊抵達順序與出發順序相同而無失序，因此也就將適用範圍做出很大的限制。

2.2 以柏努利隨機過程處理

在[5]與[6]中，其將網路傳輸情況以柏努利隨機過程的方式來處理。首先，柏努利隨機過程的敘述如下：

有一個隨機變數為 δ ，其可能的值為 0 與 1。並且分布符合

$$\begin{aligned}\text{Prob}\{\delta = 1\} &= \mathbb{E}\{\delta\} := \bar{\delta} \\ \text{Prob}\{\delta = 0\} &= 1 - \mathbb{E}\{\delta\} := 1 - \bar{\delta}\end{aligned}\quad (2.5)$$

首先，假定系統送出觀測資訊與控制命令時皆通過一有延遲之網路，該延遲的長短以系統取樣週期分割為二，一個為短於系統取樣週期，一個為長於系統取樣週期。當延遲時間短於系統取樣週期時，將其視為及時送達；當延遲時間長於系統取樣週期時，將其視為沒有送達，並且使用前一個時刻的資訊來替代。以此，將系統描述如下：

$$\begin{aligned}x_{k+1} &= Ax_k + \beta_k Bu_{k-1} + (1 - \beta_k) Bu_{c,k} \\ u_k &= \beta_k u_{k-1} + (1 - \beta_k) u_{c,k} \\ y_{c,k} &= (1 - \alpha_k) Cx_k + \alpha_k y_{c,k-1}\end{aligned}\quad (2.6)$$

其中， α_k 與 β_k 為柏努利隨機變數，其分佈符合下列表示

$$\begin{aligned}\text{Prob}\{\alpha_k = 1\} &= \mathbb{E}\{\alpha_k\} := \bar{\alpha} \\ \text{Prob}\{\alpha_k = 0\} &= 1 - \mathbb{E}\{\alpha_k\} := 1 - \bar{\alpha} \\ \text{Prob}\{\beta_k = 1\} &= \mathbb{E}\{\beta_k\} := \bar{\beta} \\ \text{Prob}\{\beta_k = 0\} &= 1 - \mathbb{E}\{\beta_k\} := 1 - \bar{\beta}\end{aligned}\quad (2.7)$$

並且根據上述之設定，[5]與[6]皆以 LMI 的方式設計了對應之 H_∞ 控制器，並且在之後證明其穩定性。

在此類網路假設中，其對於資訊延遲之處理較為單純，其只設定兩種情況，當延遲時間短於取樣週期時，視為沒有延遲，而延遲長於取樣週期時則直接將其視為沒有取

得，而馬上取前一時刻所得之資訊來使用。並且其對網路情況僅以一參數來描述，只表達比系統取樣週期長或短，無法精確描述大於一個取樣週期之延遲時間。

因此，此方式除對估測法之設計較為單純，僅僅使用即時沒有延遲之資訊，系統之取樣時間就必須配合網路情況做設定，無法在網路延遲時間多數大於系統取樣時間的情況下運作。

2.3 切換系統

在 [11][12]中，將網路化控制系統中網路介面之因素模型化為一個切換系統。並且此二篇研究所處理之系統架構為在感測器與控制器之間使用網路為傳輸介面，控制器到制動器之間則無，如下圖。

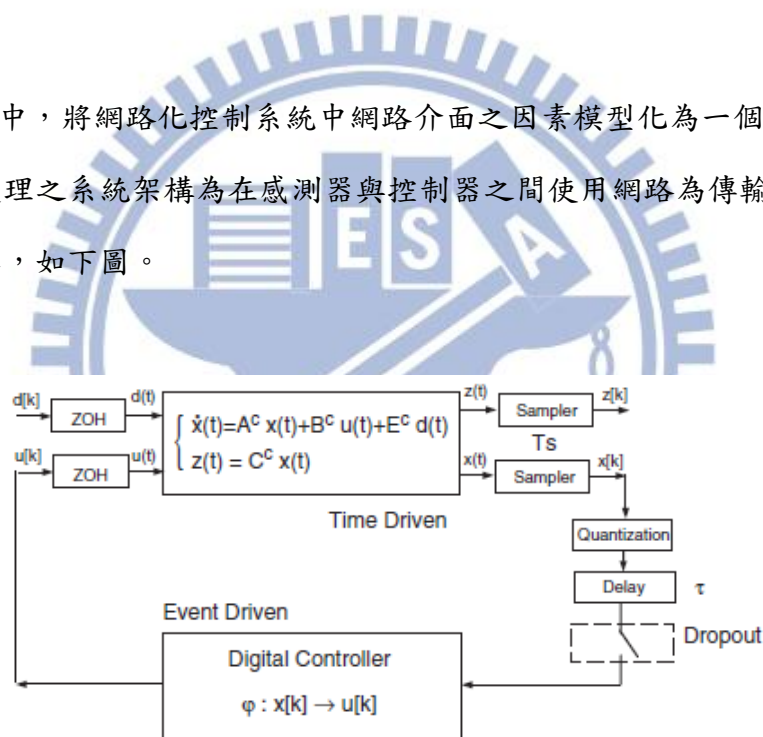


圖 2-2 [11]中所使用之系統架構

其假設感測器以 T_s 之週期送出觀測資料，當超過一個週期仍未確認資料之送達時，則放棄傳送，直接送出下一筆最新之資料。而控制器則是以 T_s/N 之週期運作，較系統為快，控制器僅收取感測器送出之最新之一筆觀測資訊，上個週期之資訊則被丟棄。當有控制器收到此最新之觀測數據時，會輸出新的控制命令給以 zero-order- hold 運作之制動器。於是將此系統表述如下：

$$x[k+1] = Ax[k] + \underbrace{[B \quad B \quad \dots \quad B]}_N \begin{bmatrix} u^1[k] \\ u^2[k] \\ \vdots \\ u^N[k] \end{bmatrix} + Ed[k]$$

$$z[k] = Cx[k]$$

X 是系統狀態，z 則是感測器之輸出，ABCE 都是固定之系統矩陣，d[k] 則是系統雜訊。由於控制器工作較快， $u^1[k], u^2[k], \dots, u^N[k]$ 即是在此系統周期內每次控制器更新時所下達之命令。由於感測器與控制器對超過一週期之資料直接以丟棄處理，因此一週期內最多僅會收取一筆資料，故分成三種情況：

1. 當網路無延遲時，整個週期內之控制命令為新的控制命令 $u[k]$ ，

$$u^1[k] = u^2[k] = \dots = u^N[k] = u[k]$$

於是系統便可寫為

$$x[k+1] = Ax[k] + N \cdot Bu[k] + Ed[k]$$

即與一般控制系統無異。

2. 當延遲時間為 $\tau = h \times Ts / N$ 時，則觀測未抵達前控制命令為 $u[k-1]$ ，抵達後為 $u[k]$ ，

$$u^1[k] = u^2[k] = \dots = u^h[k] = u[k-1]$$

$$u^{h+1}[k] = u^{h+2}[k] = \dots = u^N[k] = u[k]$$

此系統則化為

$$x[k+1] = Ax[k] + h \cdot Bu[k-1] + (N-h) \cdot Bu[k] + Ed[k]$$

3. 當完全沒有收到延遲資訊時，

$$u^1[k] = u^2[k] = \dots = u^N[k] = u[k-1]$$

於是系統便為

$$x[k+1] = Ax[k] + N \cdot Bu[k-1] + Ed[k]$$

接著並假設系統最多連續 D_{\max} 次沒有收到任何資料。並且沒有收到資料的情形是以周期 T_m 出現的，此 T_m 是 T_s/N 的倍數， $T_m/T_s=m$ 。當 $m=1$ 時沒有遺失， $m \geq 2$ 時則要同時考慮遺失與延遲之情況。依照上面所列之三種情形將 x 以每 T_s 時間間隔連續將上列之系統帶入，最後可以整理出

$$\hat{x}[k+1] = \Phi_{(m,h)}\hat{x}[k] + E_m d[k]$$

$$z[k] = [C \ 0]\hat{x}[k], \quad \hat{x}[k] = \begin{bmatrix} x(kT_m - T_s) \\ x(kT_m) \end{bmatrix},$$

$$\Phi_{(m,h)} = \begin{cases} \begin{bmatrix} 0 & I \\ hBK & A + (N-h)BK \end{bmatrix}, & m = 1 \\ \begin{bmatrix} N \sum_{i=0}^{m-2} A^i BK & A^{m-1} \\ \left(N \sum_{i=1}^{m-1} A^i + (N-h)NBK \sum_{i=0}^{m-2} A^i + h \right) BK & A^m + (N-h)BK A^{m-1} \end{bmatrix}, & m \geq 2 \end{cases}$$

$$E_m = \begin{cases} \begin{bmatrix} 0 \\ E \end{bmatrix}, & m = 1 \\ \begin{bmatrix} \sum_{i=0}^{m-2} A^i E \\ (N-h)BK \sum_{i=0}^{m-2} A^i E + \sum_{i=0}^{m-1} A^i E \end{bmatrix}, & m \geq 2. \end{cases}$$

其為 $m+h \times D_{\max}$ 個模式之切換系統。

此篇研究所提之系統架構僅有觀測端有網路連接，並且處理目標是小於一個取樣週期之延遲時間以及封包遺失。其對網路運作有設定最大連續遺失限制(D_{\max})，並沒有任何機率上的假設，依靠當前收取到之資料來決定此切換系統之切換。

然而此設計隨著 $m=T_m/T_s$ 、 D_{\max} 越大，切換系統之模式越多，也就是設計控制時手續將會更為繁複，並且此設計尚沒有考慮到控制端有網路，也是此方法有所限制之處。

2.4 相關研究之總結

經由以上所提之幾種方法可以見到，其所處理之對象通常為較短之延遲時間，使得系統運作頻率需搭配調整，並在估測時採用之手段多為直接假設當尚未接收時，直接採用前一時刻的值來當作輸入，或者給予一個依網路機率估算之增益。並且對於網路之描述甚或採用單一變數、或者依延遲時間分割為切換系統等，落於過於單純或者套用系統模型複雜。

本研究提出之網路模型可適用之延遲時間並無特別限制，並盡量一般化設定使得可用於任何表現之網路。所提之對應估測法則善加利用延遲之資料，使用上亦可容易套用於一般線性之系統，方便設計。



第三章 系統之假設與模型

本論文之探討對象為網路化控制系統，為了針對此種系統設計估測器，將網路模型化以及將其與控制系統之模型結合成一完整之描述便是研究之重點。本章節針對設計估測器方法中所需使用到之系統架構與基本假設以及受控系統模型、網路模型做說明與推導。並且在最後將受控系統與網路模型以貝式網路機率模型表示法結合表示成一個圖形式模型。

3.1 系統架構與基本假設

本節將說明本研究處理之系統架構以及其中各方塊中所代表之功能以及其基本假設。假設的產生有些是為事實之物理限制，有些則是由於估測方法的使用限制而產生，當產生理由是為後者時，無法達成該假設時，便會對估測效果產生影響。

3.1.1 系統架構

由於本研究之目標為替網路化控制系統設計估測器，因此在方法的研究中，系統內包含有：控制命令、受控系統、網路、估測器。其中控制命令採直接連接，以即時的方式傳送至受控系統，而由受控系統中安置之感測器採集之觀測資訊則透過網路傳送至估測器端。如此設定是為將研究重點放置在估測器方法，而無考慮到控制器之設計。

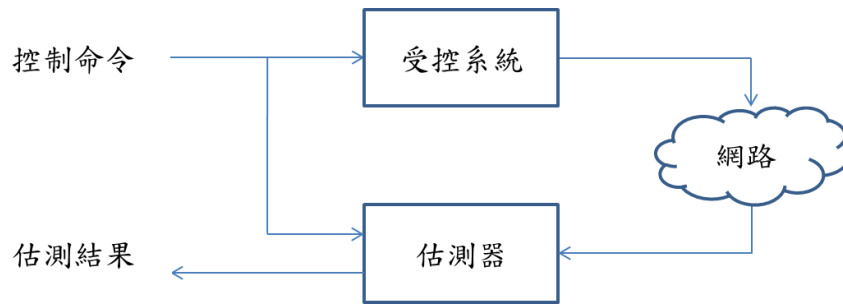


圖 3-1 系統架構圖

在實際使用上的例子有比如大樓之空調系統，空調系統的控制為直接中央控制，而溫度或濕度感應器則大量散佈在整個建築空間中，這些感應器則使用無線網路通訊；又或者如以雲端作為遠端大量端點資訊監控，各端點之系統之制動器安置在同地點，而監控資訊則透過雲端的方式送至監控端。當控制系統中觀測端點數量眾多或與估測端距離遙遠時，採用網路來傳輸觀測資訊便是一個可能的作法，因此而有了這樣的架構設定。

3.1.2 控制命令之假設

在本研究中，由於沒有控制器之設計，而將重點放在估測效果。因此，控制命令在所有公式推導中皆視為已知，並且此控制命令以即時、無透過網路的方式傳送至受控系統中之制動器。

3.1.3 受控系統之假設

本研究中所設計之估測方法以受控系統為線性系統之基本假設而推導而成。因此在此出現之受控系統須為線性系統，而此系統受到模型之不準確性以及雜訊等影響，在系統中有系統雜訊以及觀測雜訊此二雜訊影響。此系統之離散時間狀態空間表示式可表示為如下：

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k & x_k &\in \mathfrak{R}^{n_x} \\ y_k &= Cx_k + Du_k + v_k & y_k &\in \mathfrak{R}^{n_y} \end{aligned} \quad (3.1)$$

x_k 是為系統狀態， u_k 為控制輸入， y_k 為觀測輸出， A, B, C, D 為系統矩陣。

w_k 為系統狀態受到干擾或者模型不準確時，加入之系統雜訊； v_k 則為觀測雜訊。此二雜訊被假設為 White Noise，其為平均值為 0，並且共變異數矩陣(covariance matrix)分別為 Q 與 R 之高斯雜訊。

3.1.4 網路之假設

本研究所提出之估測方法須得知每一筆透過網路所獲得之資訊之個別延遲時間，並且在每一個來自同一時間之觀測端點之觀測值都抵達時，方將其視為該時間觀測資料之抵達，無資料之缺漏。若資訊之延遲時間並非實際傳輸所花費之時間，例如受控系統與估測器兩端之時間並無同步，則估測效果將會受影響。

延遲時間的取得可以透過手動在系統時間附加在封包資訊內獲得，而系統時間之同步與否除會影響估測結果外，本研究內之公式與理論之推導並無對網路所使用之各種協定與架設等有特殊之設定或要求。

3.2 系統模型

由於本研究所提出之估測方法是以粒子濾波器為基礎所發展之方法，而粒子濾波器是一個以機率方法為中心概念之估測方法。因此在接下來的推導與說明中，會把受控系統與網路轉為以機率的方式表達之數學模型，並且在最後將受控系統與網路之機率模型結合成一整體系統之機率模型，並且以貝氏網路(Bayesian Network)機率模型表示法將其

圖形化，此圖形化模型可使得判斷各變數間的條件相依性變得容易，幫助後續公式之推導。

3.2.1 受控系統機率模型

由(3.1)以及 w_k 與 v_k 的機率分布可知 x_{k+1} 為平均值為 $Ax_k + Bu_k$ ，共變異數矩陣為 Q 之高斯隨機變數； y_k 為平均值為 $Cx_k + Du_k$ ，共變異數矩陣為 R 之高斯隨機變數，表示如下：

$$\begin{aligned} x_{k+1} \Big|_{x_k} &\sim \mathbf{N}(x_{k+1} \mid Ax_k + Bu_k, Q) \\ y_k \Big|_{x_k} &\sim \mathbf{N}(y_k \mid Cx_k + Du_k, R) \end{aligned} \tag{3.2}$$

3.2.2 網路模型

3.2.2.1 網路機率模型

當感測器將資料蒐集後，將數據以封包的形式經由網路送出至估測端。而此封包在網路傳輸過程中會花費時間，其中有由於傳輸距離與頻寬所造成的物理上的最少所需時間，以及由於網路其他資訊的封包傳輸壅擠而產生的額外等待時間，或者是由於中間某段線路的不穩定而造成的封包資料遺失等情況。而最終當封包有抵達估測端時，將可由封包中附加之時間資訊獲得確定的總延遲時間。

在此設定在一個時間出發的資訊封包，隨著系統時間的前進，封包將會有三種可能的狀態，分別為接收(Receipt)、延遲(Delay)，以及遺失(Loss)。顧名思義，當封包狀態為接收時，代表封包完好到達估測器端，並且被使用於估測演算法則中、不會在之後的時間到達；當封包狀態為延遲，代表此封包仍然在網路中傳輸、尚未到達估測器端；當

封包狀態為遺失，代表此封包已經不會繼續傳輸、永遠不會到達估測器端。

依據這些網路資訊封包的物理表現，假定有一參數 S ，使用此參數來表達特定資訊封包的狀態，並且可以得到此參數之有限狀態機(finite state machine)圖形如下：

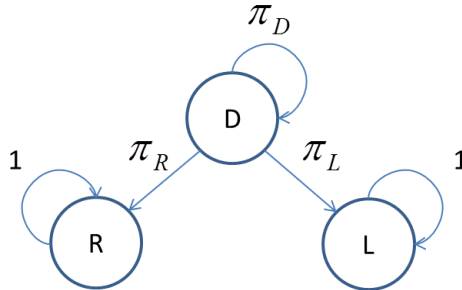


圖 3-2 封包傳輸狀態之狀態機

其中 π_R 、 π_D 、 π_L 分別為該資訊封包由延遲轉變為接收、延遲保持為延遲、由延遲轉變為遺失之機率。而此機率是一個與當前網路傳輸之品質、物理限制，以及由資訊封包出發至當前時刻已經過之時間有關之函數。於是可以將網路參數 S 定義如下：

$$S_{k,m} \in \{R, D, L\}, k \geq m \quad (3.3)$$

Transition Matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ \pi_R(k-m) & \pi_D(k-m) & \pi_L(k-m) \\ 0 & 0 & 1 \end{bmatrix}, \pi_R(k-m) + \pi_D(k-m) + \pi_L(k-m) = 1 \quad (3.4)$$

上式所代表的是：網路參數 $S_{k,m}$ 是由 m 時間出發的資訊封包，在 k 時間點時的狀態，並且其下一個時刻的狀態轉變機率視乎當前的狀態以及由 m 到 k 已經過時間而定。

