

國立交通大學

電機與控制工程學系

碩士論文

使用快速 GPB-2 演算法的即時機器人致動器錯誤偵測與識別

Real-Time Detection and Identification of Robot Actuator

Failures with Fast GPB-2 Algorithm

研究生：黃暉鈞

指導教授：蕭得聖 博士

中華民國九十七年七月

使用快速 GPB-2 演算法的即時機器人致動器錯誤偵測與識別

Real-Time Detection and Identification of Robot Actuator

Failures with Fast GPB-2 Algorithm

研究生: 黃暉鈞

Student: Huei-jyun Haung

指導教授: 蕭得聖 博士

Advisor: Dr. Te-Sheng Hsiao

國立交通大學

電機學院 電機與控制工程學系



A Thesis

Submitted to College of Electrical Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Electrical and Control Engineering

July 2008

Hsinchu, Taiwan, Republic of China

中華民國 九十七 年 七月

使用快速 GPB-2 演算法的即時機器人致動器 錯誤偵測與識別

研究生：黃暉鈞

指導教授：蕭得聖 博士

國立交通大學電機與控制工程學系碩士班

摘要

自 1970 年代始，機器人即大量運用於工業製造生產上，例如汽車、機械與半導體等產業。當時的機器人只能執行反覆單調的生產裝配等任務。隨著時間的流逝，科技日新月異，機器人也開始擁有更多的功能，甚至準備進入人類的生活之中。使機器人進入人類生活的障礙，並非機器人的功能或可提供的服務而在於其運作時的可靠度、安全性與成本。可靠度是指機器人自身能夠長期於妥善的狀況下運作，而安全性乃指機器人運作時不應對人類以及週遭環境造成傷害，成本則是指該機器人製造與後續維修所需的費用。

本論文的貢獻在於對機械臂僅裝置角度編碼器以降低成本的情況下，發展應用於機械臂的即時錯誤偵測與辨識演算法，以期能對機械臂的運作狀態進行良好估測，增加機器人運作的安全性。文中將使用非追蹤型卡曼濾波器進行機械臂角度和角速度的估測，使用期望值最大化演算法估測摩擦力，利用濾波扭力計算與多模型組態 GPB-2 演算法來判斷錯誤模型的種類，由模擬結果可看到該即時演算法的確比原始的多模型組態 GPB-2 演算法更為精確與有效率。

Real-Time Detection and Identification of Robot Actuator Failures with Fast GPB-2 Algorithm

Student: Huei-jyun Haung

Advisor: Dr. Te-Sheng Hsiao

Institute of Electrical and Control Engineering
Nation Chiao-Tung University

Abstract

Since 1970, robots have been widely used in manufacturing industries, such as automotive, mechanical, and semiconductor industries. But robots were limited to perform simple and repetitive tasks at that time until recently. Due to the advances in modern technologies, as well as the requirements for social and economic aspects, inspiring innovations in robotic technologies are emerging during the past decade. Furthermore, it is expected that robot technologies will be integrated into human's daily life in the near future. However, the obstacles for robots to enter human's daily life are not only the services they can provide, but the reliability, safety and cost of the robots. Reliability means that the robot can operate well for a long time. Safety means the robot should not hurt human or cause damage to the environment.

This thesis has a great contribution on detection and identification of the robot's actuator failures by combining UKF(Unscented Kalman Filter), GPB-2 algorithm, EM algorithm and pre-classification of fault models. Simulation results show that the proposed FDI(fault detection and identification) algorithm is better than conventional GPB-2 algorithm in terms of computational efficiency and accuracy of fault detection.

誌 謝

首先誠摯的感謝指導教授蕭得聖博士，老師悉心的指導使得我得以一窺錯誤偵測與辨識領域的深奧，不時地與我討論並指點我正確的方向，使我在碩士班的這兩年獲益匪淺，收穫頗豐。老師對作學問的要求與嚴謹，更是我學習的典範。

此外，我也要感謝幫我口試的楊谷洋老師、徐保羅老師與陳宗麟老師。各位老師在口試中對我的研究與初稿給了許多寶貴的方向與建議，讓我得以將初稿的缺點加以修正，並讓我對錯誤偵測與識別的領域有更深的體會。

兩年的日子裡，實驗室裡共同的生活點滴、學術上的討論、言不及義的閒聊瞎扯、趕作業的革命情感等，都令人難忘。感謝眾位學長、同學、學弟的共同砥礪，你們的陪伴讓兩年的研究生活變的多彩多姿。感謝永州學長不厭其煩的指導我研究上的缺失，且總能在我迷網時幫我解惑，也感謝毅泓、宗明、昭明陪我解悶，在研究上互相打氣，也感謝東欣、茂橋的幫忙，最後僅以此文現給我摯愛的雙親。

黃暉鈞 謹識

中華民國九十七年七月

新竹 交大

目錄

中文摘要	I
英文摘要	II
誌謝	III
目錄	IV
圖目錄	VI
表目錄	IX

第一章 緒論	1
1.1 研究動機與目的	1
1.1.1 背景知識與回顧	2
1.1.2 研究動機與目的	3
1.2 系統發生錯誤的情形	4
1.3 論文貢獻	5
1.4 論文架構	5
第二章 機器人系統	6
2.1 機器人系統簡介	6
2.2 機器人錯誤偵測與識別相關研究	8
第三章 錯誤偵測與識別技術	11
3.1 基於模型的錯誤偵測和識別技術	12
3.1.1 錯誤系統的模型建立	14
3.1.2 剩餘值產生器	17
3.1.3 剩餘值評估	19
3.2 觀察器設計	21
3.2.1 觀察器架構	21
3.2.2 卡曼濾波器	23
3.2.3 應用卡曼濾波器於非線性系統	25
3.3 多模型架構	30
3.4 期望值最大化	42

第四章	系統計算減量與參數估測	44
4.1	引入運動學方程式進行計算減量	44
4.2	應用期望值最大化於摩擦力估測	49
4.3	濾波扭力計算	58
4.4	分群估測	61
4.5	演算法架構圖	64
第五章	模擬分析與探討	66
5.1	單一模型狀態估測	67
5.2	摩擦力估測	72
5.3	扭力差值估測模擬	80
5.4	動力學模型與運動學模型切換模擬	86
5.5	應用於機械臂之即時演算法模擬	90
第六章	結論與未來工作	107
6.1	結論	107
6.2	未來工作	108
附錄		109
參考文獻		112



圖目錄

圖 2.1	達文西手術系統操作示意圖	7
圖 3.1	錯誤偵測架構圖	12
圖 3.2	閉迴路系統中的錯誤偵測	14
圖 3.3	簡化後的系統錯誤模型圖	15
圖 3.4	剩餘值產生器的通用架構(M. Basseville)	17
圖 3.5	剩餘值產生器的通用架構(Patton and Chen)	18
圖 3.6	觀察器作用示意圖	21
圖 3.7	蒙地卡羅、拓展型卡曼濾波器與非追蹤型卡曼濾波器估測 比較	27
圖 3.8	靜態模型下估測流程方塊圖	33
圖 3.9	GPB-1 流程方塊圖	38
圖 3.10	GPB-2 估測流程方塊圖	41
圖 4.1	機械臂示意圖	44
圖 4.2	狀態空間模型圖	50
圖 4.3	二軸機械臂錯誤模型分群估測說明圖	63
圖 4.4	演算法架構圖	64
圖 5.1	估測二軸機械臂單一狀態架構圖	68
圖 5.2	高斯雜訊下模型一狀態估測	69
圖 5.3	高斯雜訊下模型五狀態估測	69
圖 5.4	大軸慣量為原來的 1.5 倍下模型一狀態估測結果	70
圖 5.5	大軸慣量為原來的 1.5 倍下模型六狀態估測結果	71
圖 5.6	摩擦力估測架構圖	72
圖 5.7	模擬(五)的黏滯摩擦力係數估測	73

圖 5.8	模擬(五)的角度與角速度估測-----	74
圖 5.9	模擬(六)的黏滯摩擦力係數估測-----	74
圖 5.10	模擬(六)的角度與角速度估測-----	75
圖 5.11	模擬(七)的黏滯摩擦力估測-----	76
圖 5.12	模擬(七)的角度與角速度估測-----	76
圖 5.13	模擬(八)的黏滯摩擦力估測-----	77
圖 5.14	模擬(八)的角度與角速度估測-----	77
圖 5.15	模擬(九)的黏滯摩擦力係數估測-----	78
圖 5.16	模擬(九)的角度與角速度估測-----	79
圖 5.17	模擬(十)的黏滯摩擦力係數估測-----	79
圖 5.18	模擬(十)的角度與角速度估測-----	80
圖 5.19	力矩差值估測架構圖-----	81
圖 5.20	模擬(十一)的力矩差值模擬估測-----	82
圖 5.21	模擬(十一)的角度與角速度估測-----	83
圖 5.22	模擬(十二)的力矩差值模擬估測-----	83
圖 5.23	模擬(十二)的角度與角速度估測-----	84
圖 5.24	模擬(十三)的力矩差值模擬估測-----	85
圖 5.25	模擬(十三)的角度與角速度估測-----	85
圖 5.26	動力學模型與運動學模型機率估測架構圖-----	87
圖 5.27	模擬(十四)的動力學模型與運動學模型機率估測-----	88
圖 5.28	模擬(十四)的動力學模型與運動學模型狀態估測-----	88
圖 5.29	模擬(十五)的動力學模型與運動學模型機率估測-----	89
圖 5.30	模擬(十五)的動力學模型與運動學模型狀態估測-----	89
圖 5.31	應用於機械臂之即時演算法模擬架構圖-----	91
圖 5.32	模擬(十六)的機率估測-----	92
圖 5.33	模擬(十六)的模型一與模型二機率估測-----	92

圖 5.34	模擬(十六)的模型估測-----	93
圖 5.35	模擬(十六)的狀態估測-----	93
圖 5.36	模擬(十七)的機率估測-----	94
圖 5.37	模擬(十七)的模型一與模型八機率估測-----	94
圖 5.38	模擬(十七)的模型估測-----	95
圖 5.39	模擬(十七)的狀態估測-----	95
圖 5.40	模擬(十八)的群組選取估測-----	96
圖 5.41	模擬(十八)的力矩差值與模型選取估測-----	97
圖 5.42	模擬(十八)的摩擦力估測-----	97
圖 5.43	模擬(十八)的狀態估測-----	98
圖 5.44	模擬(十九)的群組選取估測-----	98
圖 5.45	模擬(十九)的力矩差值與模型選取估測-----	99
圖 5.46	模擬(十九)的摩擦力估測-----	99
圖 5.47	模擬(十九)的狀態估測-----	100
圖 5.48	模擬(二十)的群組選取估測-----	102
圖 5.49	模擬(二十)的力矩差值與模型選取估測-----	102
圖 5.50	模擬(二十)的摩擦力估測-----	103
圖 5.51	模擬(二十)的狀態估測-----	103
圖 5.52	模擬(二十一)的群組選取估測-----	104
圖 5.53	模擬(二十一)的力矩差值與模型選取估測-----	104
圖 5.54	模擬(二十一)的摩擦力估測-----	105
圖 5.55	模擬(二十一)的狀態估測-----	105

表目錄

表一：二軸機械臂相關參數-----	109
表二：非追蹤型卡曼濾波器強健度測試表-----	110
表三：模型切換強健度測試-----	111



第一章 緒論

1.1 研究動機與目的

自 1970 年代始，機器人即大量運用於工業製造生產上，當時的機器人只能執行反覆單調的生產裝配等任務。自 1990 年開始，隨著個人電腦的普及，感測、通訊、控制等技術日新月異，製造智慧型機器人漸漸不再是個夢想。另外由於人口老化與少子化的情形日趨嚴重，作為照顧老人生活的智慧型機器人的相關研究終於如火如荼的展開。

各項研究報告均指出，智慧型機器人將成為下一世代的明星產業。以機器人製造與研發大國的日本為例，日本政府已將智慧型機器人列入「新產業創造戰略」七大領域之一，其鄰近的韓國也將智慧型機器人列為「十大新世代成長動力產業」之一。至於台灣的智慧型機器人產業預估 2005 至 2008 年的產值為新台幣 300 億元，而 2009 至 2013 年的產值預估將達新台幣 900 億元以上。

目前的機器人的研究多著重於機器人的功能的設計，而較少專注於機器人運作時的可靠度與安全性。可靠度是指機器人自身能夠長期於妥善的狀況下運作，而安全性乃指機器人運作時不應對人類以及週遭環境造成傷害。這被忽略的部份，卻是智慧型機器人能否商品化的最重要關鍵，因為一個不安全不可靠的產品絕對不可能為公司帶來利潤，反而是一種累贅與商譽的損壞。

本論文的目的就是討論機器人錯誤偵測與識別 (**Fault Detection and Identification**) 的能力。藉由二軸機械臂進行錯誤偵測 (fault detection)，利用角度量測進行速度的估測再用其計算力矩與各種錯誤模式下的機率進行比較，以確定錯誤發生的原因。

1.1.1 相關背景知識與回顧

所謂的錯誤偵測與識別在一般文獻研究上稱為 FDI(Fault Detection and Identification)。顧名思義，FDI 分為兩個部份：

1. fault detection: 利用已知系統的狀態方程式，配合感測器輸出，計算出系統是否有錯誤發生。
2. fault identification: 明確指出系統發生錯誤的元件。

Fault Detection 主要可分為兩主要步驟[1]，產生剩餘值(Residuals Generation)和判斷法則(Decision Making)，又可細分如下：

1. 產生剩餘值(Residuals Generation)

- a. 觀察器法(Observer-based): 由感測器量測得到的輸出與觀察器估測之輸出的差值為觀察器法的剩餘值(Residuals)。
- b. 參數估測法(Parameter estimation): 定義的系統參數和估測出的系統參數的差值即為剩餘值(Residuals)。
- c. 等值關係(Parity relation): 同位元檢查法是在時域上取一固定長度的視窗，記錄在此視窗內的輸入與輸出訊號，經過適當的投影，使得系統動態與錯誤訊號處於互相垂直的子空間，然後判別錯誤是否在這段時間內發生。

2. 判斷法則(Decision Making)

- a. 基於模型(Model-based): 利用理論系統與實際系統的誤差判定系統是否有錯誤發生，此方法是由 Jones 於文獻[1]中提出的。其優點在於當能夠精準的獲得理論系統的參數時，也就能精確得計算剩餘值(Residuals)，此方法的抗雜訊能力較高；反之，若理論系統的參數不夠精準時，剩餘值(Residuals)計算的誤差將導致系統錯誤有可能難以判斷。

- b. 不基於模型(Model-free)：顧名思義，此方法僅由感測器的輸出作為判斷系統是否發生故障的依據。此方法於文獻[14]由 A.Ray 和 R.Luck 提出，此方法是先設定剩餘值(Residuals)的範數(Norm)大小，來定義系統發生錯誤的邊界；優點是可即時的運算和判斷，缺點則是抗雜訊能力低，剩餘值邊界範圍不好設定。

產生剩餘值(Residuals Generation)和判斷法則(Decision Making)主要可能碰到的問題有以下四種：

- a. 系統發生錯誤的種類會隨時間改變
- b. 理論系統的參數不準確。
- c. 系統雜訊和量測雜訊影響系統響應。
- d. 必須在特別要求的時間內運算完成

本論文將重心放在 d 部分的探討，至於 a, b 與 c 的部份雖然不是本論文的重心，但仍將在第五章進行模擬，以確保實際運作上該偵錯系統有一定的可靠度。

1.1.2 研究動機與改良

由於使用多模型組態[18]的錯誤偵測與識別的即時(real time)計算量太大了，所以本論文主要是著重在利用一些方法將計算量降低，以其可以達到即時運算。本論文中採用分群的原理，將各種系統錯誤發生的可能，依照機率高低加以排序，使得事前機率(prior probability)的功能發揮到最大。並運用動力學模型(Dynamic Model)和運動學模型(Kinematics Model)，減少所需估測時間。

另一方面，利用估測到的系統輸出配合濾波器的設計，重新計算系統的輸入並與經過濾波器的系統輸入進行比較，將可在計算量降低的情況下準確地分辨此時的錯誤狀況，增加錯誤偵測(Fault Detection)的準確度。

當確定錯誤後，將進行錯誤識別(fault identification)，也就是切換到決定的錯誤模型分群中。在分群中，我們仍然將經由觀測器作再次的確定，以避免誤判造成的系統響應。

1.2 系統發生錯誤的情形

在研究錯誤偵測和隔離(Fault Detection and Identification)常見的文獻裡，依照系統錯誤發生的情況分成四大類[2]：

1. 感測器錯誤(Sensor faults)
2. 致動器錯誤(Actuator faults)
3. 系統處理程序錯誤(Process faults)
4. 閉迴路或控制器錯誤(Close loop or controller faults)

若依系統錯誤訊號發生的狀況分類[1]，則可分成以下兩類：

1. 突發性錯誤訊號(Abrupt or Sudden faults):顧名思義，此為系統輸出訊號產生大幅的跳動或是脈衝性的訊號，這代表著系統極有可能已經出錯或即將發生錯誤，必須儘快地做出相對應的措施，以排除錯誤。
2. 緩慢發生錯誤訊號(incipient or slowly developing faults):由字面上的意思，這是與突發性錯誤訊號(Abrupt or Sudden faults)相對的訊號。此種訊號代表系統慢慢變化，比方說，示波器使用一段時間後需要歸零。若該訊號不隨時間變化，則稱為定值偏差錯誤訊號(dc-offset or bias signal)；若訊號隨時間變化，則稱為漂移偏差錯誤訊號(time-varying or drift signal)。

由於本論文以機械臂馬達發生故障的情形，作為電腦模擬的依據，所以此篇論文裡的錯誤指的是致動器錯誤(Actuator faults)與突發性錯誤訊號(Abrupt or Sudden faults)。

1.3 論文貢獻

經由前人的努力，線性系統的偵測與識別系統已經相當完善。但相對的機械臂為一非線性系統，其在理論與實用必然不能直接套用許多前人的理論，例如極小的力矩誤差及有可能造成不可預料的後果，因此研究上必然也會碰到許多挑戰。本論文的貢獻就在於改進「錯誤偵測與識別演算法」，使能夠更快地偵測機械臂系統上的錯誤並進行識別。在第五章中，將可藉由模擬結果看到改良後的演算法在系統識別上比原有的演算法效果更精準，效率也更提高。

1.4 論文架構

本篇論文共分為六個章節：第一章為序論，主要說明研究動機與目的；第二章介紹機器人系統與文獻，對近期發表的論文進行整理；第三章介紹錯誤偵測相關與識別相關技術，針對估測器、錯誤模型，期望值最大化等演算法等作進一步的介紹；第四章進入設定的目標「計算減量」，闡述各種方法為達到此目標的各種方法；第五章將之前的模擬和推導在二軸機械臂的平台上進行模擬，探討模擬的結果與該結果形成的原因；最後，第六章對兩年來的結果做一結論並展望未來可以進行的研究方向，希望後進能在此一基礎上更上一層樓，造福世界人類。

第二章 機器人系統

廣義的來說，機器人(Robot)是一個模仿人類行為的機器裝置。機器人可接受人類指揮，也可執行預先編排的程式，也可根據以人工智慧技術制定的法則行動。機器人的用途在於取代或是協助人類的工作，例如製造業、建築業，或是危險的工作。機器人一詞，最早出現在西元 1920 年傑克科幻小說家卡雷爾·恰佩克(Karel Čapek)「羅索姆的萬能機器人」(Rossum's Universal Robots)一書中，原文為「Robota」為捷克語的「苦工、勞動者」的意思，後來成為通行的「Robot」。本章分為兩小節，第一小節介紹機器人系統，第二小節簡介機器人錯誤偵測與辨識的相關研究。



2.1 機器人系統簡介

機器人系統種類非常多，因為論文模擬平台為機械臂系統，故只針對機械臂系統進行簡介。

機械臂系統由動力來分，可分為液壓式和電動式；由用途可區分為工業機器人、玩具機器人、仿生機器人等。由於本篇論文主要是針對致動器的錯誤偵測與識別進行討論，故就動力加以區分。

1. 液壓式機器人

液壓式機器人一般為大型機器人，因為液壓驅動之力量較大，但相對來說其作業要求往往較不精確。大家熟知的液壓機械臂系統為挖土機，也就是俗稱的「怪手」。怪手的正式名稱為液壓挖土機，其挖掘方式為反鏟和正鏟。挖土機的挖掘臂設計以人的手臂為樣本，但其主臂相當長以使力矩增加，其有三個關節，以適應各種角度挖掘。其大臂與車相連處如肩關節；大臂與副臂相接處如肘關節；副臂與挖斗相接處如腕關節，挖斗如拱起之手掌，破壞齒如手指。

2. 電動式機器人

電動式機器人其驅動方式為電動馬達，因其機構易於設計，動力來源容易取得，故運用非常廣泛。其位置控制精確度可達 $1\mu\text{m}$ 以下，常用於半導體製程或精密加工等範圍。最新的應用則為醫療用途，舉達文西手術系統(Da Vinci Surgical System)為例：

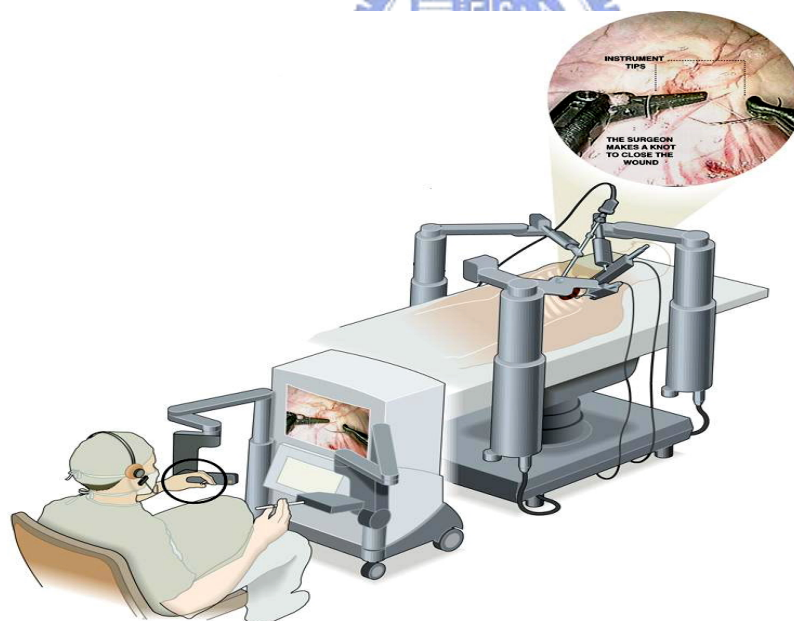


圖 2.1 達文西手術系統操作示意圖

這是一款由美國 Intuitive Surgical 公司研發的手術機器人，可用於開心、開腎、開肺等多種開刀用途，更可進行遠距醫療與同時接續四條血管，同時由於可以讓醫生坐著開刀，故可延長醫生的開刀壽命。

2.2 機器人錯誤偵測與識別相關研究

1. Fabrizio Caccavale 與 D.Walker[3]文中發表一套建構於離散時間的自我診斷錯誤系統，該系統將觀察器與實際量測的誤差值進行線性回授，再利用延遲非線性補償的方式，來提高機器人對於未知系統參數的強健度，而非將臨界門檻調高使得感測錯誤的靈敏度降低。另一方面，該系統在觀察器的設計上考慮系統離散化時的誤差，使觀察器能夠更確實地反應出系統狀況。

雖然延遲架構能增加系統的強健度，但因為其特性類似低通濾波器，能夠濾掉突變的訊號，但同時也會犧牲錯誤診斷的敏銳度。至於將觀察器設計考慮系統離散的誤差，雖然看似可行，但是在現實中，系統模型不可能完全與理論模型相同的情況下，未能完全解決系統離散化誤差的問題。



2. M.L. McIntyre ,W.E. Dixon ,D.M. Dawson 和 I.D. Walker 於[4]將機械臂錯誤情形分為致動器卡住(locked-joint fault)、不受力甩動(free-swing fault)、階梯錯誤(ramp fault)或是飽和錯誤(saturated fault)。文中發表如何經由預測值與實際值誤差和非線性觀測器與來進行轉軸致動器的錯誤偵測和識別，其主要貢獻在於利用濾波器使計算力矩不需使用到角加速度，因為若由角度作兩次差分或角速度作一次差分往往會因為雜訊的干擾而失真，造成力矩計算的極大誤差，因此利用濾波器濾掉角加速度來進行力矩計算與比較。此外由於觀察器不會受到控制器影響，故強健度較好，不容易受到雜訊干擾。

3. Jin-Ho Shin 與 Ju-Jang Lee 於 [5] 中提出適用於機器人致動器的錯誤偵測與強健的錯誤矯正控制，其討論錯誤發生為致動器錯誤並僅討論馬達自由甩動的情況，馬達卡住或飽和不在討論範圍內。

該論文首先將偵測與識別各別分為一個平台，兩個平台的方法都很直接。

錯誤偵測

由計算力矩控制器(Computed Torque Controller)事先給定軌道 q_{c_0} ，再用感測器去獲得實際的軌道 q ，若兩者相差 e_{c_0} 大於臨界值則錯誤發生。

錯誤識別

錯誤識別的平台也是檢測事先給定的系統錯誤的軌道 q_{c_i} 是否與感測器量測值相同，來確定哪個軸的致動器故障。若一 n 軸機械臂有 p 個軸故障，則事先給定的參考訊號須 C_p^n 個。

以 3 軸機械臂且有 1 軸故障為例，需要給定 3 個 ($C_1^3 = \frac{3!}{(3-1)! \times 1!}$)

錯誤軌道。

q ：量測軌道

q_{c_i} ：給定軌道

$$e_{c_i} = q - q_{c_i} \quad i=1,2,3 \quad (2.27)$$

分成以下三種者情形：

$\|e_{c_1}\| = 0$ ，第一軸錯誤發生。

$\|e_{c_2}\| = 0$ ，第二軸錯誤發生。

$\|e_{c_3}\| = 0$ ，第三軸錯誤發生。

4. De Luca, A.Mattone,R. 於 [7] 中提出一種基於廣義慣量 (generalized momenta)與經由 H_{∞} 理論設計狀態觀測器的機械臂轉軸錯誤偵測方法。該論文先將致動器發生錯誤的問題重新改寫為一針對線性系統設計觀察器的問題。因此設計一組識別器 (identifier) 代表一種錯誤狀況，每一個識別器由皆為線性濾波器，並會產生一個相對應的剩餘值訊號，來決定該錯誤發生與否。

雖然該論文最後進行實驗，的確可以決定系統錯誤的種類，但作者仍有提到這是依據量測雜訊 (measurement noise) 事先設定門檻 (threshold) 的結果，也就是門檻的設定將會影響錯誤偵測的結果。

5. Alexander B. Trunov 與 Marios M. Polycarpou於[8]提出一個應用於非線性多輸入多輸出系統 (non-linear multi-input multi-output) 的強健非線性錯誤診斷架構。這個錯誤診斷架構裡使用一組線上的近似器 (on-line approximator) 和適應性非線性濾波技術來估測錯誤的形式，整個架構先對非線性系統作一非線性近似 (例如使用類神經網路) 來估測系統錯誤發生並近似錯誤的種類，再對該錯誤進行識別和容忍 (accommodation)。有以上的論述，可見該架構可同時且獨立的估測系統狀態和出錯種類，且由於對系統做近似的關係，其計算量也比原始非線性系統觀察器來的小。

第二章大略介紹機器人相關的錯誤偵測相關研究後，在第三章將更深入地闡述各種錯誤偵測與識別技術，作為後續本論文演算法推導的基礎。

第三章 錯誤偵測與識別技術

基於模型的錯誤偵測和識別(Model-based FDI) 廣泛運用於各類型的動態系統，舉凡：航太、車輛、交通運輸、工業用機械臂等都可見其身影。基於模型的錯誤偵測和識別(Model-based FDI)利用理論系統和實際系統間的輸出不同來計算其剩餘值(Residuals)，以指出系統是否出錯。本章節會對其相關演變和背景知識作清楚的介紹，順序是先將整個的估測系統做一概略性的介紹，再分別對其各個區塊進行詳細的介紹。

第一小節介紹錯誤估測和識別系統的三大區塊，分別是：建立錯誤系統模型、剩餘值產生和剩餘值評估。建立錯誤系統模型是基於模型的錯誤偵測和識別(Model-based FDI)最重要的部份，在精確的數學模型下，系統錯誤才容易偵測和識別。剩餘值產生和評估則是一般論文主要專注的課題，這部份設計的好，將可補足模型精確度上的不足。第二小節專注於觀察器的設計，其對應到錯誤偵測與識別三大區塊的剩餘值產生的部份，觀察器的種類非常多，因為模擬對象為在高斯雜訊下的非線性系統，故選用非追蹤型卡曼濾波器(UKF, Unscented Kalman Filter)。第三小節將介紹多模型組態，這部分對應到剩餘值的評估，本小節介紹的多模型組態演算法為 GPB-1(General Pseudo Bayesian 1)與 GPB-2(General Pseudo Bayesian 2)演算法，這兩種演算法的原理是相同的，但是考量到模型估測的準確度，故本論文採用 GPB-2 演算法。最後一小節就如何增進模型的準確度，介紹期望值最大化演算法。在錯誤偵測與識別三大區塊中，最重要的就是模型的精確度，期望值最大化演算法將用於估測摩擦力係數，補償模型不精確的部份，因而增加錯誤偵測與識別的準確率。

3.1 基於模型的錯誤偵測和識別技術

基於 1.1.1 節的敘述，大概可以了解到基於模型的錯誤偵測和識別 (Model-based FDI) 是一個具有回授的技術，此技術比不基於模型的錯誤偵測和識別 (Model-free FDI) 有更好的效果。在這章節裡，將會對該種技術作更深入的探討。

基於模型的錯誤偵測和識別技術 (Model-based FDI)，大致可用一句話來做完整的描述：「事先給定系統錯誤的描述以及系統的數學模型，藉由萃取感測器量測之系統輸入與輸出，利用系統之數學模型來進行系統狀態的確認。」做法為將感測器量測之實際訊號和經過數學模型的估測訊號相減產生剩餘值 (Residual Generation)，再經由事先設定錯誤門檻 (fault threshold) 進行剩餘值評估 (Residual Evaluation)，確定是否有錯誤發生。



本論文採用的架構圖如下：

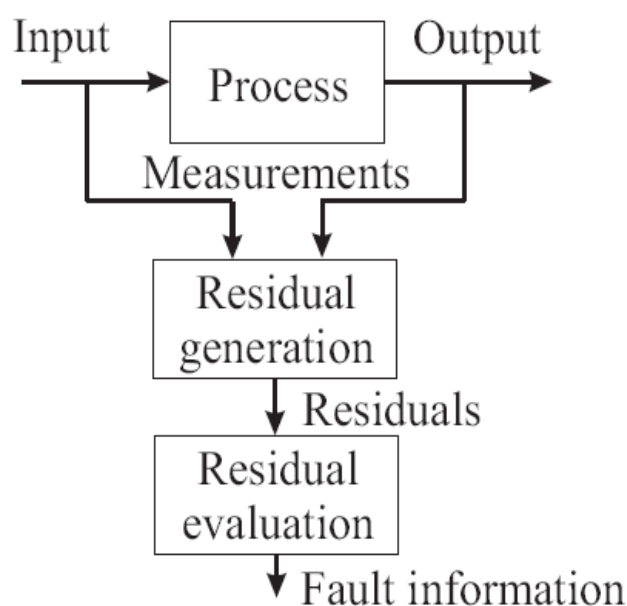


圖 3.1 錯誤偵測架構圖 [10]

該流程圖是由 chow 和 willsky [10] 於 1980 年提出的，目前已被廣泛

地用於錯誤診斷的領域。圖 3.1 可分為兩個主要步驟：剩餘值產生 (Residual Generation)與剩餘值評估 (Residual Evaluation)。

1. 剩餘值產生 (Residual Generation)：這個區塊利用待估測系統的輸入和輸出負責產生剩餘值，所計算出的剩餘值必須能夠某種程度的代表系統是否出錯。因此，剩餘值必須在正常運作的時候為零或接近零，而在系統出錯時遠大於零。理想的情況下，剩餘值的大小和系統的輸入和輸出呈獨立關係，只跟系統的狀態有關。
2. 剩餘值評估 (Residual Evaluation)：這個區塊對產生的剩餘值利用相似值 (Likelihood)或決定法則 (Decision Rule)進行評估，以確定系統是否真有錯誤發生。一般的作法是設定一個門檻 (threshold)對即時產生的剩餘值進行測試，也可對剩餘值進行移動平均 (Moving Averages)再進行測試。此外，也有一些利用統計方法的測試，例如：通用相似值比率測試 (generalized likelihood ratio testing)[19]和序列機率值比率測試 (sequential probability ratio testing)[20]。

在基於模型的錯誤偵測和識別 (model-based FDI)的領域裡的重要的貢獻大都分布在剩餘值產生 (Residual Generation)的部份，因為若產生的剩餘值能夠明確地表現系統的狀態，剩餘值評估 (Residual Evaluation)的部份將容易設計。

3.1.1 錯誤系統模型的建立

大略地了解架構後，此小節開始進行錯誤偵測的第一步：「建立錯誤系統模型。」

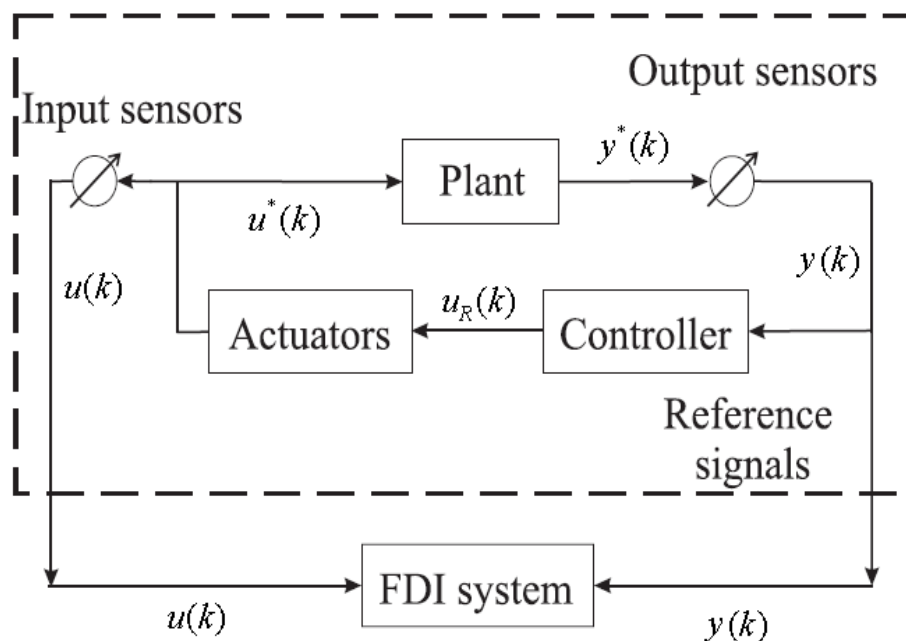


圖 3.2 閉迴路系統中的錯誤偵測[2]

圖 3.2 是一個結合錯誤偵測的閉迴路系統控制方塊圖，可見到受控廠的輸入 $u^*(k)$ 和輸出 $y^*(k)$ 分別經由輸入感測器(input sensor)和輸出感測器(output sensor)輸入錯誤偵測和識別(FDI)區塊進行處理。值得注意的是，控制器的設計對錯誤偵測有相當的影響，故做錯誤偵測時也必須考慮控制器的響應，過於強健的控制器有可能忽略系統的錯誤而使得錯誤診斷愈加困難。

假設控制器不會有錯誤發生且受控廠有錯誤發生的情況下，將圖 3.2 改畫為圖 3.3 以便將探討錯誤發生的範圍縮小。

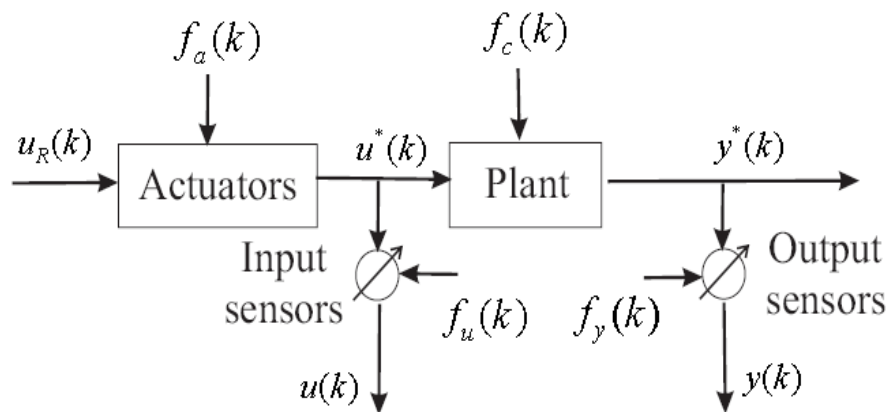


圖 3.3 簡化後的系統錯誤模型圖 [2]

圖 3.3 是一個簡易的系統錯誤模型圖，可見到假設的錯誤包括致動器錯誤、受控廠錯誤與感測器錯誤。假設此受控廠為一線性非時變 (LTI) 的離散系統，則其狀態方程式可以表示為 (3.1), (3.2)。

$$\begin{cases} x(k+1) = Ax(k) + Bu^*(k) & (3.1) \\ y^*(k) = Cx(k) & (3.2) \end{cases}$$



其中 $x(k) \in R^n$ 為系統狀態， $y^*(k) \in R^m$ 為系統輸出， $u^* \in R^r$ 為系統輸入， $A \in R^{n \times n}$ 為狀態轉移矩陣， $B \in R^{n \times r}$ 為系統輸入矩陣， $C \in R^{m \times n}$ 為系統輸出矩陣。

將受控廠錯誤 $f_c(k)$ 列入考慮，並假設其可用疊加訊號表示，則狀態方程式為 $x(k+1) = Ax(k) + Bu^*(k) + f_c(k)$ (3.3)

其中 $f_c(k) = [f_{c_1}(k) \ f_{c_2}(k) \ \dots \ f_{c_n}(k)]^T$

若不考慮感測器的動態響應，也可將感測器的量測誤差寫為疊加訊號，如 (3.4), (3.5)。(3.4) 與 (3.5) 這種表示有其實質上的好處，不但可以在處理訊號的時候統一進行處理，也可藉由等量加減消去的方式來推估錯誤的種類。

$$\begin{cases} u(k) = u^*(k) + f_u(k) & (3.4) \\ y(k) = y^*(k) + f_y(k) & (3.5) \end{cases}$$

其中 $f_u(k) = [f_{u_1}(k) \ f_{u_2}(k) \ \dots \ f_{u_r}(k)]^T$

$$f_y(k) = [f_{y_1}(k) \ f_{y_2}(k) \ \dots \ f_{y_n}(k)]^T$$

舉兩個藉由(3.4),(3.5)描述錯誤的例子：

例一、若輸入感測器的輸出因故障而固定在某個輸出值 \bar{u} ，
則錯誤可描述為 $f_u(k) = -u^*(k) + \bar{u}(k)$ 。

例二、若輸入感測器的輸出受到震動的影響使輸出值更動
為 $(1+\delta)u^*(k)$ ，錯誤可描述為 $f_u(k) = -u^*(k) + (1+\delta)u^*(k)$
 $= \delta u^*(k)$ 。

此外，致動器 (actuator) 的錯誤亦可以用疊加訊號表示為

$$u^*(k) = u_R(k) + f_a(k) \quad f_a(k) = [f_{a_1}(k) \ f_{a_2}(k) \ \dots \ f_{a_n}(k)]^T \quad (3.6)$$

一般所有的量測都會有量測誤差 v ，這個誤差同樣也可以用疊加訊號來表示，故最後感測器的完整錯誤模型描述為

$$\begin{cases} u(k) = u^*(k) + v_u(k) + f_u(k) & (3.7) \\ y(k) = y^*(k) + v_y(k) + f_y(k) & (3.8) \end{cases}$$

其中 $f_u(k) = [f_{u_1}(k) \ f_{u_2}(k) \ \dots \ f_{u_r}(k)]^T$ $f_y(k) = [f_{y_1}(k) \ f_{y_2}(k) \ \dots \ f_{y_n}(k)]^T$

$$v_u(k) = [v_{u_1}(k) \ v_{u_2}(k) \ \dots \ v_{u_r}(k)]^T \quad v_y(k) = [v_{y_1}(k) \ v_{y_2}(k) \ \dots \ v_{y_n}(k)]^T$$

整理(3.1)、(3.2)、(3.3)、(3.4)、(3.5)、(3.6)、(3.7)、(3.8)得到一組完整描述系統(圖 3.3)之方程式 (3.9)、(3.10)、(3.11)、(3.12)。

$$\begin{cases} x(k+1) = Ax(k) + Bu^*(k) + f_u(t) & (3.9) \end{cases}$$

$$\begin{cases} y(k) = Cx(k) + f_y(k) + v_y(k) & (3.10) \end{cases}$$

$$\begin{cases} u(k) = u^*(k) + f_u(k) + v_u(k) & (3.11) \end{cases}$$

$$\begin{cases} u^*(k) = u_R(k) + f_a(k) & (3.12) \end{cases}$$

3.1.2 剩餘值產生器

讀過 3.1.1 了解如何建立模型後，在錯誤偵測和識別架構中的下一個要了解的區塊是剩餘值產生器(Residual Generator)。

M. Basseville [11] 於 1988 年提出一連續時間的剩餘值產生器的通用架構(Residuals Generator General Structure)如圖 3.4 所示。

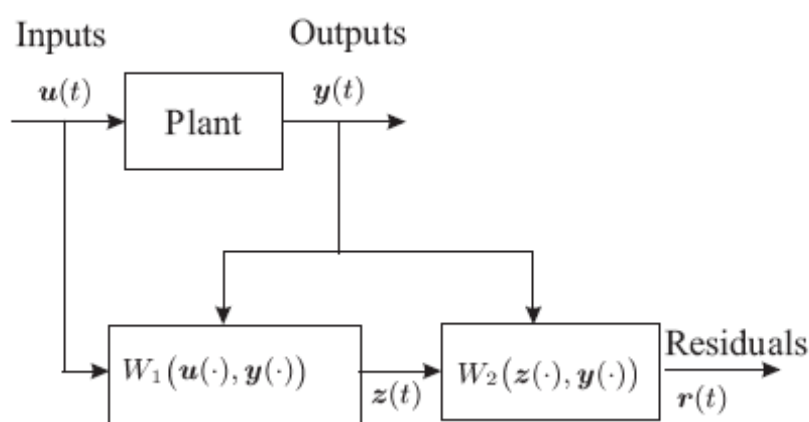


圖 3.4 剩餘值產生器的通用架構(M. Basseville) [11]

Fig 3.4 描述剩餘值 $z(t)$ 的產生方式為

$u(t)$:input $y(t)$:output $r(t)$:residuals

$$\begin{cases} z(t) = W_1(u(\cdot), y(\cdot)) & (3.13) \\ r(t) = W_2(z(\cdot), y(\cdot)) & (3.14) \end{cases}$$

當系統沒有錯誤發生的時候 $r(t) = 0$ ，反之 $r(t) \neq 0$ 。

圖 3.4 中 $W_1(t)$ 的最簡單的設計方法是將其用受控廠(plant)取代，也就是 $z(t) = W_1(u(\cdot))$ ，此時的 $z(t)$ 與 $y(t)$ 無關；我們可將 $W_1(u(\cdot))$ 視為待估測系統的模擬器(simulator)，此時設計 $r(t) = z(t) - y(t)$ 即可。

由於此為一個開迴路(open-loop)系統，故 $r(t)$ 可能發散。為了避免發散的情形，比較好的設計方法是將 $y(t)$ 列入考慮，例如：使用卡曼濾波器(Kalman Filter)等觀察器進行狀態估測。

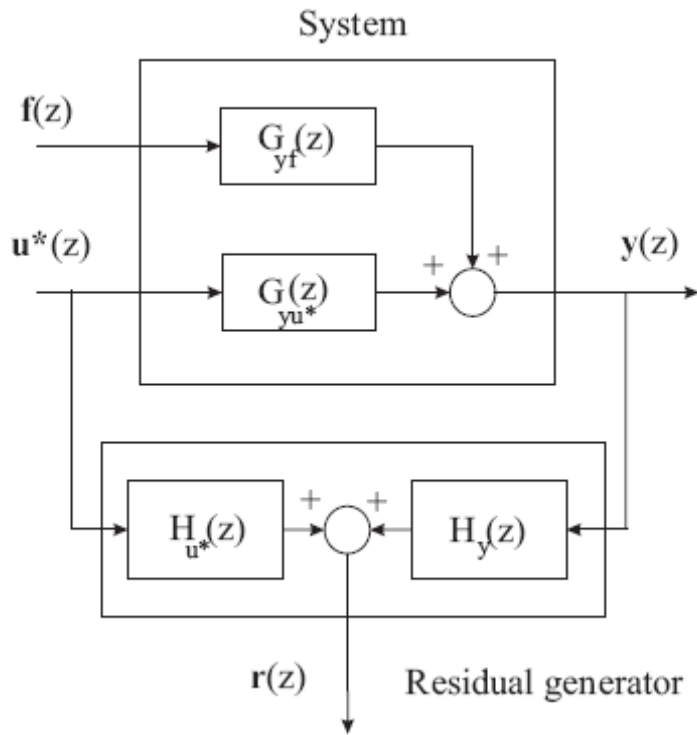


圖 3.5 剩餘值產生器的通用架構(Patton and Chen) [12]

另外一個通用的離散剩餘值產生器設計方法如 Fig 3.5 所示，這是一個由 Patton 和 Chen [12]於 1991 年提出的通用的剩餘值產生器(Residuals Generator)，此剩餘值產生器把錯誤對系統的響應用 $G_{yf}(z)$ 表示，並設計 $H_{u^*}(z)$ 和 $H_y(z)$ 來產生剩餘值。

方程式如下：

$$\begin{cases} y(z) = G_{yu^*}(z)u^*(z) + G_{yf}(z)f(z) \end{cases} \quad (3.15)$$

$$\begin{cases} r(z) = \begin{bmatrix} H_{u^*}(z) & H_y(z) \end{bmatrix} \begin{bmatrix} u^*(z) \\ y(z) \end{bmatrix} = H_{u^*}(z)u^*(z) + H_y(z)y(z) \end{cases} \quad (3.16)$$

基於剩餘值的物理意義，必須設計 $f(z)=0$ 時 $r(z)=0$ 為充分必要條件，


推導其充分必要條件如下：

$$\begin{aligned}r(z) &= H_u^*(z)u^*(z) + H_y(z)y(z) \\ &= H_u^*(z)u^*(z) + H_y(z)G_{yu^*}(z)u^*(z) \\ &= \left[H_u^*(z) + H_y(z)G_{yu^*}(z) \right] u^*(z) \\ &= 0\end{aligned}\tag{3.17}$$

$$\text{要讓(3.17)成立則 } H_u^*(z) + H_y(z)G_{yu^*}(z) = 0\tag{3.18}$$

方程式(3.18)可利用參數誤差設計的，相關的文獻可以參考Patton and Chen[12]以及Chen and Patton 1999[13]。

3.1.3 剩餘值評估



在3.1.1節了解了系統錯誤模型的建立，在3.1.2節了解了剩餘值產生器的大致流程，接著就是剩餘值評估的部份。一般用於判斷錯誤與否最簡單和廣泛使用的方法是建立一個錯誤門檻(fault threshold)，當剩餘值超過門檻的話就是有錯誤發生，反之則否。在某些情況下，甚至會對剩餘值再進行處理，使得錯誤偵測與識別的效果更好，以下舉「時間平均判定法」為例。

由於實際系統的輸出參雜有雜訊的部份，因此而使得產生的剩餘值不會為0，而為了去除雜訊最簡單的做法就是將剩餘值取平均值，此即為「時間平均判定法」。若雜訊為具有平均值為0的白雜訊(white noise)就可藉由時間平均(time average)濾掉雜訊而使錯誤偵測變得更準確；若平均值不為0，則需設定一個比此平均值大的臨界門檻來判定是否有錯誤發生。此外，如果直接設定錯誤門檻，將剩餘值取範數比較，則為「直接比較判定法」。

直接比較判定法

在文獻[14]中提到將剩餘值取範數(norm)，再藉由臨界門檻進行判定錯誤發生與否。

$\| \cdot \|$: 範數

$$\|r(t)\| \begin{cases} \geq \text{threshold, fault occurs} \\ < \text{threshold, no fault} \end{cases} \quad (3.19)$$

這種方法雖然可即時的進行錯誤檢測，但在檢測度較為不準，其臨界門檻設定至少必須比雜訊的標準差來的大，以避免將雜訊判定成系統錯誤發生。但是如此一來，當有比雜訊標準差小的錯誤發生時，就會發生誤判的情形。



時間平均判定法

$E\{ \cdot \}$: 平均值

$$E\{r(k)\} = \frac{r(k)+r(k-1)+\dots+r(k-m+1)}{m} \quad (3.20)$$

$$E\{r(k)\} \begin{cases} \geq \text{threshold, fault occurs} \\ < \text{threshold, no fault} \end{cases} \quad (3.21)$$

將剩餘值作時間平均，再與錯誤門檻進行比較。(3.20)為設定一個長度為m的取樣窗戶(sampling window)，對時間k至時間k-m+1的m個剩餘值取平均，(3.21)則是將該平均值與錯誤門檻比較評估是否有錯誤發生。

3.2 觀察器設計

在控制系統上，觀察器與受控廠的架構如圖3.6，其目的在於藉由給予實際系統的輸出和輸入來模擬實際系統而計算出實際系統的內部狀態。當獲得系統的內部狀態，則可進行控制器的設計，已達到我們希望的效果。

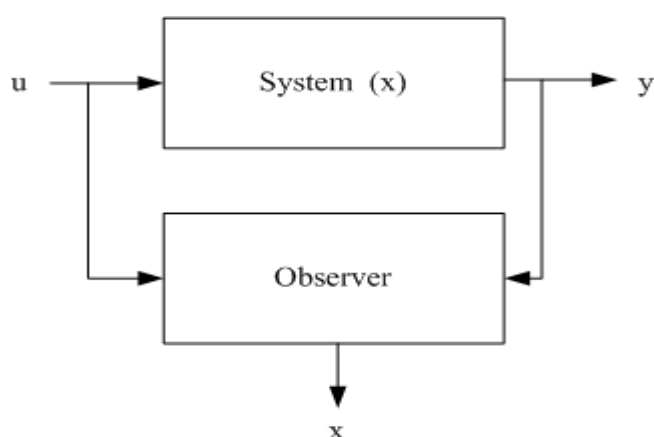


圖 3.6 觀察器作用示意圖

3.2.1 觀察器架構

假設控制廠為一線性系統狀態方程式如下，進行簡易的觀察器設計：

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) & (3.22) \\ y(k) = Cx(k) + Du(k) & (3.23) \end{cases}$$

$x(k)$: 實際系統時間 k 的內部狀態

$u(k)$: 實際系統時間 k 的輸入

$y(k)$: 實際系統時間 k 的輸出

A, B, C, D : 線性矩陣

最早的觀察器架構與想法為盧納柏格[15]提出的，故稱為盧納柏格觀察器(Luenberger observer)，其方法是設計一個矩陣 L ，以適應性的方式漸漸將估測器與實際輸出誤差值降低，其方程式如下：

$$\begin{cases} \hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L[y(k) - \hat{y}(k)] & (3.24) \\ \hat{y}(k) = C\hat{x}(k) + Du(k) & (3.25) \end{cases}$$

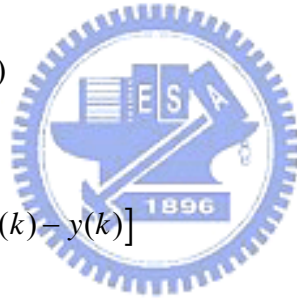
將(3.24)與(3.22)相減，得到估測與實際之間的誤差值 e ，(3.26)。

$$\begin{aligned} e(k+1) &= \hat{x}(k+1) - x(k) \\ &= \{A\hat{x}(k) + Bu(k) + L[y(k) - \hat{y}(k)]\} - \{Ax(k) + Bu(k)\} \end{aligned} \quad (3.26)$$

將(3.23),(3.25)代入(3.26)

$$\begin{aligned} e(k+1) &= \hat{x}(k+1) - x(k) \\ &= A[\hat{x}(k) - x(k)] - L[\hat{y}(k) - y(k)] \\ &= Ae(k) - L[C(\hat{x}(k) - x(k))] \\ &= (A - LC)e(k) \end{aligned} \quad (3.27)$$

由(3.27)，若矩陣 $A - LC$ 的特徵值的絕對值皆小於1，則此觀測器會收斂並穩定，可由此設計 L 矩陣。



3.2.2 卡曼濾波器

於1960年，卡曼(R.E. Kalman)發表了一篇探討線性濾波與估測問題的論文，在這篇論文中卡曼(R.E. Kalman)利用遞迴的方式來解決運算量過大的問題，並證明這是線性系統在高斯雜訊下的最佳解。由於正逢美蘇軍備競賽，因此卡曼濾波器在控制與導航領域中被大量研究與運用。

卡曼濾波器是線性系統在高斯雜訊(Gaussian noise)下的最小誤差平均平方(minimum mean square error)解。

考慮線性系統與加成型高斯雜訊如下：

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + Gv(k) & (3.28) \\ y(k) = Cx(k) + Du(k) + w(k) & (3.29) \end{cases}$$

A, B, C, D, E ：線性轉移矩陣

$u(k)$ ：時間 k 時的系統輸入

$x(k)$ ：時間 k 時的系統狀態

$y(k)$ ：時間 k 時的系統輸出

$v(k)$ ：時間 k 時的處理雜訊

$w(k)$ ：時間 k 時的量測雜訊

卡曼濾波器主要分為兩個主要步驟：

1. 狀態的預測：利用系統轉移矩陣與輸入，從此刻的狀態預測下個時間的狀態，也就是從 $\hat{x}(k|k)$ 預測 $\hat{x}(k+1|k)$ ，從 $P(k|k)$ 預測 $P(k+1|k)$ 。
2. 狀態的修正：利用系統輸出量測值修正之前的預測值，已得到更精確的估測，也就是將 $\hat{x}(k+1|k)$ 修正為 $\hat{x}(k+1|k+1)$ ，將 $P(k+1|k)$ 修正為 $P(k+1|k+1)$ 。

其中 $\hat{x}(k|k)$ 為經由 k 時間點的量測值估測得到的系統狀態估測值， P 為狀態估測值協方差矩陣 (state covariance matrix)，定義為 $P(k|k) = E\left\{\left[\hat{x}(k|k) - x(k)\right]\left[\hat{x}(k|k) - x(k)\right]^T\right\}$ ，並定義 $z(k)$ 為在 k 點的資訊 $(u(k), y(k))$ ， Z^k 為到第 k 點為止的資訊 $z(k)$ 。

整理卡曼濾波器演算法如下：

狀態估測

$$\hat{x}(k+1|k) = A\hat{x}(k|k) + Bu(k) \quad (3.30)$$

$$\begin{aligned} P(k+1|k) &= E\left\{\left[\hat{x}(k+1|k) - x(k+1)\right]\left[\hat{x}(k+1|k) - x(k+1)\right]^T\right\} \\ &= E\left\{A\left[\hat{x}(k|k) - x(k)\right]\left[\hat{x}(k|k) - x(k)\right]^T A^T + Gv(k)v^T(k)G^T\right\} \\ &= AP(k|k)A^T + GQ(k)G^T \end{aligned} \quad (3.31)$$

狀態修正

$$\begin{aligned} P(k+1|k+1) &= \left[P(k+1|k) + C^T R(k)^{-1} C\right]^{-1} \\ &= P(k+1|k) - P(k+1|k)C^T \left[R(k) + CP(k+1|k)C^T\right]^{-1} CP(k+1|k) \end{aligned} \quad (3.32)$$

$$L(k+1) = P(k+1|k+1)C^T R(k)^{-1} \quad (3.33)$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + L(k+1)[y(k) - C\hat{x}(k+1|k) + Du(k)] \quad (3.34)$$

$Q(k) = E\left[v(k)v^T(k)\right]$ 為處理雜訊的協方差矩陣

$R(k) = E\left[w(k)w^T(k)\right]$ 為量測雜訊的協方差矩陣

3.2.3 應用卡曼濾波器於非線性系統

基於卡曼濾波器的簡單、強健度與最佳化等性能，故其廣泛使用於線性系統的狀態追蹤或系統參數的估測，但是如果將卡曼濾波器應用在非線性領域則相當的困難。一般常用的作法是將非線性系統做一階線性化，再套入卡曼濾波器的步驟執行，也就是所謂的「拓展型卡曼濾波器」(Extended Kalman filter)。另一種作法，則是對系統進行「非蹤跡型轉換」(Unscented Transform)，這是一種類似蒙地卡羅(Monte-Carlo)取樣與質點濾波器(Particle Filter)的方法，將其套入卡曼濾波器則稱為「非蹤跡型卡曼濾波器」(Unscented Kalman filter)。



考慮非線性系統如下：

$$\begin{cases} x(k+1) = f[x(k), u(k), v(k), k] & (3.35) \\ y(k) = h[x(k), u(k), k] + w(k) & (3.36) \end{cases}$$

其中 f, h 皆為非線性轉換

$Q = E[v(k)v^T(k)]$ 為處理雜訊的協方差矩陣

$R = E[w(k)w^T(k)]$ 為量測雜訊的協方差矩陣

拓展型卡曼濾波器 (Extended Kalman Filter)

拓展型卡曼濾波器是做一階線性化，故將(3.35)式的非線性系統 f 作泰勒展開得到

$$\begin{aligned} f[x] &= f[\bar{x} + \delta x] \\ &= f[\bar{x}] + \nabla f \delta x + \frac{1}{2} \nabla^2 f \delta x^2 + \frac{1}{3!} \nabla^3 f \delta x^3 + \frac{1}{4!} \nabla^4 f \delta x^4 + \dots \end{aligned} \quad (3.37)$$

故

$$\bar{y} = f[\bar{x}] + \frac{1}{2} \nabla^2 f P_{xx} + \frac{1}{2} \nabla^4 f E[\delta x^4] + \dots \quad (3.38)$$

$$\begin{aligned} P_{yy} &= \nabla f P_{xx} (\nabla f)^T + \frac{1}{2 \times 4!} \nabla^2 f (E[\delta x^4] - E[\delta x^2 P_{yy}] - E[P_{yy} \delta x^2] + P_{yy}^2) (\nabla^2 f)^T \\ &\quad + \frac{1}{3!} \nabla^3 f E[\delta x^4] (\nabla^3 f)^T + \dots \end{aligned} \quad (3.39)$$

由(3.38)取 $\bar{y} = f[\bar{x}]$ 作線性近似，由(3.39)取 $P_{yy} = \nabla f P_{xx} (\nabla f)^T$ 作線性近似，接下來就是套入(3.30)、(3.31)、(3.32)、(3.33)與(3.34)進行計算。

非追蹤型卡曼濾波器 (Unscented Kalman Filter)

讓我們先來了解非追蹤型轉換 (Unscented Transform)，這是由 Simon J. Julier 與 Jeffrey K. Uhlmann[16]於1997年提出的。非追蹤型轉換 (Unscented Transform) 是一種計算隨機變數因非線性轉換時所產生的統計變化的方法，其最基本的概念在於「對輸入訊號作高斯轉換的近似比對待處理的非線性系統作近似來的容易」。由圖3.7[17]將可了解其與「拓展型卡曼濾波器」的差別。

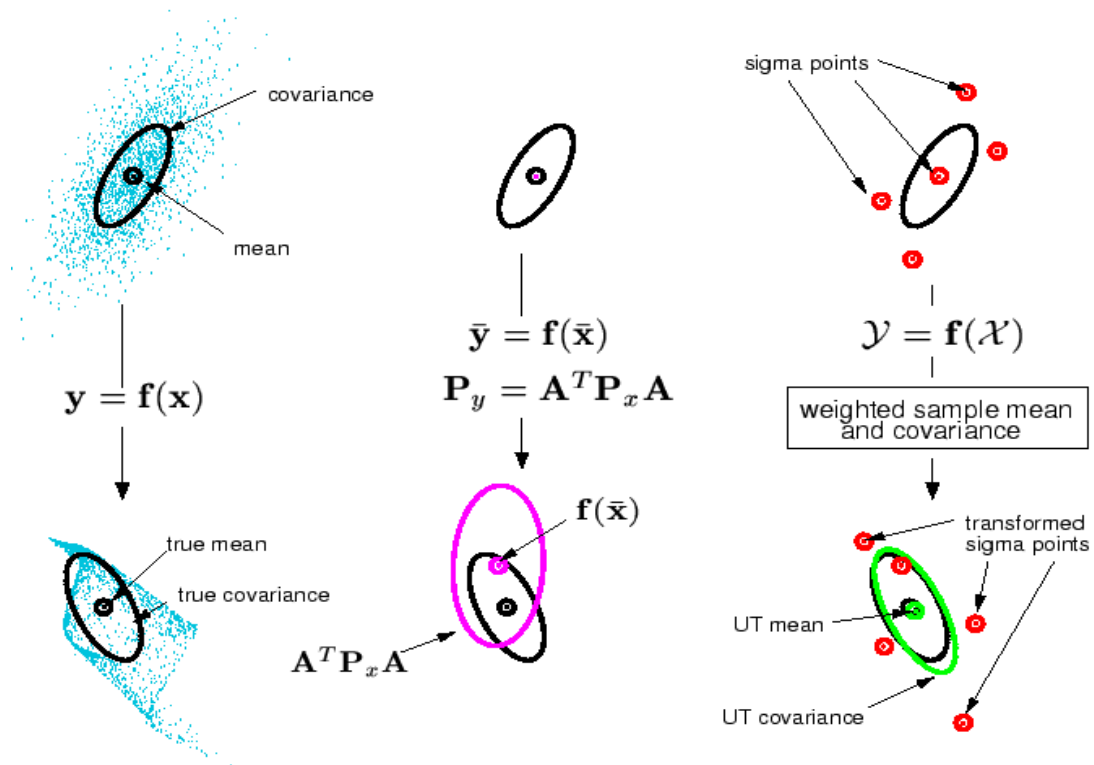


圖 3.7 蒙地卡羅、拓展型卡曼濾波器與非追蹤型卡曼濾波器估測比較

圖 3.7 左邊代表當取點很多的時候，可以準確的算出準確的平均和代表協方差的圓圈範圍；中間的圖則代表「拓展型卡曼濾波器」，也就是對非線性系統進行一階線性化，這時所算出來的平均和協方差的圓圈則大幅偏離真實的平均和協方差圓圈；右邊的圖為「非追蹤型卡曼濾波器」，這個方法是在原來的數值域取特定數量的點，在將其經過非線性轉換，並利用轉換後的點計算平均值和協方差，可發現其平均與協方差與實際的值相差不多。

公式整理

初始化

將待估測值狀態 x 與處理雜訊 v 和量測雜訊 w 組合成為一新的狀態

(3.40) 與新的協方差 (3.41)。

$$\hat{x}(0|0) = E[x(0|0)]$$

$$P(0|0) = E\left(\left[x(0|0) - \hat{x}(0|0)\right]\left[x(0|0) - \hat{x}(0|0)\right]^T\right)$$

$$\hat{x}^a(0|0) = E\left[x^a(0|0)\right] = \left[\hat{x}^T(0|0) \quad 0 \quad 0\right]^T \quad (3.40)$$

$$P^a(0|0) = E\left(\left[x^a(0|0) - \hat{x}^a(0|0)\right]\left[x^a(0|0) - \hat{x}^a(0|0)\right]^T\right) = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & P_v & 0 \\ 0 & 0 & P_w \end{bmatrix} \quad (3.41)$$

其中 P_v 為(3.35)中的 $v(k)$ 的協方差矩陣, P_w 為(3.36)中的 $w(k)$ 的協方差矩陣

取點與計算對應的權重

$\sqrt{(L+\lambda)P^a(k-1|k-1)}$ 可參照 [17] 使用 Cholesky 分解

$$x^a(k-1|k-1) = \left[\hat{x}^a(k-1|k-1) \quad \hat{x}^a(k-1|k-1) \pm \left(\sqrt{(L+\lambda)P^a(k-1|k-1)}\right)_i \right] \quad (3.42)$$

$$W_0^{(m)} = \frac{\lambda}{(L+\lambda)} \quad (3.43)$$

$$W_0^{(c)} = \frac{\lambda}{((L+\lambda) + (1-\alpha + \beta^2))} \quad (3.44)$$

$$W_i^{(m)} = W_i^{(c)} + \frac{1}{2(L+\lambda)} \quad (3.45)$$

其中 $i=0,1,\dots,2L+1$, L 為 P^a 的行或列的長度。

$\lambda = \alpha^2(L+\kappa) - L$ 為可調參數, α 通常設為極小的正數, 其代表所取的點與 \hat{x} 的分佈距離; β 用於合併之前關於 x 的資訊 (若 x 為高斯分佈, β 為 2 最佳); κ 則為次要的調整參數, 其值通常設為 0。詳細的調整方式可參照 [16], [17]。

預測

將取樣的點(3.42)輸入非線性系統(3.35),(3.36)得到(3.46),(3.47)。

$$\mathcal{X}^x(k|k-1) = f\left[\mathcal{X}^x(k-1|k-1), \mathcal{X}^v(k-1|k-1), u(k-1), k\right] \quad (3.46)$$

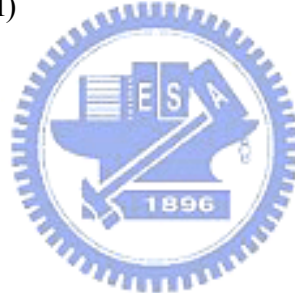
$$\mathcal{Y}(k|k-1) = h\left[\mathcal{X}^x(k|k-1), u(k-1), k\right] + \mathcal{X}^w(k-1|k-1) \quad (3.47)$$

使用(3.43),(3.44),(3.45),(3.46),(3.47)計算預測的狀態(3.48)和協方差(3.49)以及預測的輸出(3.50)。

$$\hat{x}(k|k-1) = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}_i^x(k|k-1) \quad (3.48)$$

$$P(k|k-1) = \sum_{i=0}^{2L} W_i^{(c)} \left[\mathcal{X}_i^x(k|k-1) - \hat{x}(k|k-1) \right] \left[\mathcal{X}_i^x(k|k-1) - \hat{x}(k|k-1) \right]^T \quad (3.49)$$

$$\hat{y}(k|k-1) = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_i(k|k-1) \quad (3.50)$$



修正

使用權重(3.43),(3.44),(3.45)與預測值(3.48),(3.49),(3.50)計算(3.51)和(3.52)。

$$P_{\bar{y}_k \bar{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} \left[\mathcal{Y}_i(k|k-1) - \hat{y}(k|k-1) \right] \left[\mathcal{Y}_i(k|k-1) - \hat{y}(k|k-1) \right]^T \quad (3.51)$$

$$P_{x_k y_k} = \sum_{i=0}^{2L} W_i^{(c)} \left[\mathcal{X}_i^x(k|k-1) - \hat{x}(k|k-1) \right] \left[\mathcal{Y}_i(k|k-1) - \hat{y}(k|k-1) \right]^T \quad (3.52)$$

使用(3.51),(3.52)計算改變的比例 \mathbf{K} (3.53)，並將(3.48)修正為(3.54)，(3.50)修正為(3.55)。

$$\mathbf{K} = P_{x_k y_k} P_{\bar{y}_k \bar{y}_k}^{-1} \quad (3.53)$$

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \mathbf{K} \left[y(k) - \hat{y}(k|k-1) \right] \quad (3.54)$$

$$P(k) = P(k|k-1) - \mathbf{K} P_{y_k y_k} \mathbf{K}^T \quad (3.55)$$

3.3 多模型組態估測

多模型組態估測的大體方法是先將系統依其特性分類為數個模型，這數個模型是互斥的且其機率和值為 1，因此在任意時間下，系統可用其某一模型加以描述。多模型估測的方式就是同時運行這數個模型，藉此比較其與實際系統輸出的差別，來分辨系統現在是由哪個模型描述的。

首先對各種多重模型(Multiple Model)下的名詞下定義。

定義：多重模型

當一個系統的行為是由超過一個且有限數量的模型來加以描述的話，這個系統就是一個多重模型系統。



多重模型又分成兩類：

第一類為靜態多重模型(Static Multiple Model)，第二類為動態多重模型(Dynamic Multiple Model)，定義如下：

定義：靜態多重模型

當進行估測時，模型之間沒有進行切換，則該系統為「靜態多重模型」。

定義：動態多重模型

當進行估測時，模型之間有進行切換，則該系統為「動態多重模型」。

以下再定義幾個變數，方便接下來的推導。

M_j : 第 j 個模型， $j = 1, 2, \dots, r$

$z(k)$: 第 k 個時間點時的資訊，在本論文中指的是受控場輸入量測與輸出量測

Z^k : 到第 k 個時間點時為止的所有資訊 ($z(0), z(1), \dots, z(k)$)

$\mu_j(k)$: 第 k 個時間點時，系統在第 j 個模型下的機率，故其為 $P\{M_j | Z^k\}$

$\hat{x}^j(k|k)$: 由 $z(k)$ 估測第 k 個時間點估測到的第 j 個模型的狀態

$P^j(k|k)$: 由 $z(k)$ 估測第 k 個時間點估測到的第 j 個模型的協方差

有了以上準備，接著開始進行推導。



靜態多重模型下的估測 (Static Multiple Model Estimator):

將第 k 個時間時，系統為模型 j 的機率 $\mu_j(k)$ ，利用貝氏定理 (Bayes' theorem) 與條件機率化簡如下：

$$\begin{aligned}
 \mu_j(k) &\equiv P\{M_j | Z^k\} = P\{M_j | z(k), Z^{k-1}\} \\
 &= \frac{p[z(k) | Z^{k-1}, M_j] P\{M_j | Z^{k-1}\}}{p[z(k) | Z^{k-1}]} \\
 &= \frac{p[z(k) | Z^{k-1}, M_j] P\{M_j | Z^{k-1}\}}{\sum_{i=1}^r p[z(k) | Z^{k-1}, M_i] P\{M_i | Z^{k-1}\}} \\
 &= \frac{p[z(k) | Z^{k-1}, M_j] \mu_j(k-1)}{\sum_{i=1}^r p[z(k) | Z^{k-1}, M_i] \mu_i(k-1)} \quad j = 1, \dots, r
 \end{aligned} \tag{3.56}$$

由最後的式子可看出 $\mu_j(k)$ 同時受到條件機率 $p[z(k)|Z^{k-1}, M_j]$ 、 $\mu_i(k-1)$ 與 $\mu_j(k-1)$ ， $i, j=1, 2, \dots, r$ 的影響。其中條件機率 $p[z(k)|Z^{k-1}, M_j]$ 是我們關心的，也是多模型組態偵測的重點。

$p[z(k)|Z^{k-1}, M_j]$ 代表到第 $k-1$ 的時間點時為止的所有資訊 Z^{k-1} 和經過第 k 個時間的第 j 個模型 $M_j(k)$ 下，第 k 個時間點所得到的輸出資訊為 $z(k)$ 的機率。尤其字面上來看其值越高代表系統在 k 時間是 M_j 的機率越高，因此其專有名詞為相似度(Likelihood)。

因為 UKF 是對該點進行高斯近似(Gaussian approximation)，令其相似值為 $\Lambda_j(k) \equiv p[z(k)|Z^{k-1}, M_j] = p[v_j(k)]$

$$= N[v_j(k); 0, S_j(k)] \quad (3.57)$$

其中 $v_j(k)$: innovation $S_j(k)$: covariance of innovation

看完了以上的名詞定義欲推導敘述，可以看出第 k 個時間點的狀態的機率分布函數為一 Gaussian Mixture 的形式，其狀態估測值為個別模型的輸出乘上個別模型的機率如下：

$$p[x(k)|Z^k] = \sum_{j=1}^r \mu_j(k) N[x(k); \hat{x}^j(k|k), P^j(k|k)]$$

$$\Rightarrow \hat{x}(k|k) = \sum_{j=1}^r \mu_j(k) \hat{x}^j(k|k) \quad (3.58)$$

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \{P^j(k|k) + [\hat{x}^j(k|k) - \hat{x}(k|k)][\hat{x}^j(k|k) - \hat{x}(k|k)]^T\} \quad (3.59)$$

流程方塊圖如下：

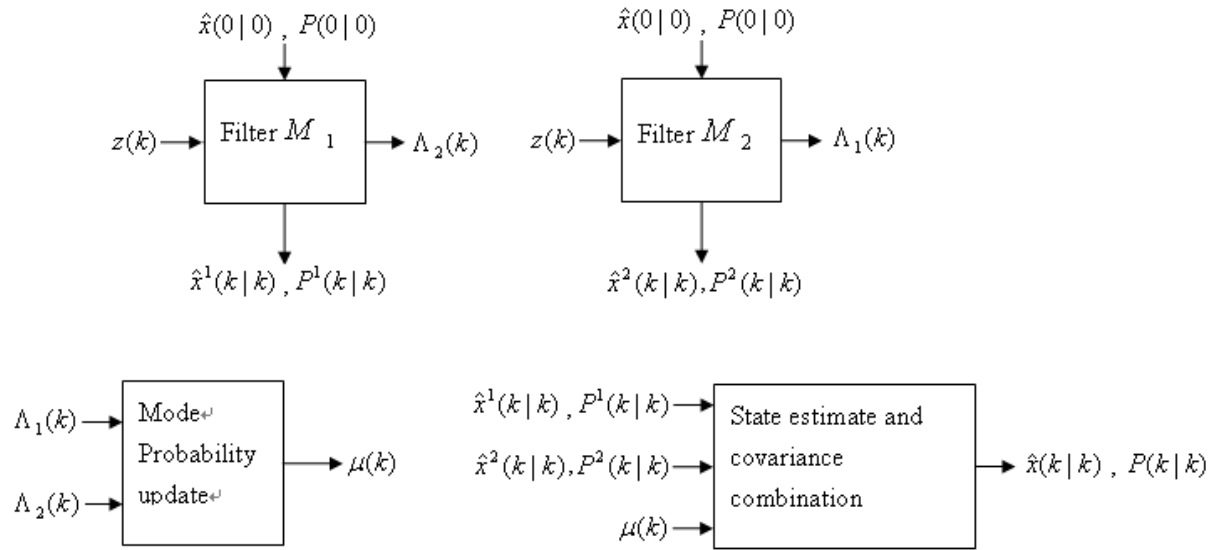


圖 3.8 靜態模型下估測流程方塊圖 [18]



動態多重模型下的估測 (Dynamic Multiple Model Estimator):

簡單來說，動態模型跟靜態模型的不同地方只在於估測時，系統有進行模型之間的切換，因此這類型的系統又稱為跳躍線性系統 (jump-linear systems)，建立動態模型方程式如 (3.60), (3.61)。

$$x(k) = F[M(k)]x(k-1) + v[k-1, M(k)] \quad (3.60)$$

$$z(k) = H[M(k)]x(k) + w[k, M(k)] \quad (3.61)$$

$F[M(k)]$, $H[M(k)]$ 為在 k 時間點的模型的系統轉移函數，若

$$M(k) \in \{M_j\}_{j=1}^r, \text{ 則 } F[M(k)] = F_j \quad j=1, 2, \dots, r, \quad H[M(k)] = H_j \quad j=1, 2, \dots, r$$

其中 F_j , $H_j \quad j=1, 2, \dots, r$ 皆為轉移函數

$v[k-1, M(k)]$, $w[k, M(k)]$ 為因為時間變動與不同模型下的處理雜訊和量測雜訊。

為了要套用靜態多重模型 (Static Multiple Model) 的公式，必須記錄每一個 model 在時間上的切換順序 (l th mode history through time k)，以作為推導的依據。

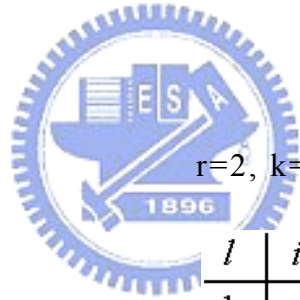
將到時間 k 的第 l 個模型歷史紀錄表示為

$M^{k,l} = \{M_{i_{1,l}}, \dots, M_{i_{k,l}}\}$ $l=1, \dots, r^k$ ， $i_{k,l}$ 為第 l 個模型歷史紀錄在第 k 個時間點對應的模型索引，其值為 $1 \sim r$ 。

舉兩個例子：

$r=2, k=2$; 則 $l=1,2,3,4$

l	$i_{1,l}$	$i_{2,l}$
1	1	1
2	1	2
3	2	1
4	2	2



$r=2, k=3$; 則 $l=1,2,\dots,8$

l	$i_{1,l}$	$i_{2,l}$	$i_{3,l}$
1	1	1	1
2	1	1	2
3	1	2	1
4	1	2	2
5	2	1	1
6	2	1	2
7	2	2	1
8	2	2	2

為了使推導簡化，假設模型切換過程為馬可夫鏈 (Markov Chain)，其中每一個模型之間切換的機率是已知，且只與上個時間的狀態有關，如 (3.62)。

$$\text{事前機率： } P\{M_j(k) | M^{k-l,s}\} = P\{M_j(k) | M_i(k-1)\} = p_{ij} \quad (3.62)$$

其中 i 為母數列 s 的最後一個模型

可以仿照靜態多重模型估測器 (Static Multiple Model Estimator) 的寫法，將給定 k 時間點之前的資訊估測 $x(k)$ 的機率化為每一個模型歷史紀錄的機率乘上其對應的相似值 (Likelihood)，如 (3.63)。

$$p[x(k) | Z^k] = \sum_{l=1}^{r^k} p[x(k) | M^{k,l}, Z^k] p\{M^{k,l} | Z^k\} \quad (3.63)$$

接下來就是分別計算 $p\{M^{k,l} | Z^k\}$ 與 $p[x(k) | M^{k,l}, Z^k]$ 。

$p\{M^{k,l} | Z^k\}$ 為第 k 個時間點，系統在第 l 個模型歷史下的機率，仿照之前 Static Model 下即為 $\mu^{k,l}$ 。

$$\begin{aligned} \mu^{k,l} &= P\{M^{k,l} | Z^k\} \\ &= P\{M^{k,l} | z(k), Z^{k-1}\} \\ &= \frac{1}{c} p[z(k) | M^{k,l}, Z^{k-1}] P\{M^{k,l} | Z^{k-1}\} \\ &= \frac{1}{c} p[z(k) | M^{k,l}, Z^{k-1}] P\{M_j(k), M^{k-1,s} | Z^{k-1}\} \\ &= \frac{1}{c} p[z(k) | M^{k,l}, Z^{k-1}] P\{M_j(k) | M^{k-1,s}, Z^{k-1}\} \mu^{k-1,s} \\ &= \frac{1}{c} p[z(k) | M^{k,l}, Z^{k-1}] P\{M_j(k) | M^{k-1,s}\} \mu^{k-1,s} \\ &= \frac{1}{c} p[z(k) | M^{k,l}, Z^{k-1}] P\{M_j(k) | M_i(k-1)\} \mu^{k-1,s} \\ &= \frac{1}{c} p[z(k) | M^{k,l}, Z^{k-1}] p_{ij} \mu^{k-1,s} \quad i, j = 1, 2, \dots, r \end{aligned} \quad (3.64)$$

c : normalization factor

令 $p[x(k)|M^{k,l}, Z^k]$ 為高斯分佈

$$p[x(k)|M^{k,l}, Z^k] = N[v_j(k); 0, S_j(k)]$$

其中 $v_j(k)$: *innovation* , $S_j(k)$: *covariance of innovation*

引入(3.64)，重寫(3.63)為(3.65)

$$\begin{aligned} p[x(k)|Z^k] &= \sum_{l=1}^{r^k} p[x(k)|M^{k,l}, Z^k] P\{M^{k,l} | Z^k\} \\ &= \sum_{l=1}^{r^k} p[x(k)|M^{k,l}, Z^k] \mu^{k,l} \end{aligned} \quad (3.65)$$

由(3.65)可看動態模型的估測會因為模型的歷史數量不停的增加，造成濾波器數量無止境的增加，因此演算法需要改進，才能進行實際的應用。



以下就是兩種改進後的演算法，其想法也都是基於把估測到的資訊統合起來做近似，使得估測計算量不會隨著時間指數增長，以便實際應用。

GPB-1(General Pseudo-Bayesian 1)