於是透過網路傳輸之資料，由於每個封包的延遲時間不一定，在同一時間內可能收到來自多個時間點的資訊。而透過網路參數 S ，可以定義某一個時間點出發的封包，在一特定時間是否抵達估測端並被納入估測計算中。

已知一個封包，其當前時刻 k 之狀態為 R (接收)，而且其前一個時刻之狀態為 D (延遲)，代表此一封包是在此時間點才抵達，並非抵達許久，因此使一封包之資訊將會被

納入估測當中。根據此推論，若有封包在 k 時間被收到，其當前網路參數必為 R ，而前一時間點網路參數須為 D ，除非其出發時間便是 k ，也就是傳輸花費時間小於一單位系統時間，此情況被視為即時獲得該資訊。

據上推論，以下定義一集合 Z_k ，其代表的是在 k 時間點所收取到的資料的集合：

$$Z_k = \left\{ y_i \left| \begin{array}{l} S_{k,i} = R, S_{k,i-1} = D, k-M \leq i < k \\ \text{or } S_{k,i} = R, i = k \end{array} \right. \right\} \quad (3.5)$$

其中 M 代表的是估測方法接受的有效視窗大小，當資料的出發時間在有效視窗內，代表該資料並不是過舊的。過舊的資料給予的狀態資訊由於受控系統不斷的變化而失去計算價值，因此有效視窗大小應視乎當前網路品質以及系統變化之速度而定。

3.2.2.2 定義收集到之資訊集合

由式子(3.5)可以注意到，收集到之資訊 Z 所代表的是一個集合，這個集合的內容含有來自何時刻的觀測值 y 是隨著對應的網路參數所改變，並且， y 值的大小則又與系統狀態 x 有關。也就是說， Z 並不是一般常見之機率分布，而是一個具有離散組合與數值的連續分布特性的複合分布，因此需要利用先前所提之基本設定以及網路之物理限制來寫出其定義，如此方可使用在估測器中使用。

Z 中可能收到總視窗長度 $M+1$ 個的 y ，是否有收到視乎網路情況，而收到的 y 值為多少則又與該筆觀測資料出發時刻之系統狀態 x 有關，因此先定義 Z 是一個與 x 和 S 有關之分布，並且可以將其機率密度函數(probability density function, PDF)展開為所有可能收取之觀測值 y 分布之連乘：

$$p(Z_k | x_{k-M:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1}) = p(y_k | x_k, S_{k,k}) \prod_{i=1}^M p(y_{k-i} | x_{k-i}, S_{k,k-i}, S_{k-1,k-i}) \quad (3.6)$$

而這當中出現之 $p(y_k | x_k, S_{k,k})$ 與 $p(y_{k-i} | x_{k-i}, S_{k,k-i}, S_{k-1,k-i})$ 則根據網路參數 S 對資料是否被接收的定義來寫出如下之定義：

$$p(y_k | x_k, S_{k,k}) = \begin{cases} \mathbf{N}(y_k | Cx_k, R) & , S_{k,k} = R \\ \delta^*(y_k) & , otherwise \end{cases} \quad (3.7)$$

$$p(y_{k-i} | x_{k-i}, S_{k,k-i}, S_{k-1,k-i}) = \begin{cases} \mathbf{N}(y_{k-i} | Cx_{k-i}, R) & , S_{k,k-i} = R, S_{k-1,k-i} = D \\ \delta^*(y_{k-i}) & , otherwise \end{cases}, 1 \leq i \leq M \quad (3.8)$$

其中 $\delta^*(y)$ 為 impulse 出現在無窮遠處之 Dirac delta 函數，其對任意包含有限值之 y 的集合積分，結果皆為零，代表有限大小的 y 出現的機率為 0。

於是，根據此定義， $\int p(y_k | x_k, S_{k,k}) dy_k$ 與 $\int p(y_{k-i} | x_{k-i}, S_{k,k-i}, S_{k-1,k-i}) dy_{k-i}$ 在給予固定之 x 與 S 之下為 1，便成為一合法之機率密度函數。

當實際計算此分布，且 Z_k 當中並沒有收到某 y_i 時，則將整體機率密度函數

$p(Z_k | x_{k-M:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1})$ 對 y_i 邊際化(marginalize)，以將此缺少之變數自公式中消去。

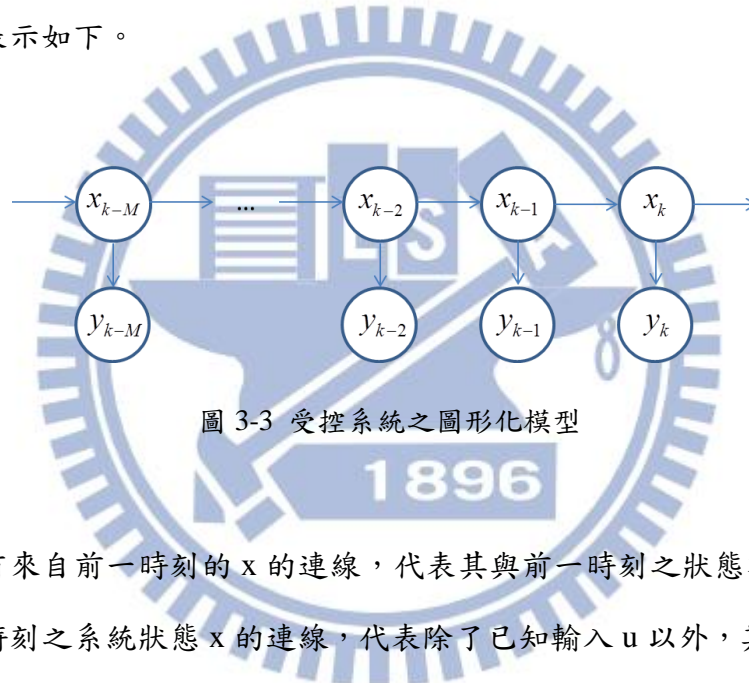
3.3 貝式網路機率模型

貝式網路機率模型表示法是一種將機率模型中出現之隨機參數以及其之間之相依性以圖形化表示的方法，也稱作圖形式模型(graphical model)。此方法以圓圈代表隨機變數，圓圈之間以連線代表參數之間存在條件相依性。在此節，將受控系統與網路模型為圖形式模型結合表示，幫助了解系統中各參數之間的關係，增進分析與推導的便利性。

3.3.1 受控系統之圖形式模型

首先，在受控系統中由於輸入命令 u 為已知，因此尚有兩個變動的隨機參數，分別為系統狀態 x ，以及觀測輸出 y 。

其中系統狀態 x 之是由前一個時刻之系統狀態以及輸入而得，由於輸入為已知，因此唯一有相依性之隨機變數僅為前一個時刻之系統狀態，與再過往之系統狀態無關，這種關係稱之為馬可夫鏈；而當前輸出 y 則是僅與當前系統狀態有關。至此，將受控系統的圖形化模型表示如下。



每個 x 皆有來自前一時刻的 x 的連線，代表其與前一時刻之狀態有關；而觀測值 y 則有來自同一時刻之系統狀態 x 的連線，代表除了已知輸入 u 以外，其僅與同一時刻之變數 x 有關聯。

3.3.2 將網路模型與受控系統模型結合

在前面的章節，網路模型部分定義出的 $p(Z_k | x_{k-M:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1})$ (式 3.5)，可以注意到當中所出現之變數僅有 Z 、 x 以及 S ，而觀測數據 y 則因為透過網路傳輸而產生了條件相依的不確定性，其與 x 的關係仍保持為高斯分布，然而 Z 卻是依靠網路參數

S 所決定之特定 y 的集合。

因此，將整體模型轉為圖形化表示的時候，y 這個隨機變數被隱藏到 Z 當中，而 Z 需要透過 x 與 S 來得到其內容，於是繪出下面的貝氏網路機率模型：

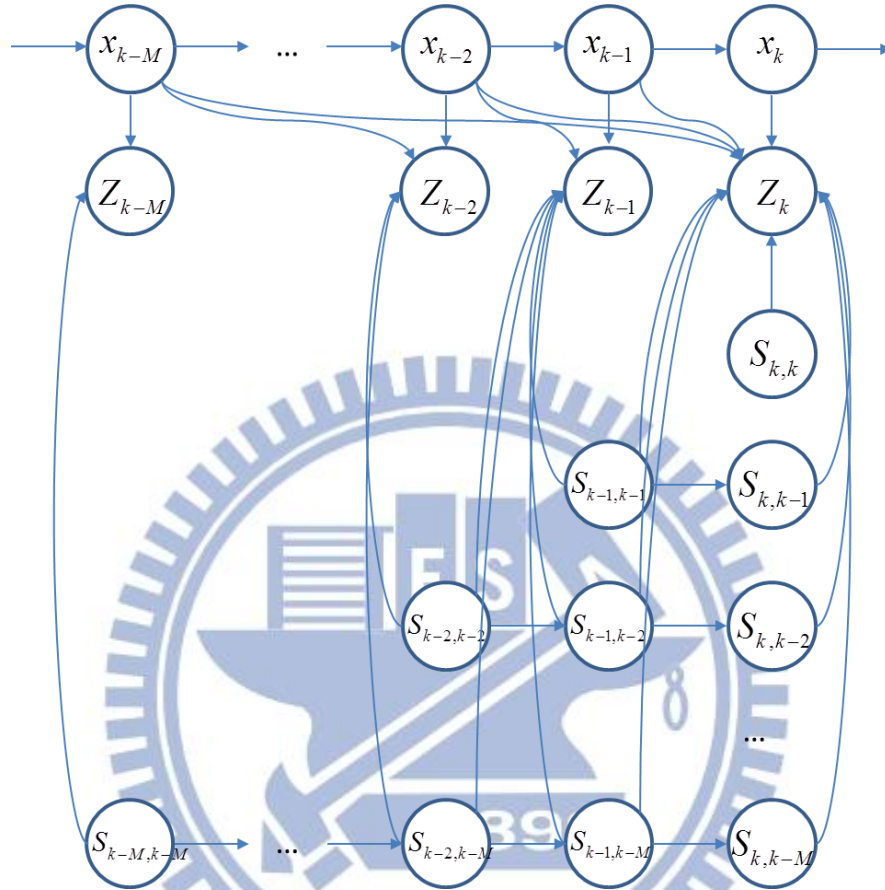


圖 3-4 整體系統之圖形化模型

此模型代表在 k 時刻，在計算的有效視窗內所出現之隨機變數之間的條件相依性。x 與 S 本身都是屬於馬可夫鏈的變數，只與前一時刻自己本身的狀態有關。而 Z 則是依該時刻與前一時刻之網路參數 S 來決定收取到的 y，以及參照那些觀測值 y 所對應之時刻的系統狀態 x，如此可以看到圖形中 Z 有來自 x 與 S 的連線，而 x 與 S 則有來自前一時刻自身的連線。如此便完成將系統各個方塊做定義與模型化，模型化之後的公式是配合為估測器所使用、以機率為基礎來描述。同時，圖形化模型則使得各個隨機變數之間的條件相依性一目了然，方便在估測器之中做進一步公式推導。

第四章 估測器設計

在本章節中，將詳述針對網路化控制系統設計之估測器方法，此方法以粒子濾波器的理論為基礎，因此將會解釋粒子濾波器之理論原理以及基本的處理過程。並且接著根據前章所闡述之網路化控制系統模型，設計其對應之估測器。除基礎之估測器公式結果外，此章節還會包含此估測方法之可能改進方式的說明以及以此得到之最終在模擬以及實驗中所使用之估測器方法。

4.1 粒子濾波器

粒子濾波器是一個以機率方法為基礎所發展之估測方法。以下將從其基本原理開始，並且講述其中一種發展作法-重要性取樣法，說明其推導過程以及運作時之流程。

4.1.1 基本原理

令系統狀態為 X_k ，其為一具有馬可夫鏈特性之隨機變數；觀測為 Y_k 。並且將其各表示為如下之機率表示式

$$\begin{aligned} X_{k+1} | X_k &\sim p(X_{k+1} | X_k) \\ Y_k | X_k &\sim p(Y_k | X_k) \end{aligned} \tag{4.1}$$

在 k 時間時，給定觀測值 $\{Y_1, Y_2, \dots, Y_k\}$ ，可將其寫成 $Y_{1:k}$ ，則其對應之最佳估測結果

$\{X_1, X_2, \dots, X_k\}$ ，或寫作 $X_{1:k}$ ，為期望值 $E[X_{1:k} | Y_{1:k}]$ 。將此條件期望值寫作

$$\bar{X}_{1:k} = \int X_{1:k} p(X_{1:k} | Y_{1:k}) dX_{1:k} \quad (4.2)$$

其中 $p(X_{1:k} | Y_{1:k})$ 代表給定 $Y_{1:k}$ 時 $X_{1:k}$ 的條件機率密度函數(conditional probability density

function)。若對 $p(X_{1:k} | Y_{1:k})$ 進行取樣，令 X_m^i , $m=1, 2, \dots, k$ 且 $i=1, 2, \dots, N_s$ ，代表 X_m 的第 i

個樣本，然後計算取樣平均值(sample mean)來代替積分，亦即

$$\bar{X}_{1:k} = \int X_{1:k} p(X_{1:k} | Y_{1:k}) dX_{1:k} \approx \frac{1}{N_s} \sum_{i=1}^{N_s} X_{1:k}^i \quad (4.3)$$

當其中之 N_s 趨近於無限大時，則其可逼近於實際之期望值條件。

4.1.2 重要性取樣法

重要性取樣法(Importance Sampling)，此方法是由於原先的條件期望值當中之

$p(X_{1:k} | Y_{1:k})$ 在實際運算時並不一定是一個可以輕易取樣的機率分佈函數，因此使用另一

容易取樣之機率分佈函數 $q(X_{1:k} | Y_{1:k})$ 來取樣，該式稱作 Importance Distribution，或

Proposal Distribution。於是，便將式(4.3) 改寫為

$$\frac{1}{N_s} \sum_{i=1}^{N_s} X_{1:k}^i = \sum_{i=1}^{N_s} w^i X_{1:k}^i, \quad w^i = \frac{p(X_{1:k}^i | Y_{1:k})}{q(X_{1:k}^i | Y_{1:k})} \quad (4.4)$$

其中 w^i 稱為權重， $\sum_{i=1}^{N_s} w^i = 1$ ，而與(4.3)同樣，當 N_s 趨近於無限大或者足夠大時，便可以

趨近實際期望值。

並且，利用(4.1)所提之 X 之馬可夫鏈特性，可將 $p(X_{1:k} | Y_{1:k})$ 進行如下之拆解

$$\begin{aligned}
& p(X_{1:k} | Y_{1:k}) \\
&= \frac{p(Y_k | X_{1:k}, Y_{1:k-1}) p(X_{1:k} | Y_{1:k-1})}{p(Y_k | Y_{1:k-1})} \\
&\propto p(Y_k | X_{1:k}, Y_{1:k-1}) p(X_{1:k} | Y_{1:k-1}) \\
&= p(Y_k | X_k) p(X_k | X_{1:k-1}, Y_{1:k-1}) p(X_{1:k-1} | Y_{1:k-1}) \\
&= p(Y_k | X_k) p(X_k | X_{k-1}) p(X_{1:k-1} | Y_{1:k-1})
\end{aligned}$$

同時，適當的選用

$$q(X_{1:k} | Y_{1:k}) = q(X_k | X_{k-1}, Y_k) q(X_{1:k-1} | Y_{1:k-1})$$

以此將權重改寫

$$w_k^i \propto \frac{p(Y_k | X_k^i) p(X_k^i | X_{k-1}^i) p(X_{1:k-1}^i | Y_{1:k-1})}{q(X_k^i | X_{k-1}^i, Y_k) q(X_{1:k-1}^i | Y_{1:k-1})} = w_{k-1}^i \times \frac{p(Y_k | X_k^i) p(X_k^i | X_{k-1}^i)}{q(X_k^i | X_{k-1}^i, Y_k)} \quad (4.5)$$

於是，整個粒子濾波器的演算便可以遞迴方式進行，並且由於使用重要性取樣法，此修改後之粒子濾波器方法亦稱為序列重要性取樣(Sequential Importance Sampling)法。其演算法流程如下：

Algorithm of Sequential Importance Sampling Method:

For Each step k:

1. Draw $X_k^i \sim q(X_k^i | X_{k-1}^i, Y_k)$
2. Calculate $\tilde{w}_k^i = w_{k-1}^i \times \frac{p(Y_k | X_k^i) p(X_k^i | X_{k-1}^i)}{q(X_k^i | X_{k-1}^i, Y_k)}$
3. Normalize $w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^{N_s} \tilde{w}_k^i}$
4. Estimate $\bar{X}_k = \sum_{i=1}^{N_s} w_k^i X_k^i$

並且，在選用重要性分布 $q(X_k | X_{k-1}, Y_k)$ 上，有一簡易的選擇是令其與狀態參數變化分布

相同，亦即 $q(X_k | X_{k-1}, Y_k) = p(X_k | X_{k-1})$ ，此方法又稱為 Bootstrap Filter[10]。若是如此選擇重要性分布時，權重之更新公式將可簡化為：

$$w_k^i \propto w_{k-1}^i \times \frac{p(Y_k | X_k^i) p(X_k^i | X_{k-1}^i)}{q(X_k^i | X_{k-1}^i, Y_k)} = w_{k-1}^i \times \frac{p(Y_k | X_k^i) p(X_k^i | X_{k-1}^i)}{p(X_k^i | X_{k-1}^i)} = w_{k-1}^i \times p(Y_k | X_k^i) \quad (4.6)$$

4.1.3 額外之取樣手段

在實際執行重要性序列取樣法的時候，由於取樣數與樣本數皆是有限的，在連續幾個遞迴步驟後，會產生一些現象降低演算法的效果，而要應付此些現象，有一些常見之額外取樣手段，以下將對這些現象與方法做說明。

4.1.3.1 重新取樣

隨著遞迴的演算法逐漸進行，每個時刻的權重會被下個時刻所繼承。在一個取樣樣本數有限以及計算精度有限的實際運作限制下，當演算進行長時間之後，權重越來越往極少數個樣本集中時，其餘大多數低權重之樣本對期望估測值幾乎沒有影響力，亦即所謂之「有效點數」降低，此現象亦稱為樣本退化(sample degeneracy)，關於有效點數，可參考[14]。

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_s} (w_k^i)^2} \quad (4.7)$$

當有效點數降低到一定門檻時，此情況便不符合此演算法中大數法則的假設。因此，在實際進行粒子濾波器方法時，都會需要用到重新取樣的方法。而重新取樣之方法有很多種[7]，在本研究中使用之重新取樣方法為 Multinomial Resampling，此重新取樣之流程