GPB-1 的方法是使用一層的濾波器，因此將其母數列用當時估測的平均值和協方差代替，使得歷史模型不會無限地增長。

配合這個概念，用 k 時間點之前的資訊估測 $x(k)$ 可近似如下：

$$p[x(k)|Z^k] = \sum_{j=1}^r p[x(k)|M_j(k), Z^k] P\{M_j(k) | Z^k\}$$

$$\begin{aligned}
&= \sum_{j=1}^r p \left[x(k) | M_j(k), z(k), Z^{k-1} \right] \mu_j(k) \\
&\approx \sum_{j=1}^r p \left[x(k) | M_j(k), z(k), x(k-1|k-1), P(k-1|k-1) \right] \mu_j(k) \quad (3.66)
\end{aligned}$$

上式的最後一行就是把 Z^k 用 $\hat{x}(k-1|k-1)$ 和 $P(k-1|k-1)$ 的高斯模型進行取代，也就有達到降低資料量的目的。

每一個 model 的估測值如下：

$$\hat{x}^j(k|k) = \hat{x}[k|k; z(k), M_j(k), \hat{x}(k-1|k-1), P(k-1|k-1)] \quad j=1, \dots, r \quad (3.67)$$

$$P^j(k|k) = P[k|k; M_j(k), P(k-1|k-1)] \quad j=1, \dots, r \quad (3.68)$$

接著計算模型機率

$$\begin{aligned}
\mu_j(k) &\triangleq P\{M_j(k) | Z^k\} \\
&= P\{M_j(k) | z(k), Z^{k-1}\} \\
&= \frac{1}{c} p \left[z(k) | M_j(k), Z^{k-1} \right] P\{M_j(k) | Z^{k-1}\} \\
&= \frac{1}{c} \Lambda_j(k) \sum_{i=1}^r P\{M_j(k) | M_i(k-1), Z^{k-1}\} P\{M_i(k-1) | Z^{k-1}\} \\
&= \frac{1}{c} \Lambda_j(k) \sum_{i=1}^r p_{ij} \mu_i(k-1) \quad (3.69)
\end{aligned}$$

c : normalization factor

將(3.66),(3.67),(3.68)和(3.69)套入(3.58),(3.59)得到(3.70),(3.71)

$$\hat{x}(k|k) = \sum_{j=1}^r \hat{x}^j(k|k) \mu_j(k) \quad (3.70)$$

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \left\{ P^j(k|k) + \left[\hat{x}^j(k|k) - \hat{x}(k|k) \right] \left[\hat{x}^j(k|k) - \hat{x}(k|k) \right]^T \right\} \quad (3.71)$$

流程方塊圖如下：

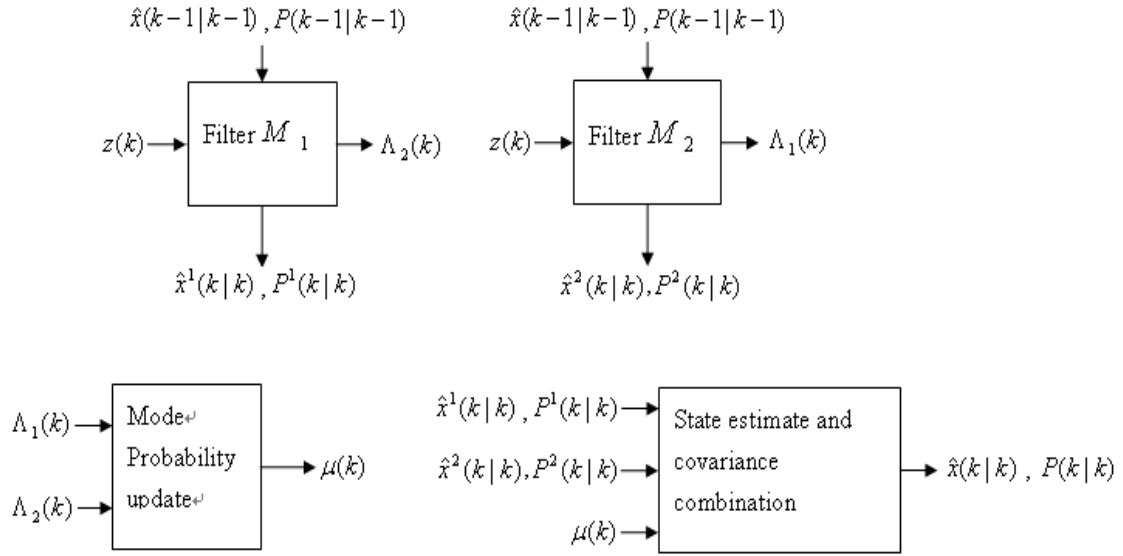


圖 3.9 GPB-1 流程方塊圖 [18]



GPB-2 (General Pseudo-Bayesian 2)

原理與 GPB-1 是一樣的，只是現在訊號經過兩層的濾波器來進行估測。由於兩層濾波器的架構，所以算出來的估測值會比 GPB-1 來的好。相對的，計算量也比 GPB-1 來的大。

$$\begin{aligned}
 p[x(k) | Z^k] &= \sum_{j=1}^r \sum_{i=1}^r p[x(k) | M_j(k), M_i(k-1), Z^k] P[M_i(k-1) | M_j(k), Z^k] \\
 &\quad \cdot P[M_j(k) | Z^k] \\
 &= \sum_{j=1}^r \sum_{i=1}^r p[x(k) | M_j(k), M_i(k-1), Z^k] \mu_{ij}(k-1|k) \mu_j(k) \\
 &\approx \sum_{j=1}^r \sum_{i=1}^r p[x(k) | M_j(k), M_i(k-1), \hat{x}^i(k-1|k-1), P^i(k-1|k-1)] \\
 &\quad \cdot \mu_{ij}(k-1|k) \mu_j(k)
 \end{aligned} \tag{3.72}$$

觀察(3.66),(3.72)，發現 GBP-2 有兩個連加符號，GBP-1 只有一個連加符號，故其所需計算量是 GBP-1 的平方倍。

每一個模型流(Model Sequence)中的每一個模型的估測：

$$\hat{x}^{ij}(k|k) \triangleq \hat{x}[k|k; M_j(k), \hat{x}^i(k-1|k-1), P^i(k-1|k-1)] \quad (3.73)$$

$$P^{ij}(k|k) \triangleq P[k|k; M_i(k), P^i(k-1|k-1)] \quad (3.74)$$

$$i, j = 1, 2, \dots, r$$

模型流整合機率(Mode Sequence Merging Probability)如下：

$$\begin{aligned} \mu_{ij}(k-1|k) &\triangleq P\{M_i(k-1) | M_j(k), Z^k\} \\ &= P\{M_i(k-1) | z(k), M_j(k), Z^{k-1}\} \\ &= \frac{1}{c_j} P[z(k), M_j(k) | M_i(k-1), Z^{k-1}] P\{M_i(k-1) | Z^{k-1}\} \\ &= \frac{1}{c_j} p[z(k) | M_j(k), M_i(k-1), Z^{k-1}] P\{M_j(k) | M_i(k-1), Z^{k-1}\} \\ &\quad \cdot P\{M_i(k-1) | Z^{k-1}\} \\ &= \frac{1}{c_j} \Lambda_{ij} p_{ij} \mu_i(k-1) \quad i, j = 1, 2, \dots, r \quad (3.75) \\ c_j &= \sum_{i=1}^r \Lambda_{ij}(k) p_{ij} \mu_i(k-1) \end{aligned}$$

算出每一個模型的估測值如下：

$$\hat{x}^j(k) = \sum_{i=1}^r \hat{x}^{ij}(k|k) \mu_{ij}(k-1|k) \quad j = 1, 2, \dots, r \quad (3.76)$$

$$P^j(k|k) = \sum_{i=1}^r \mu_{ij}(k-1|k) \left\{ P^{ij}(k|k) + [\hat{x}^{ij}(k|k) - \hat{x}^j(k|k)] [\hat{x}^{ij}(k|k) - \hat{x}^j(k|k)]^T \right\} \quad (3.77)$$

$$j = 1, 2, \dots, r$$

模型機率更新 (Mode Probability Update)

$$\begin{aligned}
 \mu_j(k) &\triangleq P\{M_j(k) | z(k), Z^{k-1}\} \\
 &= \frac{1}{c} P[z(k), M_j(k) | Z^{k-1}] \\
 &= \frac{1}{c} \sum_{i=1}^r P[z(k), M_j(k) | M_i(k-1), Z^{k-1}] P\{M_i(k-1) | Z^{k-1}\} \\
 &= \frac{1}{c} \sum_{i=1}^r p[z(k) | M_j(k), M_i(k-1), Z^{k-1}] P\{M_j(k) | M_i(k-1), Z^{k-1}\} \mu_i(k-1) \\
 & \quad j = 1, 2, \dots, r
 \end{aligned}$$

$$\text{整理為 } \mu_j(k) = \frac{1}{c} \sum_{i=1}^r \Lambda_{ij} p_{ij} \mu_i(k-1) = \frac{c_j}{c} \quad j = 1, 2, \dots, r \quad (3.78)$$

$$c = \sum_{j=1}^r c_j \quad (3.79)$$

最後計算新的狀態 (State) 和協方差 (Covariance)

$$\hat{x}(k | k) = \sum_{j=1}^r \hat{x}^j(k | k) \mu_j(k) \quad (3.80)$$

$$P(k | k) = \sum_{j=1}^r \mu_j(k) \left\{ P^j(k | k) + [\hat{x}^j(k | k) - \hat{x}(k | k)] [\hat{x}^j(k | k) - \hat{x}(k | k)]^T \right\} \quad (3.81)$$



GPB-2 流程方塊圖如下：

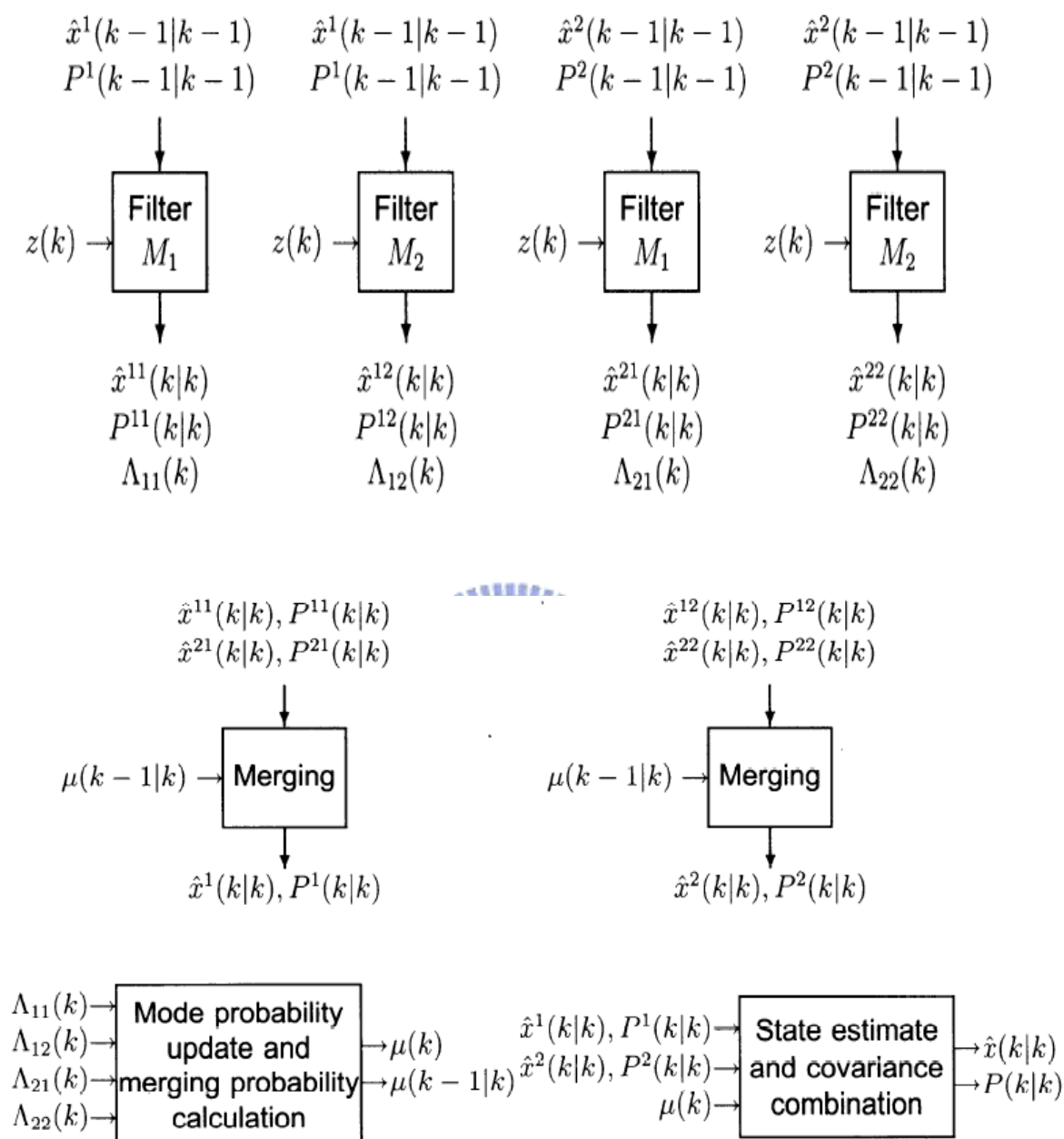


圖 3.10 GPB-2 估測流程方塊圖 [18]

3.4 期望值最大化

期望值最大化演算法(Expectation-Maximization algorithm)是在機率 (probabilistic) 模型中尋找參數最大似然估測(Maximum Likelihood Estimates)的演算法，其中機率模型依賴於無法觀測的隱藏參數 θ 。最大期望演算法經過兩個步驟交替進行計算，第一步是計算期望值(E)，方法是假設隱藏參數 θ 已知，然後計算相似值(Likelihood)的期望值；另外一步是最大化(M)，也就是計算會使 E 步上找到的相似值(Likelihood)的期望值最大的隱藏變數 θ 。接著再將 M 步上找到的隱藏參數 θ 用於下一個 E 步計算，這兩個步驟不斷交替進行，即可求得該機率模型下的準確參數。

令以下變數進行數學推導

z ：不完整的量測資訊

x ：無法獲得的資訊(隱藏參數)

θ ：隱藏變數

E-Step (估測無法獲得的資訊 z)

先利用貝氏定理計算給定不完整量測資訊 z 和隱藏變數 θ 時的機率密度函數 $p(x|z, \theta)$

$$p(x|z, \theta) = \frac{p(z, x|\theta)}{p(z|\theta)} = \frac{p(z|x, \theta)p(x|\theta)}{\int p(z|x, \theta)p(x|\theta)dx} \quad (3.82)$$

注意(3.82)的計算必須先知道 $p(x|\theta)$ 和 $p(z|x,\theta)$ 的機率函數，這代表我們必須先對系統的機率模型有一定的了解或假設。

接著估測 \hat{x} ， \hat{x} 為在 $p(x|z,\theta)$ 下的期望值

$$\hat{x} = E[x|z,\theta] = \int_x x \cdot p(x|z,\theta) dx \quad (3.83)$$

M-Step

定義在 θ 已知的情況下，使完整資訊 x 、 z 最大化的函數 $Q(\theta)$ 。

$$\begin{aligned} Q(\theta) &= E_z [\log p(z, \hat{x} | \theta) | z] \\ &= \int_{-\infty}^{\infty} p(\hat{x} | z, \theta_n) \log p(z, \hat{x} | \theta) d\hat{x} \end{aligned} \quad (3.84)$$

由於 E-step 已經估測到 \hat{x} ，代入(3.90)求得 $\theta_{n+1} = \arg \max_{\theta} Q(\theta)$ 。

再將 θ_{n+1} 代回(3.82),(3.83)執行 E-Step，彼此互相疊代，及為期望值最大化演算法。



第四章 系統計算減量

4.1 引入運動學方程式進行計算減量

待估測的機械臂系統如圖 4.1，其連續時間動力學方程式為 (4.1)，相關參數對照表(一)。

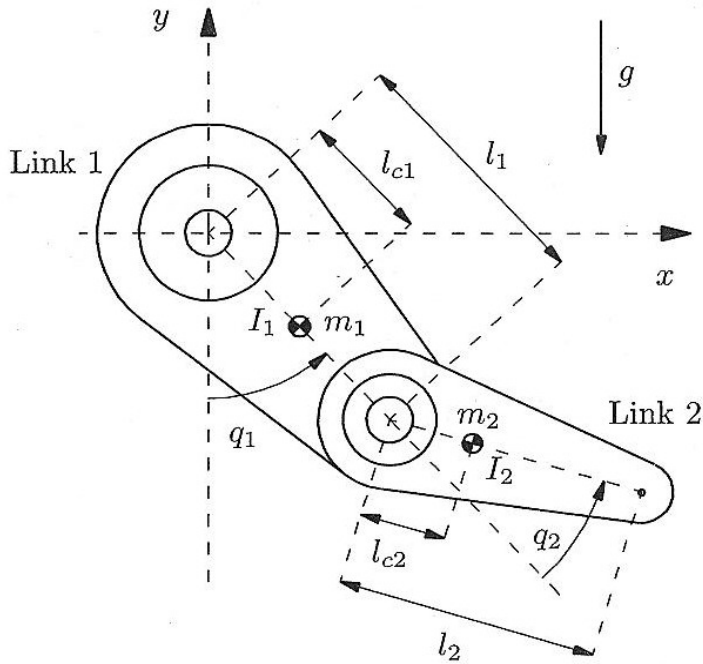


圖 4.1 機械臂示意圖[22]

機械臂動力學方程式(Dynamic Model)

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ M^{-1}(\theta) \times (-V(\theta, \dot{\theta}) - G(\theta) + \tau) \end{bmatrix} \quad (4.1)$$

將(4.1)作離算化得到離散時間方程式(4.2)與(4.3)。

$$\theta_k = \theta_{k-1} + h \times \dot{\theta}_{k-1} \quad (4.2)$$

$$\dot{\theta}_k = \dot{\theta}_{k-1} + h \times \left\{ M^{-1}(\theta_{k-1}) \times (-V(\theta_{k-1}, \dot{\theta}_{k-1}) - G(\theta_{k-1})) \right\} + h \times M^{-1}(\theta_{k-1}) \times \tau \quad (4.3)$$

其中 h 為取樣時間 (sampling time)， $\theta, \dot{\theta}$ 為 n 維的向量，表示各軸的角度、角速度。 τ 為 n 維的向量，表示各軸的致動器的力矩。 $M(\theta)$ 為機械臂的轉動慣量矩陣，維度為 $n \times n$ 。 $V(\theta, \dot{\theta})$ 、 $G(\theta)$ 都是 n 維的向量，分別代表向心力與科氏力的作用、重力的影響。

由於此項技術為針對智慧型機器人進行開發，成本必須盡量降低，因此無法要求智慧型機器人加裝額外的感測器或致動器給予錯誤診斷系統使用，故只裝設角度編碼器 (angle encoder)，在可倚靠的資訊來源受到限制下進行錯誤估測。在這種情況下，首先要進行的就是角速度的估測，再者是利用多模型組態對錯誤進行分類和選取。由於模擬系統對應的是兩軸機械臂，並將馬達分為兩種錯誤情形，**卡住**和**甩動**：卡住是指不管控制器指定馬達的力矩為多少，該軸維持固定角度；甩動是指不管控制器指定馬達的力矩為多少，該軸馬達輸出力矩皆為零，因此該軸會自由甩動。

先將(4.1)改寫為(4.4)與(4.5)，以方便待會表示不同錯誤模型的離散時間動力學方程式。

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} V_{11} \\ V_{21} \end{bmatrix} + \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

$$\Rightarrow \begin{cases} M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + V_{11} + G_{11} = \tau_1 \\ M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + V_{21} + G_{21} = \tau_2 \end{cases} \quad (4.4)$$

其中 $M_{11}, M_{12}, M_{21}, M_{22}$ 為 M 的內項， V_{11}, V_{21} 為 V 的內項， G_{11}, G_{21} 為 G 的內項， τ_1, τ_2 為 τ 的內項。

茲將兩軸機械臂的錯誤情況分成下列九種模型並表示其離散

動力學方程式：

模型一：第一軸正常、第二軸正常，離散動力學方程式為(4.2)與(4.3)。

模型二：第一軸卡住、第二軸正常。

$$\theta_{1(k)} = \theta_{1(k-1)}, \dot{\theta}_{1(k)} = 0$$

$$\theta_{2(k)} = \theta_{2(k-1)} + h \times \dot{\theta}_{2(k-1)}, \dot{\theta}_{2(k)} = \dot{\theta}_{2(k-1)} + h \times \left\{ M_{22}^{-1}(\theta_{k-1}) \times (\tau_2 - V_{21}(\theta_{k-1}, \dot{\theta}_{k-1}) - G_{21}(\theta_{k-1})) \right\}$$

模型三：第一軸正常、第二軸卡住

$$\theta_{1(k)} = \theta_{1(k-1)} + h \times \dot{\theta}_{1(k-1)}, \dot{\theta}_{1(k)} = \dot{\theta}_{1(k-1)} + h \times \left\{ M_{11}^{-1}(\theta_{k-1}) \times (\tau_1 - V_{11}(\theta_{k-1}, \dot{\theta}_{k-1}) - G_{11}(\theta_{k-1})) \right\}$$

$$\theta_{2(k)} = \theta_{2(k-1)}, \dot{\theta}_{2(k)} = 0$$

模型四：第一軸卡住、第二軸卡住

$$\theta_{1(k)} = \theta_{1(k-1)}, \dot{\theta}_{1(k)} = 0$$

$$\theta_{2(k)} = \theta_{2(k-1)}, \dot{\theta}_{2(k)} = 0$$

模型五：第一軸甩動、第二軸正常

$$\tau = [0 \quad \tau_2]^T \text{ 代入(4.2),(4.3)}$$

模型六：第一軸正常、第二軸甩動

$$\tau = [\tau_1 \quad 0]^T \text{ 代入(4.2),(4.3)}$$

模型七：第一軸甩動、第二軸甩動

$$\tau = [0 \quad 0]^T \text{ 代入(4.2),(4.3)}$$

模型八：第一軸卡住、第二軸甩動

$$\theta_{1(k)} = \theta_{1(k-1)}, \dot{\theta}_{1(k)} = 0$$

$$\theta_{2(k)} = \theta_{2(k-1)} + h \times \dot{\theta}_{2(k-1)}, \dot{\theta}_{2(k)} = \dot{\theta}_{2(k-1)} + h \times \left\{ M_{22}^{-1}(\theta_{k-1}) \times (0 - V_{21}(\theta_{k-1}, \dot{\theta}_{k-1}) - G_{21}(\theta_{k-1})) \right\}$$

模型九：第一軸甩動、第二軸卡住

$$\theta_{1(k)} = \theta_{1(k-1)} + h \times \dot{\theta}_{1(k-1)}, \dot{\theta}_{1(k)} = \dot{\theta}_{1(k-1)} + h \times \left\{ M_{11}^{-1}(\theta_{k-1}) \times (0 - V_{11}(\theta_{k-1}, \dot{\theta}_{k-1}) - G_{11}(\theta_{k-1})) \right\}$$

$$\theta_{2(k)} = \theta_{2(k-1)}, \dot{\theta}_{2(k)} = 0$$

由於二軸機系臂為非線性系統，故使用非追蹤型卡曼濾波器 (UKF Unscented Kalman Filter) 進行估測。其估測狀態為角度和角速度，皆為一 2×1 向量，處理雜訊 (Process Noise) 為 4×1 向量與量測雜訊 (Measurement Noise) 為 2×1 向量。

在多模型組態分類方面，由於將錯誤狀況分為 9 個模型，故如果使用 GPB-1 則需要 $9(9^1)$ 個估測器，GPB-2 則需要 $81(9^2)$ 個估測器，這對即時作業來講是非常大的負擔。而且隨著機系臂維度的增加，計算量將呈指數遞增 (exponential increase)，故計算減量在即時作業下是非常重要的。

設計方法是在這九種模型之上，引入運動學模型 (Kinematic Model)。由於運動學模型只需考慮角加速度為角速度微分，角速度為角度微分，而無須考慮受力的影響，故不受致動器錯誤影響。也因此只要角度量測器不發生故障，運動學模型估測的系統狀態 (角度、角速度) 就永遠是對的，將運動學模型 (4.6) 與動力學模型 (4.1), (4.2) 作為 GPB-2 多模型組態的兩個估測器，進行比較。因此不管是幾維的機械臂，皆只使用兩個估測器來判斷系統錯誤與否，以降低計算量。

運動學模型 (Kinematic Model)

$$\theta_k = \theta_{k-1} + h \times \dot{\theta}_{k-1} + w_{1(k)}$$

$$\begin{aligned}\dot{\theta}_k &= \frac{\theta_k - \theta_{k-1}}{h} + w_{2(k)} \\ &= \frac{\theta_{k-1} + h \times \dot{\theta}_{k-1} + w_{1(k)} - \theta_{k-1}}{h} + w_{2(k)} \\ &= \dot{\theta}_{k-1} + \frac{1}{h} w_{1(k)} + w_{2(k)}\end{aligned}$$

$$\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}_k = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}_{k-1} + \begin{bmatrix} 1 & 0 \\ 1/h & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_k \quad (4.6)$$

由上式看到運動學模型為線性方程式，故只需使用卡曼濾波器 (Kalman Filter)，也可降低計算量。

在估測的效果上，由於動力學模型 (Dynamic Model) 是將整隻機械臂的慣量、質量、質心位置等變數都納入考慮，故運動學模型 (Kinematic Model) 在系統正常作情況下其精確度不及動力學模型。但當有錯誤發生的時候，動力學模型會大幅的偏離正確的角度和角速度，而運動學模型雖然精確度較差，但卻始終跟著正確的角度和角速度的，所以此時運動學模型會主宰整個估測的輸出。

4.2 應用期望值最大化進行摩擦力估測

由於系統模型的精確於否，對於系統錯誤偵測與識別(Fault Detection & Identification)有著重大的影響，因此必須對影響系統精確度的摩擦力進行估測，以期能夠補償系統的不準確部份，使錯誤偵測與識別達到一較好的水平。由於機械臂系統中有兩個參數(角速度、摩擦力係數)未知，故使用期望值最大化演算法(Expectation Maximization Algorithm)進行估測。該演算法的特點在於當系統內有兩個未知參數時，在期望值和最大化兩個步驟中皆假定其中一個參數已知，並不斷交替進行這兩個步驟以求得兩個準確的參數，本小節則將其應用在摩擦力的估測上。本論文假設的摩擦力為庫倫摩擦力與黏滯摩擦力，黏滯摩擦力為黏滯摩擦力係數乘上角速度，庫倫摩擦力為庫倫摩擦力係數乘上角速度的方向性，在此摩擦力係數皆假設為常數。兩者在 EM 演算法的推導上都是一樣的，以下皆以庫倫摩擦力為推導對象。

期望值最大化演算法的目標是對 $\log p(x, z | \lambda)$ 作最大化，其中 z 為量測資訊， x 為非量測資訊， λ 為待定系統參數。以本論文模擬平台二軸機械臂而言， x 為角速度， z 為角度 y 與輸入力矩 u ， λ 為摩擦力係數。

為了版面方便將符號重新定義如下： $z(k) \rightarrow z_k$ 、 $Z^k \rightarrow z_{1:k}$ 、 $\hat{x}(k|k) \rightarrow \hat{x}_{k|k}$ 、 $\hat{x}(k-1|k) \rightarrow \hat{x}_{k-1|k}$ 、 $\{\hat{x}(1|k), \hat{x}(2|k), \hat{x}(k|k)\} \rightarrow \hat{x}_{1:k}$ 、 $y(k) \rightarrow y_k$ 、 $\{y(1), y(2), \dots, y(k)\} \rightarrow y_{1:k}$ 、 $u(k-1) \rightarrow u_k$ 。

其中 E-Step 的部份為假設 λ 已知，量測獲得 z 估測 x 。這部份可由非追蹤型卡曼濾波器(UKF)進行，如(4.7), (4.8)。

$$\hat{x}_{k|k} \leftarrow E(x_k | z_{1:k}) = \hat{x}(k|k) \quad (4.7)$$

$$\hat{x}_{k-1|k} \leftarrow E(x_{k-1} | z_{1:k}) = \hat{x}(k-1|k) \quad (4.8)$$

接著進行 M-step，也就是相似值最大化的部份。

首先將相似值 $p(x_{1:k}, z_{1:k} | \lambda)$ 拆解開來，再對系統參數 λ 微分求其最大值。

$$\begin{aligned} & p(x_{1:k}, z_{1:k} | \lambda) \\ &= p(z_k | x_{1:k}, z_{1:k-1}, \lambda) p(x_{1:k}, z_{1:k-1} | \lambda) \\ &= p(z_k | x_{1:k}, z_{1:k-1}, \lambda) p(x_k | x_{1:k-1}, z_{1:k-1}, \lambda) p(x_{1:k-1}, z_{1:k-1} | \lambda) \end{aligned} \quad (4.9)$$

觀察(4.9)發現該式具有遞迴的特性，但是 x ， z 與 λ 之間的關係並沒有顯然的函數描述。引入狀態空間模型(SSM, State Space Model)，將有助於釐清此一問題。

狀態空間模型(State Space Model)如下：

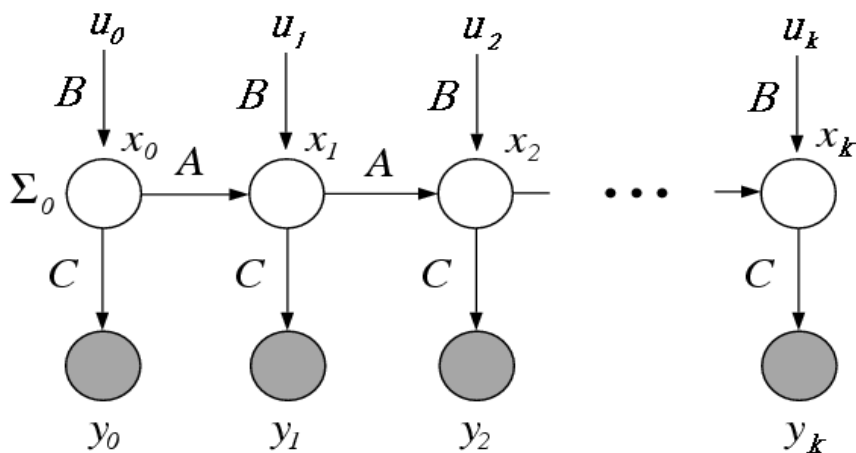


圖 4.2 狀態空間模型圖

觀察圖 4.2 發現有以下兩個性質

性質 1: 由於 $y_k = Cx_k$ ，且矩陣 C 已知。故知道每一個時間的 x 就可知道每一個時間的 y 。

性質 2: 因為 $x_k = Ax_{k-1} + Bu_{k-1}$ ，且矩陣 A, B 已知。故知道現在時間的 x 和 u 就能知道下一個時間的 x 。

將(4.9)利用狀態空間模型(SSM, State Space Model)性質(1),(2)

$$\begin{aligned}
 & p(x_{1:k}, z_{1:k} | \lambda) \\
 &= p(z_k | x_{1:k}, z_{1:k-1}, \lambda) p(x_k | x_{1:k-1}, z_{1:k-1}, \lambda) p(x_{1:k-1}, z_{1:k-1} | \lambda) \\
 &= p(z_k | x_k, \lambda) p(x_k | x_{k-1}, \lambda) p(x_{1:k-1}, z_{1:k-1} | \lambda)
 \end{aligned} \tag{4.10}$$

(4.10)利用遞迴特性，並將(4.7),(4.8)代入得到(4.11)

$$\begin{aligned}
 & p(x_{1:k}, z_{1:k} | \lambda) \\
 &= p(z_k | x_k, \lambda) p(x_k | x_{k-1}, \lambda) p(x_{1:k-1}, z_{1:k-1} | \lambda) \\
 &= p(z_k | x_k, \lambda) p(x_k | x_{k-1}, \lambda) \cdots p(z_0 | x_0, \lambda) p(x_0 | x_{-1}, \lambda) \\
 &= p(z_k | \hat{x}_{k|k}, \lambda) p(\hat{x}_{k|k} | \hat{x}_{k-1|k}, \lambda) \cdots p(z_0 | \hat{x}_0, \lambda) p(\hat{x}_0 | \hat{x}_{-1}, \lambda)
 \end{aligned} \tag{4.11}$$

因為 UKF 會將每一個時間點的估測近似為高斯分佈且 $z_k = (u_k, y_k)$ ，而 $\hat{x}_{k|k}$ 已經隱含了 u_k 的資訊，又處理雜訊(Process Noise(Q))和量測雜訊(Measurement Noise(R))也為高斯分佈表示為(4.12),(4.13)。

$$p(z_k | \hat{x}_{k|k}, \lambda) = p(y_k | \hat{x}_{k|k}, \lambda) = N(C \hat{x}_{k|k}, R) \tag{4.12}$$

$$p(\hat{x}_{k|k} | \hat{x}_{k-1|k}, \lambda) \approx N(A \hat{x}_{k-1|k}, \tau_{k-1}, \lambda), Q \tag{4.13}$$

A 為機械臂的狀態轉移方程式

將(4.12),(4.13)代回(4.11)得到(4.14)

$$\begin{aligned}
& p(x_{1:k}, z_{1:k} | \lambda) \\
&= p(z_k | \hat{x}_{k|k}, \lambda) p(\hat{x}_{k|k} | \hat{x}_{k-1|k}, \lambda) \cdots p(z_0 | \hat{x}_0, \lambda) p(\hat{x}_0 | \hat{x}_{-1}, \lambda) \\
&\approx N(C \hat{x}_{k|k}, R) \cdot N(A(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda), Q) \cdots N(C \hat{x}_{0|k}, R) \cdot N(A(\hat{x}_{-1|k}, \tau_{-1}, \lambda), Q) \quad (4.14)
\end{aligned}$$

將 Q 中的雜訊項假設為非相關，故 Q 為對角矩陣 $\begin{bmatrix} q_{11} & 0 & 0 & 0 \\ 0 & q_{22} & 0 & 0 \\ 0 & 0 & q_{33} & 0 \\ 0 & 0 & 0 & q_{44} \end{bmatrix}$ 。

(4.14)為多個高斯分佈相乘，對其取對數 \ln 則為多個平方式相加。

離散機械臂方程式 A 如下：

$$\begin{aligned}
\frac{d}{dt} \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix} &= \begin{bmatrix} \dot{\theta} \\ M^{-1}(\theta) \times (-V(\theta, \dot{\theta}) - G(\theta) + \tau') \end{bmatrix} \\
\Rightarrow \theta_k &= \theta_{k-1} + h \times \dot{\theta}_{k-1} \quad (4.15)
\end{aligned}$$

$$\begin{aligned}
\dot{\theta}_k &= \dot{\theta}_{k-1} + h \times \left\{ M^{-1}(\theta_{k-1}) \times (-V(\theta_{k-1}, \dot{\theta}_{k-1}) - G(\theta_{k-1})) \right\} + h \times M^{-1}(\theta_{k-1}) \times \tau' \\
&= \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}_{k-1} + h \times \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}_{k-1} \times \begin{bmatrix} \tau_{1(k-1)} - f_1 \operatorname{sgn}(\dot{\theta}_{1(k-1)}) \\ \tau_{2(k-1)} - f_2 \operatorname{sgn}(\dot{\theta}_{2(k-1)}) \end{bmatrix} \quad (4.16)
\end{aligned}$$

$$\text{其中 } \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \dot{\theta}_{k-1} + h \times \left\{ M^{-1}(\theta_{k-1}) \times (-V(\theta_{k-1}, \dot{\theta}_{k-1}) - G(\theta_{k-1})) \right\}$$

有了(4.14),(4.15),(4.16)，並假設摩擦力假設為 $f_i = f_1 \operatorname{sgn}(\dot{\theta}_1)$,

$f_2 = f_2 \operatorname{sgn}(\dot{\theta}_2)$ ，需要估測的值為摩擦力係數 f_1, f_2 ，就可以進行一個點的

最大相似值(Likelihood)的推導，之後再推廣至 k 個點即可。

由於(4.14)的 $p(z_k | \hat{x}_{k|k}, \lambda)$ 為 $N(C \hat{x}_{k|k}, R)$ ，與 λ 無關，故在微分的時候不必納入計算。

$$\begin{aligned}
& \ln p(\hat{x}_{k|k} | \hat{x}_{k-1|k}) \\
& \approx \ln N(F(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda), Q) \\
& = \ln \left\{ \frac{1}{(2\pi)^{n/2} \cdot \det(Q)^{1/2}} \exp \left[-\frac{1}{2} (\hat{x}_{k|k} - A(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda))^T Q^{-1} (\hat{x}_{k|k} - A(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda)) \right] \right\} \\
& = -\frac{n}{2} \ln[2\pi] - \frac{1}{2} \ln[\det(Q)] - \frac{1}{2} [\hat{x}_{k|k} - A(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda)]^T Q^{-1} [\hat{x}_{k|k} - A(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda)] \\
& (4.17)
\end{aligned}$$

將(4.15),(4.16)代入(4.17)作 f_1 的偏微分，得到(4.18)

$$\begin{aligned}
& \frac{\partial}{\partial f_1} \ln p(\hat{x}_{k|k} | \hat{x}_{k-1|k}) \\
& \approx \frac{\partial}{\partial f_1} \left(-\frac{1}{2} [\hat{x}_{k|k} - A(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda)]^T Q^{-1} [\hat{x}_{k|k} - A(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda)] \right) = 0 \quad (4.18)
\end{aligned}$$

其中

$$\begin{aligned}
\hat{x}_{k|k} & = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}_{k|k} \\
A(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda) & = \begin{bmatrix} \theta_{k-1|k} + h \times \dot{\theta}_{k-1|k} \\ \left\{ \begin{array}{l} \dot{\theta}_{k-1|k} + h \times \left\{ M^{-1}(\theta_{k-1|k}) \times (-V(\theta_{k-1|k}, \dot{\theta}_{k-1|k}) - G(\theta_{k-1|k})) \right\} \\ + h \times M^{-1}(\theta_{k-1|k}) \times \begin{bmatrix} \tau_1 - f_1 \operatorname{sgn}(\dot{\theta}_1) \\ \tau_2 - f_2 \operatorname{sgn}(\dot{\theta}_2) \end{bmatrix}_{k-1|k} \end{array} \right\} \end{bmatrix}
\end{aligned}$$

由於 f_1 只跟加速度項有關，將(4.18)化簡如下：

$$\frac{\partial}{\partial f_1} \left\{ \begin{array}{l} -\frac{1}{2} \left(\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}_{k|k} - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}_{k-1|k} - h \times \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}_{k-1|k} \times \begin{bmatrix} \tau_1 - f_1 \operatorname{sgn}(\dot{\theta}_1) \\ \tau_2 - f_2 \operatorname{sgn}(\dot{\theta}_2) \end{bmatrix}_{k-1|k} \right)^T \\ \begin{bmatrix} q_{33} & 0 \\ 0 & q_{44} \end{bmatrix}^{-1} \left(\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}_{k|k} - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}_{k-1|k} - h \times \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}_{k-1|k} \times \begin{bmatrix} \tau_1 - f_1 \operatorname{sgn}(\dot{\theta}_1) \\ \tau_2 - f_2 \operatorname{sgn}(\dot{\theta}_2) \end{bmatrix}_{k-1|k} \right) \end{array} \right\} = 0 \quad (4.19)$$

同理可得

$$\frac{\partial}{\partial f_2} \left\{ \begin{array}{l} -\frac{1}{2} \left(\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}_{k|k} - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}_{k-1|k} - h \times \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}_{k-1|k} \times \begin{bmatrix} \tau_1 - f_1 \operatorname{sgn}(\dot{\theta}_1) \\ \tau_2 - f_2 \operatorname{sgn}(\dot{\theta}_2) \end{bmatrix}_{k-1|k} \right)^T \\ \begin{bmatrix} q_{33} & 0 \\ 0 & q_{44} \end{bmatrix}^{-1} \left(\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}_{k|k} - \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}_{k-1|k} - h \times \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}_{k-1|k} \times \begin{bmatrix} \tau_1 - f_1 \operatorname{sgn}(\dot{\theta}_1) \\ \tau_2 - f_2 \operatorname{sgn}(\dot{\theta}_2) \end{bmatrix}_{k-1|k} \right) \end{array} \right\} = 0 \quad (4.20)$$

將(4.19)，(4.20)內項乘開解聯立方程式，得(4.21),(4.22)

$$q_{33}^{-1} \left\{ \begin{array}{l} \left[\begin{array}{l} \dot{\theta}_{1(k|k)} - c_{1(k-1|k)} - hw_{11(k-1|k)}\tau_1 - hw_{12(k-1|k)}\tau_2 \\ +hw_{11(k-1|k)}f_1 \operatorname{sgn}(\dot{\theta}_{1(k-1|k)}) + hw_{12(k-1|k)}f_2 \operatorname{sgn}(\dot{\theta}_{2(k-1|k)}) \end{array} \right] hw_{11(k-1|k)} \end{array} \right\} + \\ q_{44}^{-1} \left\{ \begin{array}{l} \left[\begin{array}{l} \dot{\theta}_{2(k|k)} - c_{2(k-1|k)} - hw_{21(k-1|k)}\tau_1 - hw_{22(k-1|k)}\tau_2 \\ +hw_{21(k-1|k)}f_1 \operatorname{sgn}(\dot{\theta}_{1(k-1|k)}) + hw_{22(k-1|k)}f_2 \operatorname{sgn}(\dot{\theta}_{2(k-1|k)}) \end{array} \right] hw_{21(k-1|k)} \end{array} \right\} = 0 \quad (4.21)$$

$$q_{33}^{-1} \left\{ \begin{array}{l} \left[\begin{array}{l} \dot{\theta}_{1(k|k)} - c_{1(k-1|k)} - hw_{11(k-1|k)}\tau_1 - hw_{12(k-1|k)}\tau_2 \\ +hw_{11(k-1|k)}f_1 \operatorname{sgn}(\dot{\theta}_{1(k-1|k)}) + hw_{12(k-1|k)}f_2 \operatorname{sgn}(\dot{\theta}_{2(k-1|k)}) \end{array} \right] hw_{12(k-1|k)} \end{array} \right\} + \\ q_{44}^{-1} \left\{ \begin{array}{l} \left[\begin{array}{l} \dot{\theta}_{2(k|k)} - c_{2(k-1|k)} - hw_{21(k-1|k)}\tau_1 - hw_{22(k-1|k)}\tau_2 \\ +hw_{21(k-1|k)}f_1 \operatorname{sgn}(\dot{\theta}_{1(k-1|k)}) + hw_{22(k-1|k)}f_2 \operatorname{sgn}(\dot{\theta}_{2(k-1|k)}) \end{array} \right] hw_{22(k-1|k)} \end{array} \right\} = 0 \quad (4.22)$$

將(4.21),(4.22)移項整理，並寫成矩陣形式如(4.23)

$$\begin{aligned}
& \left[\begin{array}{cc} \left\{ \begin{array}{l} q^{-1}_{33} h^2 w^2_{11(k-1|k)} \operatorname{sgn}(\dot{\theta}_{1(k-1|k)}) \\ + q^{-1}_{44} h^2 w^2_{21(k-1|k)} \operatorname{sgn}(\dot{\theta}_{1(k-1|k)}) \end{array} \right\} & \left\{ \begin{array}{l} q^{-1}_{33} h^2 w_{11(k-1|k)} w_{12(k-1|k)} \operatorname{sgn}(\dot{\theta}_{2(k-1|k)}) \\ + q^{-1}_{44} h^2 w_{21(k-1|k)} w_{22(k-1|k)} \operatorname{sgn}(\dot{\theta}_{2(k-1|k)}) \end{array} \right\} \\ \left\{ \begin{array}{l} q^{-1}_{33} h^2 w_{11(k-1|k)} w_{12(k-1|k)} \operatorname{sgn}(\dot{\theta}_{1(k-1|k)}) \\ + q^{-1}_{44} h^2 w_{21(k-1|k)} w_{22(k-1|k)} \operatorname{sgn}(\dot{\theta}_{1(k-1|k)}) \end{array} \right\} & \left\{ \begin{array}{l} q^{-1}_{33} h^2 w^2_{12(k-1|k)} \operatorname{sgn}(\dot{\theta}_{2(k-1|k)}) \\ + q^{-1}_{44} h^2 w^2_{22(k-1|k)} \operatorname{sgn}(\dot{\theta}_{2(k-1|k)}) \end{array} \right\} \end{array} \right] \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \\
& = - \begin{bmatrix} q^{-1}_{33} h w_{11(k-1|k)} & q^{-1}_{44} h w_{21(k-1|k)} \\ q^{-1}_{33} h w_{12(k-1|k)} & q^{-1}_{44} h w_{22(k-1|k)} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{1(k|k)} - c_{1(k-1|k)} - h w_{11(k-1|k)} \tau_1 - h w_{12(k-1|k)} \tau_2 \\ \dot{\theta}_{2(k|k)} - c_{2(k-1|k)} - h w_{21(k-1|k)} \tau_1 - h w_{22(k-1|k)} \tau_2 \end{bmatrix} \quad (4.23)
\end{aligned}$$

觀察(4.23)具有對稱的性質，可將其改寫為(4.24)方便分析。

$$D_{k-1|k}^T R_{k-1|k} Q_L^{-1} R_{k-1|k}^T D_{k-1|k} F = -\frac{1}{h} D_{k-1|k}^T R_{k-1|k} Q_L^{-1} B_{(k,k-1|k)} \quad (4.24)$$

其中 h : 取樣時間

$$F = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

$$R_{k-1|k} = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix}_{k-1|k}$$

$$Q_L = \begin{bmatrix} q_{33} & 0 \\ 0 & q_{44} \end{bmatrix}$$

$$D_{k-1|k} = \begin{bmatrix} \operatorname{sgn}(\dot{\theta}_1) & 0 \\ 0 & \operatorname{sgn}(\dot{\theta}_2) \end{bmatrix}_{k-1|k}$$

$$B_{(k,k-1|k)} = \begin{bmatrix} \dot{\theta}_{1(k|k)} - c_{1(k-1|k)} - h w_{11(k-1|k)} \tau_1 - h w_{12(k-1|k)} \tau_2 \\ \dot{\theta}_{2(k|k)} - c_{2(k-1|k)} - h w_{21(k-1|k)} \tau_1 - h w_{22(k-1|k)} \tau_2 \end{bmatrix}$$

作完一個時間點的最大相似值(Maximum Likelihood)推導，接著進行兩個時間點的推導。

$$\begin{aligned}
& \ln p(\hat{x}_{k|k} | \hat{x}_{k-1|k}) p(\hat{x}_{k-1|k} | \hat{x}_{k-2|k}) \approx \ln N(f(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda), Q) \cdot N(f(\hat{x}_{k-2|k}, \tau_{k-2}, \lambda), Q) \\
& = \ln \left\{ \frac{1}{(2\pi)^{-n/2} \sqrt{\det(Q)}} \exp \left[-\frac{1}{2} (\hat{x}_{k|k} - f(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda)) Q^{-1} (\hat{x}_{k|k} - f(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda)) \right] \times \right. \\
& \quad \left. \frac{1}{(2\pi)^{-n/2} \sqrt{\det(Q)}} \exp \left[-\frac{1}{2} (\hat{x}_{k-1|k} - f(\hat{x}_{k-2|k}, \tau_{k-2}, \lambda)) Q^{-1} (\hat{x}_{k-1|k} - f(\hat{x}_{k-2|k}, \tau_{k-2}, \lambda)) \right] \right\} \\
& = -\frac{n}{2} \ln [2\pi] - \frac{1}{2} \ln [\det(Q)] - \frac{1}{2} [\hat{x}_{k|k} - f(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda)] Q^{-1} [\hat{x}_{k|k} - f(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda)] \\
& \quad - \frac{n}{2} \ln [2\pi] - \frac{1}{2} \ln [\det(Q)] - \frac{1}{2} [\hat{x}_{k-1|k} - f(\hat{x}_{k-2|k}, \tau_{k-2}, \lambda)] Q^{-1} [\hat{x}_{k-1|k} - f(\hat{x}_{k-2|k}, \tau_{k-2}, \lambda)] \\
& (4.25)
\end{aligned}$$

對(4.25)作微分得到(4.26)

$$\begin{aligned}
& \frac{\partial}{\partial f_1} \ln p(\hat{x}_{k|k} | \hat{x}_{k-1|k}) p(\hat{x}_{k-1|k} | \hat{x}_{k-2|k}) \\
& \approx \frac{\partial}{\partial f_1} \left\{ \begin{array}{l} -\frac{1}{2} [\hat{x}_{k|k} - f(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda)] Q^{-1} [\hat{x}_{k|k} - f(\hat{x}_{k-1|k}, \tau_{k-1}, \lambda)] \\ -\frac{1}{2} [\hat{x}_{k-1|k} - f(\hat{x}_{k-2|k}, \tau_{k-2}, \lambda)] Q^{-1} [\hat{x}_{k-1|k} - f(\hat{x}_{k-2|k}, \tau_{k-2}, \lambda)] \end{array} \right\} = 0 \quad (4.26)
\end{aligned}$$

由(4.18),(4.24),(4.26)可見使兩個時間點的對數期望值最大的參數 F 滿足(4.27)。

$$\begin{aligned}
& (R_{k-1|k} Q_L^{-1} R_{k-1|k}^T D_{k-1|k} + R_{k-2|k} Q_L^{-1} R_{k-2|k}^T D_{k-2|k}) F \\
& = -\frac{1}{h} R_{k-1|k} Q_L^{-1} B_{(k,k-1|k)} - \frac{1}{h} R_{k-2|k} Q_L^{-1} B_{(k-1,k-2|k)} \quad (4.27)
\end{aligned}$$

將(4.27)推廣到 k 個點，並寫成矩陣形式即為(4.28)

$$\begin{aligned}
 & \left[D_{k-1|k}^T R_{k-1|k} Q_L^{-1} \quad D_{k-2|k}^T R_{k-2|k} Q_L^{-1} \quad \dots \quad D_{0|k}^T R_{0|k} Q_L^{-1} \right] \begin{bmatrix} R_{k-1|k}^T D_{k-1|k} \\ R_{k-2|k}^T D_{k-2|k} \\ \vdots \\ R_{0|k}^T D_{0|k} \end{bmatrix} F \\
 & = -\frac{1}{h} \left[D_{k-1|k}^T R_{k-1|k} Q_L^{-1} \quad D_{k-2|k}^T R_{k-2|k} Q_L^{-1} \quad \dots \quad D_{0|k}^T R_{0|k} Q_L^{-1} \right] \begin{bmatrix} B_{(k,k-1)|k} \\ B_{(k-1,k-1)|k} \\ \vdots \\ B_{(1,0)|k} \end{bmatrix} \quad (4.28)
 \end{aligned}$$

其中 F, R, Q_L, D, B 請參照(4.24)。

4.3 濾波力矩計算

在模擬的過程中我們發現，在運動學模型和動力學模型的多模型組態 GPB-2 的架構下，其角度和角速度的估測皆能達到相當好的品質。因此若能將估測的角速度經由差分獲得角加速度，再將其代回計算力矩並與原始輸入力矩做比較(驗算)，即可判斷哪一個軸的致動器發生故障。

但是在實際使用角速度差分獲得角加速度，進行力矩計算之後，發現效果不好。原因是因為差分會使固有雜訊放大，而使得角加速度相當不準確，造成力矩計算的極大誤差，因此這是一條不可行的路。

但在 [4], [9] 中 Warren E. Dixon 提出一個利用一階濾波器濾掉角加速度，再將其與原本的濾波力矩作比較的錯誤偵測方法。由於沒有了角加速度的影響，使得力矩驗算的想法可行。

推導

n 軸自由度的機器臂動力學方程式如 (4.1)，但 (4.1) 並未包括致動器錯誤，重寫包含致動器錯誤的 n 軸自由度的機器臂方程式如下：

$$M(\theta)\ddot{\theta}(t) + N(\theta, \dot{\theta}) + u_{-1}(t - T_f)\zeta(t) = \tau(t) \quad (4.29)$$

其中 $\theta, \dot{\theta}, \ddot{\theta} \in R^n$ 為各別轉軸的變數(角度、角速度、角加速度)， $\tau \in R^n$ 分別對應各別轉軸的力矩， M 為 $n \times n$ 的對稱正定矩陣，其代表每個軸對其他軸的慣量。 N 為 (4.1) 中的 $V+G$ ，寫成 N 是為了推導方便，其為由質心造成的離心力與重力加速度 g 的效應所組成，為一 $n \times 1$ 向量。

$\zeta(t)$ 代表致動器的錯誤發生， $u_{-1}(t-T_f)$ 為一在 T_f 啟動的脈波訊號(step function)，故 $u_{-1}(t-T_f)\bar{\zeta}(t)$ 代表制動器錯誤發生在 T_f 之後。

比較(4.1)與(4.29)可發現，當系統錯誤發生後，必須將錯誤的情形考慮進力矩的部份。反之，應該可以由力矩的變化，偵測系統是否有錯誤發生。

將 $\tau(t)$ 與 $f(t)$ 兩者作卷積(convolution)得到

$$\tau_f(t) = f(t) * \tau(t) \quad (4.30)$$

將 $\tau(t)$ 以(4.29)代入(4.30)右半邊

$$\begin{aligned} & f(t) * \tau(t) \\ &= f(t) * \{M(\theta)\ddot{\theta} + N(\theta, \dot{\theta}) + u_{-1}(t-T_f)\zeta(t)\} \\ &= f(t) * \{M(\theta)\ddot{\theta}\} + f(t) * N(\theta, \dot{\theta}) + f(t) * u_{-1}(t-T_f)\zeta(t) \\ &= M(\theta) \{f(t) * \ddot{\theta}\} + f(t) * N(\theta, \dot{\theta}) + f(t) * u_{-1}(t-T_f)\zeta(t) \end{aligned} \quad (4.31)$$

觀察(4.31)發現只有第一項帶有角加速度，令 $f(t)$ 為一可濾掉角加速度的線性濾波器(4.32)。

$$\text{令 } f(t) = \alpha \exp(-\beta t) \quad (4.32)$$

$$\dot{\theta}(t) = y(t) \quad (4.33)$$

且令 $f(t)$ 的拉式轉換為 $F(s)$ ， $y(t)$ 的拉式轉換為 $Y(s)$ ，如(4.34),(4.35)。

$$f(t) \xrightarrow{\text{Laplace}} F(s) \quad (4.34)$$

$$\dot{q}(t) = y(t) \xrightarrow{\text{Laplace}} Y(s) \quad (4.35)$$

將(4.32),(4.33)代入(4.31)第一項 $f(t)*\ddot{q}$ 得到(4.36)。

$$\begin{aligned}
 & f(t)*\ddot{q}(t) \\
 &= f(t)*\left\{\frac{d}{dt}\dot{q}(t)\right\} \\
 &= f(t)*\left\{\frac{d}{dt}y(t)\right\} \tag{4.36}
 \end{aligned}$$

將(4.36)配合(4.34),(4.35)做拉式轉換得到(4.37)

$$\begin{aligned}
 f(t)*\left\{\frac{d}{dt}y(t)\right\} &\xrightarrow{Laplace} F(s)\cdot[sY(s)-y(0)] \\
 &= [sF(s)]Y(s)-F(s)y(0) \\
 &= [sF(s)-F(0)]Y(s)+F(0)Y(s)-F(s)y(0) \tag{4.37}
 \end{aligned}$$

再將(4.37)經由反拉式轉換獲得時域上的表示式(4.38)

$$\begin{aligned}
 & [sF(s)-F(0)]Y(s)+F(0)Y(s)-F(s)y(0) \\
 &\xrightarrow{L^{-1}} \dot{f}(t)*y(t)+f(0)y(t)-f(t)y(0) \\
 &= \dot{f}(t)*\dot{q}(t)+f(0)\dot{q}(t)-f(t)\dot{q}(0) \tag{4.38}
 \end{aligned}$$

將(4.38)代回(4.31)，整理得到(4.39)

$$\tau_f(t) = Y_f(\theta, \dot{\theta}) + \zeta_f(t) \tag{4.39}$$

其中 $Y_f(\theta, \dot{\theta}) = M(\theta)\left[\dot{f}(t)*\dot{\theta}(t)+f(0)\dot{\theta}(t)-f(t)\dot{\theta}(0)\right]+f(t)*N(\theta(t), \dot{\theta}(t))$

$$\zeta_f(t) = f(t)*u_{-1}(t-T_f)\zeta(\theta(t), \dot{\theta}(t))$$

錯誤偵測

在(4.39)的等號右邊可由估測的角度 $\hat{\theta}$ 、角速度 $\hat{\dot{\theta}}$ 帶入計算得到 $Y_f(\hat{\theta}, \hat{\dot{\theta}})$ ，左邊則直接由量測力矩和濾波器卷積產生。將兩者相減為剩餘值 $\varepsilon(t)$ 。

$$\varepsilon(t) = \tau_f(t) - \hat{\tau}_{f(t)} = \tau_f(t) - Y_f(\hat{\theta}, \hat{\dot{\theta}}) = \varsigma_f(t) \quad (4.40)$$

$$D[\varepsilon] = \begin{cases} 1 & \text{if } \|\varepsilon\| > \text{threshold} \\ 0 & \text{if } \|\varepsilon\| \leq \text{threshold} \end{cases} \quad (4.41)$$

由 D 的數值為 0 或 1 就可決定故障發生與否。

4.4 分群估測

在傳統的多模型組態估測演算法 GPB-2 裡，必須將所有的錯誤模型一起估測，造成計算量過大而無法實際應用到多軸機械臂系統。本小節則提出將錯誤模型事先分群以降低計算量的概念，並配合 4.3 小節的濾波力矩計算，以判斷當下系統模型所處的群組。如此一來，將可由減少 GPB-2 演算法所計算的模型數達到降低計算量的需求。

二軸機械臂運作模型如下所示：

模型一：第一軸正常、第二軸正常。

模型二：第一軸卡住、第二軸正常。

模型三：第一軸正常、第二軸卡住。

模型四：第一軸卡住、第二軸卡住。

模型五：第一軸甩動、第二軸正常。

模型六：第一軸正常、第二軸甩動。

模型七：第一軸甩動、第二軸甩動。

模型八：第一軸卡住、第二軸甩動。

模型九：第一軸甩動、第二軸卡住。

配合 4.3 小節的濾波力矩計算，將模型二、模型五分為群組一（只有第一軸故障），模型三、模型六分為群組二（只有第二軸故障），模型四、模型七、模型八、模型九分為群組三（兩軸皆故障）。

在適當的分群後，再依據(4.40)定義 $\varepsilon_n(t)$ 代表濾波力矩差值 $\varepsilon(t)$ 中的第 n 個狀態，由(4.42)中對應各軸濾波力矩差值的元素，即可判斷現在哪一個軸發生錯誤。

$$I[\varepsilon_n(t)] = \begin{cases} 1 & \text{if } \|\varepsilon(t)_n\| > \text{threshold} \\ 0 & \text{if } \|\varepsilon(t)_n\| \leq \text{threshold} \end{cases} \quad (4.42)$$

I 為 $n \times 1$ 的向量空間，其每一個元素對應一個軸的狀態，由該元素數值為 0 或 1 即可判斷該軸是否發生錯誤。

圖示如下：

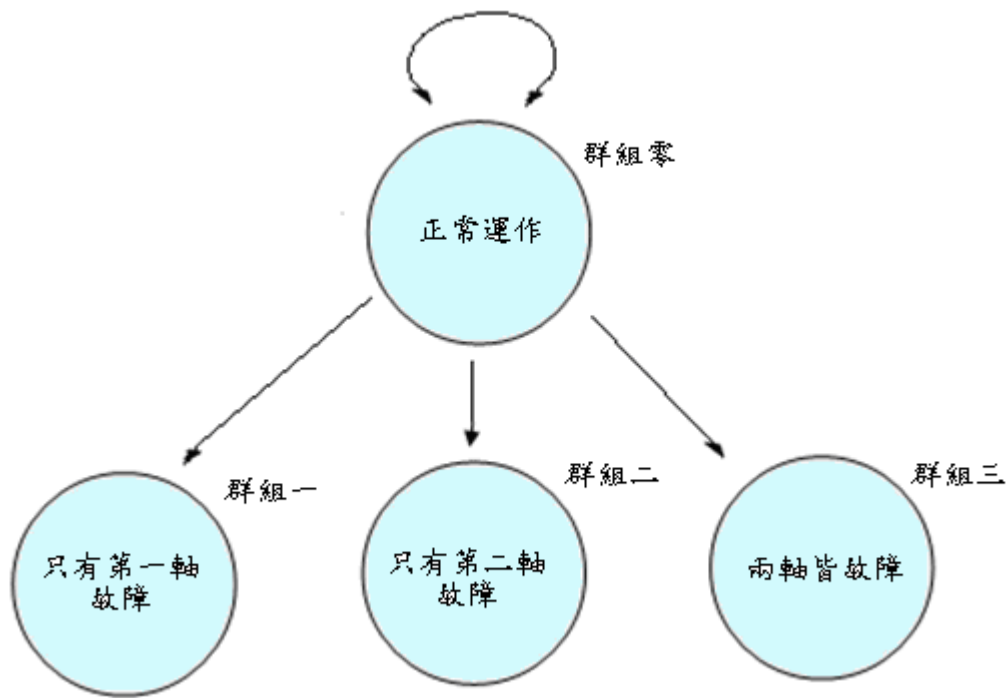


圖 4.3 二軸機械臂錯誤模型分群估測說明圖

由圖 4.3 可見剛開始系統為正常運作(群組零)，若沒有錯誤發生則繼續維持在群組零，當有錯誤發生則會跳至群組一、二或三。在群組一、二與三中，GPB-2 演算法會指出目前機械臂所運作的模型。

這種分群的概念，除了大大縮減了 GPB-2 演算法同時所需估測的模型數目而減少計算量外，更因為模型數目減少而使得模型識別的準確度大幅提升，在第五章中看到完整演算法的良好的模擬結果。

4.5 演算法架構圖

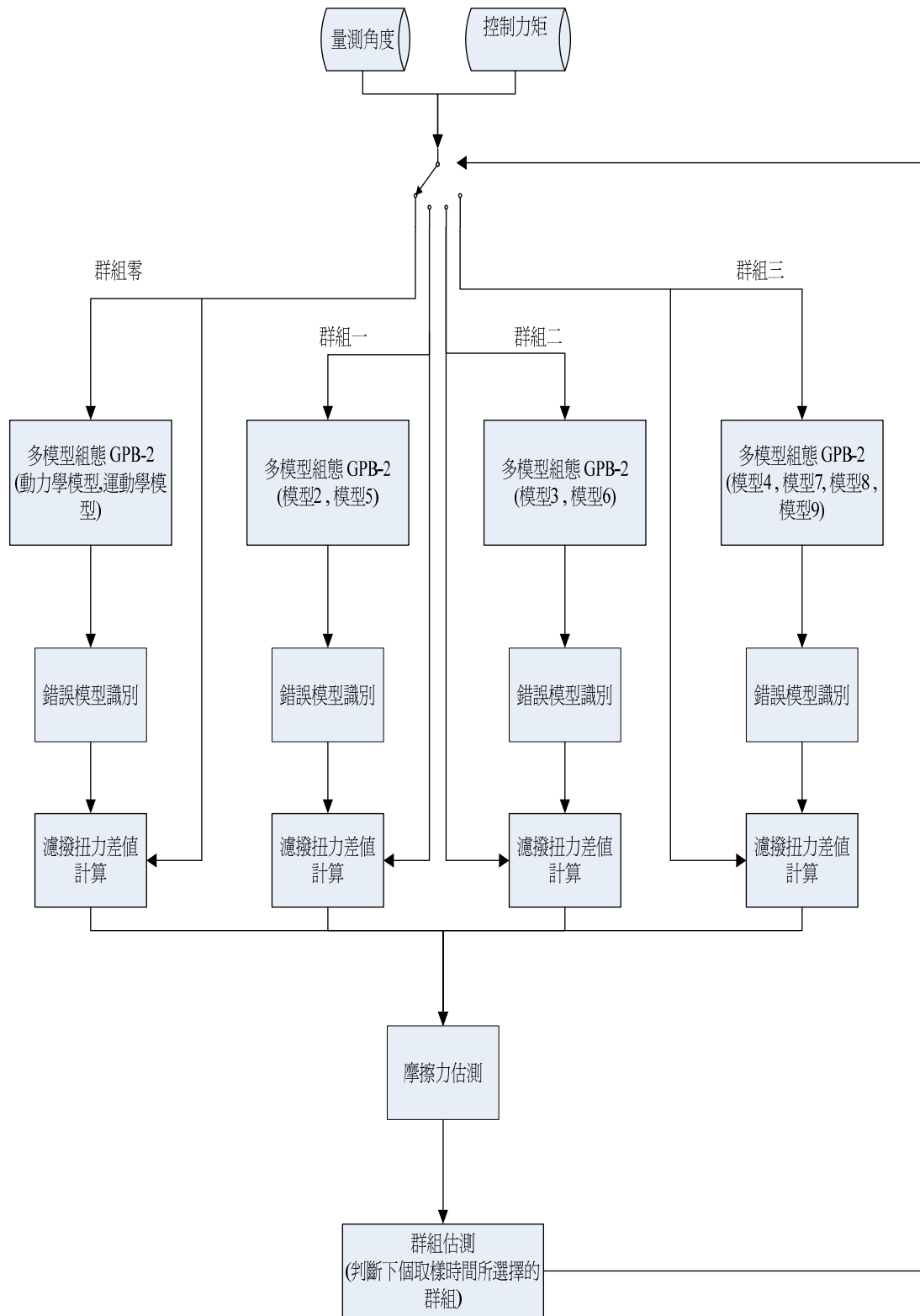


圖 4.4 演算法架構圖

圖 4.4 為完整演算法的架構圖，可看到共分為群組零、一、二、三等四個群組。演算法預設為群組零，藉由輸入量測角度與控制力矩，利用 GPB-2 演算法進行錯誤識別，並利用力矩差值計算作為下個取樣點的群組選擇依據。在進行群組估測前，也藉由估測狀態(角度、角速度)使用期望值最大化演算法進行摩擦力估測，以其使群組估測更準確。



第五章 模擬分析與探討

為了驗證本論文提出的即時錯誤偵測與識別法，本章將利用電腦軟體 Matlab 進行模擬驗證，並繪出相關的模擬圖表，取樣時間皆為 0.001 秒。本論文的模擬平台為二軸機械臂，藉由演算法估測角度與角速度，並同時估測機械臂正常工作下的黏滯摩擦力係數與庫倫摩擦力係數，其相關參數可參照附錄表(一)。論文中將馬達分為兩種錯誤情形，**卡住**和**自由甩動**：卡住是指不管控制器指定馬達的力矩為多少，該軸維持定固定角度；自由甩動是指不管控制器指定馬達的力矩為多少，該軸馬達輸出力矩皆為零，因此該軸會自由甩動。演算法將隨時指出兩軸馬達的情形，也就是**正常**，**卡住**與**自由甩動**三種狀態。

本章第一小節進行單一模型的模擬，藉由單一模型的模擬確保非追蹤型卡曼濾波器(UKF, Unscented Kalman Filter)有良好的抗雜訊能力，以作為後續演算法的基礎。第二小節為估測機械臂正常工作下的黏滯摩擦力係數，摩擦力估測是系統抗雜訊的一環，準確的摩擦力估測對後續的錯誤偵測與識別演算法相當重要。第三小節為力矩差值計算，將藉由模型的切換看到力矩差值的變化。第四小節模擬動力學模型與運動學模型之間的切換，檢視系統錯誤發生時，該演算法產生的剩餘值(Residual)是否可以指出錯誤發生。最後一小節則將本論文的演算法套用到機械臂系統上，並藉由與原始 GPB-2 演算法比較，以確認新演算法的優良性能。

5.1 單一模型狀態估測

這小節將利用電腦模擬非追蹤型卡曼濾波器(UKF, Unscented kalman filter)在二軸機械臂系統上的狀態估測效果，將看到非追蹤型卡曼濾波器(UKF)是一個具有相當好的抗雜訊能力的觀察器(observer)。

茲將兩軸機械臂的錯誤情況分成下列九種模型：

模型一：第一軸正常、第二軸正常

模型二：第一軸卡住、第二軸正常

模型三：第一軸正常、第二軸卡住

模型四：第一軸卡住、第二軸卡住

模型五：第一軸甩動、第二軸正常

模型六：第一軸正常、第二軸甩動

模型七：第一軸甩動、第二軸甩動

模型八：第一軸卡住、第二軸甩動

模型九：第一軸甩動、第二軸卡住

以上九種模型的離散動力學方程式可參照第四章第一小節，由於模型為數眾多，故本節只取模型一、五進行模擬。

模擬條件：

令處理雜訊(process noise)與量測雜訊(measurement noise)分別為平均值為 0 標準差為 0.1 與 0.001 的高斯雜訊。其餘參數設為

$$\tau_1 = 8\sin(3t), \quad \tau_2 = 0.5\sin(10t), \quad f_1 = 0.5 \times \dot{\theta}_1, \quad f_2 = 0.1 \times \dot{\theta}_2。$$

Simulink 架構圖：

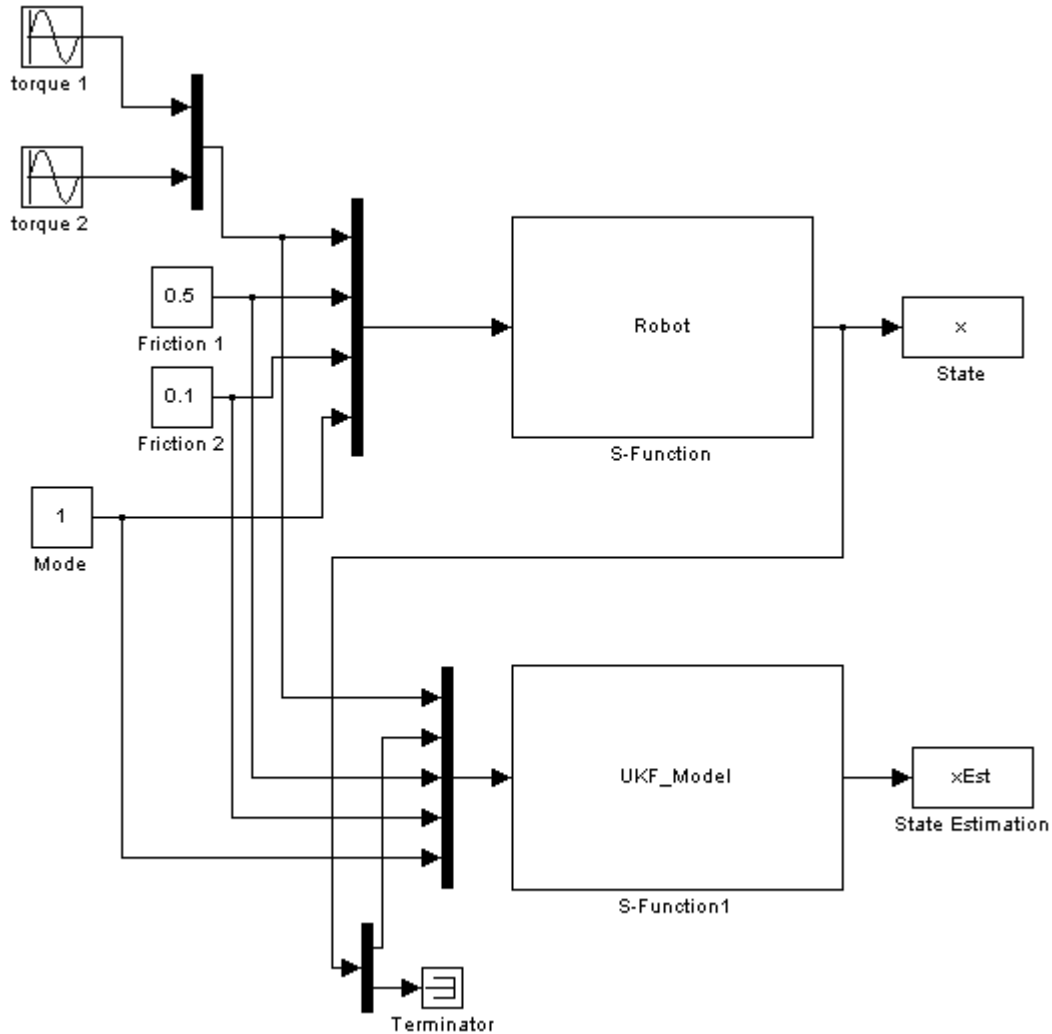


圖 5.1 估測二軸機械臂單一狀態架構圖

圖 5.1 的構成分兩大區塊，Robot 區塊代表二軸機械臂的模型，其輸入端順序依次為力矩、摩擦力一、摩擦力二與模型選取，模型選取的範圍為 1~9，其輸出為角度與角速度。UKF_Model 的區塊為非追蹤型卡曼濾波器(UKF)，其輸入端順序依次為力矩、量測角度、摩擦力一、摩擦力二與模型選取，其輸出值為角度與角速度估測值。

模擬結果

模擬(一)

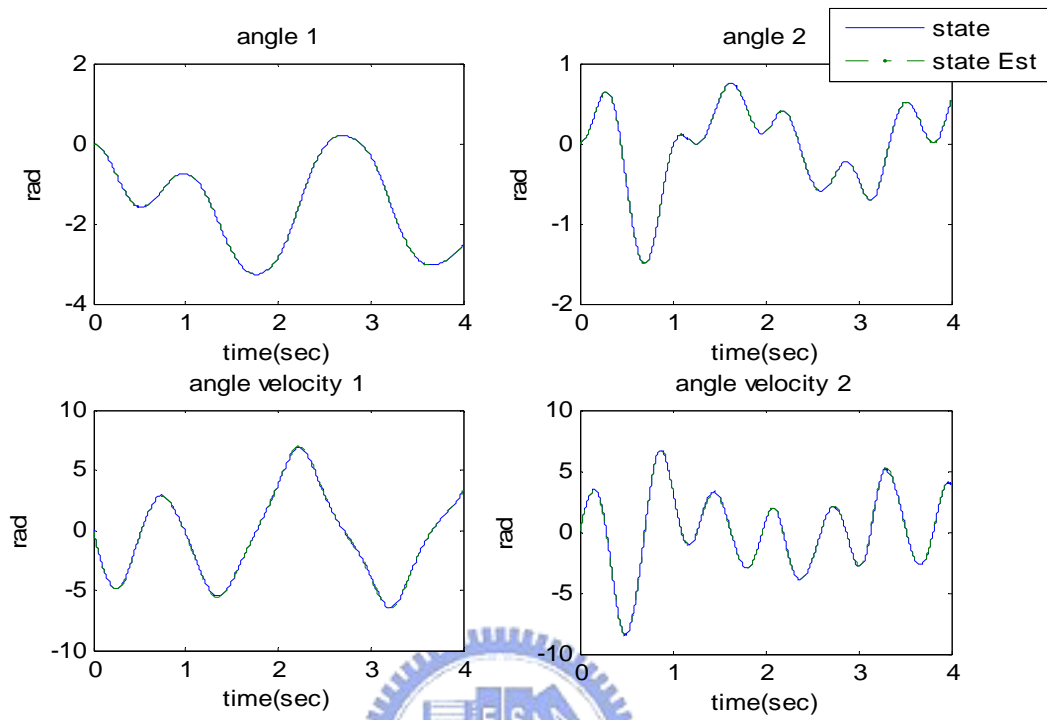


圖 5.2 高斯雜訊下模型一狀態估測

模擬(二)

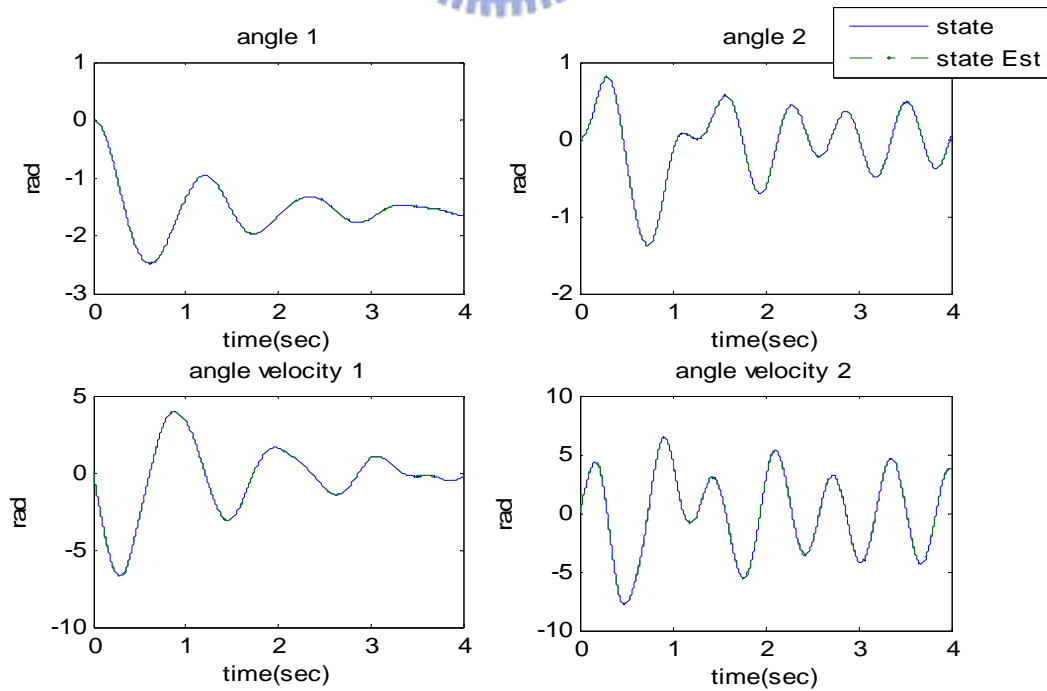


圖 5.3 高斯雜訊下模型五狀態估測

由圖 5.1, 圖 5.2, 可見到非追蹤型卡曼濾波器(UKF)在高斯雜訊的干擾下, 仍然可估測到每個模型的狀態, 估測與實際完全吻合。接著測試非追蹤型卡曼濾波器(UKF)在模型不精確下的估測效果, 將以改變慣量的方式來進行模擬。

模擬條件:

令處理雜訊(process noise)與量測雜訊(measurement noise)分別為平均值為 0 標準差為 0.1 與 0.001 的高斯雜訊, $\tau_1 = 8\sin(3t)$, $\tau_2 = 0.5\sin(10t)$, $f_1 = 0.5 \times \dot{\theta}_1$, $f_2 = 0.1 \times \dot{\theta}_2$ 。但改變大軸(圖 4.1 的第一軸)慣量為原始的 1.5 倍, 取模型一、模型六進行模擬。

模擬結果:

模擬(三)

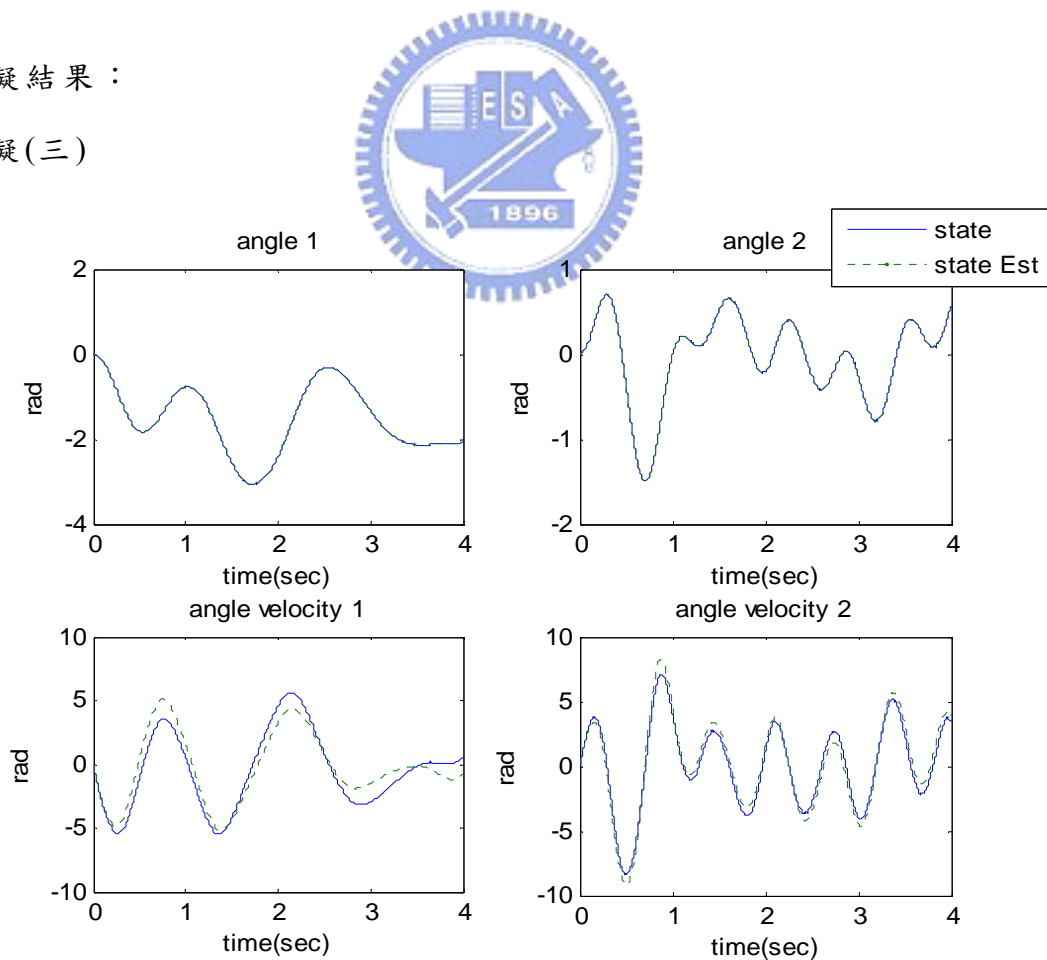


圖 5.4 大軸慣量為原來的 1.5 倍下模型一狀態估測結果

模擬(四)