如下：

Algorithm of Multinomial Resampling

For each particle i

1. Draw $U^i \sim \mathbf{U}(0,1]$, where U is the uniform distribution.
2. $j = \text{Di}(U^i)$, where Di returns the index of the Empirical Function of w_k .
3. $\tilde{x}_{k-M:k}^i = x_{k-M:k}^j$

After all N_s particles are assigned

4. Reset all $w_k^i = \frac{1}{N_s}$

此重新取樣之方法是為將原先權重集中在少數的幾個樣本，依照其權重比例，重新打散為 N_s 個樣本，使有效樣本數再度回到 N_s 。

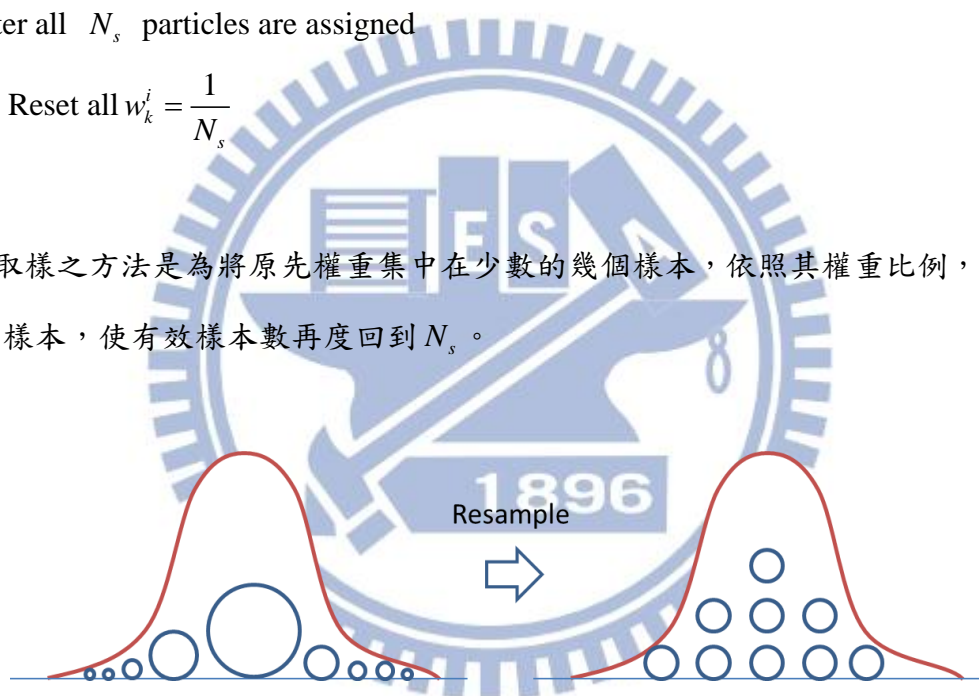


圖 4-1 重新取樣示意圖

如上圖，藍色圓圈代表一個樣本，圓圈的大小代表其權重的高低。左邊的圖代表權重集中在中間幾個樣本中，其他多數樣本權重極低。而重新取樣後則將原先高權重的單一樣本依照其權重所占比例複製幾份，並且將樣本之權重重設。

4.1.3.2 正規化

而由於重新取樣後之結果，是從將原先過度集中的少數幾個樣本重複進行複製的動作，因此就結果而言是將樣本的多樣性減少，而當樣本之多樣性太低值時，接下來的取樣結果仍然會過度集中，也就發生無法正確反映真實分布的情況，使得收斂結果不如預期，此現象稱為樣本貧化(Sample Impoverishment)現象。

針對此現象有一對應之處理方式，稱之為正規化(Regularization)，正規化的基本概念是：計算重新取樣前之樣本共變異數(sample covariance)，並在重新取樣後之樣本上重新加入此樣本共變異數的一定比例。並且參考[8]獲得建議的比例係數 h_{opt} ，而實際使用時再對其進行微調。

Algorithm of Regularization

1. Calculated $\mathbf{S}_k =$ sample covariance of x_{k-1}^i , and \mathbf{D}_k such that $\mathbf{D}_k \mathbf{D}_k^T = \mathbf{S}_k$.
2. After Resampling,

Draw $\varepsilon_k^i \sim$ Epanechnikov Kernel or Gaussian Kernel

$$\tilde{x}_{k-1}^i = \tilde{x}_{k-1}^i + h_{opt} \varepsilon_k^i \mathbf{D}_k$$

$$\text{For Epanechnikov Kernel: } h_{opt} = \left[\frac{8n_x(n_x+2)(n_x+4)(2\sqrt{\pi})^{n_x}}{N_s(2n_x+1)c_{n_x}} \right]^{1/(n_x+4)}$$

$$\text{For Gaussian Kernel: } h_{opt} = \left[\frac{4}{N_s(2n_x+1)} \right]^{1/(n_x+4)}$$

c_{n_x} is the volume of the unit n_x -dimensional sphere.

下圖是為正規化方法前後之樣本分布狀況之示意圖，其中重複出現之同樣之樣本做微幅

之擾動後，使原先對應數值較為單調的樣本分布變得更為多樣。

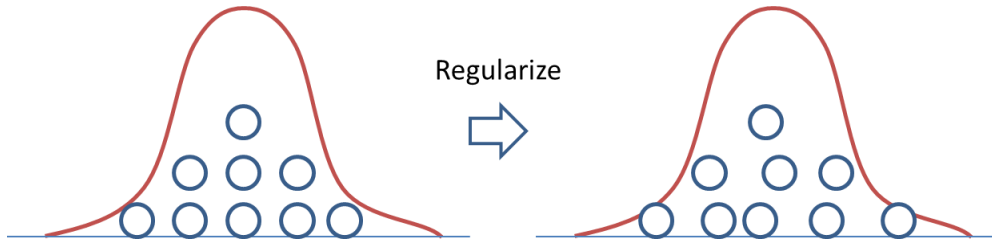


圖 4-2 正規化示意圖

如上圖所示，將原先重新取樣後重複的粒子做微幅的擾動，使其數值並非完全相同，使重新取樣造成之樣本多樣性減少現象能夠有所改善。

4.2 根據網路化控制系統模型設計估測器

首先，回顧在第三章所推導所得之網路化控制系統模型。其中，受控系統是為線性系統狀態，其系統狀態 x_k 是一個具有馬可夫鏈性質的隨機變數。

$$\begin{aligned} x_{k+1} | x_k &\sim \mathbf{N}(x_{k+1} | Ax_k + Bu_k, Q) \\ y_k | x_k &\sim \mathbf{N}(y_k | Cx_k + Du_k, R) \end{aligned} \quad (4.8)$$

而感測器輸出 y_k ，透過網路傳送至估測器端。

$$Z_k = \left\{ \begin{array}{l} y_i \left| \begin{array}{l} S_{k,i} = R, S_{k,i-1} = D, k-M \leq i < k \\ or \quad S_{k,i} = R, i = k \end{array} \right. \end{array} \right\} \quad (4.9)$$

每個時間 k 在估測器端都會根據個別資訊封包之狀態來決定確實被收到之觀測值，該時間內收到之資訊集合稱為 Z_k 。

$$p(Z_k | x_{k-M:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1}) = p(y_k | x_k, S_{k,k}) \prod_{i=1}^M p(y_{k-i} | x_{k-i}, S_{k,k-i}, S_{k-1,k-i}) \quad (4.10)$$

可以注意到，在網路化控制系統當中，系統之輸出的定義已經與原先在粒子濾波器

當中所推導之定義有些微之不同。因此在進行估測器的演算法時，其計算所使用之公式必須經過重新推導與檢查，而此時便是在第三章結尾所獲得之圖形式表示模型帶來益處的時候，配合貝式球(Bayes Ball Algorithm)的規則[9]，在判斷各個隨機變數間的條件相依性的相當有助益。

4.2.1 使用網路化控制系統模型公式推導

首先，從 $\bar{X}_{1:k} = \int X_{1:k} p(X_{1:k} | Y_{1:k}) dX_{1:k} \approx \frac{1}{N_s} \sum_{i=1}^{N_s} X_{1:k}^i$ 當中之 $p(X_{1:k} | Y_{1:k})$ 開始，在網路化控

制系統之模型中，由於在經過網路後所得觀測資訊是 Z ，因此將 $p(X_{1:k} | Y_{1:k})$ 寫作

$$p(x_{1:k} | Z_{1:k}) \quad (4.11)$$

並且，同樣利用得到(4.5)之前置處理方式，由於 x 是馬可夫鏈隨機變數

$$\begin{aligned} & p(x_{1:k} | Z_{1:k}) \\ &= \frac{p(Z_k | x_{1:k}, Z_{1:k-1}) p(x_{1:k} | Z_{1:k-1})}{p(Z_k | Z_{1:k-1})} \\ &\propto p(Z_k | x_{1:k}, Z_{1:k-1}) p(x_{1:k} | Z_{1:k-1}) \\ &= p(Z_k | x_{1:k}, Z_{1:k-1}) p(x_k | x_{1:k-1}, Z_{1:k-1}) p(x_{1:k-1} | Z_{1:k-1}) \\ &= p(Z_k | x_{1:k}, Z_{1:k-1}) p(x_k | x_{k-1}) p(x_{1:k-1} | Z_{1:k-1}) \end{aligned} \quad (4.12)$$

並且選用重要性分布 $q(x_{1:k} | Z_{1:k})$ 其符合特性

$$q(x_{1:k} | Z_{1:k}) = q(x_k | x_{k-1}, Z_k) q(x_{1:k-1} | Z_{1:k-1}) \quad (4.13)$$

於是

$$\begin{aligned}
w_k^i &= \frac{p(x_{1:k}^i | Z_{1:k})}{q(x_{1:k}^i | Z_{1:k})} \\
&\propto \frac{p(Z_k | x_{1:k}^i, Z_{1:k-1}) p(x_k^i | x_{k-1}^i) p(x_{1:k-1}^i | Z_{1:k-1})}{q(x_k^i | x_{k-1}^i, Z_k) q(x_{1:k-1}^i | Z_{1:k-1})} \\
&= w_{k-1}^i \times \frac{p(Z_k | x_{1:k}^i, Z_{1:k-1}) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, Z_k)}
\end{aligned} \tag{4.14}$$

注意到 $p(Z_k | x_{1:k}^i, Z_{1:k-1})$ 並不是如同原始序列重要性取樣濾波器當中出現之 $p(Y_k | X_k^i)$ ，其較為複雜，因為其中出現的 Z 這個複合型的隨機變數的關係，為了實際運算時使用，必須設法將其化為常見、已定義的機率分布。由於考慮到運作時之有效視窗大小為 M ，如第三章所提，此 M 代表估測器視多久以前之資料有計算價值，因此令

$$p(Z_k | x_{1:k}^i, Z_{1:k-1}) \cong p(Z_k | x_{k-M:k}^i, Z_{k-M:k-1}) \tag{4.15}$$

參考(4.8)之定義，使用邊際化(Marginize)方式補上此處所缺少之網路參數 S

$$\begin{aligned}
&p(Z_k | x_{k-M:k}^i, Z_{k-M:k-1}) \\
&= \sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[\begin{aligned} &p(Z_k | x_{k-M:k}^i, Z_{k-M:k-1}, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M:k-M}) \\ &\times p(S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M:k-M} | x_{k-M:k}^i, Z_{k-M:k-1}) \end{aligned} \right] \\
&= \sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[\begin{aligned} &p(Z_k | x_{k-M:k}^i, Z_{k-M:k-1}, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M:k-M}) \\ &\times p(S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M:k-M} | Z_{k-M:k-1}) \end{aligned} \right]
\end{aligned}$$

其中 $p(Z_k | x_{k-M:k}^i, Z_{k-M:k-1}, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M:k-M})$ ，根據圖形式模型，當給予

$x_{k-M:k}^i, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M:k-M}$ 時， Z_k 與 $Z_{k-M:k-1}$ 彼此不相關，因此

$$p(Z_k | x_{k-M:k}^i, Z_{k-M:k-1}) = \sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[\begin{aligned} &p(Z_k | x_{k-M:k}^i, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M:k-M}) \\ &\times p(S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M:k-M} | Z_{k-M:k-1}) \end{aligned} \right]$$

若將第三章定義之(3.6)式代入其中，則可得到

$$p(Z_k | x_{k-M:k}^i, Z_{k-M:k-1}) = \sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[\begin{aligned} &p(y_k | x_k, S_{k,k}) \prod_{i=1}^M p(y_{k-i} | x_{k-i}, S_{k,k-i}, S_{k-1,k-i}) \\ &\times p(S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M:k-M} | Z_{k-M:k-1}) \end{aligned} \right] \tag{4.16}$$

得到此式後，便可在確定 Z_k 之後進行計算。

4.2.2 網路參數之處理

在上面最後所得到之(4.14)式，其中出現之 $p(S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M:k-M} | Z_{k-M:k-1})$

項目，由於當 $Z_{k-M:k}$ 確定時，可以透過其中所包含之 y 來將對應之

$S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M:k-M}$ 分成是「R」還是「D或L」兩種情形。

並且， $p(y_k | x_k, S_{k,k}) \prod_{i=1}^M p(y_{k-i} | x_{k-i}, S_{k,k-i}, S_{k-1,k-i})$ 在定義中，若是確實有收到某一時間之資料 y_i ，然而對應之 $S_{k,i}, S_{k-1,k-i}$ 卻為 D 或 L 時，則出現 $\delta^*(y_i)$ ，使得整個項目化為 0。

因此只有當全部之網路參數 S 與 Z 當中顯示之 y 的情況相符合時，稱這個網路參數組合為 S' ，以及此時有收到的 y 對應的 x 為 x' ，舉例而言，當 $Z_k = \{y_k, y_{k-1}\}$ 時，使

$p(y_k | x_k, S_{k,k}) \prod_{i=1}^M p(y_{k-i} | x_{k-i}, S_{k,k-i}, S_{k-1,k-i})$ 不等於 0 的 S 組合其中必含有

$S_{k,k} = R, S_{k,k-1} = R, S_{k-1,k-1} = D$ ，並以此可得到

$$\begin{aligned} & p(y_k | x_k, S_{k,k} = R) p(y_{k-1} | x_{k-1}, S_{k,k-1} = R, S_{k-1,k-1} = D) \\ & = \mathbf{N}(y_k | Cx_k, R) \mathbf{N}(y_{k-1} | Cx_{k-1}, R) \end{aligned}$$

將 y 值帶入則可將其轉為 x' 的高斯分布相乘，亦即

$$\begin{aligned} & \mathbf{N}(y_k | Cx_k, R) \mathbf{N}(y_{k-1} | Cx_{k-1}, R) \\ & = \mathbf{N}(x_k | C^{-1}y_k, C^T R^{-1}C) \mathbf{N}(x_{k-1} | C^{-1}y_{k-1}, C^T R^{-1}C) \end{aligned}$$

將此結果表示為 $p(x' | u', \Sigma')$ 。

於是，將這個非 0 的 $p(x' | u', \Sigma')$ 高斯 PDF 提出，也就相當於：

$$\begin{aligned}
& p(Z_k | x_{k-M:k}, Z_{k-M:k-1}) \\
&= \sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[p(y_k | x_k, S_{k,k}) \prod_{i=1}^M p(y_{k-i} | x_{k-i}, S_{k,k-i}, S_{k-1,k-i}) \right] \\
&\quad \times p(S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M:k-M} | Z_{k-M:k-1}) \\
&= 0 + \dots + p(x' | u', \Sigma') p(S' | Z_{k-M:k-1}) + \dots + p(x' | u', \Sigma') p(S' | Z_{k-M:k-1}) + \dots + 0 + \dots \\
&= p(x' | u', \Sigma') \sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} [p(S' | Z_{k-M:k-1})] \tag{4.17}
\end{aligned}$$

可以發現，後半部網路參數的部分與樣本 x' 無關。

因此在權重的標準化時

$$\begin{aligned}
w_k^i &= \frac{\tilde{w}_k^i}{\sum_{i=1}^{N_s} \tilde{w}_k^i} \\
w_{k-1}^i &\times \frac{p(x^i | u', \Sigma') \sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} [p(S' | Z_{k-M:k-1})]}{q(x_k^i | x_{k-1}^i, Z_k)} \\
&= \frac{\sum_{i=1}^{N_s} \left\{ w_{k-1}^i \times \frac{p(x^i | u', \Sigma') \sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} [p(S' | Z_{k-M:k-1})]}{q(x_k^i | x_{k-1}^i, Z_k)} \right\}}{\sum_{i=1}^{N_s} \left[w_{k-1}^i \times \frac{p(x^i | u', \Sigma')}{q(x_k^i | x_{k-1}^i, Z_k)} \right]} \times \frac{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} [p(S' | Z_{k-M:k-1})]}{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} [p(S' | Z_{k-M:k-1})]} \\
&= \frac{w_{k-1}^i \times \frac{p(x^i | u', \Sigma')}{q(x_k^i | x_{k-1}^i, Z_k)}}{\sum_{i=1}^{N_s} \left[w_{k-1}^i \times \frac{p(x^i | u', \Sigma')}{q(x_k^i | x_{k-1}^i, Z_k)} \right]} \times \frac{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} [p(S' | Z_{k-M:k-1})]}{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} [p(S' | Z_{k-M:k-1})]} \\
&= \frac{w_{k-1}^i \times \frac{p(x^i | u', \Sigma')}{q(x_k^i | x_{k-1}^i, Z_k)}}{\sum_{i=1}^{N_s} \left[w_{k-1}^i \times \frac{p(x^i | u', \Sigma')}{q(x_k^i | x_{k-1}^i, Z_k)} \right]} \tag{4.18}
\end{aligned}$$

可以發現，網路參數 S 的狀態變化機率實際上並不會包含在計算當中，換句話說，估測器在計算權重時，注重的是收取到之資料內容與該資料之延遲時間，在某一資料尚未被收取到時，估測器無法對該資料做任何利用。

4.3 改進估測器方法

雖然已經將網路化控制系統之模型套用到粒子濾波器上，但是此方式仍然存有許多改進空間。如重要性分布(Importance Distribution)的選取、演算公式的修改等，以下逐步對各個手段說明。