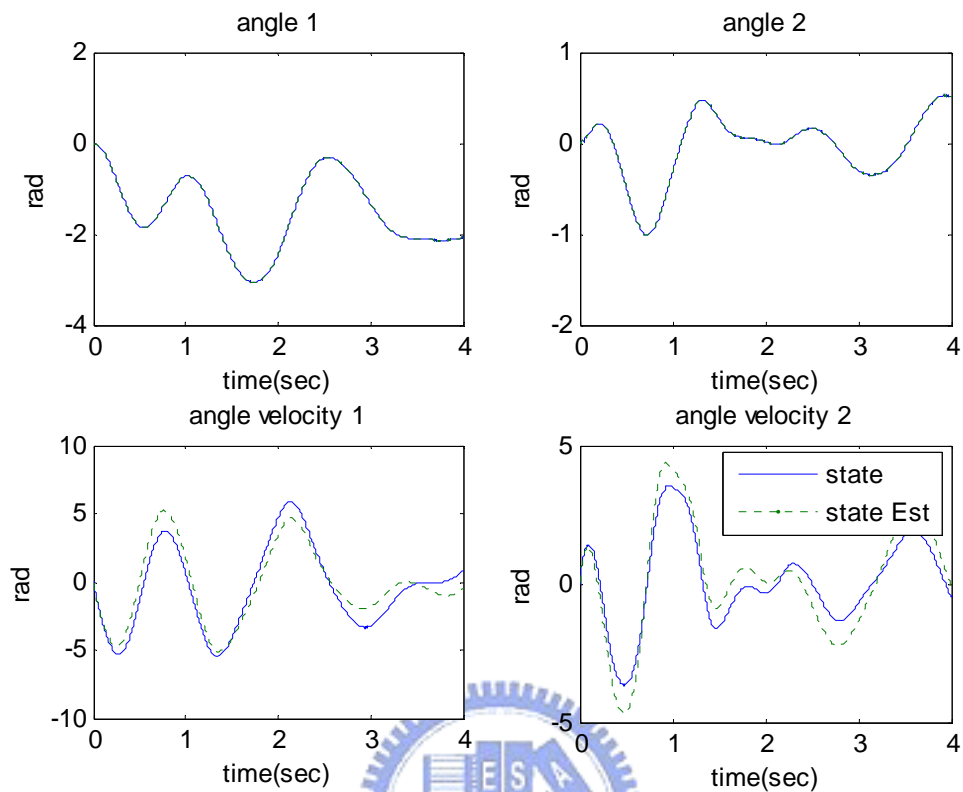


圖 5.5 大軸慣量為原來的 1.5 倍下模型六狀態估測結果

圖 5.4, 圖 5.5 中實線為實際值, 虛線為估測值。觀察後可發現當大軸的質量與慣量改變, 狀態估測中的角速度會明顯地變差, 角度的部份由於可以量測到, 故估測結果始終不錯。若改變小軸慣量, 也會有類似的模擬結果, 相關數據可參考附錄表(二), 在此就不贅述。

5.2 摩擦力估測

摩擦力係數估測是藉由機械臂系統狀態，角度和角速度的變化，輔以機械臂的動力學方程式，利用第四章所推導的期望值最大化演算法 (Expectation-Maximization Algorithm) 進行計算。

模擬條件：

令處理雜訊 (process noise) 與量測雜訊 (measurement noise) 分別為平均值為 0 標準差為 0.1 與 0.001 的高斯雜訊。

Simulink 架構圖：

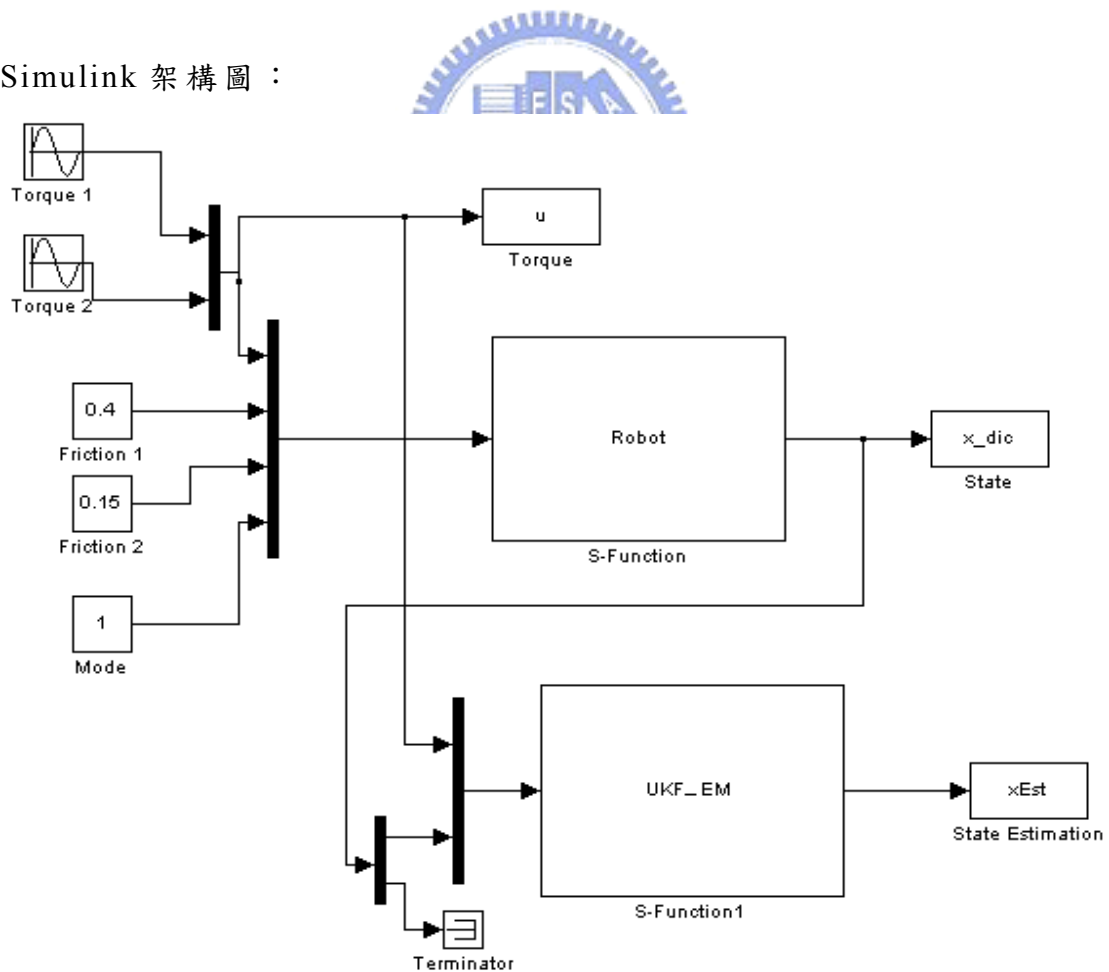


圖 5.6 摩擦力估測架構圖

圖 5.6 的構成分兩大區塊，Robot 區塊代表二軸機械臂的模型，其輸入端順序依次為力矩、摩擦力一、摩擦力二與模型選取，模型選取的範圍為 1~9，其輸出為角度與角速度。UKF_EM 的區塊使用期望值最大化演算法(EM Algorithm)與非追蹤型卡曼濾波器(UKF)計算摩擦力係數並估測系統狀態，其輸入端順序依次為力矩、量測角度，其輸出值為角度、角速度與摩擦力係數估測值。在模擬中，將可看到輸入力矩對使用動力學模型進行摩擦力係數估測有一定的影響，這也是之後使用運動學方程式進行摩擦力估測的理由之一。

模擬結果

動力學模型估測摩擦力係數模擬結果

模擬(五)： 參數設為 $\tau_1 = 8\sin(3t)$ ， $\tau_2 = 0.5\sin(10t)$ ， $f_1 = 0.4 \times \dot{\theta}_1$ ， $f_2 = 0.15 \times \dot{\theta}_2$

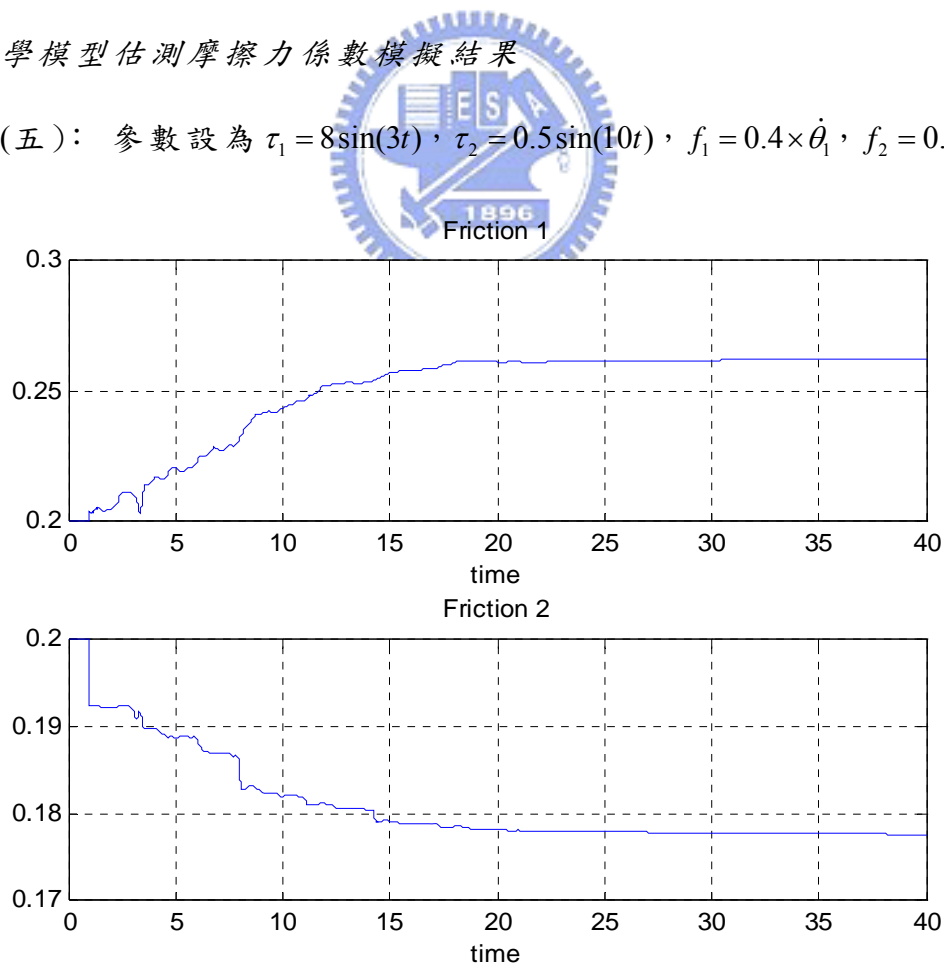


圖 5.7 模擬(五)的黏滯摩擦力係數估測

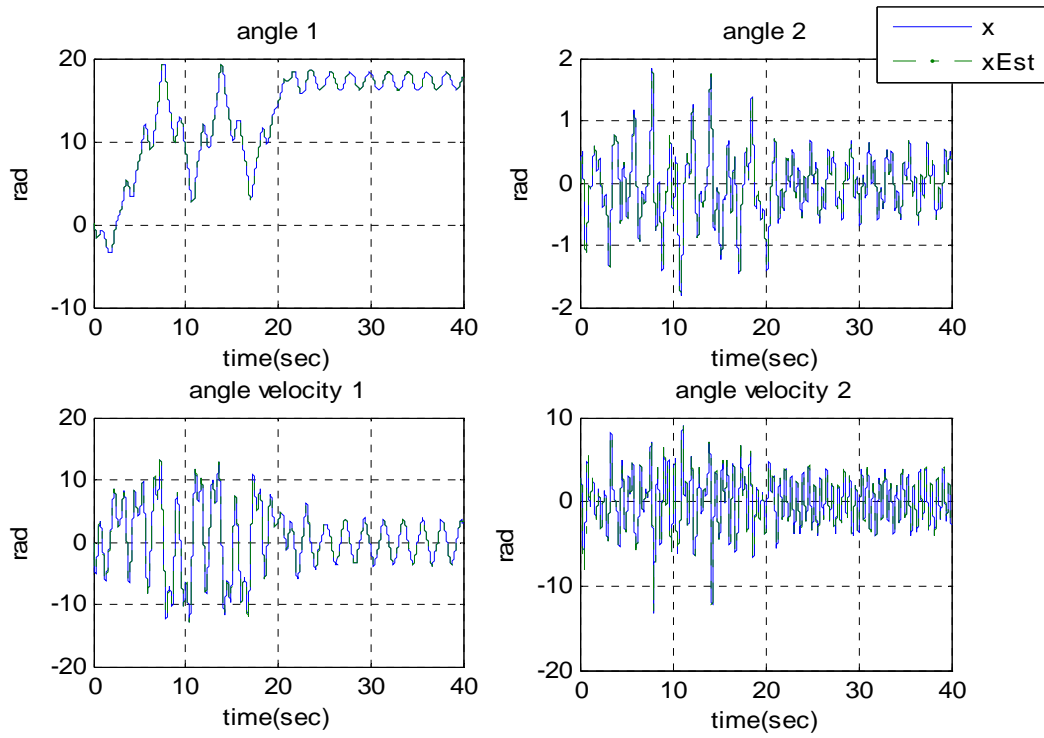


圖 5.8 模擬(五)的角度與角速度估測

模擬(六)： 參數設為 $\tau_1 = 8\sin(t)$ ， $\tau_2 = 0.5\sin(10t)$ ， $f_1 = 0.4 \times \dot{\theta}_1$ ， $f_2 = 0.15 \times \dot{\theta}_2$

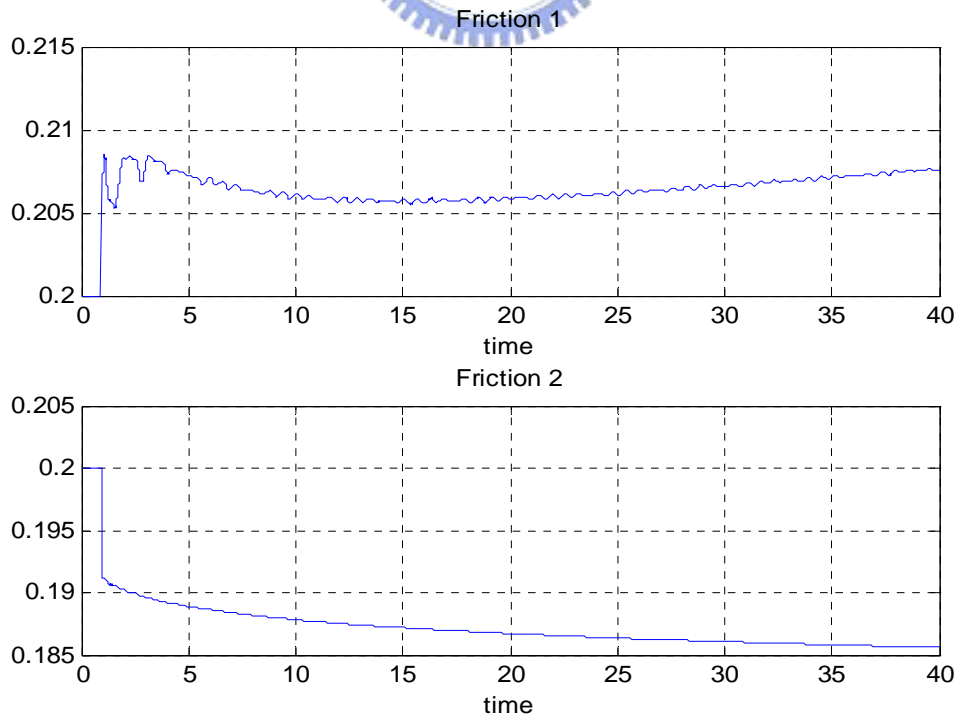


圖 5.9 模擬(六)的黏滯摩擦力係數估測

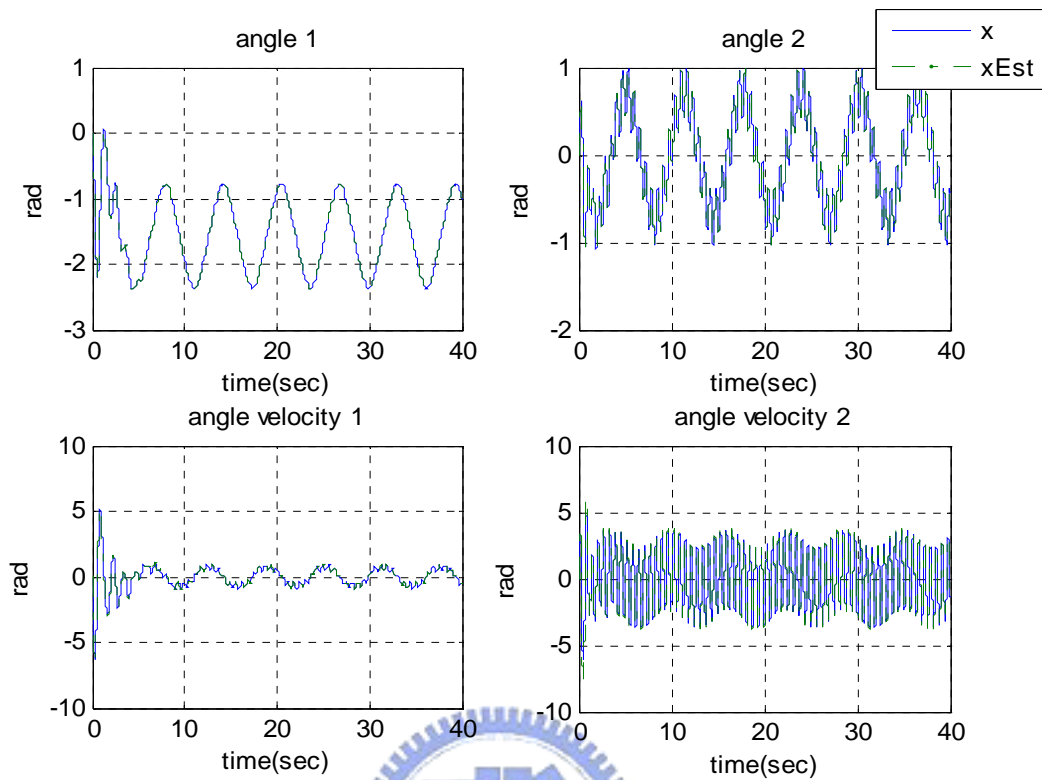


圖 5.10 模擬(六)的角度與角速度估測

將圖 5.7 與 5.9 比較原始的黏滯摩擦力係數設定，發現估測值會接近但並沒有收斂到真正的黏滯摩擦力係數，而且比較圖 5.7 與圖 5.9 會發現力矩的給予也會影響摩擦力估測的結果。因此改用運動學模型計算角度與角速度，並使用一階低通巴特沃茲濾波器(Butterworth filter)過濾，再代入最大化演算法計算黏滯摩擦力係數。

運動學模型估測摩擦力係數模擬結果

模擬(七)：參數設為 $\tau_1 = 8\sin(3t)$ ， $\tau_2 = 0.5\sin(10t)$ ， $f_1 = 0.4 \times \dot{\theta}_1$ ， $f_2 = 0.15 \times \dot{\theta}_2$

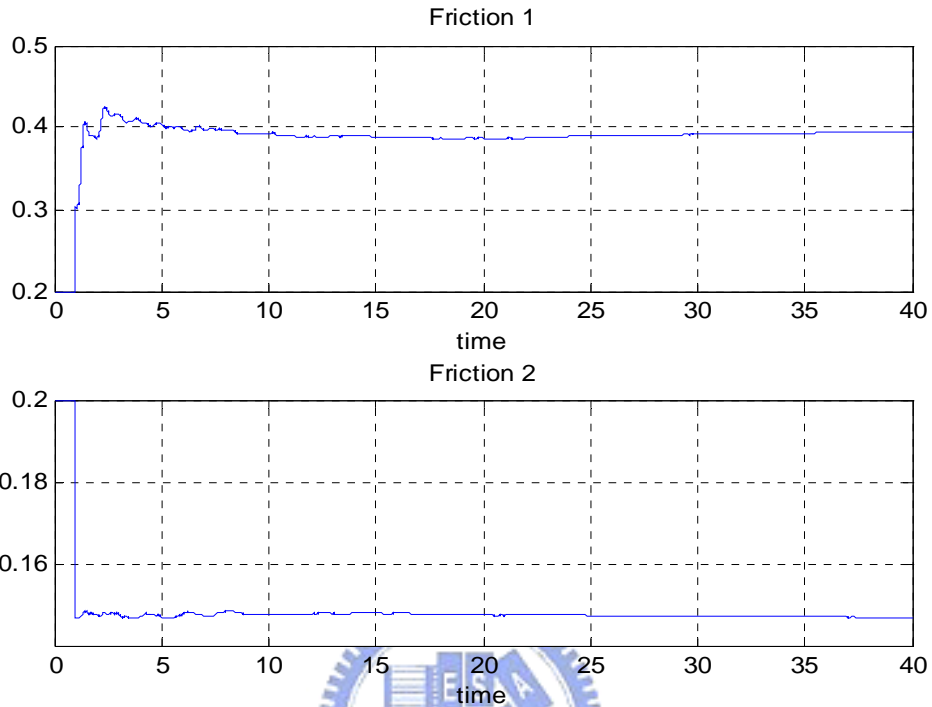


圖 5.11 模擬(七)的黏滯摩擦力估測

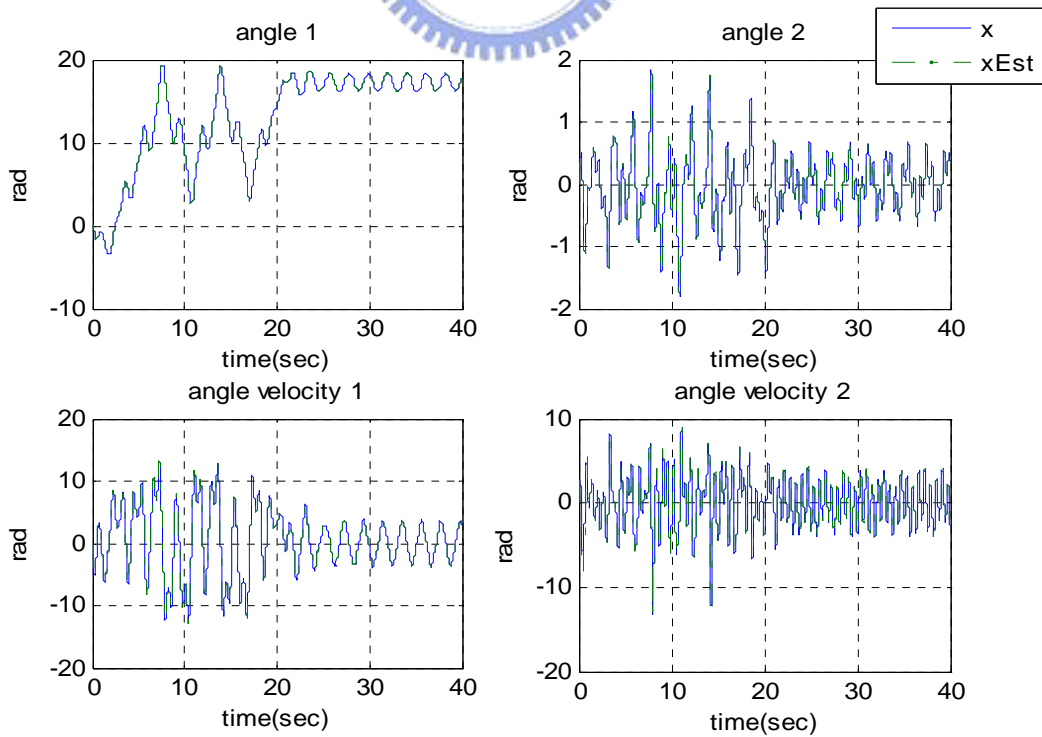


圖 5.12 模擬(七)的角度與角速度估測

模擬(八)：參數設為 $\tau_1 = 8\sin(t)$ ， $\tau_2 = 0.5\sin(10t)$ ， $f_1 = 0.4 \times \dot{\theta}_1$ ， $f_2 = 0.15 \times \dot{\theta}_2$

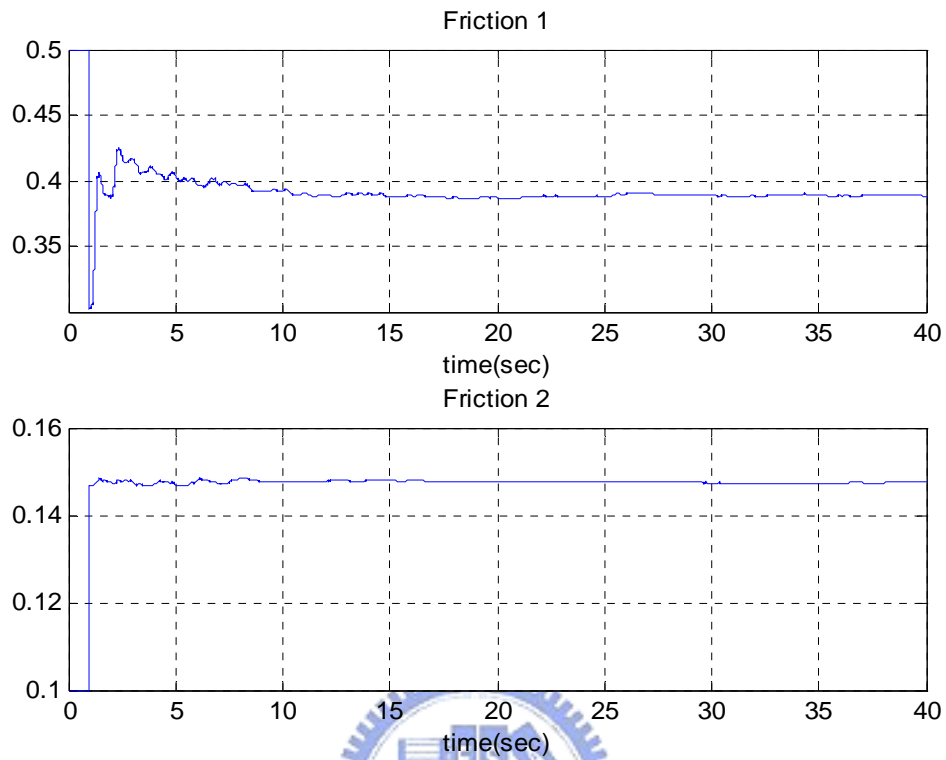


圖 5.13 模擬(八)的黏滯摩擦力估測

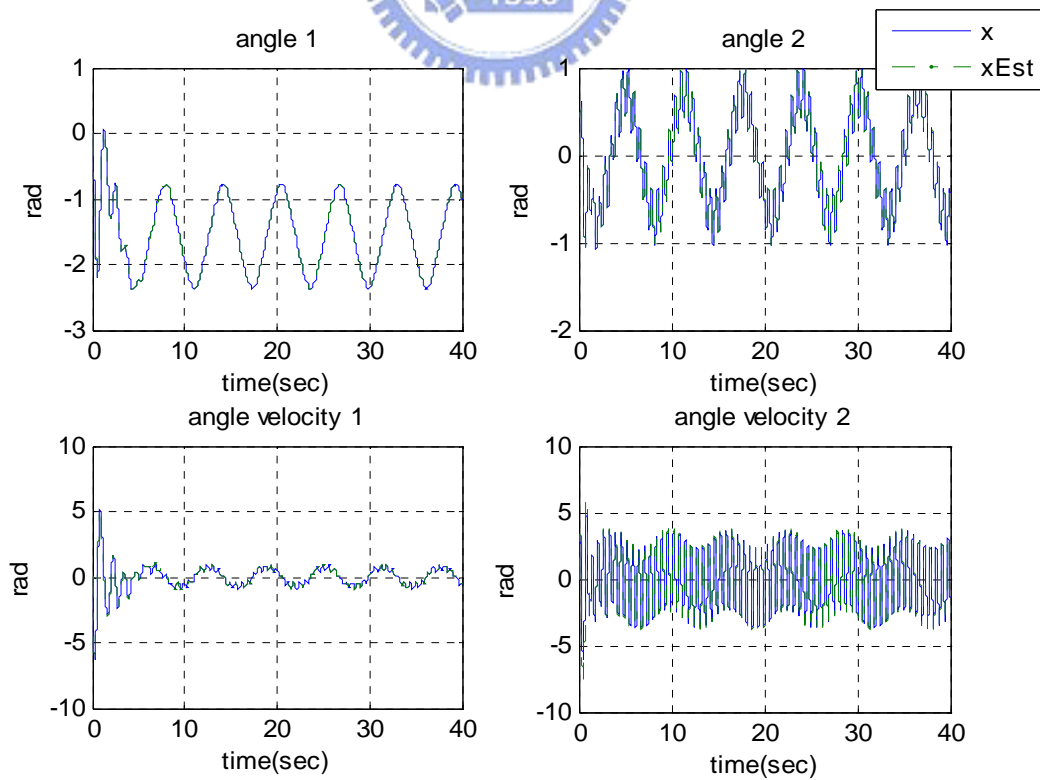


圖 5.14 模擬(八)的角度與角速度估測

將圖 5.7、圖 5.9 與圖 5.11、5.13 對比，可看到改良後的摩擦力演算法估測較準確，可達到模型精確的要求。比較圖 5.8、圖 5.10 與圖 5.12、5.14 發現在摩擦力估測準確的情形下，其系統狀態估測也較準確。

接著就測試在不精確的動力學模型與運動學模型下的摩擦力係數估測狀況，將大軸慣量變為原來的 1.5 倍：

不精確模型下動力學模型估測摩擦力係數模擬結果

模擬(九)：參數設為 $\tau_1 = 8\sin(3t)$ ， $\tau_2 = 0.5\sin(10t)$ ， $f_1 = 0.4 \times \dot{\theta}_1$ ， $f_2 = 0.15 \times \dot{\theta}_2$

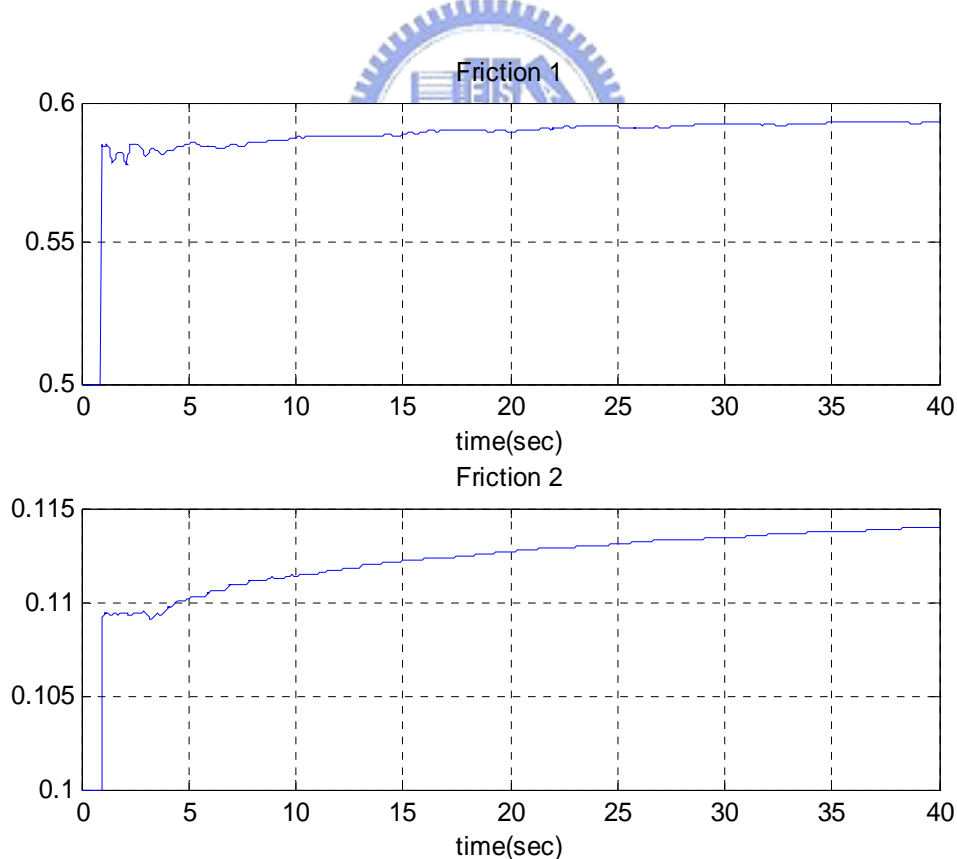


圖 5.15 模擬(九)的黏滯摩擦力係數估測

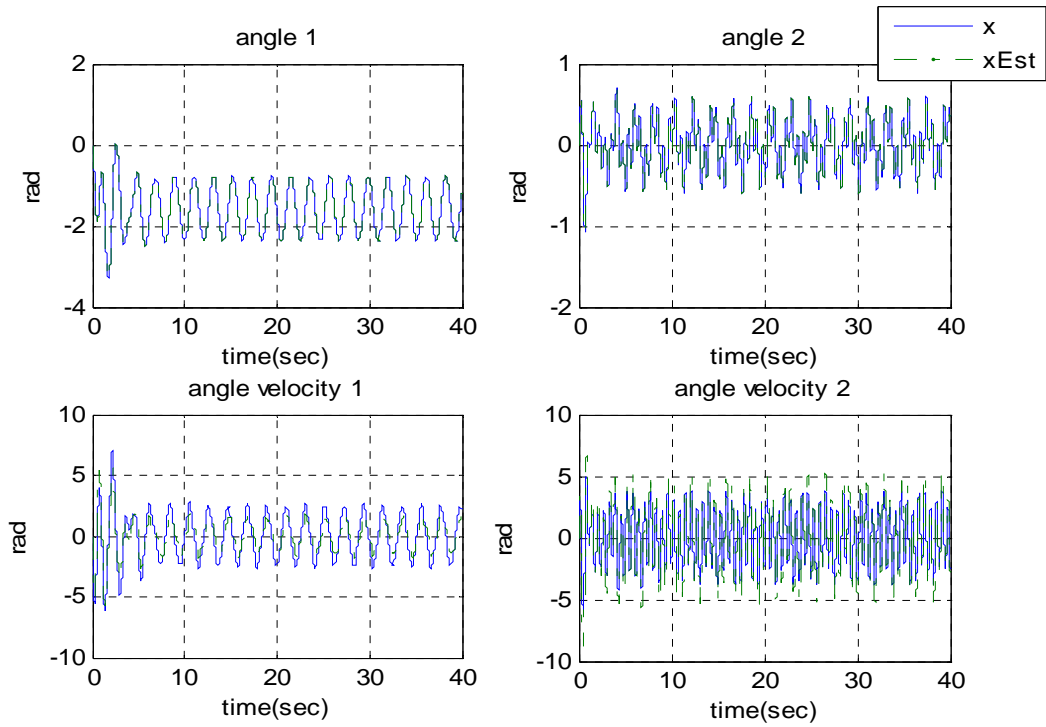


圖 5.16 模擬(九)的角度與角速度估測

不精確模型下運動學模型估測摩擦力係數模擬結果

模擬(十)：參數設為 $\tau_1 = 8\sin(3t)$ ， $\tau_2 = 0.5\sin(10t)$ ， $f_1 = 0.4 \times \dot{\theta}_1$ ， $f_2 = 0.15 \times \dot{\theta}_2$

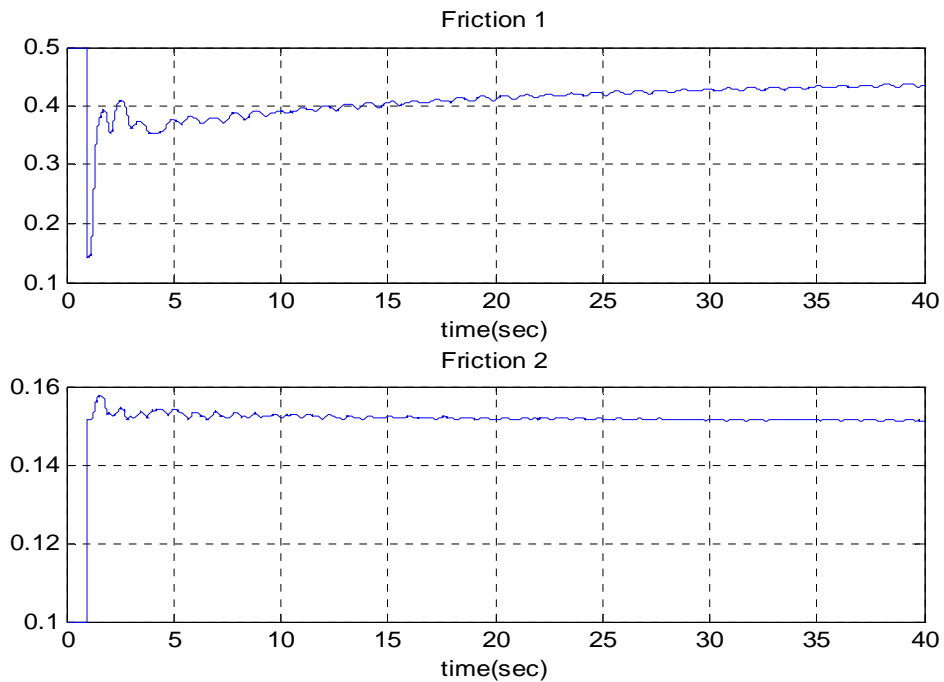


圖 5.17 模擬(十)的黏滯摩擦力係數估測

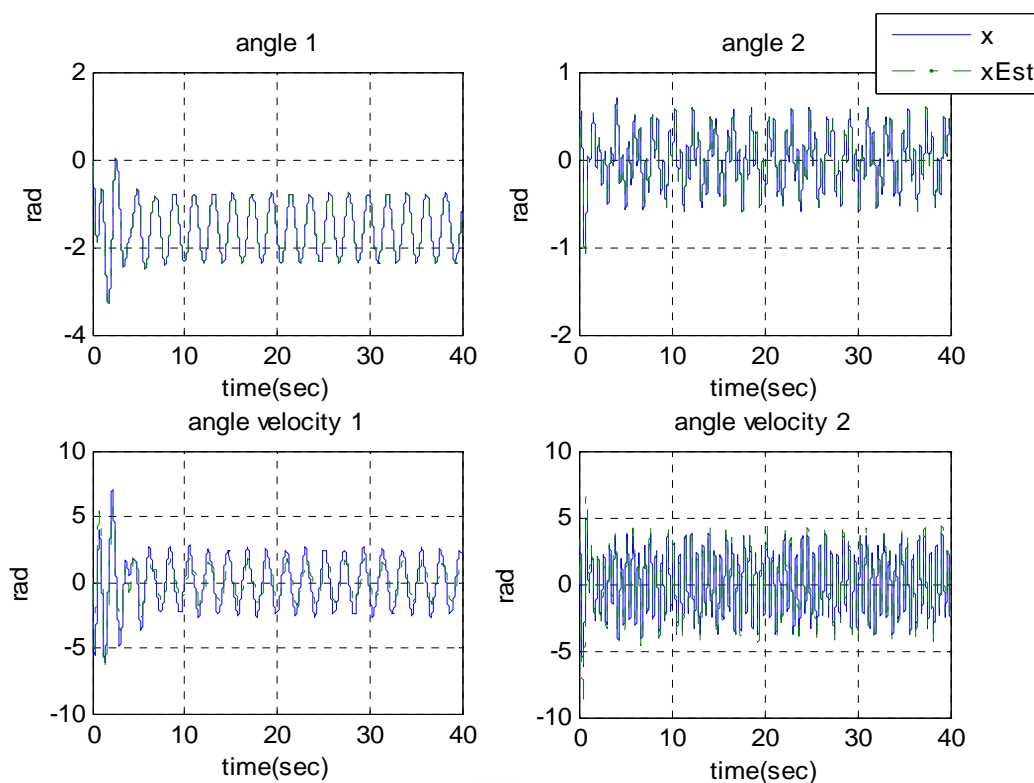


圖 5.18 模擬(十)的角度與角速度估測

從圖 5.15,5.17 比較可看出，使用運動學模型的摩擦力係數仍較使用動力學模型的摩擦力係數估測來的好。相對應的，由圖 5.16, 圖 5.18 可看出改良後的演算法在角度和角速度估測上較為精準。若將小軸慣量改變也有類似的結果，就不贅述。

5.3 力矩差值估測模擬

這小節中將模擬模型切換下力矩差值的變化，力矩差值是將非追蹤型卡曼濾波器所估測到的系統狀態(角度、角速度)代回(4.39), (4.30),(4.41)獲得。當系統沒有錯誤發生時，差值的絕對值小於錯誤門檻(fault threshold)，反之則大於錯誤門檻。

模擬條件：

令處理雜訊(process noise)與量測雜訊(measurement noise)分別為平均值為 0 標準差為 0.1 與 0.001 的高斯雜訊， $\tau_1 = 8\sin(3t)$ ， $\tau_2 = 0.5\sin(10t)$ ， $f_1 = 0.5 \times \dot{\theta}_1$ ， $f_2 = 0.1 \times \dot{\theta}_2$ 。慣量、質量與質心位置等系統參數皆與預設相同。

Simulink 架構圖

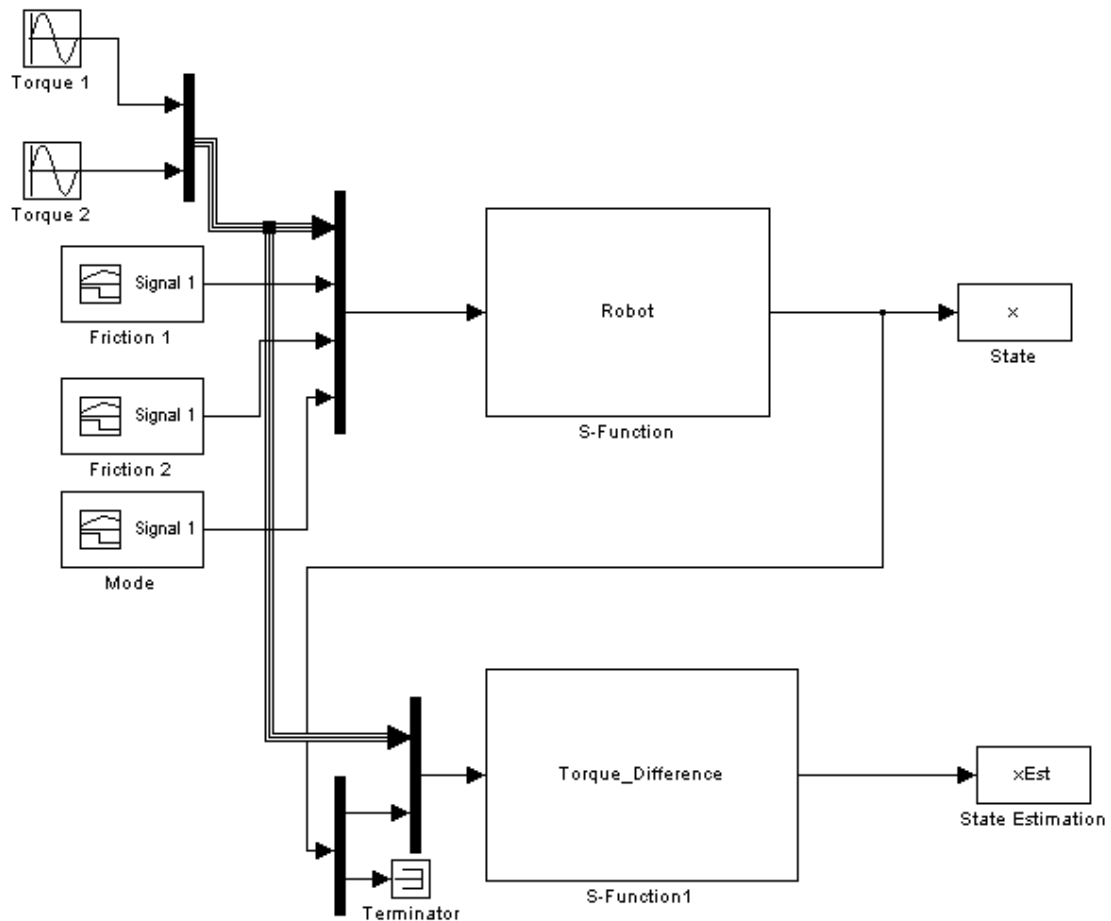


圖 5.19 力矩差值估測架構圖

圖 5.19 分為 Robot 和 Torque_Difference 兩大區塊。Robot 區塊的輸入端順序依次為力矩、摩擦力一、摩擦力二與模型選取，模型選取的範圍為 1~9，其輸出為角度與角速度。Torque_Difference 區塊的輸入端順序為依次為力矩與量測角度，其輸出值為系統狀態(角度、角速度)與計算力矩與濾波力矩差值(4.40)。

力矩差值模擬結果

模擬(十一)：在時間兩秒時由模型一跳到模型二

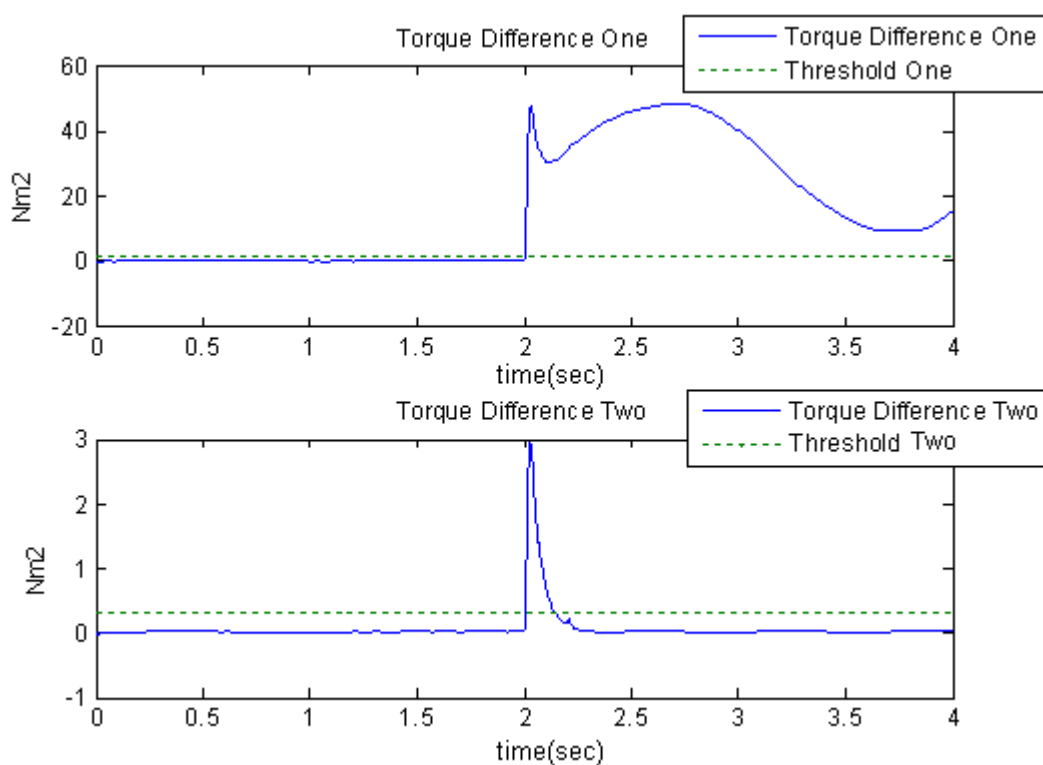


圖 5.20 模擬(十一)的力矩差值模擬估測

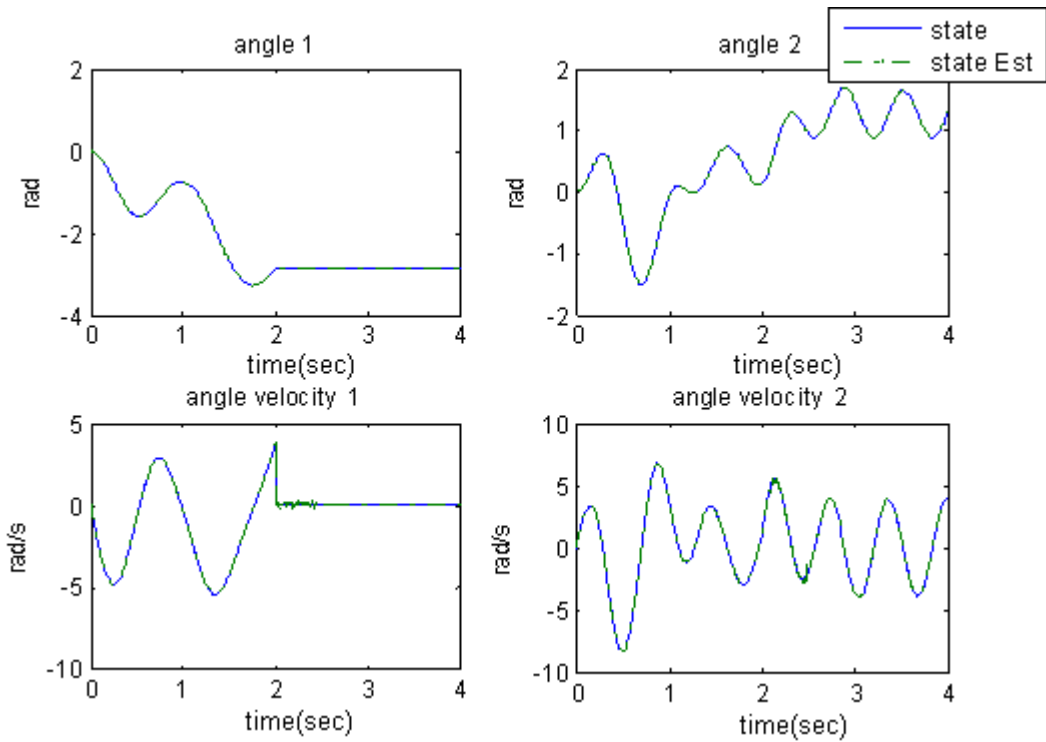


圖 5.21 模擬(十一)的角度與角速度估測

模擬(十二)：在時間兩秒時由模型一跳到模型四

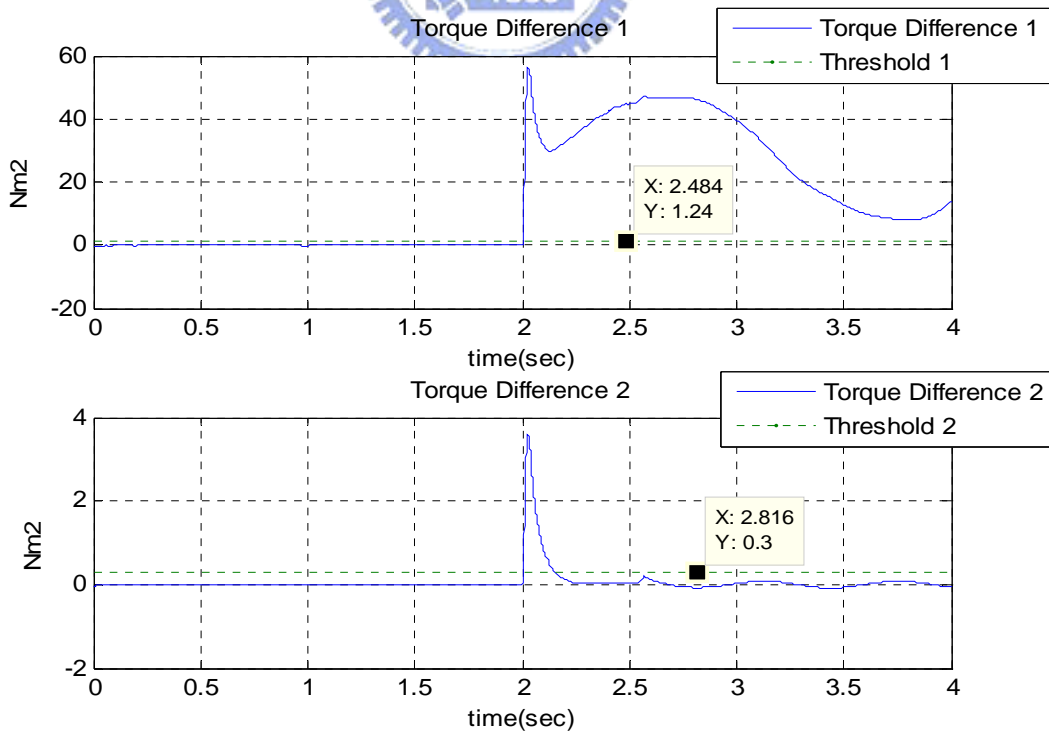


圖 5.22 模擬(十二)的力矩差值模擬估測

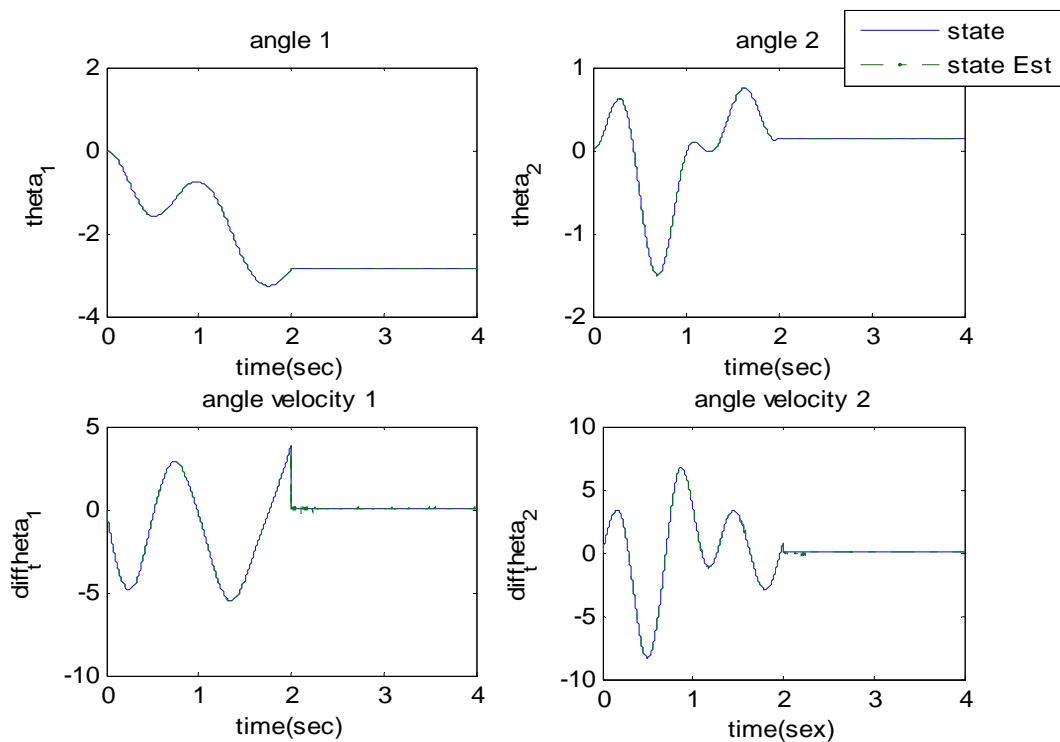


圖 5.23 模擬(十二)的角度與角速度估測

由圖 5.20 發現在時間 2 秒時，由於有錯誤發生，力矩差值一(實線, Torque Difference One)與力矩差值二(實線, Torque Difference Two)皆急速的增加，分別都超過設定的門檻一(數值為 1.24, 虛線, Threshold One)與門檻二(數值為 0.3, 虛線, Threshold Two)，所以可以確定有錯誤發生；圖 5.22 也有相同的狀況。但是比較圖 5.20 與圖 5.22 發現，圖 5.20 的力矩差值二(Torque Difference Two)的絕對值最終會回到零附近，也就在錯誤門檻二(Threshold Two)以下，而圖 5.22 的力矩差值二的絕對值則始終處於震盪。這是因為圖 5.20 是由模型一跳至模型二，只有第一軸發生故障，第二軸始終正常工作，故力矩差值二突然地超過錯誤門檻二為數值計算上所造成的暫態響應，穩態則回至零附近；反之，圖 5.22 是由模型一跳至模型四，兩軸皆進入錯誤狀態，故穩態不會回到零的附近，由圖 5.21 與圖 5.23 的角度與角速度變化也可對應到模型的切換。

接著進行模型參數不準確時的模擬，將大軸慣量變為 1.5 倍。

模擬(十三)：在時間兩秒時由模型一跳到模型二

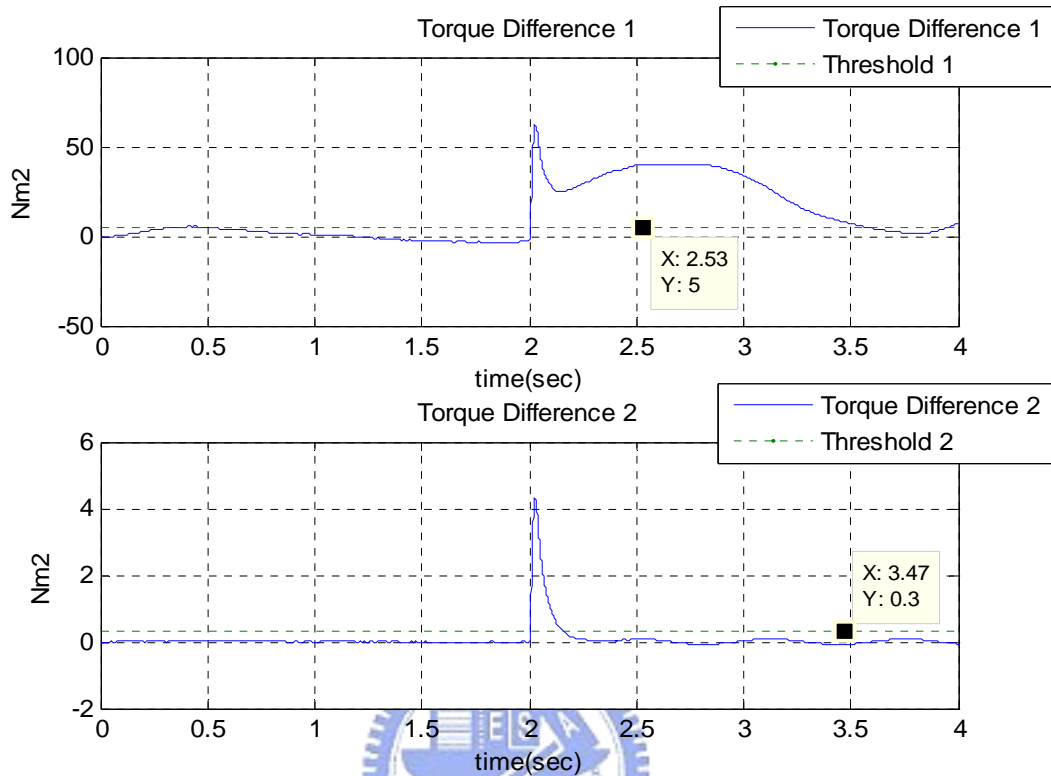


圖 5.24 模擬(十三)的力矩差值模擬估測

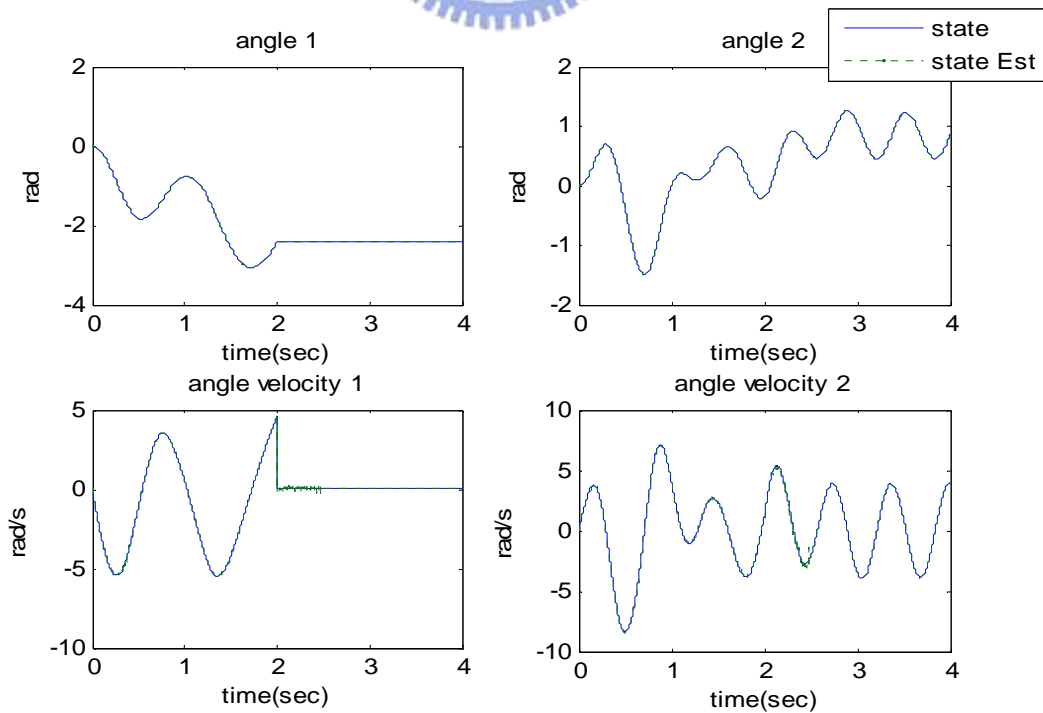


圖 5.25 模擬(十三)的角度與角速度估測

由圖 5.24 可看設定的門檻一由 1.24 調高調高為 5(虛線, ThresholdOne)與錯誤門檻二則維持原值數值 0.3(虛線, Threshold Two)。這是因為模型不準確下，若是維持原來的門檻(大軸 1.24,小軸 0.3)則有可能在 0.3 秒就發生錯誤的判斷，為了避免誤判發生，故將錯誤門檻調高，由圖 5.25 的角度與角速度可看到模型切換的過程。

5.4 動力學模型與運動學模型切換模擬



本論文錯誤偵測與識別演算法的前端為動力學模型(Dynamic Model)與運動學模型(Kinematics Model)的多模型組態 GPB-2，以降低計算量。本小節即是將模擬當錯誤發生時，動力學模型與運動學模型之間的機率改變與狀態估測結果。

模擬條件：

令處理雜訊(process noise)與量測雜訊(measurement noise)分別為平均值為 0 標準差為 0.1 與 0.001 的高斯雜訊， $\tau_1 = 8\sin(3t)$ ， $\tau_2 = 0.5\sin(10t)$ ， $f_1 = 0.5 \times \dot{\theta}_1$ ， $f_2 = 0.1 \times \dot{\theta}_2$ 。慣量、質量與質心位置等系統參數皆與預設相同。

Simulink 架構圖：

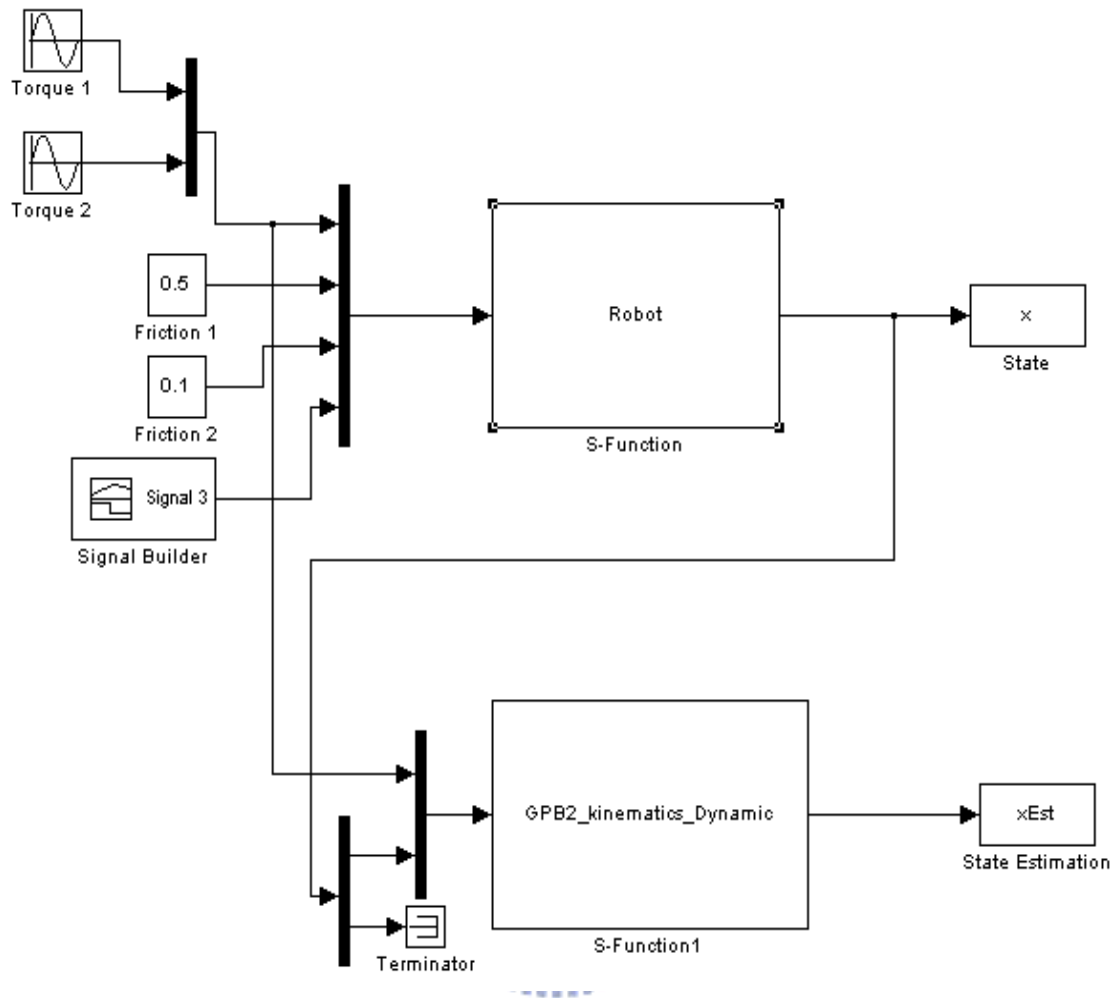


圖 5.26 動力學模型與運動學模型機率估測架構圖

圖 5.26 的分為 Robot 和 GPB2_Kinematics_Dynamic 兩個區塊。Robot 區塊代表二軸機械臂的模型，其輸入端順序依次為力矩、摩擦力一、摩擦力二與模型選取，模型選取的範圍為 1~9，其輸出為角度與角速度。GPB2_Kinematics_Dynamic 的區塊輸入端順序依次為力矩、量測角度，摩擦力係數已經配合選取的摩擦力係數寫入區塊 GPB2_Kinematics_Dynamic，其輸出值為角度、角速度估測值與模型機率。

模擬結果

模擬(十四)：於 2 秒的時候由模型一跳至模型二

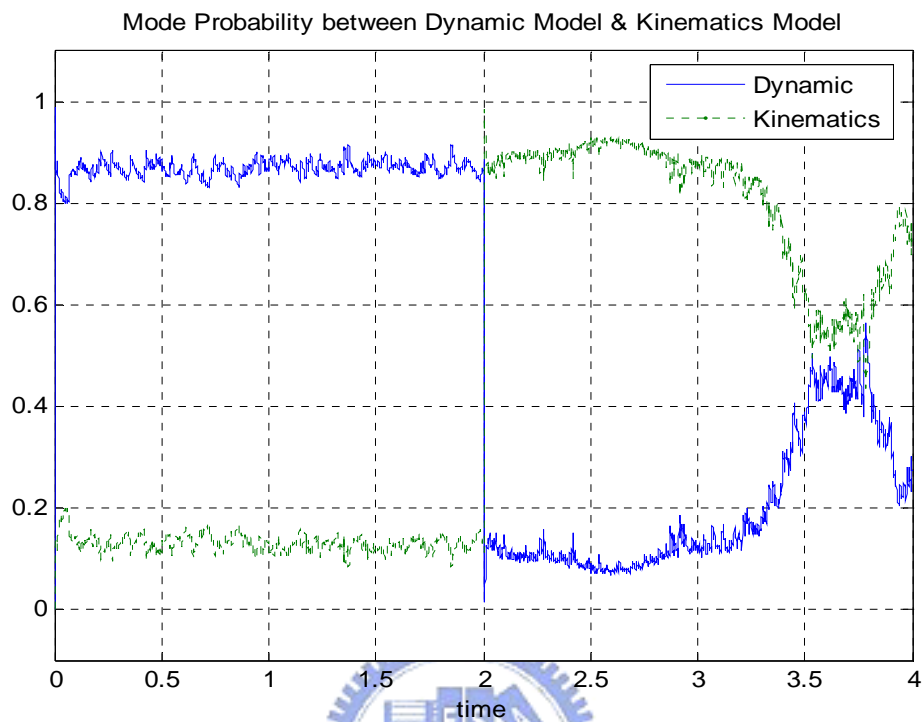


圖 5.27 模擬(十四)的動力學模型與運動學模型機率估測

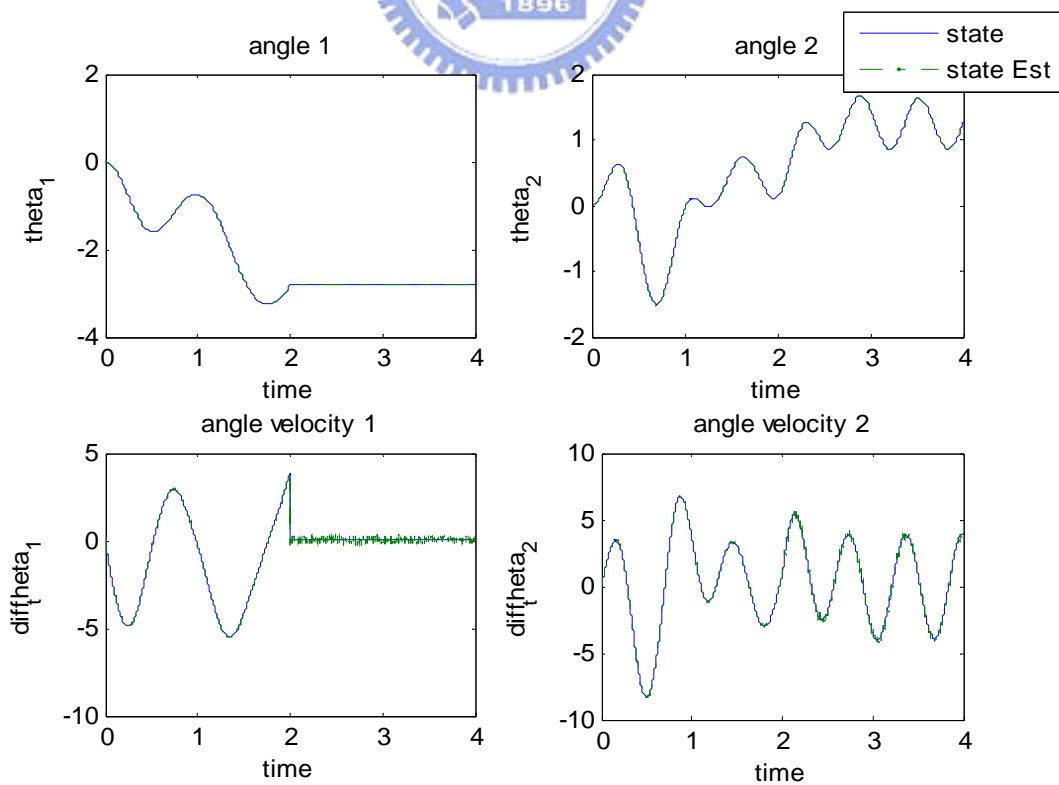


圖 5.28 模擬(十四)的動力學模型與運動學模型狀態估測

模擬(十五)：於 2 秒的時候由模型一跳至模型四

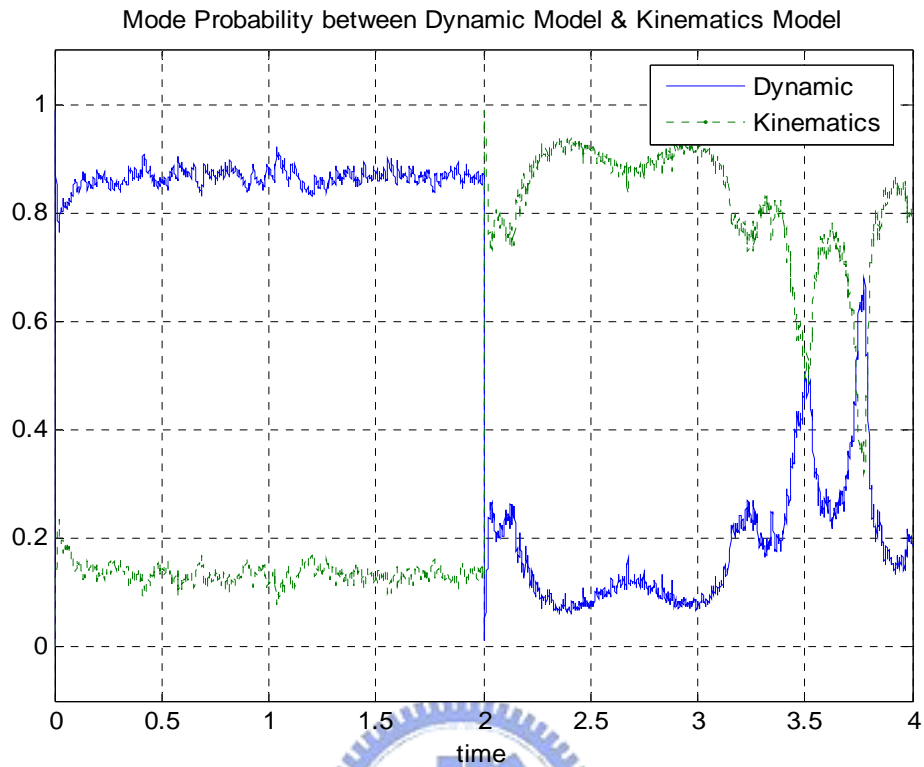


圖 5.29 模擬(十五)的動力學模型與運動學模型機率估測

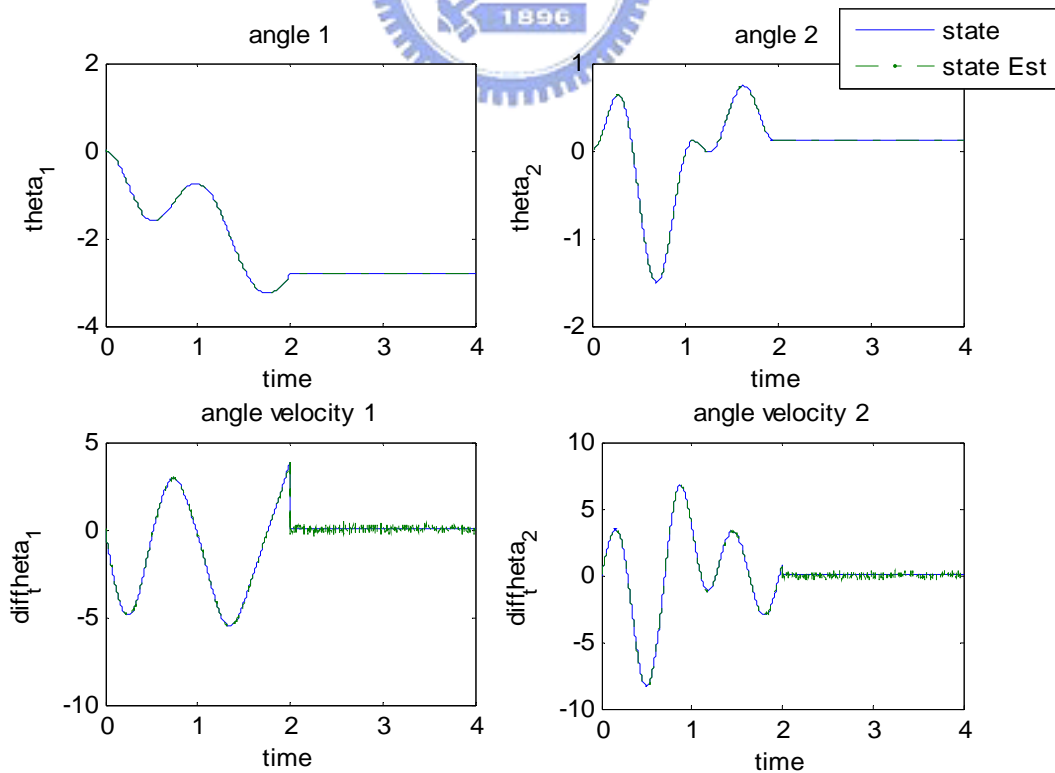


圖 5.30 模擬(十五)的動力學模型與運動學模型狀態估測

由圖 5.27、圖 5.29 可見到當模型切換時，動力學模型(Dynamic Model)與運動學模型(Kinematics Model)的機率幾乎會瞬間交換，指出現在有錯誤發生。這是因為當系統沒有錯誤發生時，動力學模型較運動學模型精準，反之則否：故錯誤發生前由動力學模型機率主導估測輸出，之後則由運動學模型主導估測輸出。觀察圖 5.28 與圖 5.30 可發現在 2 秒之後，角速度的估測有相當大的雜訊，2 秒之前則雜訊較小，這就是動力學模型較運動學模型精準的證據。

5.5 應用於機械臂之即時演算法模擬

由 5.1 至 5.4 的四個小節，可看出各個演算法組成的各個區塊的模擬都有相當好的結果，接著就要對整合後的演算法進行模擬。但在這之前，先看看原始的 GPB-2 演算法在模擬上的結果，再進行改良演算法的模擬，以便將兩者比較優缺。在模擬(十六)與模擬(十七)中，標示的「模擬所需時間」代表在本機上跑完這 4 秒所需的時間。另外定義「誤判率」為錯誤判斷的取樣點除上整段模擬的取樣點數，在本模擬中由於模擬時間 4 秒鐘而取樣頻率為 1K(Hz)，故總取樣點數為 4000 點。

模擬條件：

原始的和改良後的演算法皆令處理雜訊(process noise)與量測雜訊(measurement noise)分別為平均值為 0 標準差為 0.1 與 0.001 的高斯雜訊， $\tau_1 = 8\sin(3t)$ ， $\tau_2 = 0.5\sin(10t)$ ， $f_1 = 0.5 \times \dot{\theta}_1$ ， $f_2 = 0.1 \times \dot{\theta}_2$ 。慣量、質量與質心位置等系統參數皆與預設相同。