4.3.1 重要性分布的選取

在粒子濾波器中，重要性分布的功能是用來抽取樣本估測值，而當抽取到的樣本範圍越接近真實分布時，自然會增加演算法所得之估測結果的收斂速度。

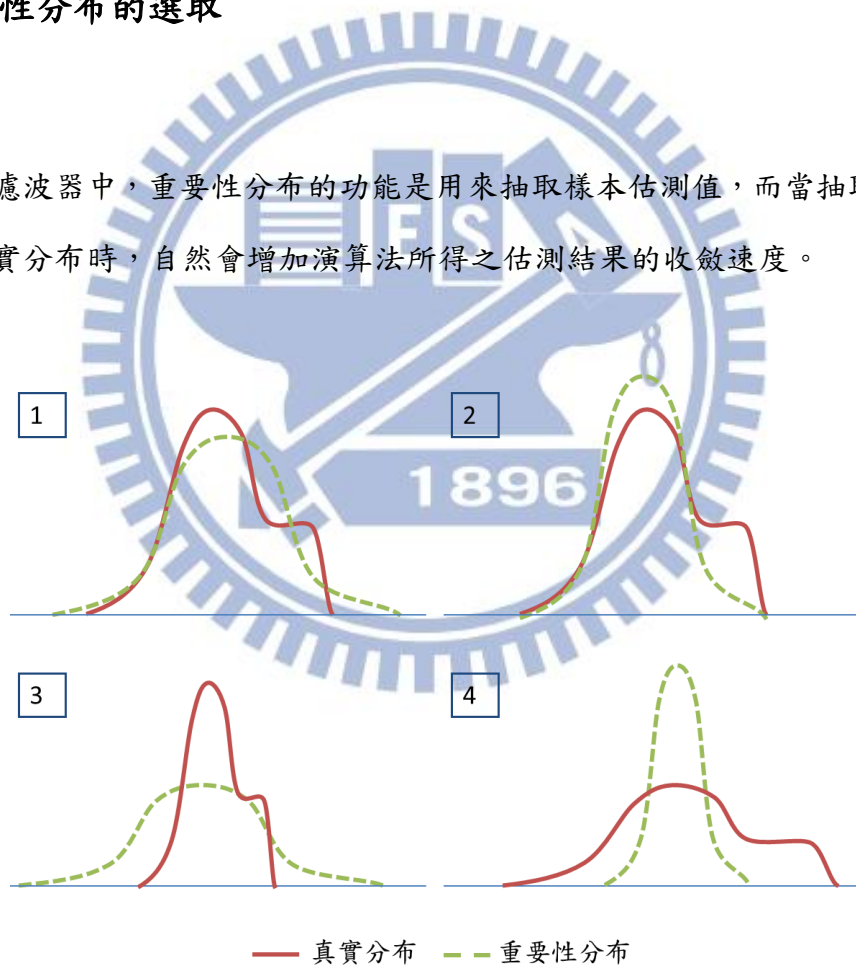


圖 4-3 重要性分布選取示意圖

上圖是一個簡易的示意圖，使用實線來代表真實分布，虛線是選取的重要性分布，其中四張圖各代表的情形說明如下：

1：雖然真實分布複雜，但是選取的重要性分布仍適當涵蓋了大部分的真實分布，並且

也沒有過度地包含到多餘的範圍，算是適當的分布。

2：選取的重要性分布無法確實表達到全部的真實分布，當估測目標在特定範圍內時可能使得收斂速度較慢。

3：選取的重要性分布涵蓋太多多餘的範圍，大部分的取樣落在真實分布之外，使得收斂較慢，假若取樣都落在真實範圍之外時更會使得情況惡化。

4：選取的重要性分布範圍太小，使得取樣過於集中，無法確實反映真實的分布，在系統不準確性高時收斂效果緩慢。

當設定 $q(X_k | X_{k-1}, Y_k) = p(X_k | X_{k-1})$ 時，由於取樣時僅僅只依靠 X_{k-1} ，假若模型之不準確性較大，或者複雜度較高時，可能取樣出來的 X_k 的範圍並不能有效涵蓋住真實的 X_k 分布範圍，使得演算法沒有辦法迅速收斂到較佳的結果。

配合網路化控制系統模型所得之濾波器當中，在套用下一章加熱系統模型進行數據模擬時，由於系統階數較高，使用 $q(x_k | x_{k-1}, Z_k) = p(x_k | x_{k-1})$ 時無法確實取樣到足夠多且接近真實系統狀態的樣本，因此遭遇收斂結果緩慢之情形。

於是，設想一重要性取樣分布

$$q(x_k | x_{k-1}, Z_k) = p(x_k | x_{k-1}, Z_k) \quad (4.17)$$

此分布由於參考到當前收取到的資料來取樣，因此理論上新的取樣值會朝向最新的觀測結果靠攏，使得取樣值更能落在真實分布內。

$$\begin{aligned} & q(x_k | x_{k-1}, Z_k) \\ &= p(x_k | x_{k-1}, Z_k) \\ &= \frac{p(Z_k | x_{k-1}, x_k) p(x_k | x_{k-1})}{p(Z_k | x_{k-1})} \end{aligned}$$

設法調整公式使其可以套用(3.6)式之定義，使用邊際化方法補上網路參數 S

$$= \frac{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[p(Z_k | x_{k-1:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1}) p(S_{k,k-M:k}, S_{k-1,k-M:k-1} | x_{k-1:k}) \right]}{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[p(Z_k | x_k, S_{k,k-M:k}, S_{k-1,k-M:k-1}) p(S_{k,k-M:k}, S_{k-1,k-M:k-1} | x_k) \right]} p(x_k | x_{k-1})$$

同樣方法，補上缺少的 x 項目

$$= \frac{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[\int \int \dots \int p(Z_k | x_{k-M:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1}) p(x_{k-M:k-2} | x_{k-1:k}) dx_{k-M} \dots dx_{k-3} dx_{k-2} \right]}{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[\int \int \dots \int p(Z_k | x_{k-M:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1}) p(x_{k-M:k-2}, x_k | x_{k-1}) dx_{k-M} \dots dx_{k-2} dx_k \right]} p(x_k | x_{k-1})$$

接著利用圖形式模型判斷其中各項中之變數之間的相依性做化簡，得到

$$= \frac{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[\int \int \dots \int p(Z_k | x_{k-M:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1}) p(x_{k-M:k-2} | x_{k-1}) dx_{k-M} \dots dx_{k-3} dx_{k-2} \right]}{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[\int \int \dots \int p(Z_k | x_{k-M:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1}) p(x_{k-M:k-2} | x_{k-1}) p(x_k | x_{k-1}) dx_{k-M} \dots dx_{k-2} dx_k \right]} p(x_k | x_{k-1})$$

將定義代入後並整理

$$= \frac{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left\{ \frac{p(y_k | x_k, S_{k,k}) \int \dots \int \prod_{i=1}^M p(y_{k-i} | x_{k-i}, S_{k,k-i}, S_{k-1,k-i}) p(x_{k-M:k-2} | x_{k-1}) dx_{k-M} \dots dx_{k-3} dx_{k-2}}{p(S_{k,k-M:k}, S_{k-1,k-M:k-1})} \right\}}{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left\{ \frac{\int p(y_k | x_k, S_{k,k}) \int \dots \int \prod_{i=1}^M p(y_{k-i} | x_{k-i}, S_{k,k-i}, S_{k-1,k-i}) p(x_{k-M:k-2} | x_{k-1}) dx_{k-M} \dots dx_{k-2} p(x_k | x_{k-1}) dx_k}{p(S_{k,k-M:k}, S_{k-1,k-M:k-1})} \right\}} p(x_k | x_{k-1})$$

其中由於 x_{k-1} 為已知項，故上下約去

$$= \frac{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[p(y_k | x_k, S_{k,k}) p(S_{k,k-M:k}, S_{k-1,k-M:k-1}) \right]}{\sum_{S_{k,k}} \sum_{S_{k,k-1}} \dots \sum_{S_{k-M:k-M}} \left[\int p(y_k | x_k, S_{k,k}) p(x_k | x_{k-1}) dx_k p(S_{k,k-M:k}, S_{k-1,k-M:k-1}) \right]} p(x_k | x_{k-1})$$

最後整理得到

$$= \frac{\sum_{S_{k,k}} \left[p(y_k | x_k, S_{k,k}) p(S_{k,k}) \right]}{\sum_{S_{k,k}} \left[\int p(y_k | x_k, S_{k,k}) p(x_k | x_{k-1}) dx_k p(S_{k,k}) \right]} p(x_k | x_{k-1}) \quad (4.18)$$

觀察此結果可以了解，當當前收取之資料含有即時到達之 y_k 時，可以使得 x_k 的取樣範圍往此筆觀測結果靠攏，在一定程度上增加收斂速度，改善估測結果。

4.3.2 演算公式的修改

正如前章節所述，當取樣出來之樣本的大多數不在系統的真实分布的範圍內時，其計算出來的權重將會趨近於 0，而使得該樣本對估測期望值的影響力減少，而除了長時間運作會導致權重分布不均外，系統過於複雜也是產生此現象的原因。

在網路化控制系統當中，在一時間內抵達的觀測資料集合不一定是僅來自一個時間，而有可能含過去數筆的觀測資料，相當於增加系統的階數。但是取樣時僅僅取樣最新時間的樣本，也就是過往時刻之樣本其實是沒有參考到最新之資訊而直接以系統模型做取樣，使得其分部範圍遠大於觀測資訊分布範圍，若無法將取樣集中在真實取樣範圍內，相當於減少取樣之有效點數。

另一方面，當估測法所參考的有效視窗範圍逐漸增加時，不只是對應之系統複雜度增加，同時使演算法占用掉更多的記憶空間，也需要更多的計算能力。理論上，取樣的樣本數目必須要倍數增加，才能達到同樣之估算效果，而這點卻是在有限硬體能力上無法達到，在模擬實驗中估算效果大為降低。

例如，在本系統中，單一時刻之觀測值為 n_y 維度，當某一時間點收到過往之多筆觀測資料時，該筆觀測資料之集合之維度便會上升，如(4.20)，最多可以成為 $n_y(M+1)$ 維。

$$P(y_k | x_k, S_{k,k}) \prod_{i=1}^M P(y_{k-i} | x_{k-i}, S_{k,k-i}, S_{k-1,k-i}) \rightarrow \mathfrak{R}^{n_y} \cdot \mathfrak{R}^{n_y} \cdot \mathfrak{R}^{n_y} \dots \cdot \mathfrak{R}^{n_y} \rightarrow \mathfrak{R}^{n_y \cdot (M+1)} \quad (4.19)$$

以下提出一權重更新公式的近似式，該近似式有著與原公式相同的變化趨勢，但是分布範圍較大，相對來說對樣本的多樣性要求較為輕微。使得在有限的硬體計算能力下，使用較少的樣本數就能達到原先預期的效果。並且由於樣本僅使用 x_k ，不須保留舊的樣本，因此亦可節省記憶空間。

$$\begin{aligned}
 & p(Z_k | x_{k-M:k}, Z_{k-M:k-1}, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M,k-M}) \\
 & \cong p(Z_k | x_k, Z_{k-M:k-1}, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M,k-M})
 \end{aligned}
 \tag{4.20}$$

以下透過觀察此二個機率密度函數的差別，藉以發現當分布過於集中使得有效取樣過低時發生的情況。利用模擬驗證時提出之系統模型計算的對應之機率密度函數，在不同的網路情況下。

接著有四張圖，由左至右、由上至下分別為網路無延遲、網路延遲 1 時間單位、網路延遲 2~3 時間單位、以及網路延遲 3~5 時間單位情況下，原始的機率密度函數與近似的機率密度函數比較圖。由於 Z 本身是多筆 y 的集合，是屬於高維度空間，因此採用將 y 從平均值給予兩個標準差範圍的偏移，代入此二機率密度函數後繪出其結果，便以一維的方式對其觀察。

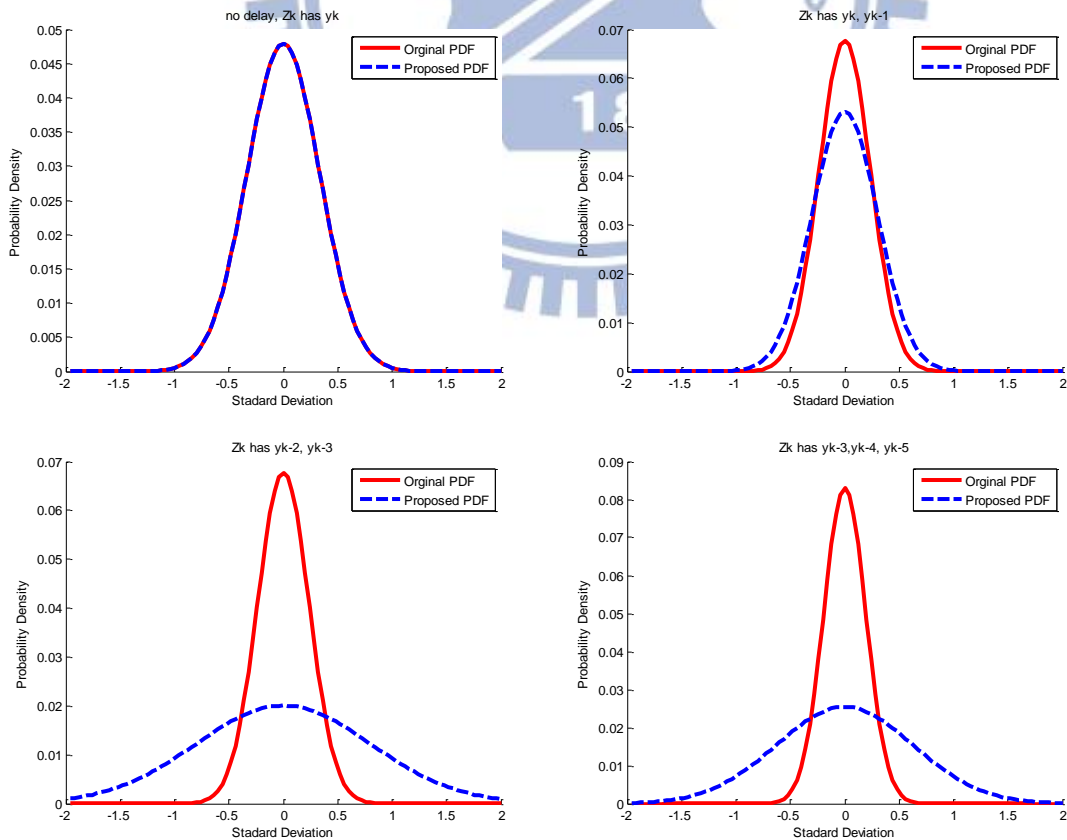


圖 4-4 不同網路情況下之機率密度函數比較圖

可以看到在網路資料中若含有即時之資訊時，此二機率密度函數皆是相當集中之分布，亦即在此情況下，取樣需要特別經過處理，使其往估測平均值的集中區域靠攏，也就同時驗證了前面章節所述之重要性取樣分布的特別選取的必要性。

而當隨著接收到的資料越舊，原始之機率密度函數分布仍然相當集中，在樣本數有限的情況下，難以將足夠多之樣本集中，在集中範圍之外的樣本權重趨近於零，使得估測無法得到正確的收斂效果。

而近似後之機率密度函數較為平均散佈開來，有著較大的樣本變異性，因此當取樣到的樣本不是相當集中在理想估測值附近時，仍然有一定之權重，使其可以將估測平均值往中央逐漸推進，達到逐漸收斂之效果。

為了估算此近似後對有效點數產生的改進效果，使用第五章所提之模擬驗證用的系統矩陣，系統雜訊之協方差為 1，以此產生一組 $x_{k-M:k}^{1:1000}$ 樣本。並且假設網路情況穩定，使在當前收取資料之前之過往資料皆已被接收，調整當前接收到的資料集合後，並假設前一時刻之權重皆相同，則可以得到此一時間不同更新公式對權重更新後之有效點數 (參考公式 4.7) 百分比。

表 4-1 某一時刻以原分布與近似分布計算之有效點數占總點數百分比

接收之資料	原分布有效點數百分比	近似有效點數百分比
即時	26.48%	26.48%
即時、延遲 1 時間	3.47%	6.85%
延遲 2 時間、延遲 3 時間	4.94%	64.54%
延遲 3 時間、延遲 4 時間、延遲 5 時間	0.79%	42.89%

可以看到原分布之有效點數在接收到過往多筆資料時會顯著降低，而近似之結果則在短延遲時間時表現較差；長延遲時間時，近似的分布則能改善有效點數。

於是對此近似之機率密度分布函數進行處理

$$p(Z_k | x_k, Z_{k-M:k-1}, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M,k-M})$$

近似後缺少的 x 項目以積分的方式邊際化處理

$$\begin{aligned} &= \iint \dots \int \left[\frac{p(Z_k | x_{k-M:k}, Z_{k-M:k-1}, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M,k-M})}{\times p(x_{k-M:k-1} | x_k, Z_{k-M:k-1}, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M,k-M})} \right] dx_{k-M} \dots dx_{k-2} dx_{k-1} \\ &= \iint \dots \int \left[\frac{p(Z_k | x_{k-M:k}, Z_{k-M:k-1}, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M,k-M})}{\times \frac{p(Z_{k-M:k-1} | x_{k-M:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M,k-M}) p(x_{k-M:k-1} | x_k, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M,k-M})}{p(Z_{k-M:k-1} | x_k, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M,k-M})}} \right] dx_{k-M} \dots dx_{k-2} dx_{k-1} \\ &= \frac{\iint \dots \int \left[\frac{p(Z_k | x_{k-M:k}, Z_{k-M:k-1}, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M,k-M})}{\times p(Z_{k-M:k-1} | x_{k-M:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M,k-M})} \right] dx_{k-M} \dots dx_{k-2} dx_{k-1}}{p(Z_{k-M:k-1} | x_k, S_{k,k-M:k}, S_{k-1,k-M:k-1}, \dots, S_{k-M,k-M})} \end{aligned}$$

將分子項目依據隨機變數之間條件相依性整理，分母亦使用同樣方式整理

$$\begin{aligned} &\iint \dots \int \left[\frac{p(Z_k | x_{k-M:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1})}{\times \prod_{i=1}^M p(Z_{k-i} | x_{k-M:k-i}, S_{k-i,k-M:k-i}, S_{k-i-1,k-M:k-i-1})} \right] dx_{k-M} \dots dx_{k-2} dx_{k-1} \\ &= \frac{\iint \dots \int \left[\frac{p(Z_k | x_{k-M:k}, S_{k,k-M:k}, S_{k-1,k-M:k-1})}{\times p(x_{k-M:k-1} | x_k)} \right] dx_{k-M} \dots dx_{k-2} dx_{k-1}}{\iint \dots \int \left[\frac{\prod_{i=1}^M p(Z_{k-i} | x_{k-M:k-i}, S_{k-i,k-M:k-i}, S_{k-i-1,k-M:k-i-1})}{\times p(x_{k-M:k-1} | x_k)} \right] dx_{k-M} \dots dx_{k-2} dx_{k-1}} \end{aligned} \quad (4.21)$$