模擬架構：

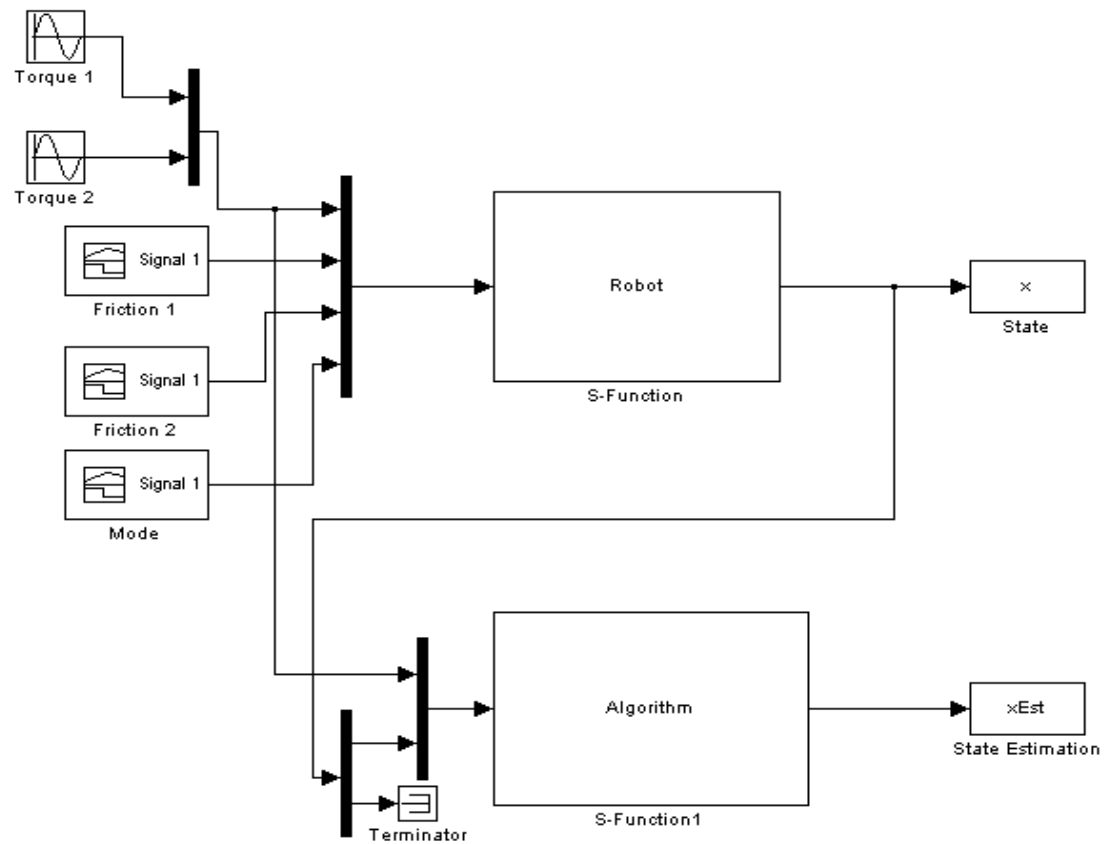


圖 5.31 應用於機械臂之即時演算法模擬架構圖

圖 5.31 的分為 Robot 和 Algorithm 兩個區塊。Robot 區塊代表二軸機械臂的模型，其輸入端順序依次為力矩、摩擦力一、摩擦力二與模型選取，模型選取的範圍為 1~9，其輸出為角度與角速度。這時的摩擦力一與摩擦力二皆改為可變訊號而非定值，這是為了要模擬摩擦力係數變化下的估測結果。Algorithm 的區塊不管原始 GPB-2 演算法或改良後的演算法，其輸入端順序皆依次為力矩、量測角度，其輸出值為角度、角速度估測值、模型機率，並判斷現在系統在哪個模型下運作，新演算法則增加摩擦力估測值。

舊演算法模擬結果

模擬(十六)：系統於 2 秒時由模型一跳至模型二

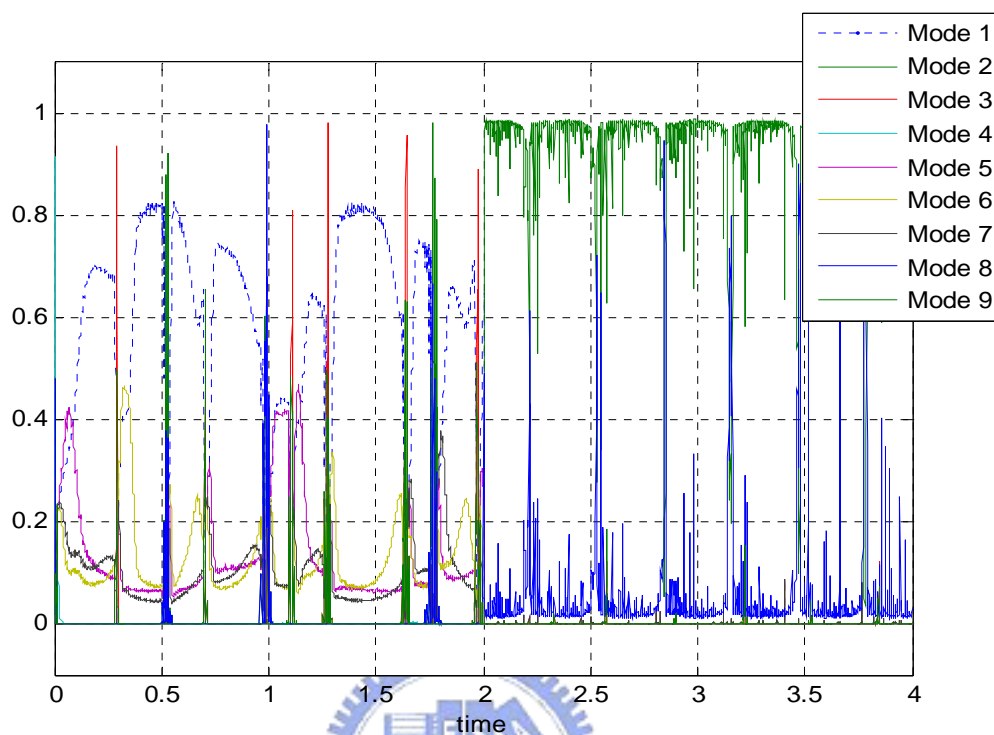


圖 5.32 模擬(十六)的機率估測

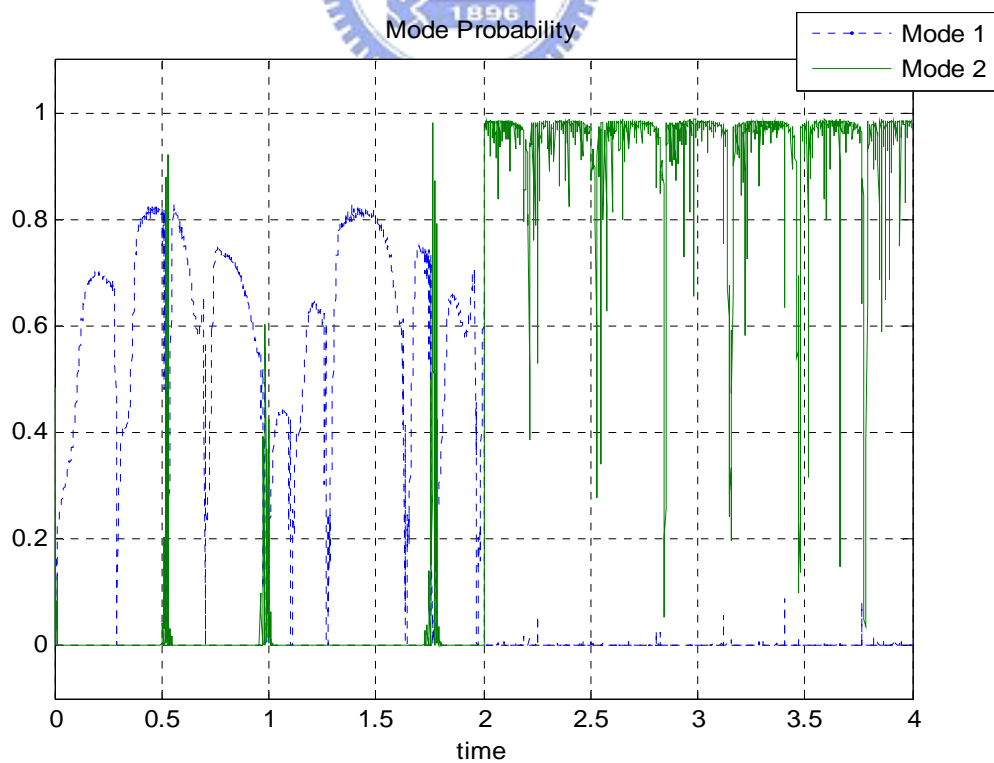


圖 5.33 模擬(十六)的模型一與模型二機率估測

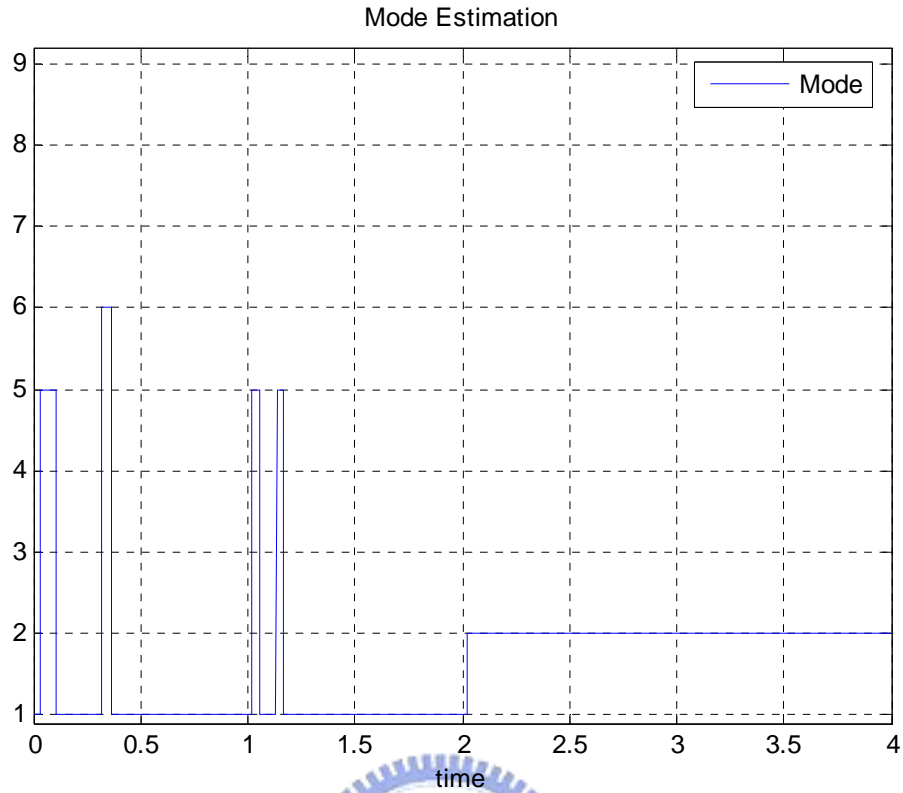


圖 5.34 模擬(十六)的模型估測

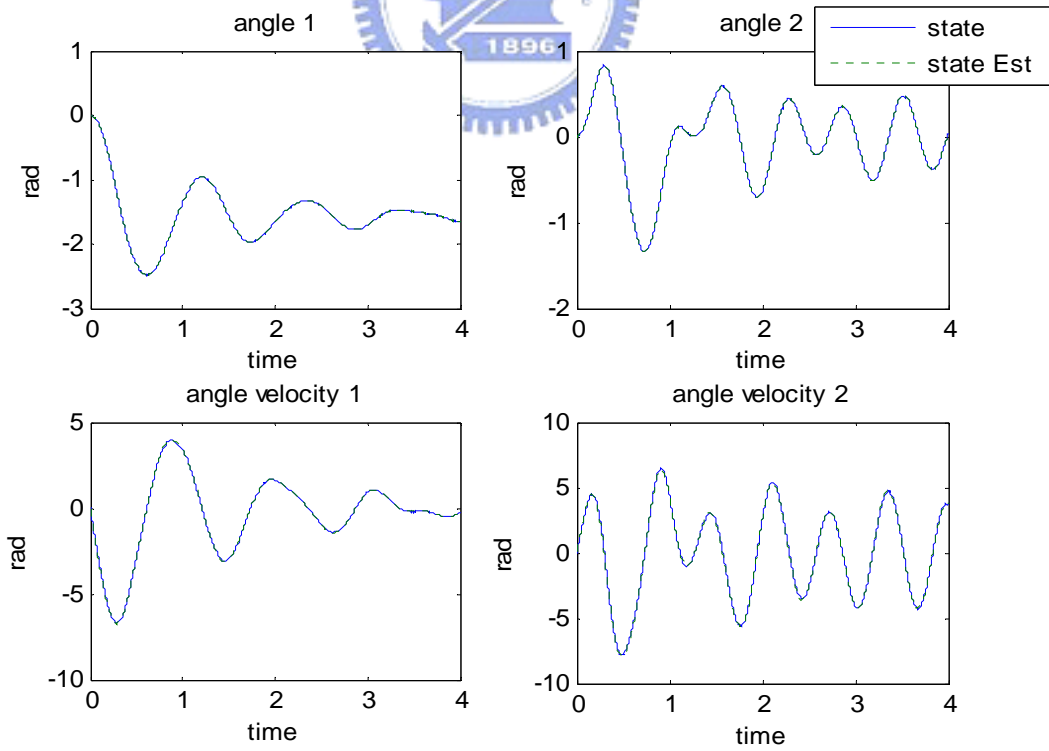


圖 5.35 模擬(十六)的狀態估測

模擬所需時間：15 分 23 秒

誤判率：8.9%

模擬(十七)：系統於 2 秒時由模型一跳至模型八

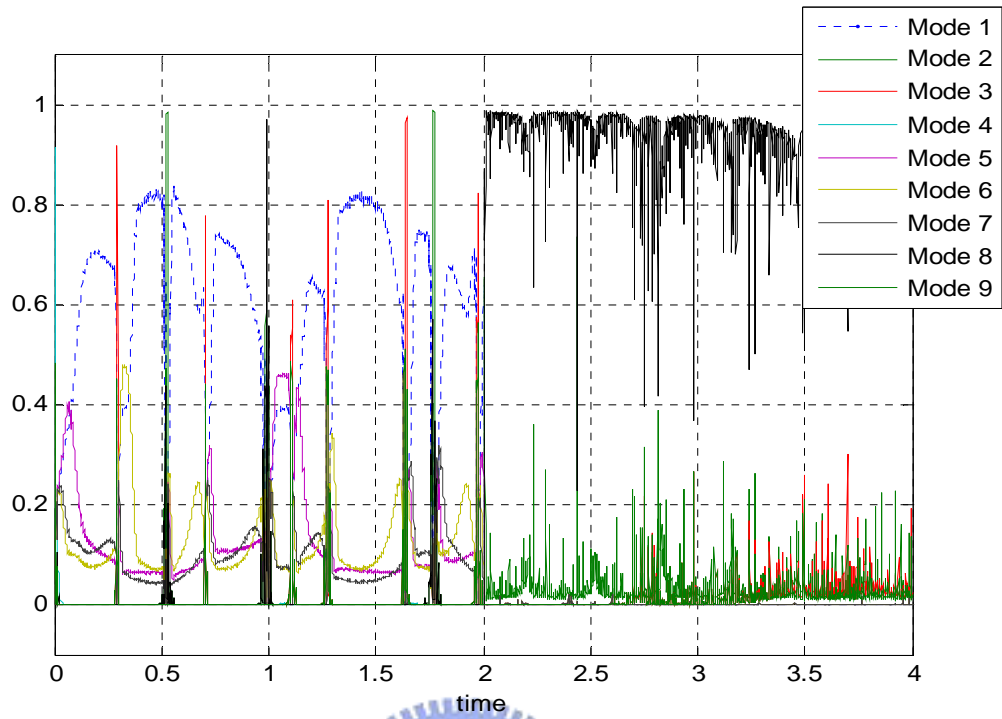


圖 5.36 模擬(十七)的機率估測

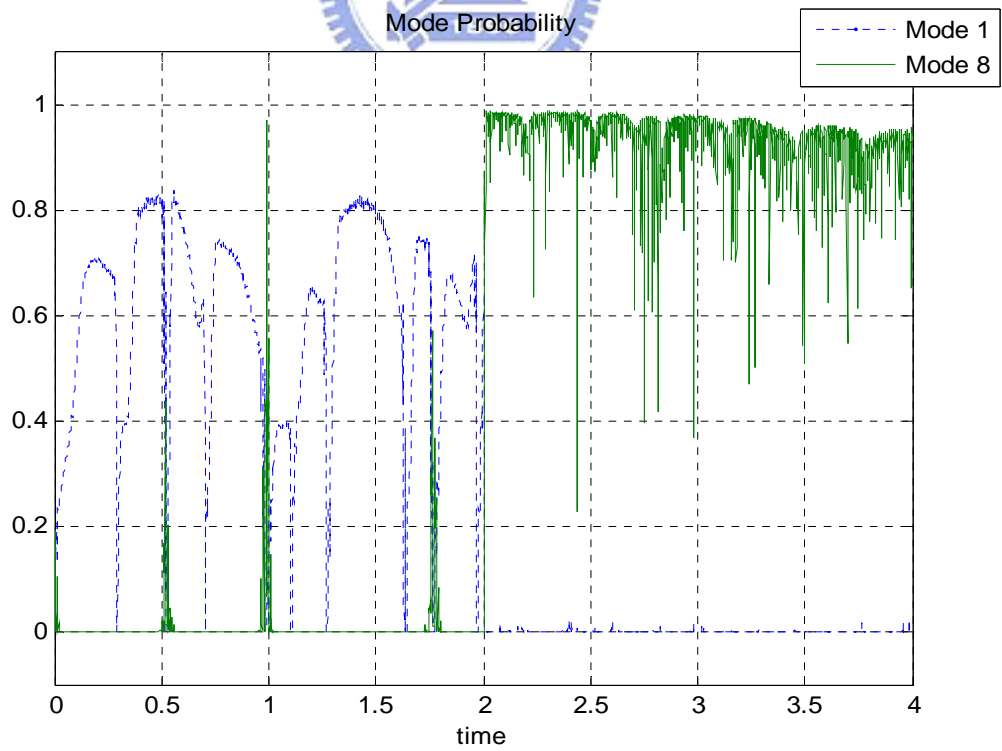


圖 5.37 模擬(十七)的模型一與模型八機率估測

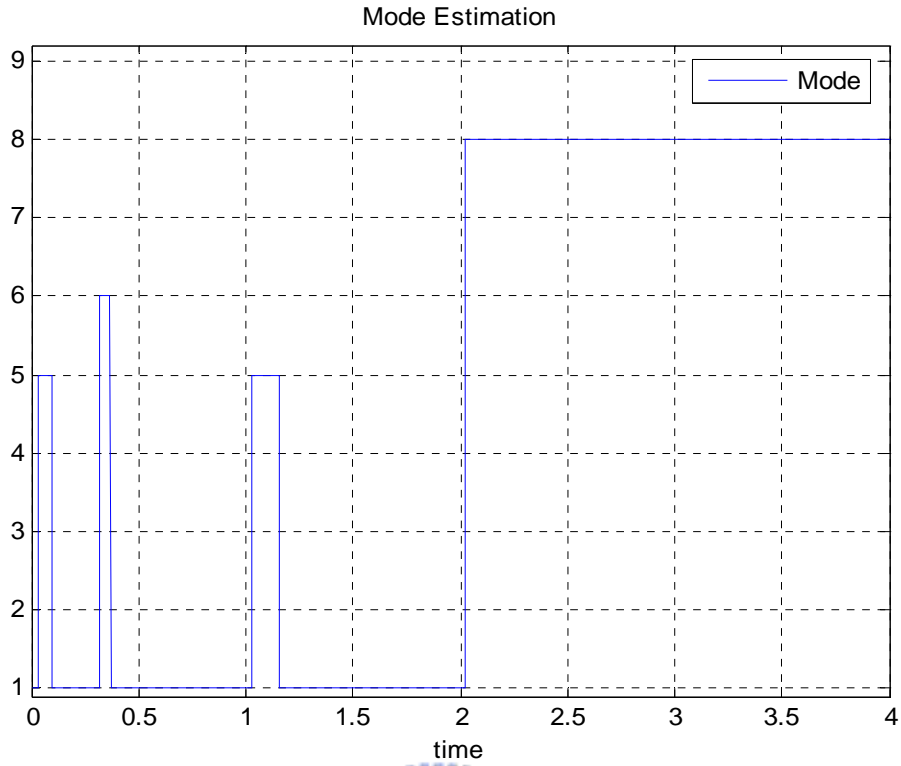


圖 5.38 模擬(十七)的模型估測

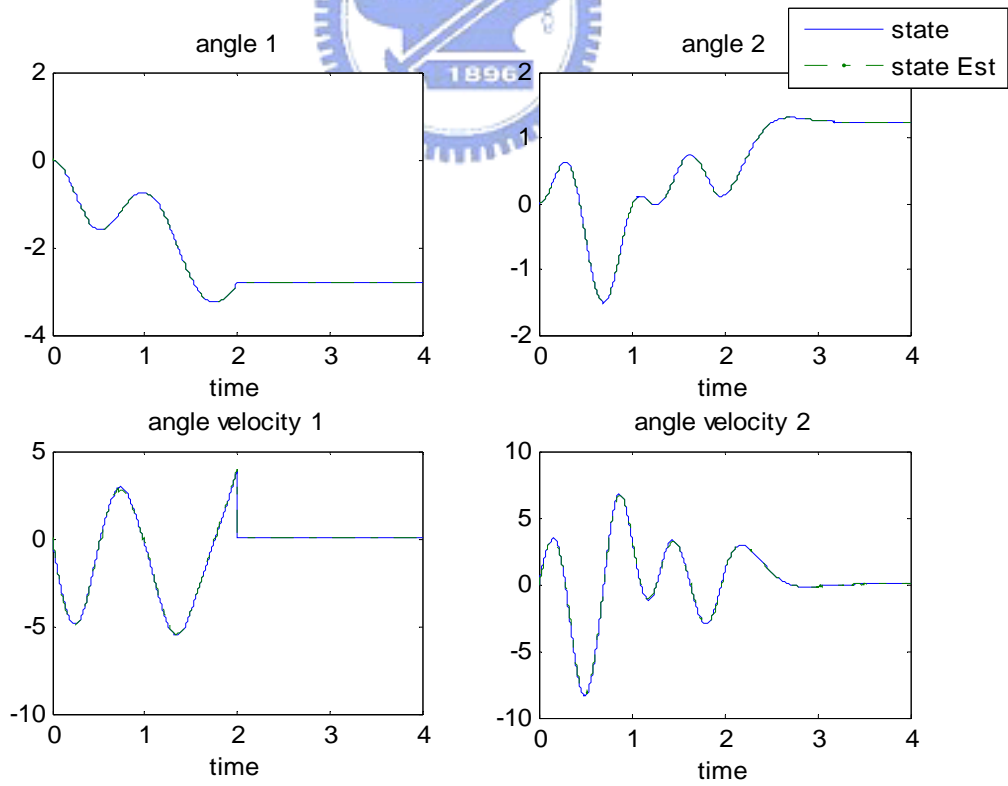


圖 5.39 模擬(十七)的狀態估測

模擬所需時間：15 分 6 秒

誤判率：10.8%

由模擬(十六)與模擬(十七)的結果，可以發現舊演算法在狀態估測上具有優異的性質(圖 5.35,圖 5.39)，但比照圖 5.32,圖 5.34,圖 5.36 與圖 5.38 即可發現該演算法在模型機率估測與模型估測上卻常有發生誤判的情形。這是因為模型太多的情況下，容易發生該時間點的某些模型的輸出狀態非常相似，以至於計算之相似值(Likelihood)過於接近而產生誤判，其誤判率分別為 8.9%與 10.8%。此外，舊演算法太耗計算量，以本電腦的配備為 P4-3.0G(Hz)，短短的 4 秒模擬，模擬(十六)與模擬(十七)仍分別需要 15 分 23 秒與 15 分 6 秒才能跑完，因此幾乎不可能達到即時運算的要求。此外，模擬(十六)與模擬(十七)皆在 2 秒有錯誤發生，比較圖 5.33 與圖 5.37，錯誤發生後相對應的錯誤模型的機率立即上升，也就是成功偵測錯誤，由圖 5.34, 圖 5.38 也可看到對應的模型估測。



新演算法模擬結果

模擬(十八)：於 2 秒的時候由模型一跳至模型二

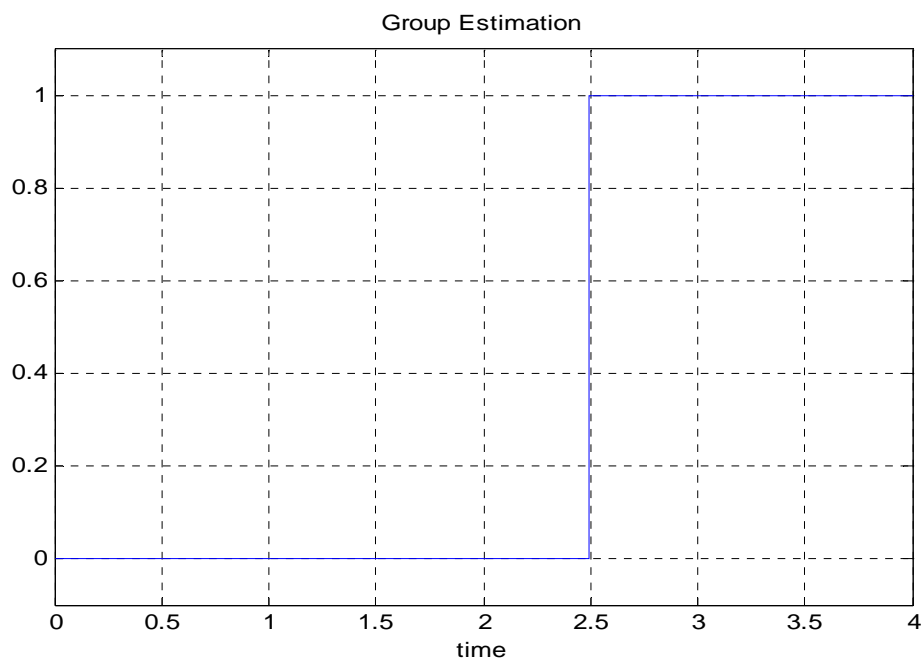


圖 5.40 模擬(十八)的群組選取估測

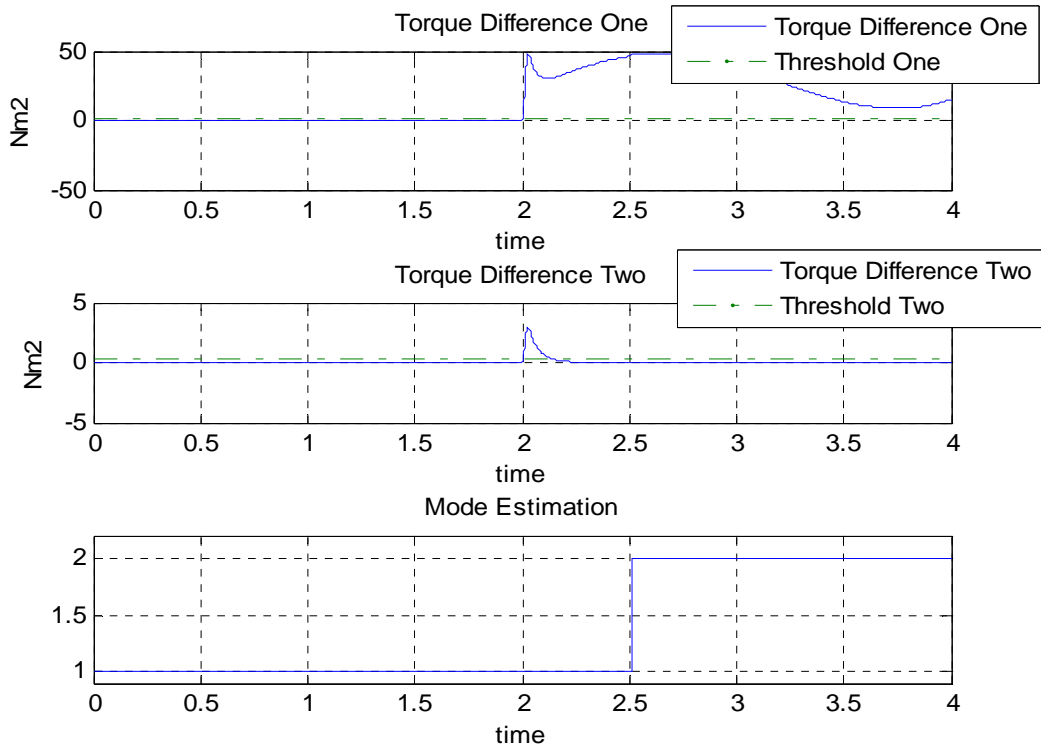


圖 5.41 模擬(十八)的力矩差值與模型選取估測

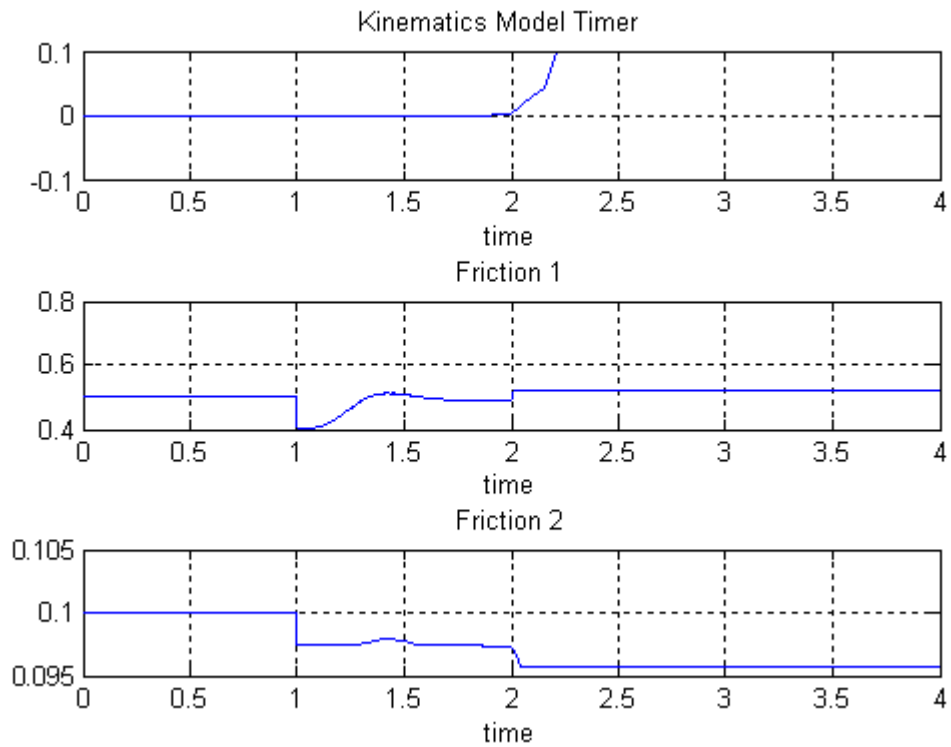


圖 5.42 模擬(十八)的摩擦力估測

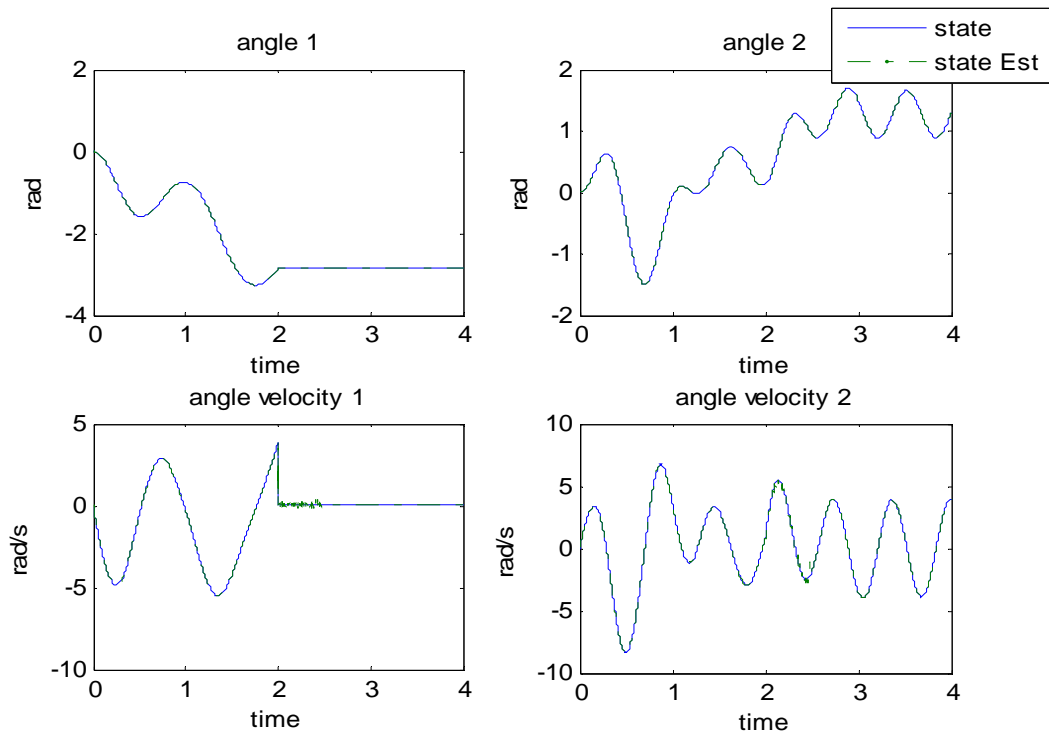


圖 5.43 模擬(十八)的狀態估測

所需時間：38 秒

誤判率：12.5%

模擬(十九)：於 2 秒的時候由模型一跳至模型八

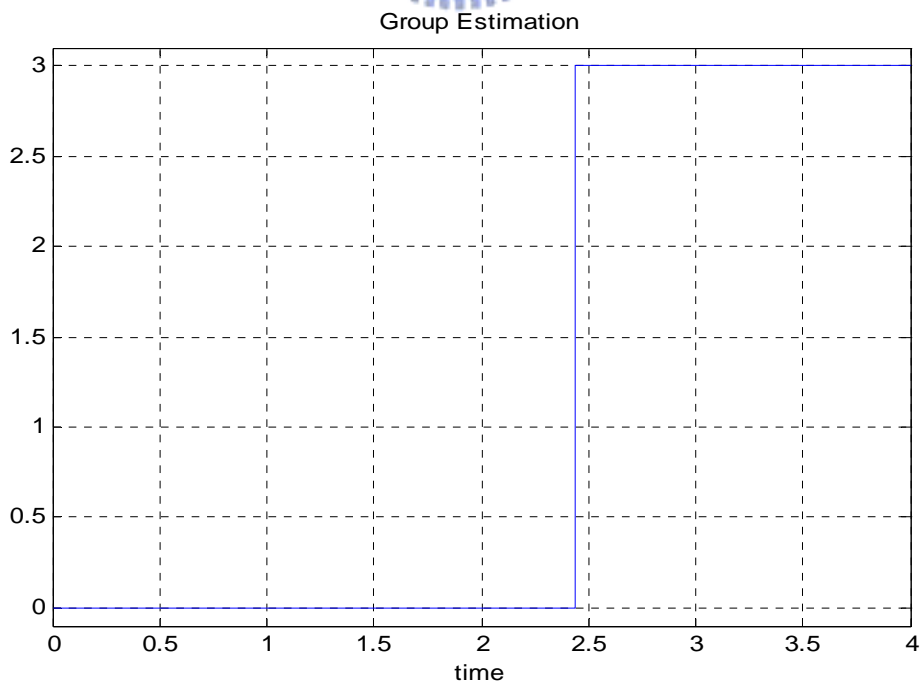


圖 5.44 模擬(十九)的群組選取估測

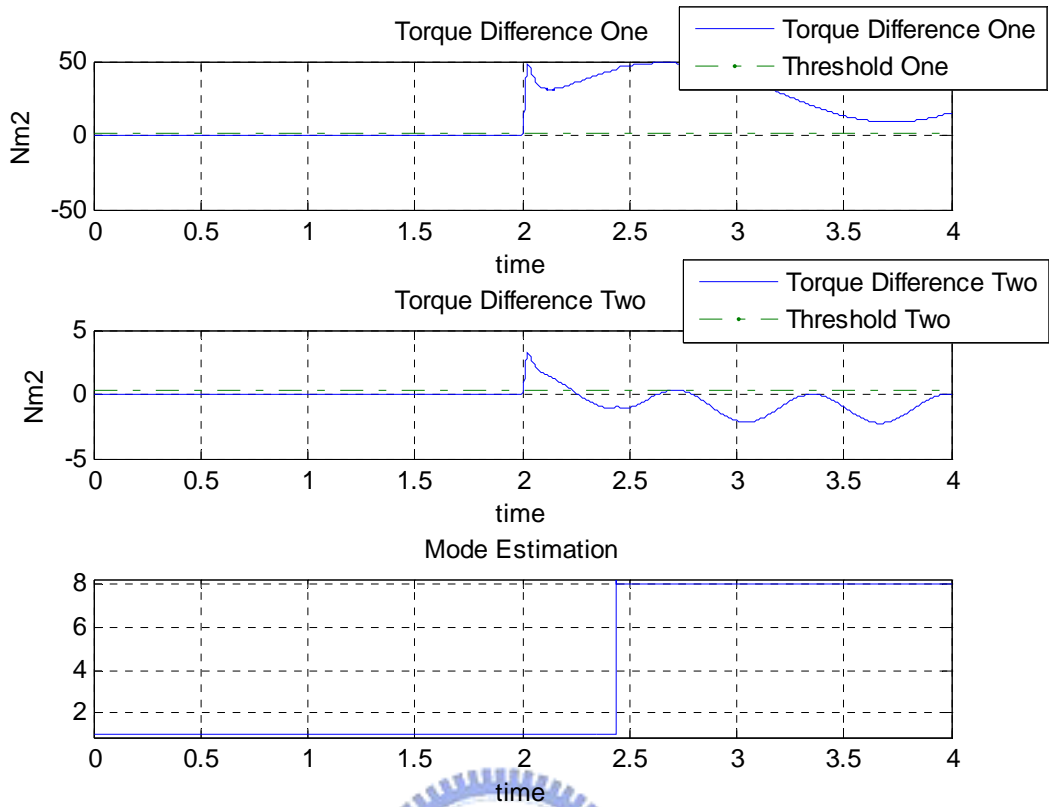


圖 5.45 模擬(十九)的力矩差值與模型選取估測

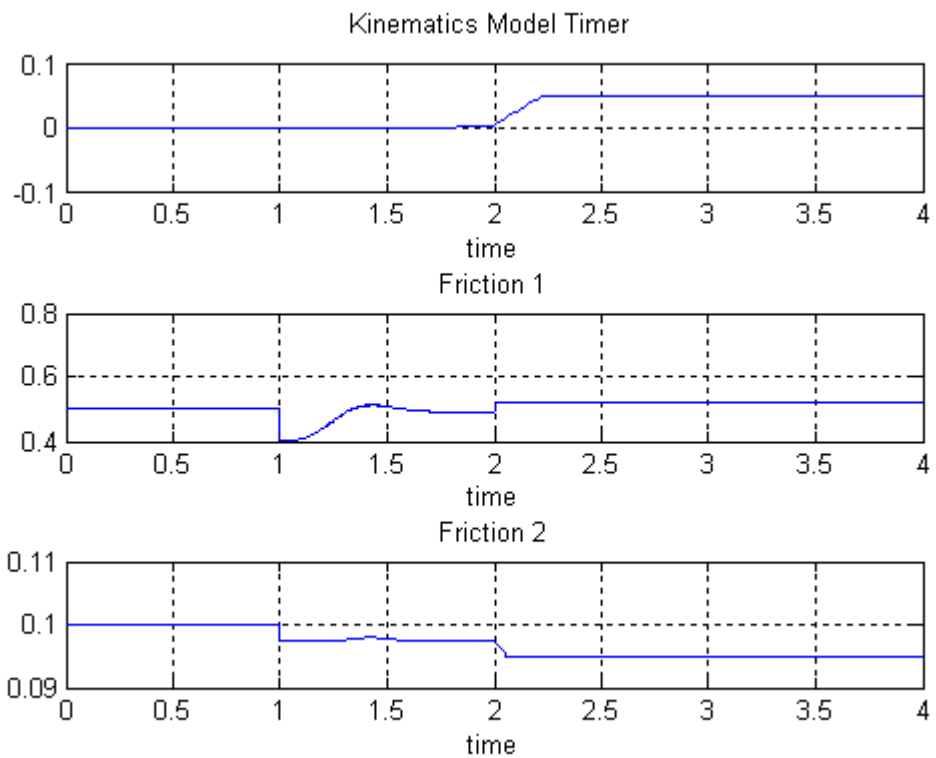


圖 5.46 模擬(十九)的摩擦力估測

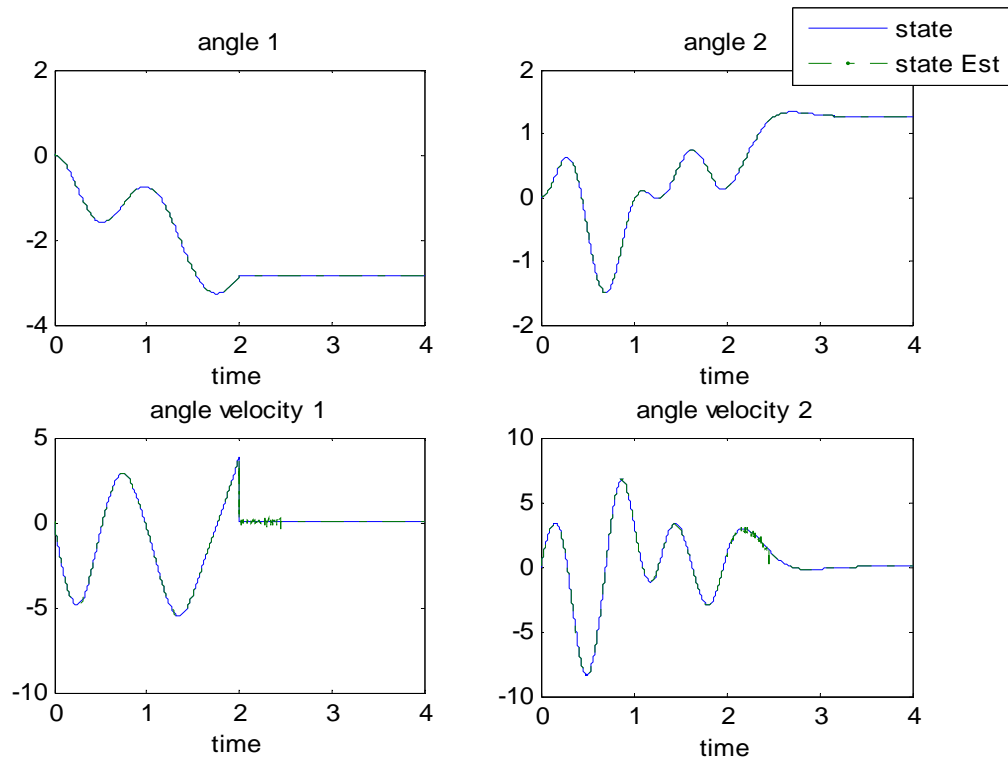


圖 5.47 模擬(十九)的狀態估測

所需時間：1 分 9 秒

誤判率：10.8%



由模擬(十八)與模擬(十九)驗證此即時演算法的確能在系統錯誤發生的情況下，即時地偵測是否有錯誤發生並識別錯誤的種類。模擬(十八)為系統在兩秒時發生錯誤(第一軸發生卡住錯誤、第二軸仍然正常工作)，因此由圖 5.41 能發現第二軸的力矩差值(Torque Difference)的絕對值，穩態時會回到錯誤門檻的虛線以下，第一軸的力矩差值的絕對值，穩態時則不會永遠地回到錯誤門檻之下。由圖 5.40 與圖 5.41 能看到演算法將錯誤分類在第一群組(只有第一軸壞掉)和模型二，這與實際錯誤發生的種類是吻合的。

同理，在模擬(十九)中，系統在兩秒時由模型一跳至模型八，也就是由兩軸正常工作轉變成第一軸卡住與第二軸自由甩動，由圖 5.45 可觀察到兩軸的力矩差值的絕對值的穩態皆不會永遠地在錯誤門檻

(Threshold)之下。由圖 5.44 與圖 5.45 可見到演算法將錯誤分類在第三群組(兩軸都壞掉)和模型八，這與實際錯誤發生的種類是吻合的。

由圖 5.42 與圖 5.46 可看到在 1 秒之前，摩擦力係數預設為與實際相同，在 1 秒的時候再啟動摩擦力估測，在啟動摩擦力估測後到 2 秒時系統發生錯誤之間，摩擦力係數估測與實際值非常接近。在 2 秒時，由於運動學模型計時器(Kinematics Model Timer，圖 5.42,圖 5.46 第一小圖，此計時器在運動學模型的機率連續地大於 0.8 時，則開始計算運動學模型連續地主導狀態輸出的時間，在模擬中設定 0.015 秒為進入群組分類與停止摩擦力估測之錯誤門檻)開始增加，故判定有錯誤發生，而停止摩擦力估測改以之前估測的摩擦力係數取代。

比較模擬(十六)、模擬(十八)與模擬(十七)、模擬(十九)，可見不論是新舊演算法在狀態估測上都相當不錯(圖 5.29 與圖 5.37，圖 5.39 與圖 5.47)。在錯誤發生後的模型偵測上，新演算法由模擬(十六)的 24 個取樣點升至模擬(十八)的 500 個取樣點，由模擬(十七)的 26 個取樣點升至模擬(十九)的 431 個取樣點，這是因為新演算法需要更多取樣點來由力矩變化判斷所屬群組。此外，雖然模擬(十六)、模擬(十八)與模擬(十七)、模擬(十九)的誤判率皆差不多，但因分群後同時估測的模型數變少，故誤判只會發生在錯誤模型切換的過程，而不會在模型沒有切換時發生，總計算量也因分群也大幅縮減。第一軸卡住故障的總模擬時間由模擬(十六)的 15 分 23 秒縮短為模擬(十八)的 38 秒，兩軸皆故障的總模擬時間由模擬(十七)的 15 分 6 秒縮短為模擬(十九)的 1 分 9 秒。

接著測試當模型不準確的情況下，本論文演算法的估測效果。模擬(二十)將大軸慣量變為原來的 1.5 倍，小軸慣量不變；模擬(二十一)則將小軸慣量調為 1.2 倍，大軸慣量不變。依照模擬(十三)將大軸錯誤門檻調高為 $5(N \cdot m^2)$ ，小軸錯誤門檻仍維持原來的 $0.3(N \cdot m^2)$ 。

模擬(二十)：於 2 秒時由模型一跳至模型二，大軸慣量變為 1.5 倍

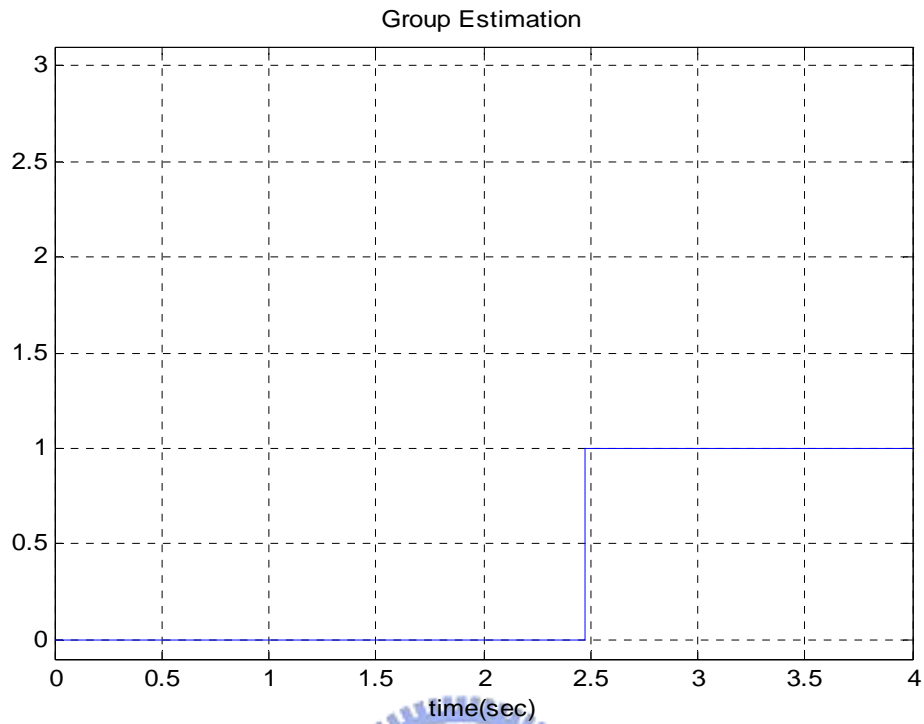


圖 5.48 模擬(二十)的群組選取估測

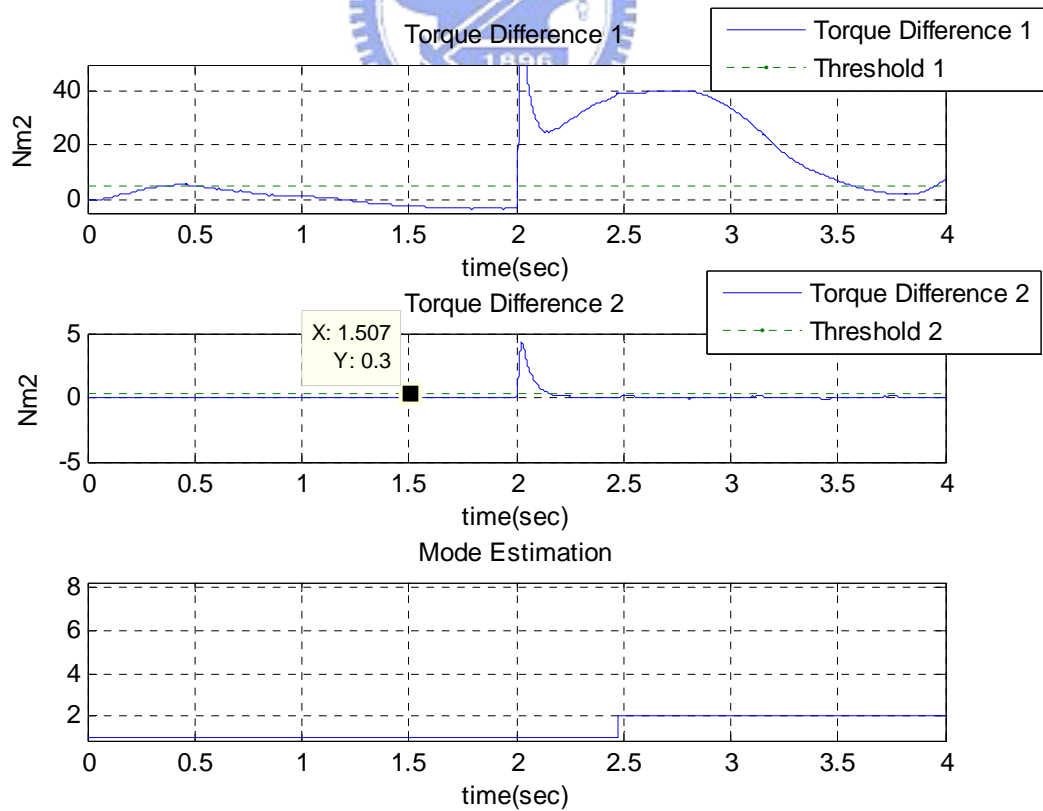


圖 5.49 模擬(二十)的力矩差值與模型選取估測

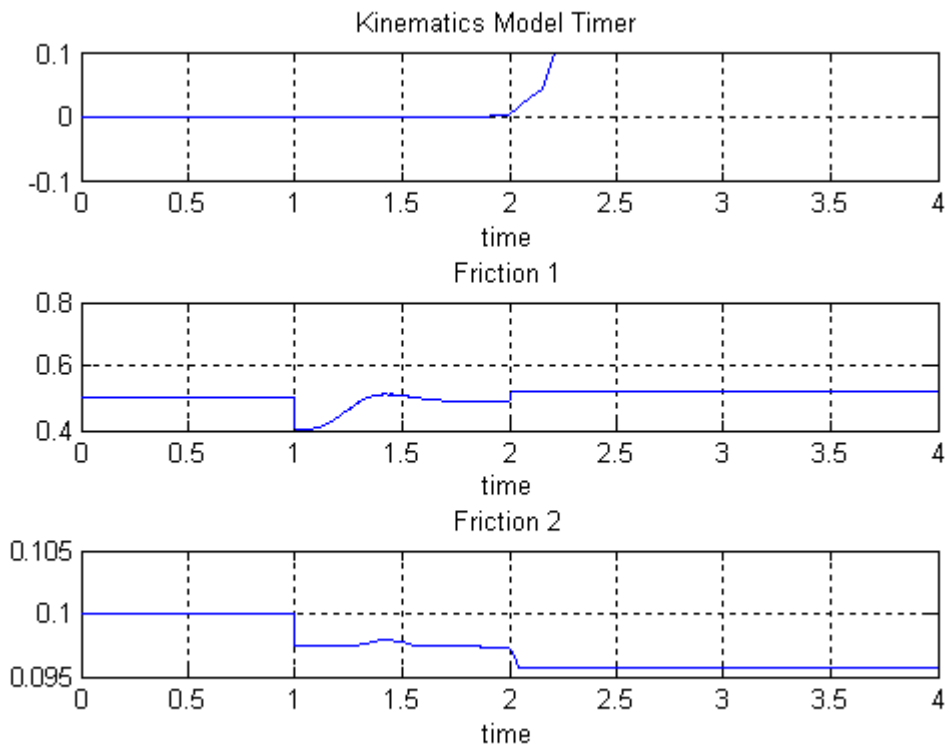


圖 5.50 模擬(二十)的摩擦力估測

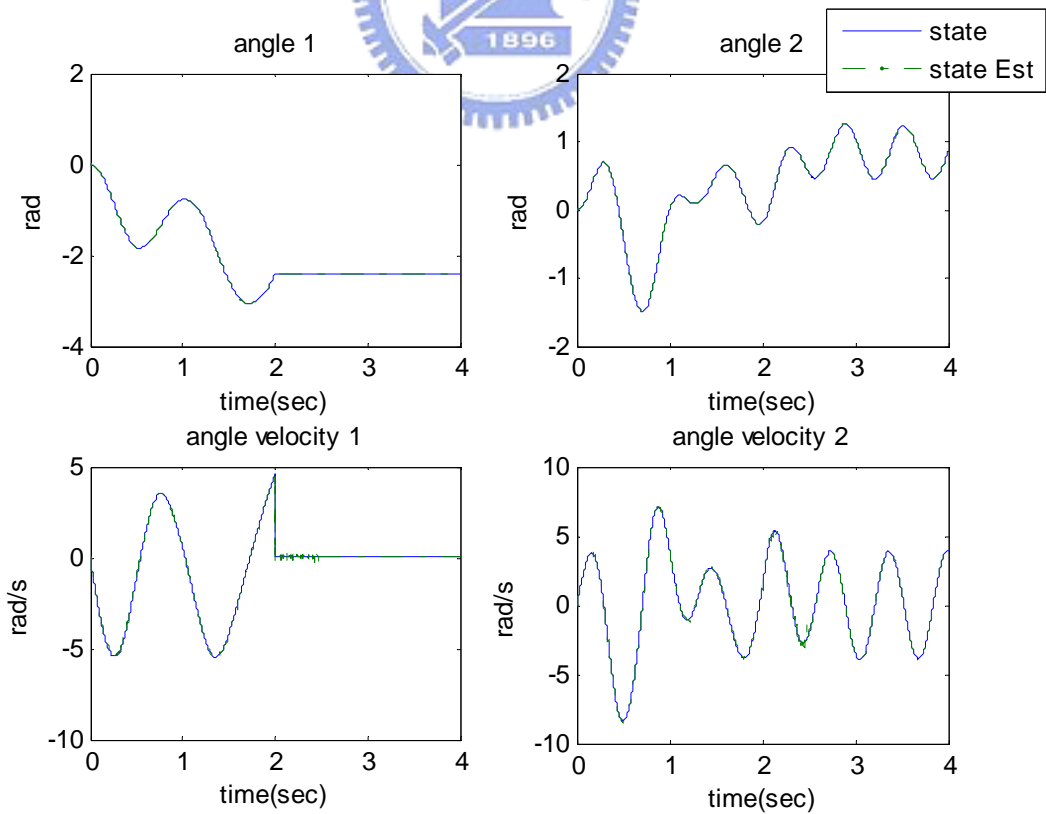


圖 5.51 模擬(二十)的狀態估測

模擬(二十一)：於 2 秒時由模型一跳至模型八，小軸慣量變為 1.2 倍

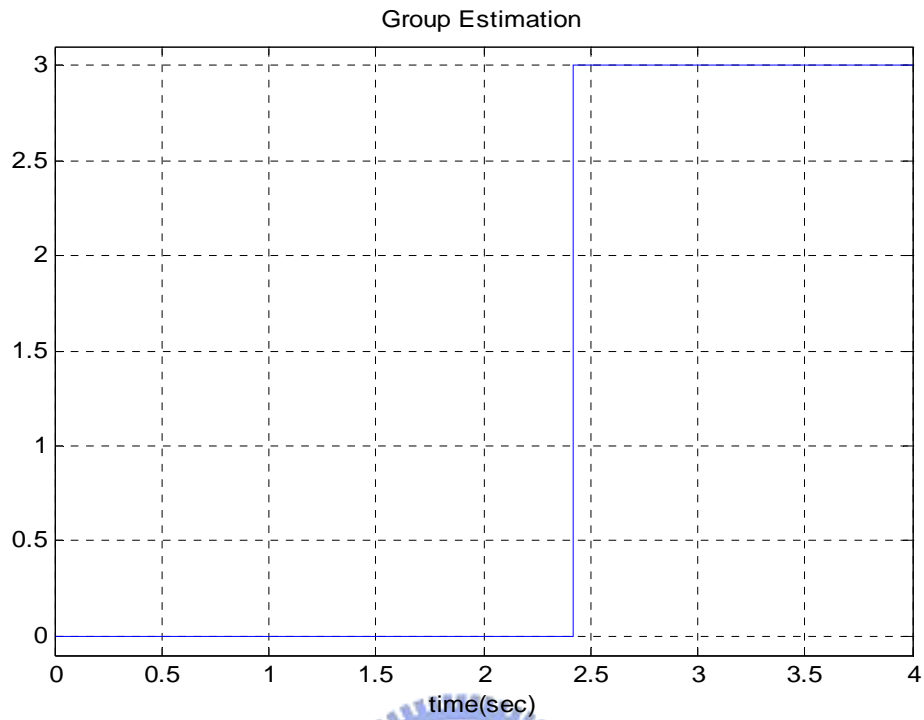


圖 5.52 模擬(二十一)的群組選取估測

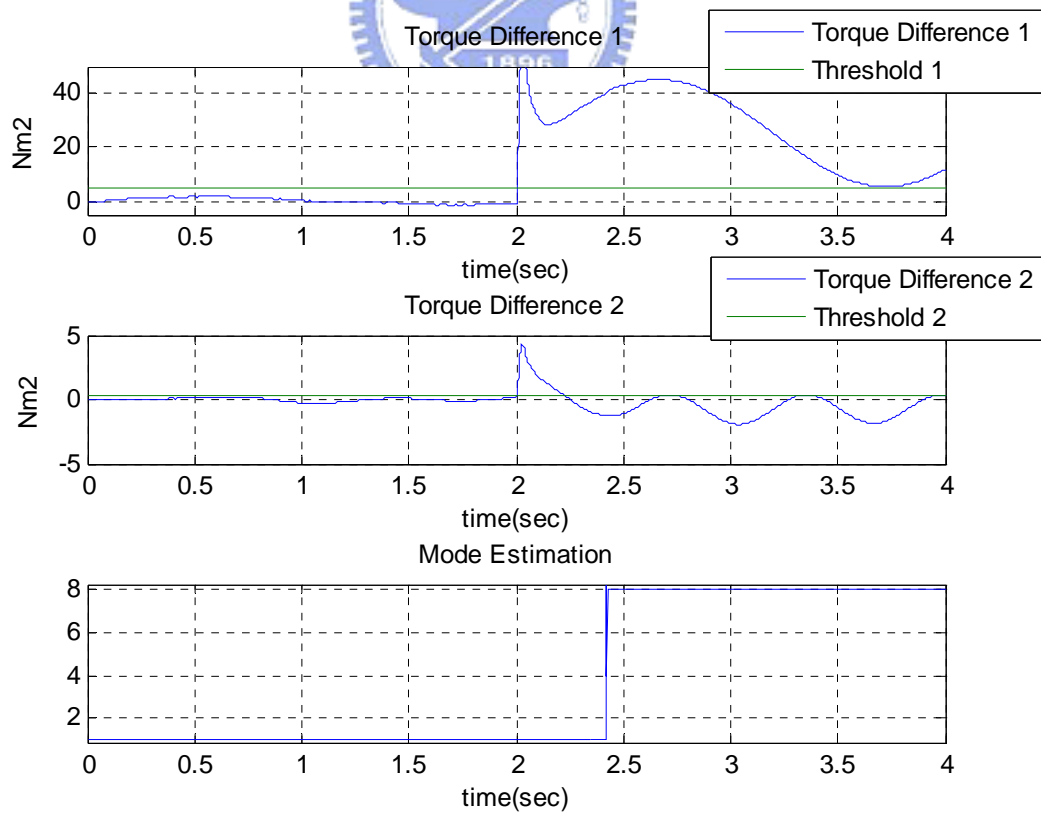


圖 5.53 模擬(二十一)的力矩差值與模型選取估測

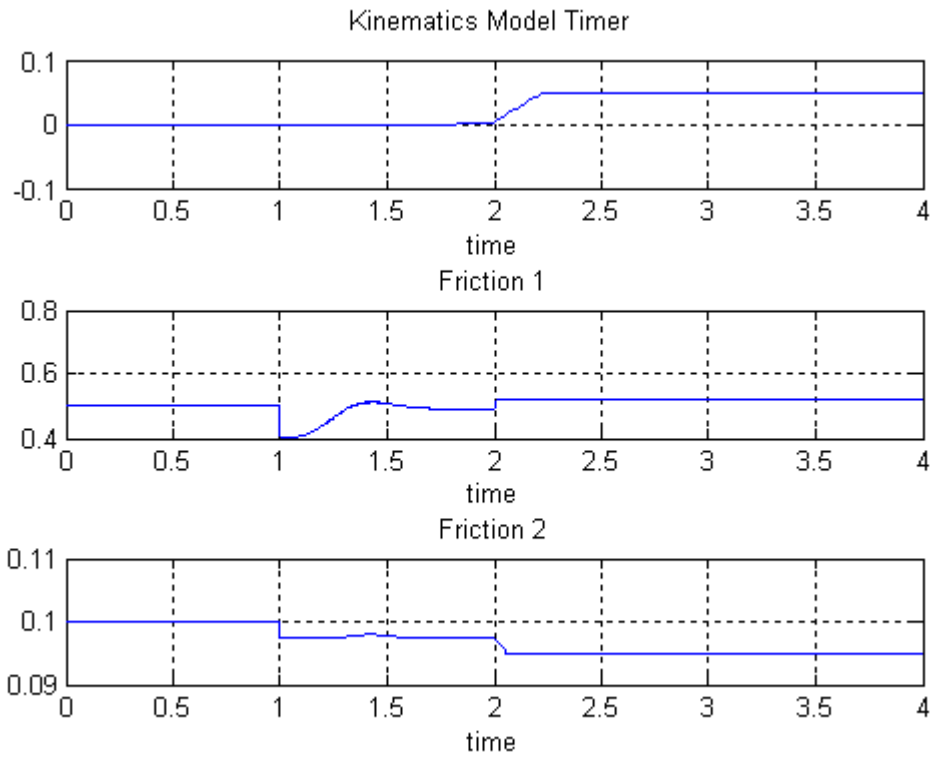


圖 5.54 模擬(二十一)的摩擦力估測

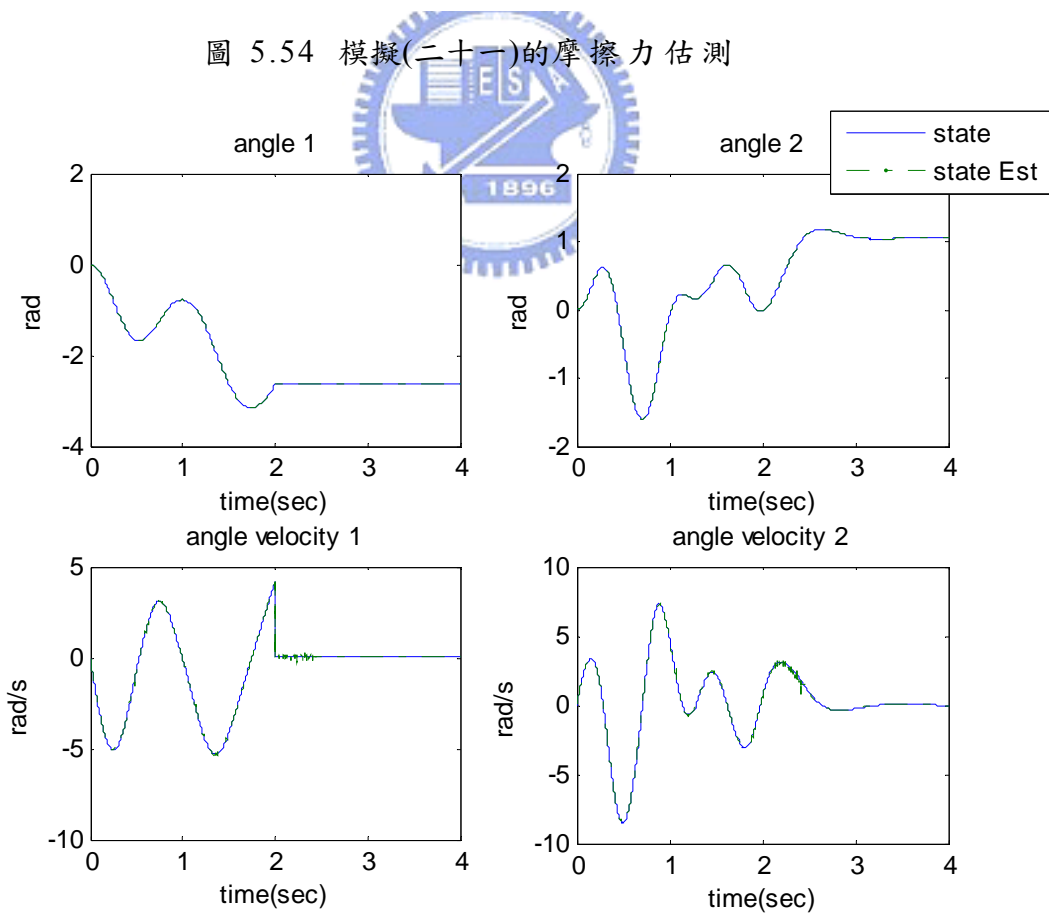


圖 5.55 模擬(二十一)的狀態估測

比較圖 5.41 與圖 5.49，圖 5.45 與圖 5.53 可看到在模型不準確的情況下，力矩差值會有相當的變化，而因應的方法是將錯誤門檻調高，以避免發生誤判。由圖 5.48 與圖 5.49，圖 5.52 與圖 5.53 的錯誤模型的估測並沒有誤判，可見到調高錯誤門檻的確有其實際的效用。而由於摩擦力估測採用運動學模型，故圖 5.50 與圖 5.54 皆不受動力學模型參數不準確的影響。至於角度與角速度的估測上，可由圖 5.51 與圖 5.55 看出實際值與估測值幾乎重疊。其他關於強健度的測試可參照附錄表(三)，在表(三)中並未同時改變大軸和小軸的慣量，因為此時的估測結果非常糟糕，即使調整錯誤門檻也無法補救。

經過上述比較，若以模型估測正確性與即時作業為前提下，新的演算法的確較舊演算法更為較為優良。



第六章 結論與未來工作

6.1 結論

於第一章序論即指出，智慧型機器人將成為下一世代的明星產業，但目前的機器人的研究多著重於機器人的功能的設計，而較少專注於機器人運作時的可靠度與安全性，而這被忽略的部份卻是智慧型機器人能否商品化的最重要關鍵。本論文即針對如何增加機器人運作時的安全性，發展一套應用於機械臂的偵測與識別錯誤(FDI, Fault Detection & Identification)之即時演算法。

本演算法採用圖 3.1 的剩餘值產生架構[10]，由於機械臂為非線性系統而使用非追蹤型卡曼濾波器(UKF)[16][17]，為了增加錯誤種類的識別度而採用 GPB-2 演算法[18]，引入運動學方程式[21]降低計算量，並提出獨創的分群概念配合力矩回算[4][9]，改良開發而成。由於計算量降低，故可即時作業而大幅提升機器人的安全性。此外，該演算法僅需量測角度輔以控制器輸入的訊號即可進行偵錯，如此將大幅降低商業機器人的成本，使其得以普及運用。

總結來說，本論文提出的演算法降低計算量與成本，同時也增加了智慧型機器人進入人類生活的可能性。

6.2 未來工作

針對本演算法的不足與要補強的部份，未來工作大致可歸類為下列三點：

1. 考慮控制器的響應：由於本論文的演算法並未考慮控制器的響應，但其實控制器對錯誤偵測的結果相當的影響，故完整的演算法必須將控制器的響應納入錯誤偵測與識別架構中，並進行模擬與實驗。
2. 適應性的調整錯誤門檻：本論文的模擬皆以固定的錯誤門檻進行偵測與識別，但是為了增加錯誤偵測與識別的強健度，必須提供一適應性的門檻調整方法。
3. 建立實驗平台：由於模擬與實作必定存在一定的差距，故必須藉由建立實驗平台，由實驗中發現演算法的不足，再加以改良。

附 錄

Description	Notation	Value	Units
Length of Link 1	l_1	0.26	m
Length of Link 2	l_2	0.26	m
Distance to the c.g. of Link 1	l_{c1}	0.0983	m
Distance to the c.g. of Link 2	l_{c2}	0.0229	m
Mass of Link 1	m_1	6.5225	kg
Mass of Link 2	m_2	2.0458	kg
Inertial rel. to the c.g. of link 1	I_1	0.1213	kg-m ²
Inertial rel. to the c.g. of link 2	I_2	0.0116	kg-m ²
Gravity Acceleration	g	9.81	m/s ²

表一：二軸機械臂相關參數[22]



模擬條件： $\tau_1 = 8 \sin(3t)$, $\tau_2 = 0.5 \sin(10t)$, $f_1 = 0.5 \times \dot{\theta}_1$, $f_2 = 0.1 \times \dot{\theta}_2$

原始參數	模型一	模型二	模型三	模型四	模型五	模型六	模型七	模型八	模型九
第一軸角度誤差的平方平均	1.50E-05	2.74E-06	2.91E-04	2.06E-06	7.40E-06	1.49E-05	7.60E-06	7.60E-06	1.27E-05
第二軸角度誤差的平方平均	1.22E-05	3.84E-05	9.84E-07	1.93E-06	1.43E-05	4.55E-06	4.37E-06	4.37E-06	1.04E-06
第一軸速度誤差的平方平均	0.0017212	1.00E-06	0.0051246	1.02E-06	0.00025664	0.0017965	0.00025376	0.00025376	0.0014687
第二軸速度誤差的平方平均	0.0051606	0.0039093	9.91E-07	9.90E-07	0.0061692	0.0013055	0.00034791	0.00034791	9.74E-07
I_1 變為 1.5 倍	模型一	模型二	模型三	模型四	模型五	模型六	模型七	模型八	模型九
第一軸角度誤差的平方平均	1.11E-05	1.71E-06	6.14E-02	1.91E-06	8.73E-06	1.13E-05	9.05E-06	9.70E-07	0.003255
第二軸角度誤差的平方平均	1.30E-05	4.89E-05	6.05E-06	2.10E-06	1.51E-05	4.70E-06	5.10E-06	2.15E-06	1.25E-06
第一軸速度誤差的平方平均	0.98109	1.05E-06	0.84397	1.00E-06	0.031632	0.95158	0.027532	9.88E-07	0.10415
第二軸速度誤差的平方平均	0.40787	0.0030714	1.02E-06	1.00E-06	0.024315	0.44985	0.020322	0.00082601	1.00E-06
I_2 變為 1.2 倍	模型一	模型二	模型三	模型四	模型五	模型六	模型七	模型八	模型九
第一軸角度誤差的平方平均	1.24E-05	1.01E-04	1.26E-02	2.05E-06	7.64E-06	1.25E-05	7.82E-06	1.02E-04	0.00055511
第二軸角度誤差的平方平均	1.20E-05	1.80E-03	1.89E-06	1.95E-06	1.48E-05	4.95E-06	5.62E-06	7.85E-04	1.06E-06
第一軸速度誤差的平方平均	0.13062	1.06E-06	1.66E-01	1.01E-06	0.00084058	0.13077	0.001867	9.58E-07	0.018846
第二軸速度誤差的平方平均	0.099438	0.11765	9.65E-07	9.95E-07	0.12625	0.047025	0.050259	0.023521	9.99E-07
I_1 變為 1.5 倍, I_2 變為 1.2 倍	模型一	模型二	模型三	模型四	模型五	模型六	模型七	模型八	模型九
第一軸角度誤差的平方平均	1.06E-05	0.00012681	0.086973	2.03E-06	8.99E-06	1.10E-05	9.19E-06	0.00010231	0.0059708
第二軸角度誤差的平方平均	1.35E-05	0.0022361	9.01E-06	2.04E-06	1.62E-05	5.90E-06	6.71E-06	0.00078646	1.33E-06
第一軸速度誤差的平方平均	1.5211	1.04E-06	1.2119	1.04E-06	0.038688	1.52	0.040539	1.02E-06	0.19438
第二軸速度誤差的平方平均	0.61285	0.12967	9.88E-07	1.02E-06	0.22424	0.59056	0.087794	0.023198	1.03E-06

表二：非追蹤型卡曼濾波器強健度測試表

模擬條件： $\tau_1 = 8\sin(3t)$ ， $\tau_2 = 0.5\sin(10t)$ ， $f_1 = 0.5 \times \dot{\theta}_1$ ， $f_2 = 0.1 \times \dot{\theta}_2$

模擬時間 4 秒，切換時間點皆在 2 秒處

1 → 2: 由模型一切換至模型二，其餘類推

原本的	1 → 2	1 → 3	1 → 4	1 → ->5	1 → 6	1 → 7	1 → 8	1 → 9
第一軸角度誤差的平方平均	2.86E-06	3.64E-05	3.15E-06	9.74E-06	3.97E-05	1.29E-05	2.86E-06	1.40E-05
第二軸角度誤差的平方平均	0.00082738	1.64E-06	1.95E-06	0.00089276	0.00012736	6.40E-05	1.53E-05	1.66E-06
第一軸速度誤差的平方平均	0.0035239	0.0041336	0.0092158	0.013077	0.0027544	0.011049	0.0036347	0.0085865
第二軸速度誤差的平方平均	0.014578	0.0018095	0.0099739	0.038458	0.042541	0.01839	0.0094317	0.0020182
I_1 變為 1.5 倍	1 → 2	1 → 3	1 → 4	1 → 5	1 → 6	1 → 7	1 → 8	1 → 9
第一軸角度誤差的平方平均	3.20E-06	2.83E-05	3.48E-06	1.37E-05	0.00010945	1.46E-05	3.21E-06	1.76E-05
第二軸角度誤差的平方平均	0.00079093	1.79E-06	2.09E-06	7.70E-04	0.00030102	5.72E-05	1.34E-05	1.80E-06
第一軸速度誤差的平方平均	0.0052217	0.65496	0.01143	0.055223	2.0514	0.011048	0.0055746	0.008754
第二軸速度誤差的平方平均	0.015811	3.7749	0.010827	0.097244	3.1202	0.022806	0.010041	0.0027042
I_2 變為 1.1 倍	1 → 2	1 → 3	1 → 4	1 → 5	1 → 6	1 → 7	1 → 8	1 → 9
第一軸角度誤差的平方平均	3.15E-05	3.38E-05	3.19E-06	1.23E-05	3.45E-05	1.29E-05	2.90E-06	1.38E-05
第二軸角度誤差的平方平均	1.48E-06	1.54E-06	1.85E-06	0.00063883	0.00018233	5.25E-05	1.15E-05	1.49E-06
第一軸速度誤差的平方平均	0.004102	0.0029046	0.0092615	0.034822	0.012939	0.009333	0.0036832	0.0072199
第二軸速度誤差的平方平均	0.0058269	0.0036624	0.014367	2.9616	0.042124	0.024887	0.012628	0.0055979

表三：模型切換強健度測試

參考文獻

- [1] Jones, H. L. , “ Failure detection in linear systems.“ ,Dept. of Aeronautics, M.I.T., Cambridge, Mass, PhD thesis, 1973.
- [2] Silvio Simani ,Cesare Fantuzzi and Ron J. Patton , Model-Based Fault Diagnosis in dynamic systems using identification techniques, Springer-Verlag, Spring 2002
- [3] Fabrizio Caccavale' and Ian D. Walker ,“Observer-based Fault Detection for Robot Manipulators”, Proceedings of the 1997 IEEE International Conference on Robotics and Automation Albuquerque, New Mexico, April 1997
- [4] M. L. McIntyre,t W. E. Dixon,t D. M. Dawson,t and I. D. Walked, “Fault Detection and Identification for Robot Manipulators”, Proceedings of the 2004 IEEE International Conference on Robotics & Automation ,New Orleans, April 2004
- [5] Jin-Ho Shin and Ju-Jang Lee,” Fault Detection and Robust Fault Recovery Control for Robot Manipulators with Actuator Failures”, Proceedings of the 1999 IEEE International Conference on Robotics & Automation Detroit, Michigan ,May 1999
- [6] Patton, R., Lopez-Toribio, C., and Uppal, F. “Artificial Intelligence Approaches to fault diagnosis for dynamic systems.” Journal of Applied Mathematics and Computer Science, pp. 471-518, April 1999.
- [7] De Luca, A. , Mattone, R. , “An Identification Scheme for Robot Actuator Faults”, Intelligent Robots and Systems(IROS), IEEE/RSJ International Conference on Publication Date: 2-6, pp. 1127- 1131 , Aug 2005.

- [8] Alexander B. Trunov and Marios M. Polycarpou ,”Robust Nonlinear Fault Diagnosis: Application to Robotic Systems”, Proceedings of the 1999 BEE International Conference on Control Applications Kohala Coast-Island of Hawai'i. Hawai'i. USA - August 22-27, 1999.
- [9] Warren E. Dixon, Ian D. Walker, Darren M. Dawson, and John P. Hartranft , “Fault Detection for Robot Manipulators with Parametric Uncertainty: A Prediction-Error-Based Approach” , IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 16, NO. 6, DECEMBER 2000.
- [10] Chow, E. Y. and Willsky, A. S., “Issue in the development of a general algorithm for reliable failure detection.” , Proc. of the 19th Conf. on Decision & Control, Albuquerque, NM ,1980.
- [11] Basseville. M., “Detecting changes in signals and systems: A survey.” Automatica, 24(3) pp: 309 – 326,1988.
- [12] Patton, R. J. and Chen, J. ,“A review of parity space approaches to fault diagnosis.” , IFAC Symposium SAFEPROCESS '91, Baden-Baden, 1991.
- [13] Chen, J. and Patton, R. J. , “Robust Model Based Fault Diagnosis for Dynamic Systems.” Kluwer Academic, 1999.
- [14] Asok Ray and Rogelio Luck , “An introduction to sensor signal validation in redundant measurement systems”, IEEE control System, pp.44-49,1991.
- [15] Sontag, Eduardo , “Mathematical Control Theory: Deterministic Finite Dimensional Systems. Second Edition.” Springer ISBN 0-387-984895, 1998.
- [16] Simon J. Julier Jeffrey K. Uhlmann , “A New Extension of the Kalman Filter to Nonlinear Systems” ,Proceeding of Aerosense : the 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls, Orlando, Florida, p182-193, 1997.

- [17] Rudolph van der Menve and Eric A. Wan ,“THE SQUARE-ROOT UNSCENTED KALMAN FILTER FOR STATE AND PARAMETER-ESTIMATION”,
Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001
IEEE International Conference on Publication Date: 2001 Volume: 6, On page(s):
3461-3464.
- [18] Yaakov Bar-Shalom, X. Rong Li, Thiagalingam Kirubarajan,
“Estimation with Applications to Tracking and Navigation: *Theory Algorithms
and Software*”, 2001 John Wiley & Sons, Inc,ISBNs: 0-471-41655-X ,p441-p466.
- [19] Isermann, R. , “Supervision, fault detection and fault diagnosis methods:
an introduction.”, Control Engineering Practice, pp:639:652.,1997.
- [20] Willsky, A. S., “A survey of design methods for failure detection in
dynamic systems. “Automatica, pp:601:611,1976.
- [21] J. D. English and A. A. Maciejewski, “Fault Tolerance for Kinematically
Redundant Manipulators: Anticipating Free-Swing Joint Failures,”, *IEEE
Transactions on Robotics and Automation*, Vol. 14, No. 4, pp566~575, 1998.
- [22] R. Kelly, V. Santibanez and A. Loria, “Control of Robot Manipulators in Joint
Space” , Springer, May 2005.