可以看到，最後獲得之式子之分子分母皆有核心為

$$p(Z_{k-i} | x_{k-M:k-i}, S_{k-i,k-M:k-i}, S_{k-i-1,k-M:k-i-1}),$$

亦即此項目是一個依據給予之變數 S 而產生高斯分布亦或者產生定義之 δ 函數而產生 0 值的複合分布函數。

4.4 最終版本之粒子濾波器

在使用到前面章節所述之粒子濾波器改進方法後，稱使用了正規化動作之粒子濾波器為正規化粒子濾波器(Regularized Particle Filter)並且其運作之演算方式如下：

Algorithm of Regularized Particle Filter:

For each step k

1. Calculate effective particle count $N_{eff} = \frac{1}{\sum_{i=1}^{N_s} (w_k^i)^2}$

2. If $N_{eff} < N_{threshold}$

Do Resample and Regularization

3. Draw $x_k^i \sim q(x_k^i | x_{k-1}^i, Z_k)$

4. Calculate $\tilde{w}_k^i = w_{k-1}^i \times \frac{p(Z_k | x_k^i, Z_{1:k-1}) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, Z_k)}$

5. Normalize $w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^{N_s} \tilde{w}_k^i}$

6. Estimate $\bar{x}_k = \sum_{i=1}^{N_s} w_k^i x_k^i$

第五章 數據模擬

在本章節中，將依據本研究中所提之網路化控制系統之假設選擇適當之受控系統。並透過數據模擬的方式，驗證所提出之估測方法是否具有預期中之效果。

5.1 受控系統之選擇

依據第一章與第三章中之假設，在受控系統方面，該系統需要具備有觀測端點多、應用情境可以為遠距離、容易操作呈現的特點。在第三章中，基於控制器端採直接連接，而觀測端則透過網路送出大量觀測資訊，提到之可能應用情境是為溫度監控系統。

設想在一溫度監控系統中，控制器採用直接連結至中央控制端點，溫度感應器則大量散佈在部分系統區域中，並且將採集到之溫度資訊透過無線網路傳輸。而若無線網路之品質較不穩定，則使得資料延遲時間不定、並且有可能遺失。於是估測器端便要藉由這些延遲或者偶爾遺失的資訊，來估測沒有安裝溫度感應器區域的溫度。

5.2 整體系統架構

為了在實驗中獲得較為容易控制且快速的溫度響應，因此實驗時選擇使用一銅板來代表欲進行監控的空間/物體，並且使用陶瓷發熱管來做為熱源。對此銅板加熱的同時，使用溫度感應器來獲取銅板部分之溫度分布，並且將此獲得之資訊透過網路傳輸至估測器端進行估測。於是繪出系統架構如下：

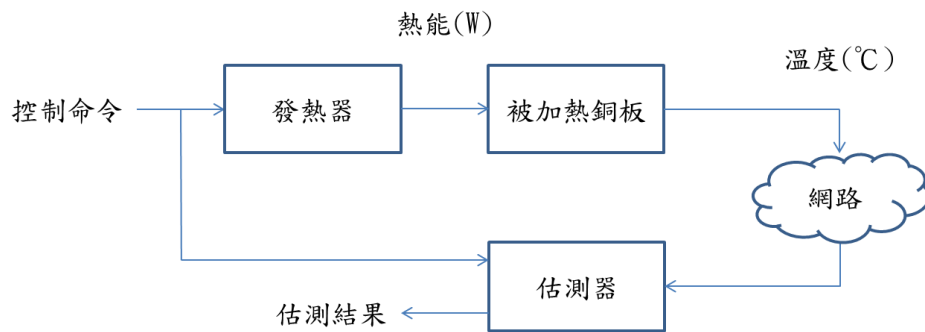


圖 5-1 模擬用系統架構

5.3 受控系統模型

5.3.1 暫態熱傳導

為了將整個加熱系統化為一線性狀態空間表示，故從其內部之暫態熱傳導開始。加熱目標是為一銅板，假設此銅板夠薄，將其平放，其垂直方向溫度為等溫，使其成為內部暫態溫度變化成為二維問題。於是，使用二維之暫態熱傳導公式：

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \quad (5.1)$$

T : 該處之溫度

α : 熱擴散率(Thermal Diffusivity)，是熱導率與單位體積熱容量之比。

此公式代表物體中某一位置之溫度變化，是該位置之溫度對其周圍方向的二次微分乘以熱擴散率係數。

接著令加熱時採用點熱源加熱，則可表示為

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + Q(t) \delta(x - x_h, y - y_h) \quad (5.2)$$

$Q(t)$:發熱體每秒提供之熱量

x_h, y_h :發熱體之座標位置

並且由於空氣是熱的不良導體，銅板熱量以極緩慢之速度流失至空氣中，因此將其忽略，於是銅板除了邊界不導熱，亦不向垂直方向的空氣流失熱量。故於此二維暫態熱傳導式中，令其邊界條件為

$$\left. \frac{\partial T}{\partial x} \right|_{x=\text{boundary}} = 0, \left. \frac{\partial T}{\partial y} \right|_{y=\text{boundary}} = 0 \quad (5.3)$$

於是獲得此銅板加熱時之暫態熱傳導公式，但實際使用時須要將其化為空間上離散之系統。

5.3.2 離散化

將銅板依 x 與 y 方向分割為有限數量之格點，並且使用有限差分法(Finite Difference Method)，將暫態熱傳導中之二次微分與邊界條件替換，

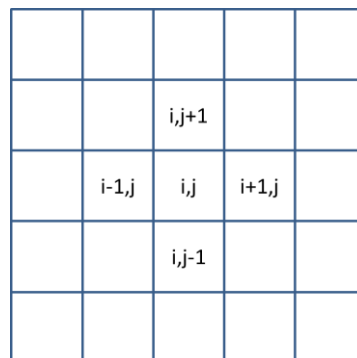


圖 5-2 有限差分法示意圖

依照有限差分法對二次微分與一次微分之定義：

將原先連續之平面改為格點，在 i, j 位置座標之端點其周圍之平均溫度為 $T_{i,j}$ ，於是改寫

暫態熱傳導中端點與邊界條件項目：

各端點(二次微分)：

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}}{\Delta x^2}, \quad \frac{\partial^2 T}{\partial y^2} = \frac{T_{i,j-1} - 2T_{i,j} + T_{i,j+1}}{\Delta y^2} \quad (5.4)$$

邊界條件(一次微分)：

$$\frac{\partial T}{\partial x} = \frac{T_{i+1,j} - T_{i-1,j}}{2\Delta x}, \quad \frac{\partial T}{\partial y} = \frac{T_{i,j+1} - T_{i,j-1}}{2\Delta y} \quad (5.5)$$

$\Delta x, \Delta y$: 切割後格點之間距離

於是便得到離散化之暫態熱傳導公式：

$$\frac{\partial T_{i,j}}{\partial t} = \alpha \left(\frac{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}}{\Delta x^2} + \frac{T_{i,j-1} - 2T_{i,j} + T_{i,j+1}}{\Delta y^2} \right) + \frac{Q(t)}{cp\rho\Delta x\Delta y h} \delta(i-i_h, j-j_h) \quad (5.6)$$

$$\left. \frac{T_{i+1,j} - T_{i-1,j}}{2\Delta x} \right|_{i=boundary}, \quad \left. \frac{T_{i,j+1} - T_{i,j-1}}{2\Delta y} \right|_{j=boundary} = 0$$

cp : 熱容量

ρ : 銅板密度

h : 銅板厚度

於是將各係數代入後，便可得到

$$\begin{aligned}
\begin{bmatrix} \dot{T}_{1,1} \\ \dot{T}_{1,2} \\ \dot{T}_{1,3} \\ \dots \end{bmatrix} &= \begin{bmatrix} \alpha\left(\frac{-2}{\Delta x^2} + \frac{-2}{\Delta y^2}\right) - \bar{H}_h \delta(1-i_h, 1-j_h) & \frac{2\alpha}{\Delta y^2} & 0 & \dots \\ \frac{\alpha}{\Delta y^2} & \alpha\left(\frac{-2}{\Delta x^2} + \frac{-2}{\Delta y^2}\right) - \bar{H}_h \delta(1-i_h, 2-j_h) & \frac{\alpha}{\Delta y^2} & \dots \\ 0 & \frac{\alpha}{\Delta y^2} & \alpha\left(\frac{-2}{\Delta x^2} + \frac{-2}{\Delta y^2}\right) - \bar{H}_h \delta(1-i_h, 3-j_h) & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} T_{1,1} \\ T_{1,2} \\ T_{1,3} \\ \dots \end{bmatrix} \\
+ \begin{bmatrix} \delta(1-i_h, 1-j_h) \\ \delta(1-i_h, 2-j_h) \\ \delta(1-i_h, 3-j_h) \\ \dots \end{bmatrix} & \frac{Q(t)}{cp\rho\Delta x\Delta y h}
\end{aligned} \tag{5.7}$$

由上式即可見連續時間線性系統之狀態空間表示式之 A、B 矩陣。

並且，透過有安裝溫度感應器之格點座標可得到 C 矩陣。

$$\begin{bmatrix} T_{out1} \\ T_{out2} \\ T_{out3} \\ \dots \end{bmatrix} = \begin{bmatrix} \dots & 1 & \dots & \dots & \dots \\ \dots & \dots & 1 & \dots & \dots \\ \dots & \dots & \dots & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} T_{1,1} \\ T_{1,2} \\ T_{1,3} \\ \dots \end{bmatrix} \tag{5.8}$$

接著將此連續時間之狀態空間表示式轉換為離散時間，即可在估測器中使用。

5.4 模擬的設定

5.4.1 加熱系統設定

在模擬當中，使用一 18×18 平方公分大小之銅板，給予三個熱源加熱。並且將該銅板劃分為 25 個格點，觀察其中九個格點之溫度，以沒有觀察到之三個端點為估測目標。並且模擬進行 60 秒，取樣週期為 0.5 秒，共 120 個時間單位。

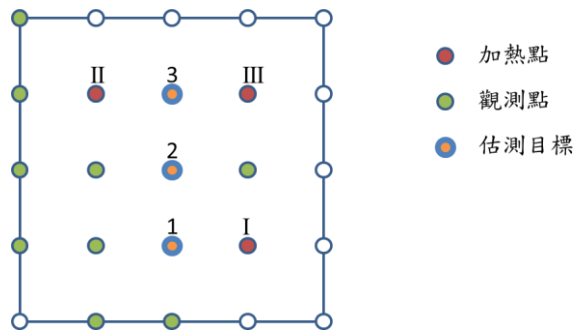


圖 5-3 模擬用加熱系統平面圖

並且其中 I 與 III 號熱源輸入每 7.5 秒為一週期之方波，其大小為 80 瓦；II 號熱源則為週期 15 秒，大小為 80 瓦。

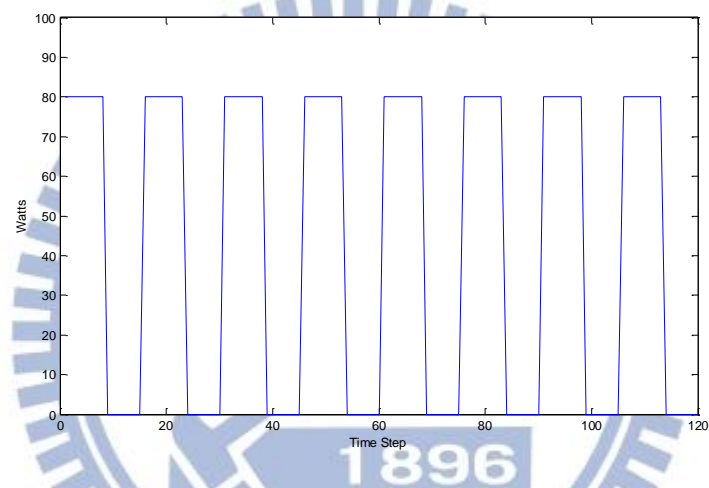


圖 5-4 I 與 III 號熱源之輸入

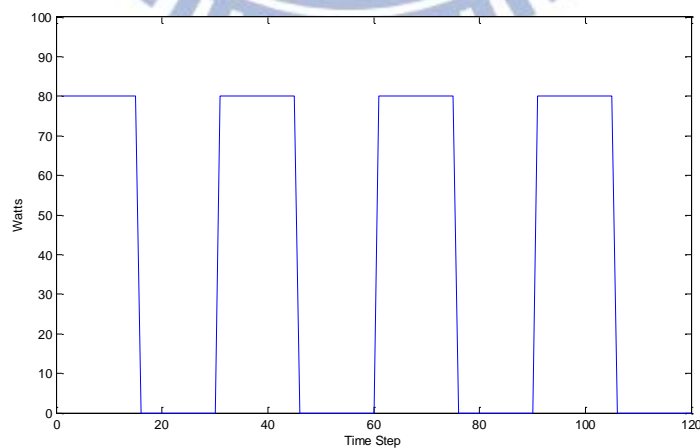


圖 5-5 II 號熱源之輸入

5.4.2 網路設定

調整網路參數之轉換機率，可以組合出想要之網路狀況。在數據模擬中，使用四種網路情況進行，稱其為 π_1 、 π_2 、 π_3 、 π_4 ，並且將其表示為下列之形式：

$$\pi = \begin{bmatrix} \pi_R(0) & \pi_R(1) & \pi_R(2) & \pi_R(3) & \pi_R(4) & \pi_R(5) & \pi_R(6) & \pi_R(7) & \dots \\ \pi_D(0) & \pi_D(1) & \pi_D(2) & \pi_D(3) & \pi_D(4) & \pi_D(5) & \pi_D(6) & \pi_D(7) & \dots \\ \pi_L(0) & \pi_L(1) & \pi_L(2) & \pi_L(3) & \pi_L(4) & \pi_L(5) & \pi_L(6) & \pi_L(7) & \dots \end{bmatrix} \quad (5.9)$$

上式是將網路參數 $S_{k,m}$ 由 D 轉移至 R、D、L 之機率 $\pi_R(k-m)$ 、 $\pi_D(k-m)$ 、 $\pi_L(k-m)$ 表示在單一矩陣中之表示式。並且由於 R 轉變為 R 之機率為 1，L 轉變為 L 之機率為 1，因此不用特別寫在此式中。

於是，設定 π_1 、 π_2 、 π_3 、 π_4 如下

$$\pi_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \pi_2 = \begin{bmatrix} 0 & 0.2 & 0.8 & 1 \\ 1 & 0.8 & 0.2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \pi_3 = \begin{bmatrix} 0 & 0 & 0 & 0.2 & 0.8 & 1 \\ 1 & 1 & 1 & 0.8 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\pi_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0.2 & 0.8 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0.8 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.10)$$

可以看出， π_1 是一個無延遲的網路，資料可以在出發後同一取樣時間被收到；而 π_2 則是一個大多數資料都會延遲兩個取樣時間才被收到的網路； π_3 則是延遲四個取樣時間； π_4 則是六個。

在這四個設定中皆沒有設定資料遺失，其原因是因為當資料被遺失時，可以視為其擁有相當長之延遲時間、該延遲時間超過估測之有效視窗範圍。因此若要對估測進行測試，僅需要逐步增加延遲時間，當某筆資料之延遲時間過長超過估測有效視窗時，在模擬上便與遺失無異。基於此理由，為了確實控制變因，因此僅針對延遲時間做設計。

5.4.3 模擬之傳輸延遲時間

於是依照這四個網路參數，模擬出網路傳輸情況，得到以下四個不同的延遲時間情形，其中當延遲時間單位數大於等於 6 時，代表資訊是在有效視窗範圍之外才會到達、或者是被遺失了。並且由於 π_1 是完全無延遲之網路情況，因此其延遲時間皆為 0，而其餘網路情況延遲時間如下所見。

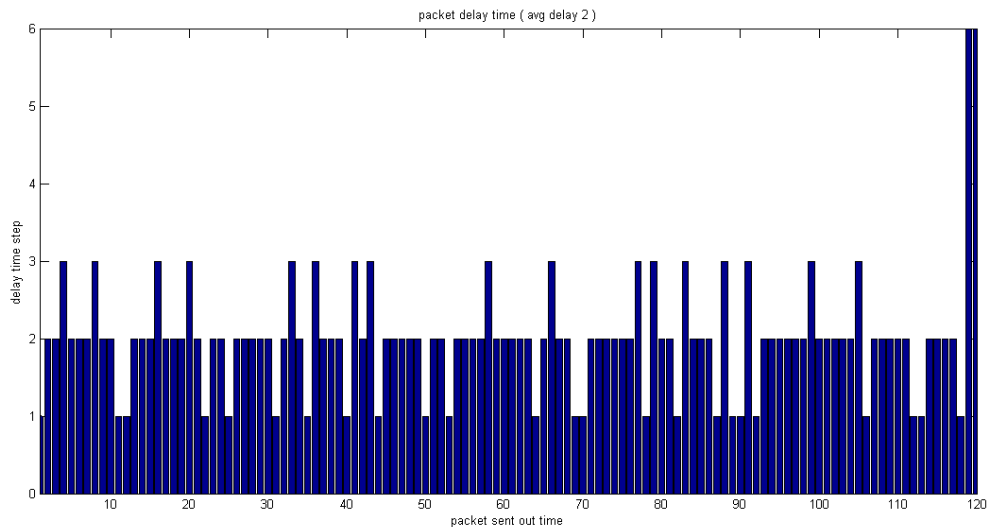


圖 5-6 網路情況 π_2 產生之資訊延遲時間

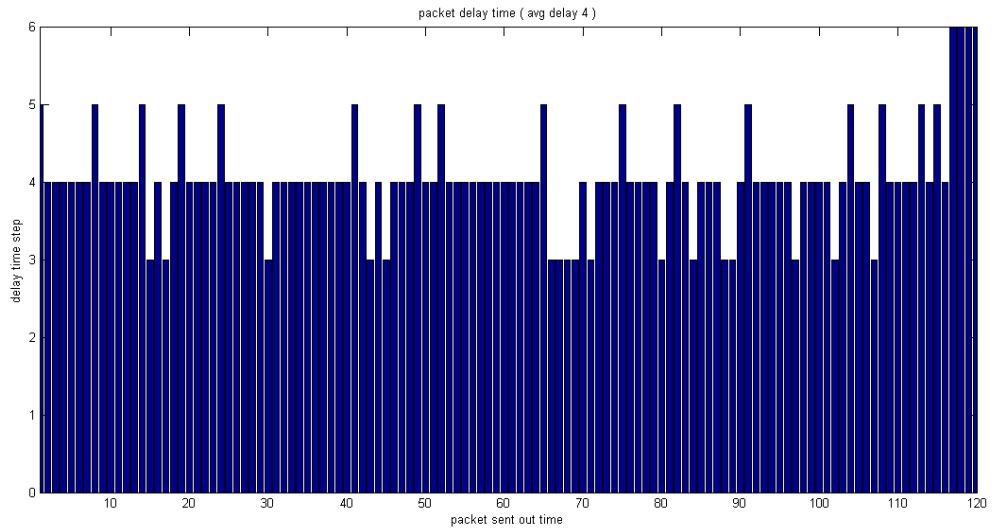


圖 5-7 網路情況 π_3 產生之資訊延遲時間

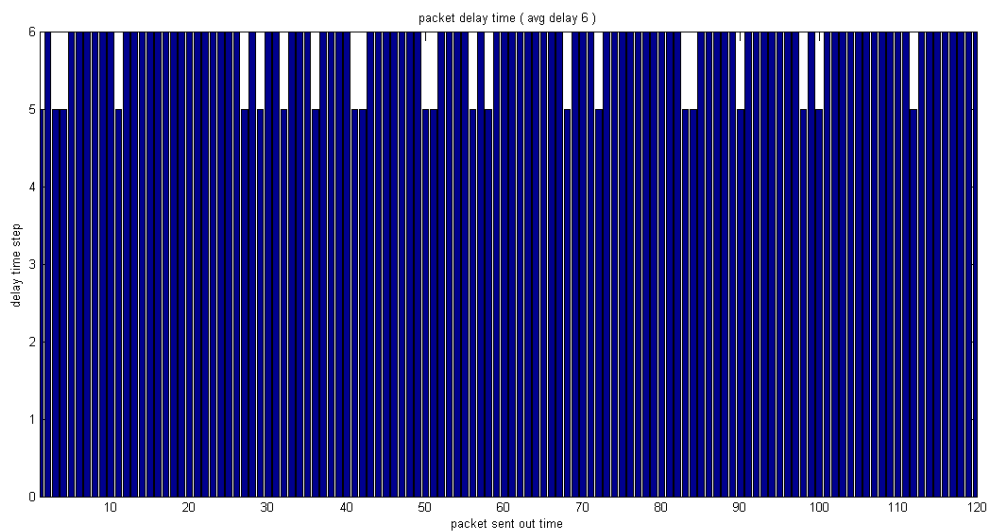


圖 5-8 網路情況 π_4 產生之資訊延遲時間

以這些網路情況條件所產生之模擬網路傳輸資訊送入估測器，便可進行估測的計算。

5.4.4 有效視窗大小選擇

為了確認有效視窗大小的設定對此估測結果的影響，選擇了固定的網路情況 π_3 來測試， π_3 的平均延遲時間是 4 個時間單位。令有效視窗大小為 1、3、5、7 來測試粒子濾波器效果，也就是分別代表有效視窗過小、有效視窗包含部分觀測、有效視窗包含多數

觀測以及有效視窗過大的情形。一樣將估測動作進行 10 次後統計，計算方均根誤差，便獲得如下之結果。

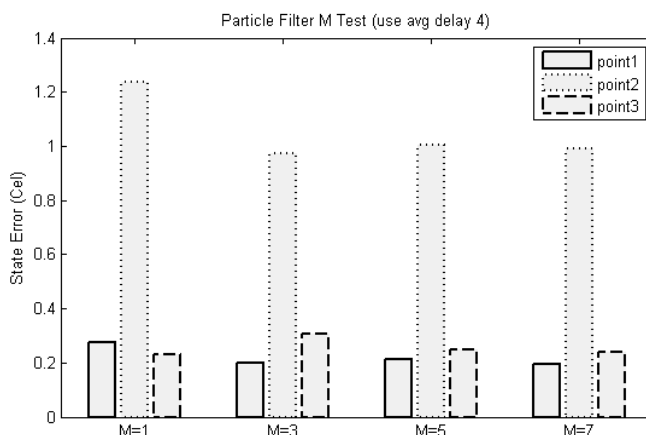


圖 5-9 不同有效視窗大小結果

可以看到當選擇的有效視窗過小時，由於完全沒有收取到任何觀測資訊，因此估測結果較差；而有效視窗大小能涵蓋收取到的觀測資訊時，估測結果稍有改善；而當有效視窗繼續加大時，由於並不會獲得任何額外觀測資訊，因此估測結果不會有多少改變。

於是在選擇有效視窗大小時，應該參考到使用的網路情況，應該選擇有效視窗大小能夠適當的涵蓋接收到的資料延遲時間範圍，不應過小，過大雖不會使估測效果降低，但相對會做多餘的運算。

5.5 模擬結果

考慮到以卡曼濾波器發展之用於網路化控制系統之控制與估測[13]。在這裡參考第二章相關研究中所提之對網路延遲時估測器的做法，加上卡曼濾波器做比較，修改卡曼濾波器採納之觀測資料方式：當觀測資料延遲時，卡曼濾波器會將上次所收取到之最近資料作為觀測值進行估測。並且在模擬中，給予卡曼濾波器與粒子濾波器之估測器參數設定如下：

State noise covariance: 1

Observation noise covariance: 0.1

Particle filter particle count: 10000

$M = 5$

依據前述之加熱系統與網路設定所產生之資料，經過卡曼濾波器與所提之估測器方法估測，以下是其中幾筆的估測目標真實溫度與估測溫度，實線的是不含雜訊的真實系統值，短虛線的是粒子濾波器估測之估測結果，而長虛線的則是卡曼濾波器之估測結果。

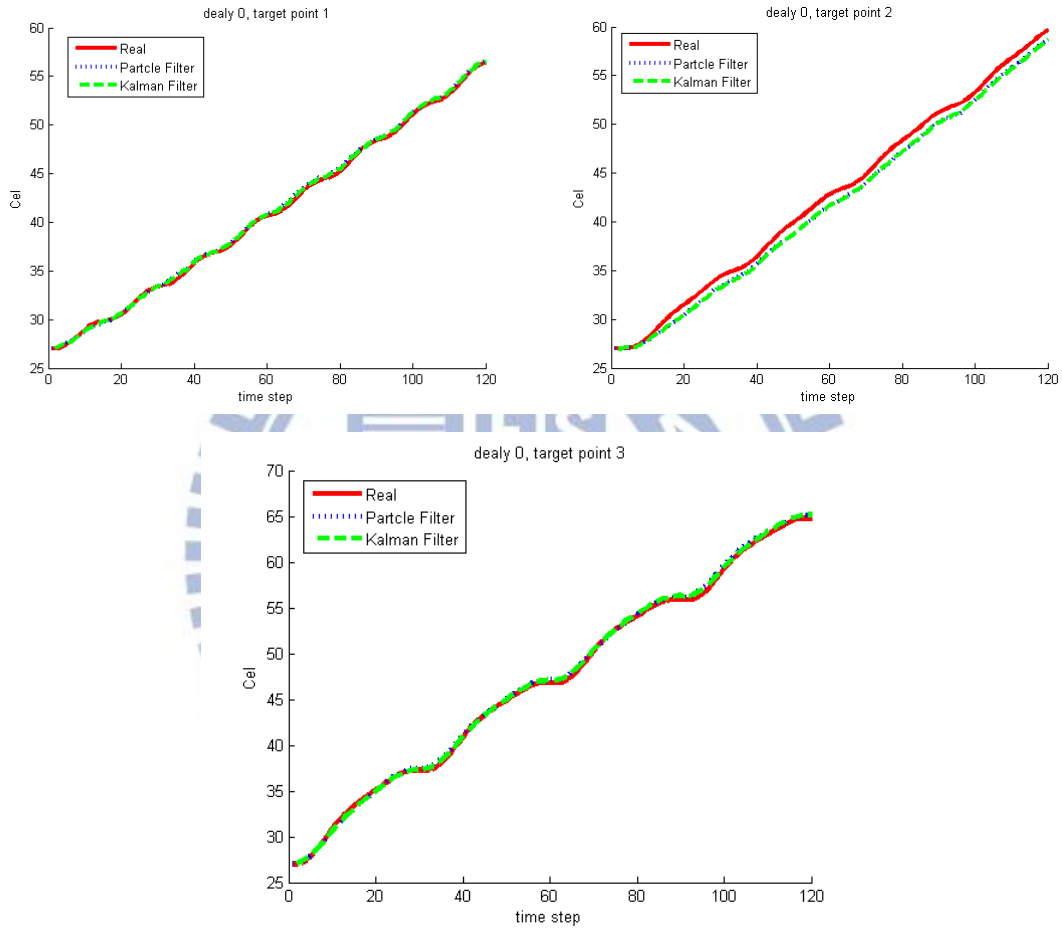


圖 5-10 網路情況沒有延遲(π_1)之估測目標溫度

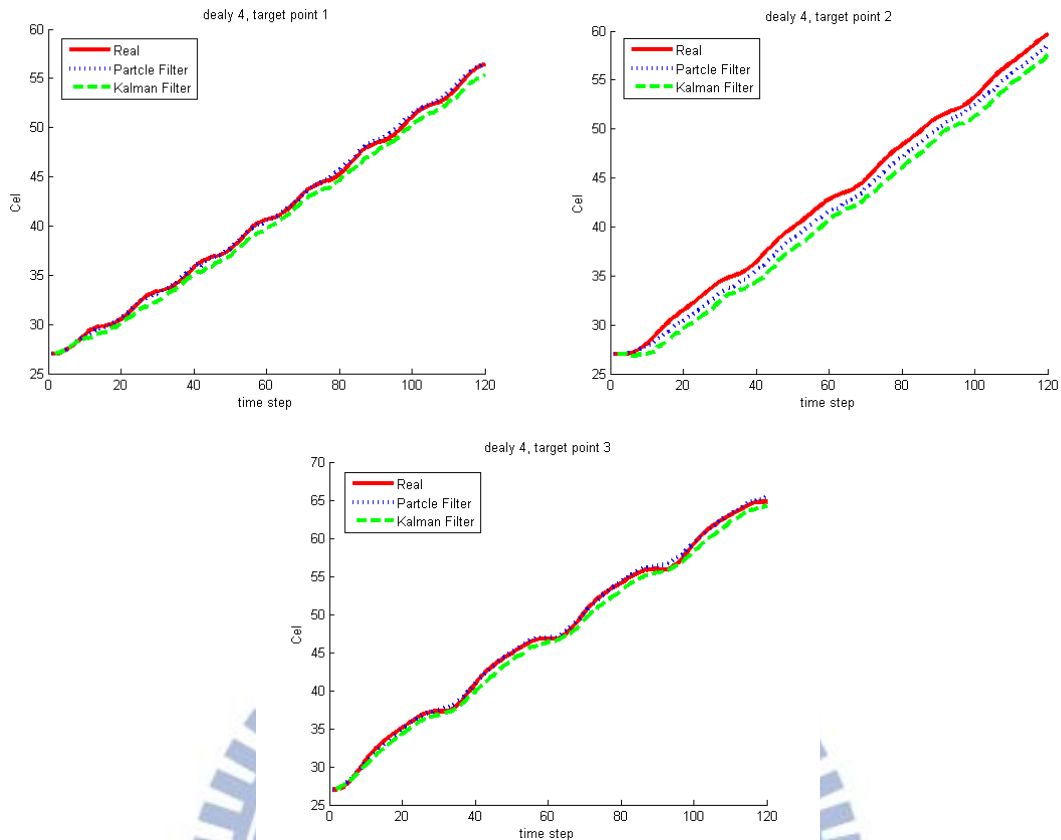


圖 5-11 網路情況平均延遲 4 時間(π_3)之估測目標溫度

雖然延遲時間較長的圖 5-11 稍微可以看出估測之差距，但由於粒子濾波器方法是一個隨機的方法，因此將這估測的動作進行 10 次，取方均根誤差，使用此統計後之結果進行比較。

5.5.1 四個網路情況之結果統計圖

以下為將觀測通過四種網路情況之結果，有三個誤差值，分別對應三個估測點。其全部資料之平均延遲時間單位分別為 0 個時間單位(無延遲)、2 個時間單位、4 個時間單位以及 6 個時間單位。

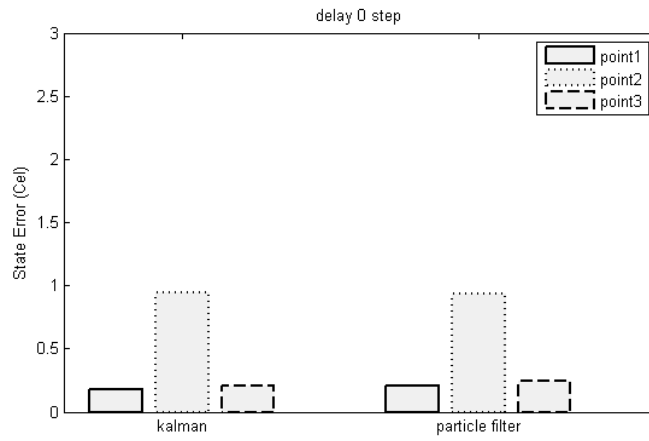


圖 5-12 網路情況 π_1 沒有延遲

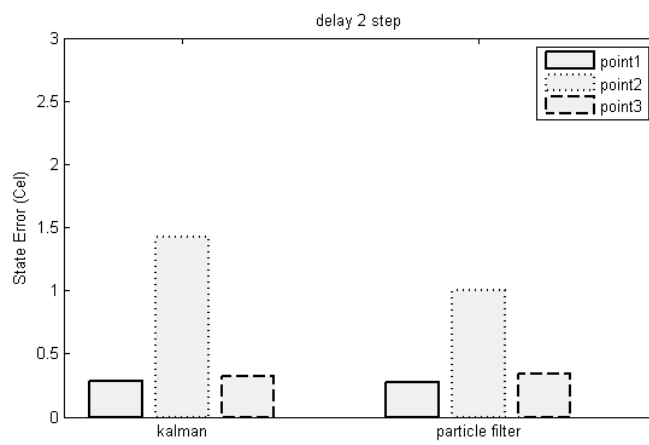


圖 5-13 網路情況 π_2 平均延遲兩個時間單位

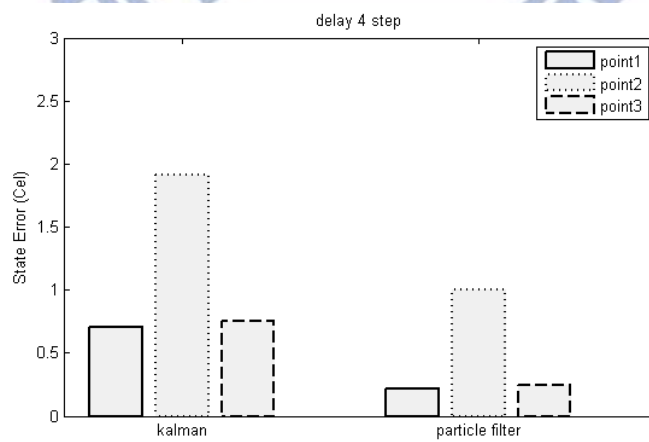


圖 5-14 網路情況 π_3 平均延遲四個時間單位

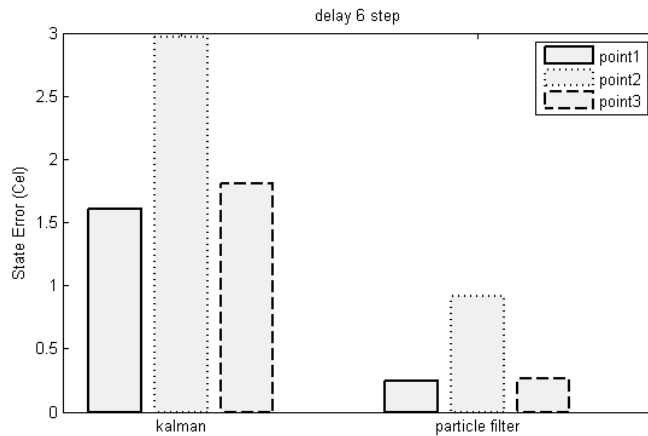


圖 5-15 網路情況 π_4 平均延遲六個時間單位

5.5.2 統計結果說明

可以由圖 5-11 ~ 5-14 看到，當網路傳輸沒有延遲時間時，卡曼與粒子濾波器之結果相當接近。而隨著延遲時間逐漸增加時，卡曼濾波器受到延遲之觀測資訊的影響而估測出比較差的結果；粒子濾波器則仍維持穩定之估測效果。也就是說，粒子濾波器收到延遲之資料可以依其所獲知之延遲時間資訊判斷其在估測中之價值，藉以維持一定之估測效果。而傳統之卡曼濾波器則因為無法利用此延遲之資訊而造成其估測效果的低落。

其中尚可發現到網路情況 π_2 的估測結果誤差稍微高了一些，這是因為在平均延遲 2 時間的情況下，無法利用到所提之重要性分布取樣帶來之好處，以及因為近似之分布在短延遲時所提供之有效點數改善效果較不明顯所致。

第六章 實驗驗證

為了驗證本研究所提出之網路模型與估測器效果，將建構如第五章所述之數據模擬之系統，並且使用實際之網路作為傳輸介面，將接收到之觀測資訊送入估測器中做估測。

6.1 實驗平台

實驗時將使用可控溫鐳槍內之陶瓷發熱體對銅板進行加熱，並且使用熱電偶進行溫度量測。量測後之數據經過類比轉數位後送至電腦端，後經過網路採集延遲資訊。下面為實驗平台之系統架構以及完成之實物圖。

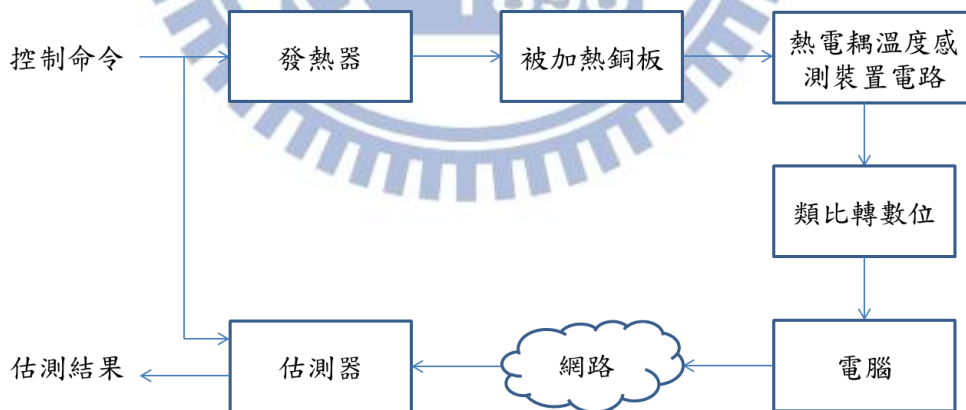


圖 6-1 實驗之系統架構

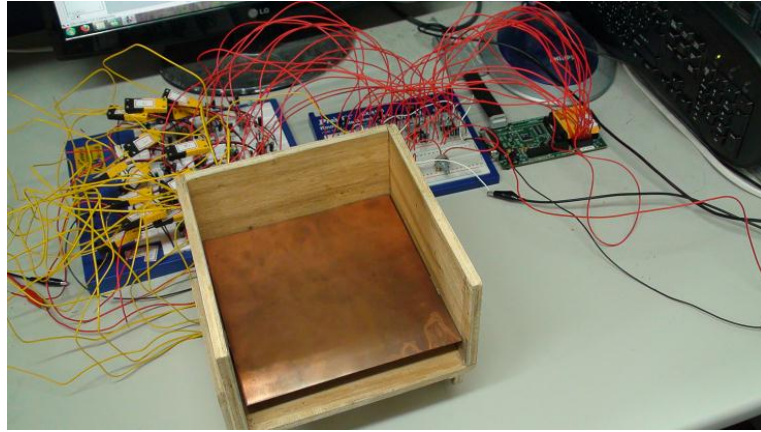


圖 6-2 實驗平台之實物

6.2 平台中各裝置

6.2.1 發熱器

為了增加發熱器每秒所能輸出之熱量，直接使用市售可控溫之鐸槍，並且將其外部拆卸後，直接利用其中心之陶瓷發熱體與銅板接觸後加熱。使用之鐸槍與控溫裝置型號為 Hako 936，其可控溫範圍為 200~480 攝氏溫度，在實驗時將其目標溫度設為攝氏 480 度，然由於陶瓷發熱體之各位置溫度並不均是目標溫度，因此仍需在溫度數據採集後對其做參數估測，才可作為已知之控制命令。

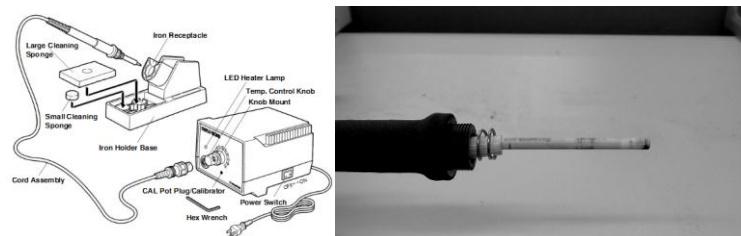


圖 6-3 使用之可控溫鐸槍與其陶瓷發熱體

6.2.2 被加熱銅板

被加熱銅板為 18×18 平方公分、厚 0.5 公厘之平板。一般來說銅金屬具有下列之物理特性：

表 6-1 金屬銅之物理特性

密度	8940 kg/m^3
比熱	$385 \text{ J/kg} \cdot \text{K}$
熱擴散率(熱傳導率/單位體積熱容量)	$112.34 \times 10^{-6} \text{ m}^2/\text{s}$

由於實際使用之銅材通常為合金，並不一定具有完全相同之物理特性，因此此數據乃一約略之範圍，之後仍需對此些係數作微調。

6.2.3 熱電耦

熱電耦是常用於高溫度範圍的溫度感測裝置，其感測之原理是當一金屬被施加以熱梯度時，其會產生電壓，此效應稱為熱電效應，而當使用兩種不同金屬相連接時，其之間之電壓差便可經過測量與比較而得出量測端點之溫度。

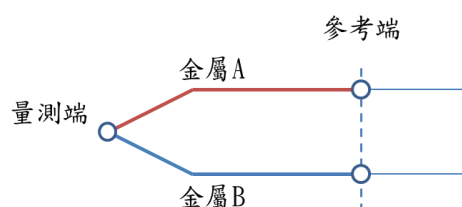


圖 6-4 熱電耦裝置

實驗中使用 J 型熱電耦，該熱電耦是由鐵與銅鎳合金兩種金屬線相連而成，適用之溫度量測範圍為攝氏-40 度至攝氏 750 度左右，受限於金屬鐵之磁性轉變點溫度。並且當環境參考點溫度為攝氏零度時，其電壓/攝氏溫度之敏感度約為 $55\mu V / ^\circ C$ 。若環境參考端溫度非攝氏零度時，須將此電壓接往有精準調校之放大晶片，獲得與環境參考端溫度無關之固定敏感度。

6.2.4 熱電耦用放大與校正晶片

使用 Analog Device 出產之 AD8494 晶片，此晶片內建有參考端溫度量測與校正功能，用來將 J 型熱電耦輸出之電壓進行放大與校正，並以此輸出一個敏感度固定為 $5mV / ^\circ C$ 之電壓值。同時，為了減少無線電波等雜訊干擾量測，可搭配簡易之低頻濾波器進行工作。

表 6-2 AD8494 特性

工作環境溫度	$0^\circ C \sim 50^\circ C$
供應電壓	2.7V~36V
量測範圍	完整 J 型熱電耦量測範圍
輸出	$5mV / ^\circ C$

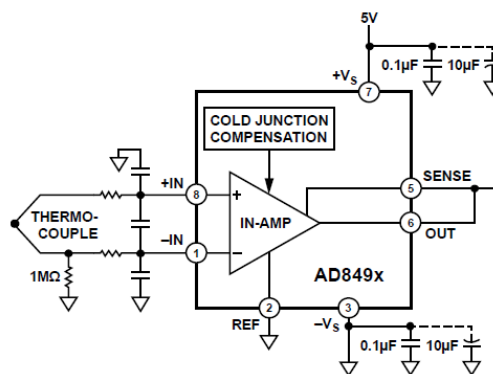


圖 6-5 單組 AD8494 晶片之應用電路

6.2.5 抗交疊濾波器(anti-aliasing filter)

AD8494 晶片輸出之電壓訊號是一類比訊號，在進行類比轉數位之動作前，使其通過抗交疊(anti-aliasing)電路，濾除掉高頻之訊號。

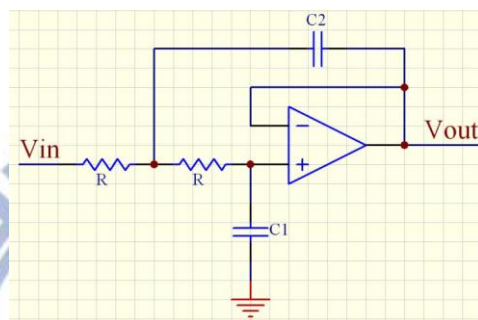


圖 6-6 抗交疊濾波器電路圖

使用二階低通巴特沃茲濾波器(Butterworth Filter)，由於系統之取樣頻率為 2Hz，依照取樣定理須濾除 1Hz 以上之訊號，因此選用

$$R=300K \Omega$$

$$C1=0.47 \mu F$$

$$C2=2C1$$

$$\text{截止頻率為 } f_{cut} = \frac{1}{2\pi R\sqrt{C1C2}} = 0.7982Hz。 \quad (6.1)$$

6.2.6 F2812 數位訊號處理器

使用 Texas Instrument 所製之 TM320F2812，此裝置以核心速度 150Mhz 工作，其開發板 eZdsp F2812 可接受 0~3.3 伏之電壓輸入，以 12bit 之類比轉數位裝置得到目標數

值。將 AD8494 晶片經抗交疊電路後之訊號接往其類比轉數位接腳後，以每 0.5 秒進行一次轉換的方式採集最多 16 個類比訊號輸入，將其轉成 0~4095 間之數位數值。

6.3 網路傳輸紀錄

由於在本研究所提出之估測法中，需要使用到每筆接收到之觀測資料的延遲時間，且為了確保每次封包內容之資料無誤，從常見之 TCP、UDP 協定中選擇了 TCP 協定。TCP 協定帶來的好處是每筆資料的正確，以及另一方面會自動在資料抵達後回傳確認資訊。確認資訊的傳送會使得每一筆資料都會被正確傳遞。雖然因此沒有辦法直接呈現資料之遺失情形，但是隨著延遲時間的改變，許久沒有收取到的資料亦有相當於遺失的效果。

使用 c++ 語言編寫一程式，該程式在兩電腦間建立 TCP 連線，將每個時間之觀測資料逐步送出，並且在資料送出時在觀測資料串之後方附註上出發當地端時間。然而，如同在第二章所提之資訊出發端與接收端的同步問題，若能使出發與接收端同步，則估測結果將能確實反映在該網路下之效果。

於是，將實驗之觀測端與估測端程式安置在同一電腦，資料自觀測端送至遠方之第二電腦，第二電腦直接將資料完整無更動回傳，在出發之電腦處再度收到該資料時，記上收到時間。如此便可確保所紀錄之延遲時間沒有含有任何系統時間誤差。

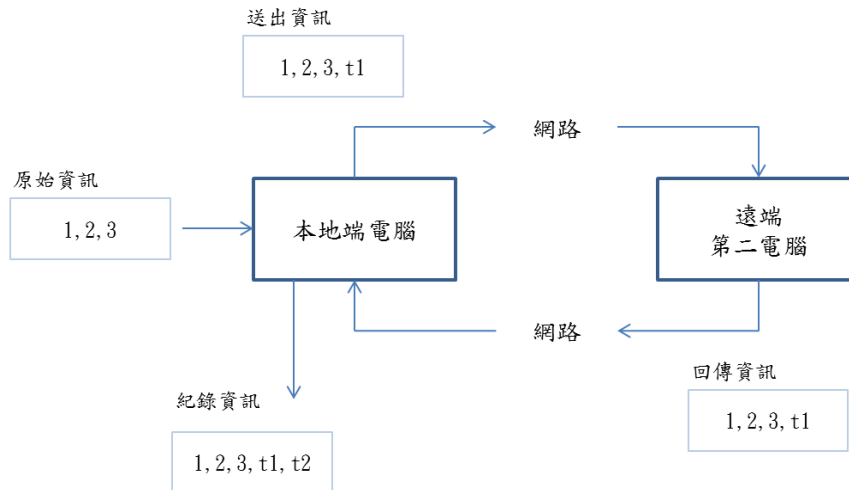


圖 6-7 實驗之網路紀錄方式示意圖

如圖 6-，利用出發端附加出發時間資訊 t_1 ，遠端電腦將其原封不動回傳，接收時記
下接收時間 t_2 ，換算即可得知延遲多少取樣時間。

$$Delay\ Step: m = \left\lceil \frac{t_2 - t_1}{sampling\ period} \right\rceil \quad (6.2)$$

此方式出發與接收是同一電腦，其目的是為了使延遲時間可以被確實記錄，藉以使估測
結果不受兩端點之系統時間不同步之影響。

6.4 加熱方程式之改動

在模擬中所提到之暫態熱傳導公式中所出現之點熱源項，由於實際執行時並無法直
接控制所提供之熱能，亦即無法以 $Q(t)$ 函數表示。因此為了確實描述實際加熱時之情
況，將暫態熱傳導之公式改為：

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + H_h (T_h - T) \delta(x - x_h, y - y_h) \quad (6.3)$$

α : 熱擴散率

將其離散化後為

$$\frac{\partial T_{i,j}}{\partial t} = \alpha \left(\frac{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}}{\Delta x^2} + \frac{T_{i,j-1} - 2T_{i,j} + T_{i,j+1}}{\Delta y^2} \right) + \bar{H}_h (T_h - T) \delta(i - i_h, j - j_h) \quad (6.4)$$

$$\bar{H}_h = \frac{H_h}{cp \cdot \rho \cdot \Delta x \cdot \Delta y \cdot h}$$

H_h : 接觸面之熱交換係數

cp : 比熱

ρ : 密度

$\Delta x, \Delta y$: 端點間之距離

h : 銅板厚度

依此改動後，則控制項目從固定瓦數改為發熱體之溫度，在模型中亦多了 \bar{H}_h 參數需要做參數估測。



6.5 實驗進行之設定

6.5.1 加熱系統設定

在數次測試中，由於發熱體與銅板間之接觸面積極小，使得其熱交換係數小，亦即若要提供足夠之熱量使銅板溫度在數個取樣周期內有大幅變化，需要相當高溫之發熱體。然而經過數次嘗試，溫度仍顯不足。

因此最後採用將發熱器改為橫向放置，也就是與銅板之接觸非單點而為一長型面積，並且以兩相鄰之點來近似此加熱方式。

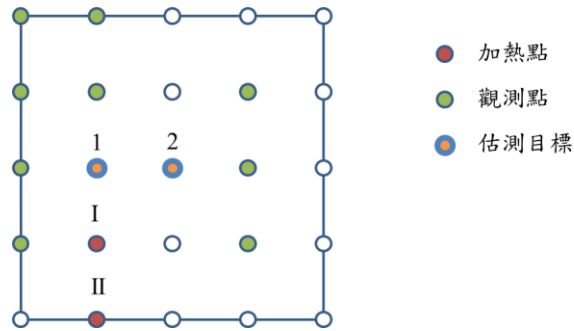


圖 6-8 實驗之加熱系統配置

將發熱器置於加熱點 I 與 II 之處，至此全部模型當中出現需要參數估測之參數：

α : 熱擴散率

$cp \cdot \rho \cdot \Delta x \cdot \Delta y \cdot h$: 端點方格範圍內熱容量

\bar{H}_1, \bar{H}_2 : 兩個接觸點之熱交換係數

為了獲得實際參數，在採集完溫度數據後，根據參數之理想值進行參數微調，使模型產生之溫度與採集到之真實溫度數據之均方根誤差最小。在調整過數組可能範圍後，選取其中各端點誤差差距較小、並與理想值較為接近的一組，如下表之數據，乘號前為原理想值。

表 6-3 實際銅板之參數

熱擴散率	112.34×0.9001
熱容量	34.85×0.983
接觸面熱交換係數 \bar{H}_1	0.0116
接觸面熱交換係數 \bar{H}_2	0.0601

6.5.2 加熱命令

實驗共進行一分鐘，以 2Hz 採集溫度，發熱器自 0 秒開始即放上銅板進行加熱，並且由於發熱體本身溫度不均勻，可高於其尖端溫度攝氏 480 度，由於其逼近類比轉數位電路之最大電壓，因此改為在溫度數據採集後估算，分別得到攝氏 519.25 度以及攝氏 508.65 度。

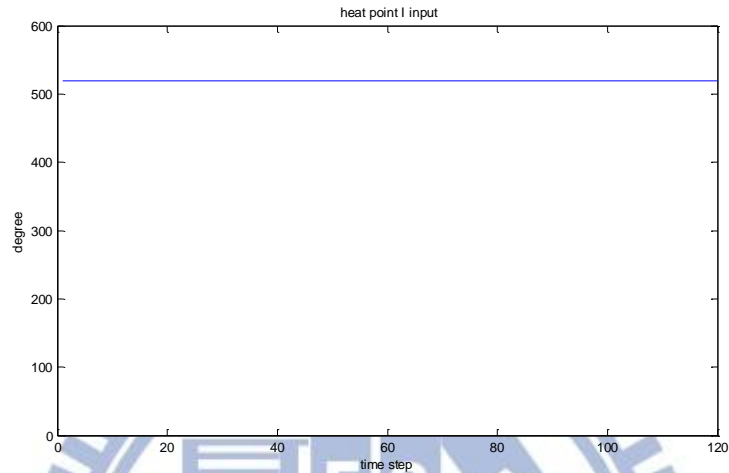


圖 6-9 加熱點 II 之輸入溫度

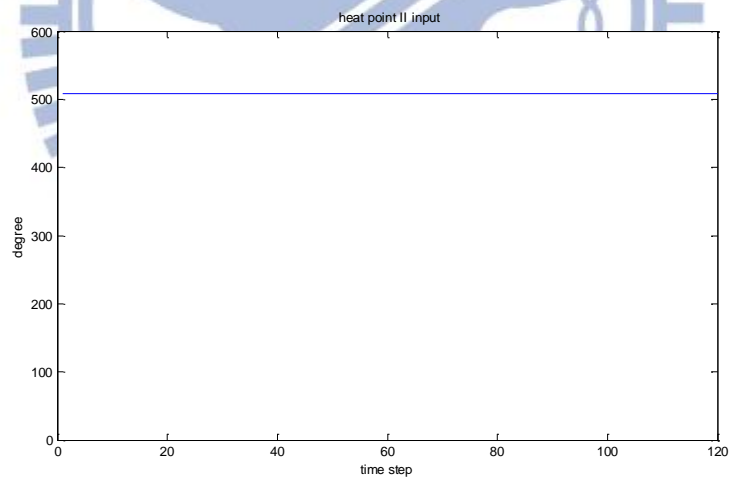


圖 6-10 加熱點 II 之輸入溫度

6.5.3 網路情況

採集之資訊透過網路在校內實驗室與台北遠端之電腦傳遞，並且以控制上傳負載量

的方式在三種網路情況下數據的採集了數據，分別是低負載量、中-高負載量以及高負載量。由於是在同一線路下以網路負載量來改變傳輸情況，因此低至中高負載量下，延遲時間並無太大之改變，而一旦網路負載接近滿載時，延遲時間則大幅增加。

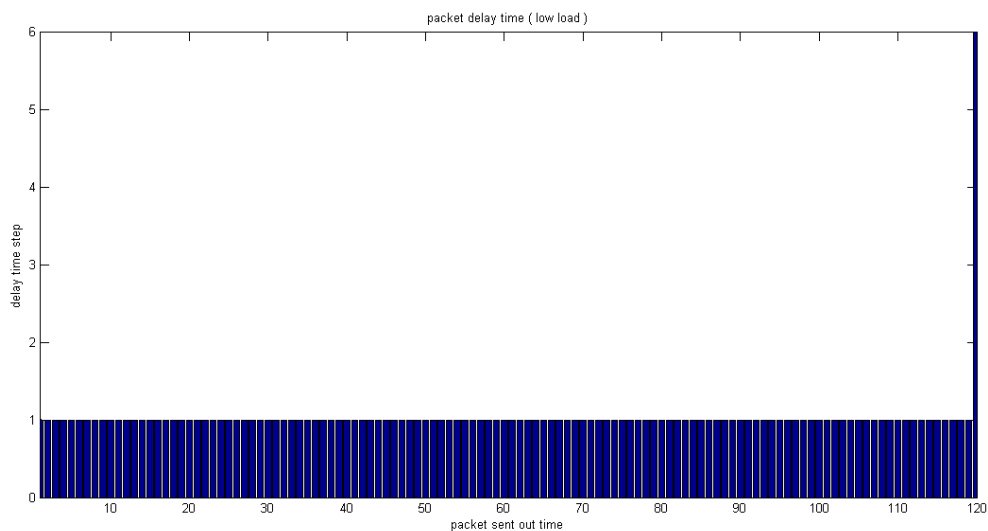


圖 6-11 網路低負載時之延遲時間單位

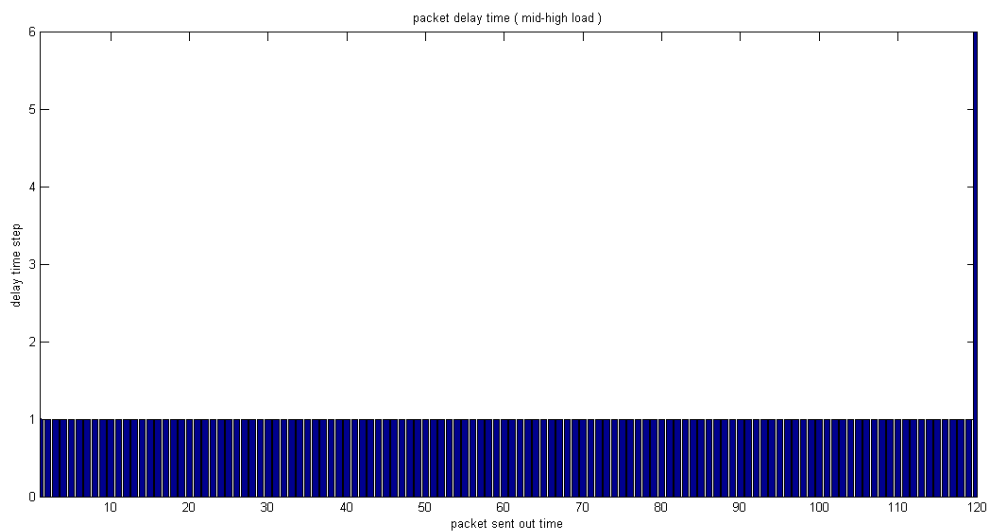


圖 6-12 網路中-高負載之延遲時間單位

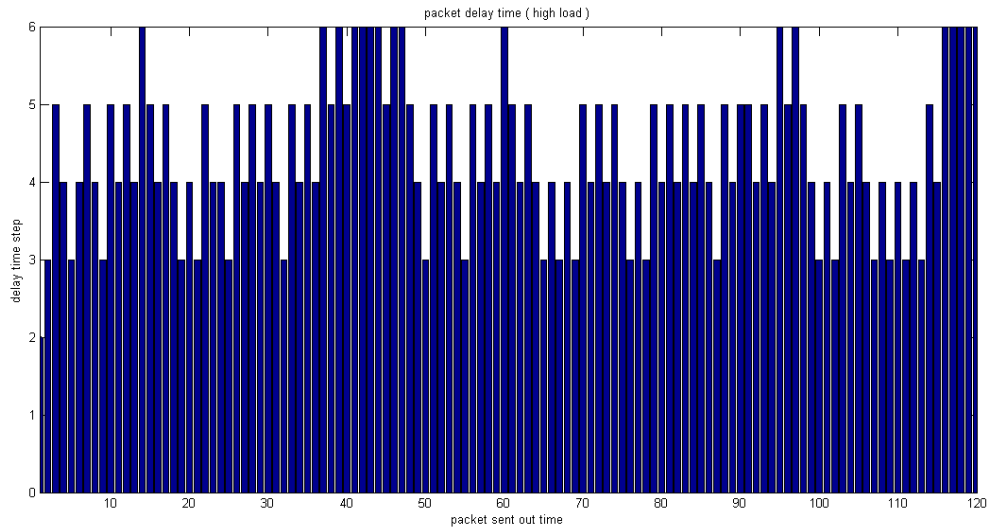


圖 6-13 網路高負載之延遲時間單位

以上是採集到的資料之延遲時間單位，並且其平均延遲時間與平均延遲時間單位為

表 6-4 三種網路情況平均延遲

網路上傳負載	平均延遲時間	延遲之取樣時間單位
低	66.85ms	1
中-高	116.5ms	1
高	1959ms	4

透過這三種網路情況傳輸觀測資料至估測端收集。由於演算法對計算力之需求尚大，尚無法即時運作，在此處採離線的方式對收取到之資料進行估測。

6.6 實驗結果

給予估測器的設定如下，由於網路延遲時間最多在 4 單位左右，因此仍然採用有效視窗 M 為 5。

State noise covariance: 0.5

Observation noise covariance: 0.07

Particle filter particle count: 10000

M=5

依此設定獲得估測結果，底下是其中幾次的估測結果，實線為實際觀測值因此含有雜訊，短虛線為粒子濾波器之估測結果，長虛線為卡曼濾波器之估測結果。與模擬相同，為了確實比較，仍然需要進行 10 次並且統計。

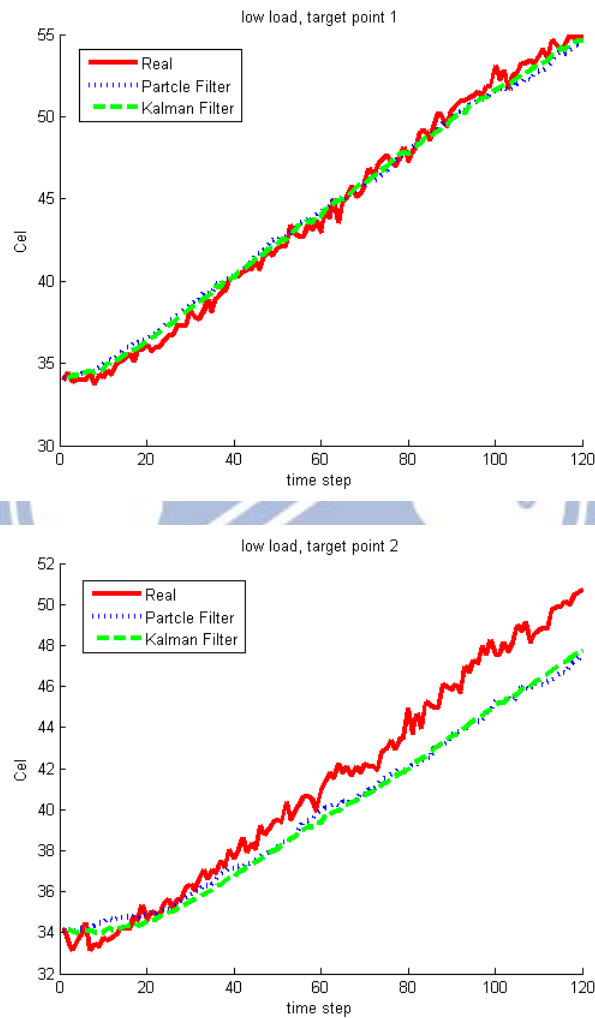


圖 6-14 低負載網路之估測結果

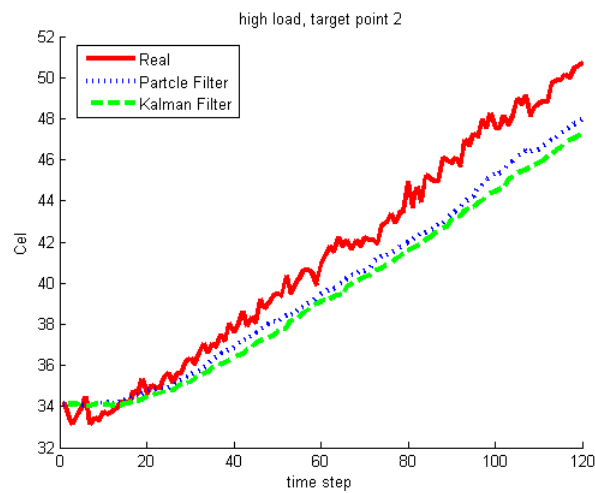
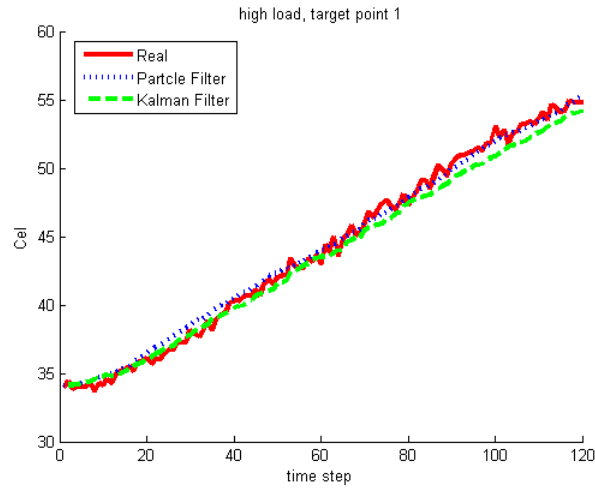


圖 6-15 網路高負載之估測結果

可以看到估測目標點一的估測較好，是因為估測目標點一離熱源較近，且其周邊之觀測點溫度變化較大，相對也就是信噪比較高，因此可以得到較好之估測。

6.6.1 三個網路情況之結果統計圖

進行 10 次估測後，將估測目標點之方均根誤差作時間上之平均，將其繪出，實線為估測目標 1 之誤差，短虛線為估測目標 2 之誤差。估測目標 1 離熱源與變化大之觀測點較近，得到較低之平均誤差，目標 2 離熱源與變化大之觀測點較遠，得到較高之平均誤差。

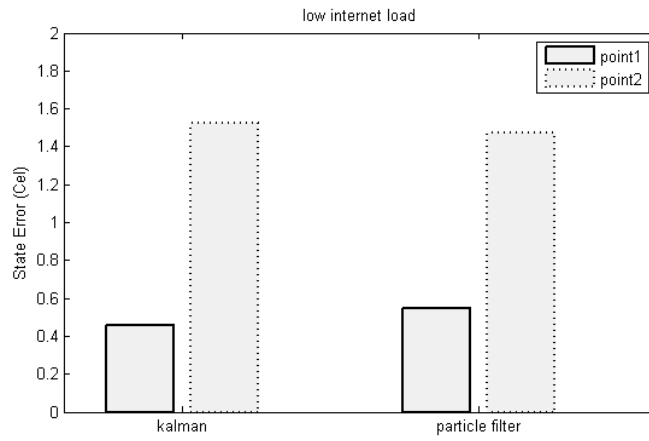


圖 6-16 低負載網路之估測結果

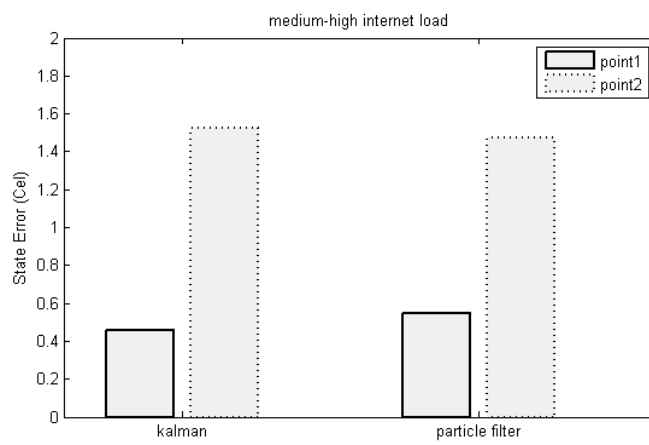


圖 6-17 中-高負載網路之估測結果

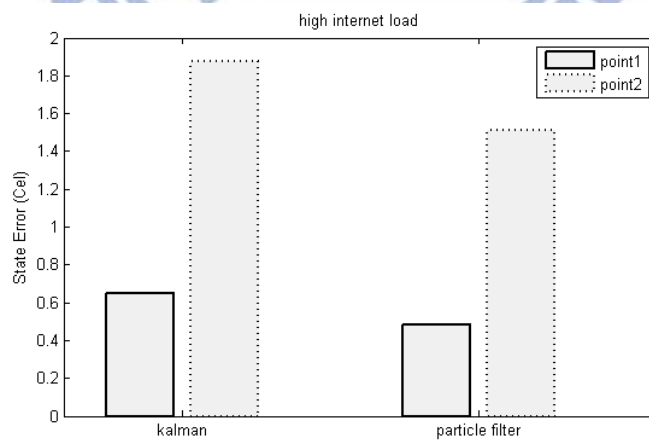


圖 6-18 高負載網路之估測結果

6.6.2 結果說明

低負載網路與中-高負載網路由於延遲時間都是一個取樣週期，因此結果相同，可以看到卡曼濾波器與粒子濾波器得到相當之結果。而當網路負載高時，由於延遲時間拉長到 4 個取樣週期，因此使得卡曼濾波器之估測效果變差，而粒子濾波器由於能對延遲時間之資料做正確之處理，因此使得其估測效果沒有受到影響。

此實驗結果與模擬之結果相仿，可以了解到粒子濾波器在低延遲時估測效果與傳統卡曼濾波器同等；隨著延遲時間逐漸拉長，粒子濾波器能夠正確利用延遲之資料，從而維持其估測效果，相對的傳統卡曼濾波器則受到資料有延遲的影響，做出之估測有所偏差。



第七章 結論與未來工作

7.1 結論

本研究為了建構一適用於網路化控制系統之估測器，對網路與受控系統建立了以機率為基礎之模型。其中使用馬可夫鏈變數描述資訊封包送出後在網路內之傳遞情況變化，並且此描述本身對網路之最大延遲時間並無限制，在每個時間點，封包資訊之可能變化各使用一個參數來描述。並且，將受控系統依其系統與觀測雜訊描述成複合的高斯分布。

基於此模型方式，因粒子濾波器方法是一個以機率期望值為基礎發展之估測方法，故以此法為基礎，將網路與受控系統模型結合入其演算法中，得到一個適用於網路化控制系統的粒子濾波器。

本研究亦應用粒子濾波器方法中廣為使用的重新取樣、正規化方法於此估測中做為改良。並從一般常見之 Bootstrap Filter 方法改為使用與現有之觀測有關之重要性取樣方式，以及對權重更新公式做修改，減少計算需求以及使用之記憶空間，進而使得此演算法在同等計算力的情況下能獲得更好的效果。

依本研究對系統架構之假設，設想一加熱裝置情境，依此進行模擬與實驗，對不同長度之延遲時間獲得較傳統無考慮資訊延遲之卡曼濾波器為穩定之估測效果。進而說明對網路化控制系統設計對應之估測器是有其使用之意義與一定程度之成效。

7.2 未來工作

在本研究中對網路資訊傳輸之參數依實際可能狀況設定有三個值，分別為接收、延遲與遺失，然而目前在進行狀態估測時，對應於封包遺失之轉移機率對狀態估測沒有影響，因此未來可以考慮簡化網路模型之設定，或者探索封包遺失機率在其他網路化控制系統之應用的可行性。

由於本研究並無做控制，因而無考慮控制端亦連接網路之情形，若在控制端加入網路時，由於無法即時得知控制命令是否送達、其延遲之時間，可將其資訊透過觀測端線路送出，估測器尚未得知此訊息時可利用網路機率模型做預估，在得到此資訊後則可應用本論文所提出之演算法進行狀態估測。



參考文獻

- [1] Rachana Ashok Gupta and Mo-Yuen Chow, “Networked Control System: Overview and Research Trends”, IEEE Transactions on Industrial Electronics, Vol.57, No.7, pp.1249-1258, July 2010.
- [2] Johan Nilsson, “Real-Time Control Systems with Delays” Ph.D. dissertation, Lund Institute of Technology, Lund, Sweden, 1998.
- [3] Yang Shi and Bo Yu, “Output Feedback Stabilization of Networked Control Systems with Random Delays Modeled by Markov Chains”, IEEE Transactions on Automatic Control, Vol.54, No.7, pp.1668-1674, July 2009.
- [4] Jing Wu and Tongwen Chen, “Design of Networked Control Systems with Packet Dropouts”, IEEE Transactions on Automatic Control, Vol.52, No.7, pp.1314-1319, JULY 2007.
- [5] Fuwen Yang, Zidong Wang, Y. S. Hung and Mahbub Gani, “ H_{∞} control for networked systems with random communication delays”, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL.51, NO.3, pp.511-517, MARCH 2006.
- [6] X. Fang and J. Wang, “Stochastic Observer-based Guaranteed Cost Control for Networked Control Systems with Packet Dropouts”, IET Control Theory Appl., Vol.2, No.11, pp.980-989, 2008.
- [7] Ecole Polytech., Palaiseau and Cappe, O., “Comparison of Resampling Schemes for Particle Filtering”, Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, pp.64-69, 2005.
- [8] Deok-Jin Lee, NonLinear Bayesian Filtering with Applications to Estimation and Navigation, 2005.

- [9] Ross D. Shachter, "Bayes-Ball: The Rational Pastime (for Determining Irrelevance and Requisite Information in Belief Networks and Influence Diagrams)", Proceedings of the Fourteenth Conference in Uncertainty in Artificial Intelligence, pp.480-487, 1998.
- [10] N. Gordon, D. Salmond and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", IEEE Proc.F. Radar Signal Process., vol.140, pp.107-113, 1993.
- [11] H. Lin and P. J. Antsaklis, "Stability and persistent disturbance attenuation properties for a class of networked control systems: Switched system approach", Int. J. Contr., vol.78, no.18, pp.1447-1458, 2005.
- [12] M. Yu, L. Wang, G. M. Xie and T. G. Chu, "Stabilization of networked control systems with data packet dropout via switched system approach," in Proc. IEEE Int. Symp. Comput. Aided Control Syst. Design, pp.362-367, 2004
- [13] Bruno Sinopoli, Luca Schenato, Massimo Franceschetti, Kameshwar Poolla, Michael I. Jordan and Shankar S. Sastry, "Kalman Filtering with Intermittent Observations", IEEE Trans. Autom. Control, 49, pp.1453-1464, 2004.
- [14] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon and Tim Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", IEEE Transactions on Signal Processing, vol.50, no.2, pp.174-188, Feb 2002.