

# 行政院國家科學委員會專題研究計畫成果報告

## 應用離散事件控制理論及先進資訊電子技術之開放式控制器平台研究(2/3) On The Study of Open Architecture Controllers Using Discrete Event Control Theory and State-of-the-art Information and Electronic Design Technology

計畫編號：NSC 89-2218-E-009-036-

執行期限：88 年 8 月 1 日至 89 年 7 月 31 日

主持人：胡竹生教授 jshu@cn.nctu.edu.tw

執行單位：國立交通大學電機與控制工程系

### 一、中文摘要(關鍵詞：開放式架構控制器， 數位訊號處理器，離散事件系統)

本研究報告主要是提出統一化的控制系統與自動化軟體架構，使應用離散事件系統規格於監控器合成的方法能同時應用於單一機器監督器的設計，與多台機器整合時監控模組的設計。在文中應用物件導向的技術來分析設計系統中的基本類別及其相互關係，設計其軟體框架(framework)，並以統一模型化語言(UML)來表示，再配合設計樣板(Design Pattern)的應用，使整個架構更具有彈性。最後配合分散式物件的技術來做為實際應用時的實作方式。

**英文摘要** (Keywords: Open Architecture Controllers, Digital Signal Processors, Discrete Event Systems)

In this report, we propose a unified framework for control system and automation software design. Using this framework, supervisory synthesis methodology with DES specification can be used in design control software not only for single machine but also for machines' integration. In the architectural design, basic classes of our proposed framework were model by Object-Oriented Analysis and Design Technique and presented in UML notation. Furthermore, in the mechanistic design, design pattern was used to improve the modality of our

framework and make it more flexible. At last, distributed object technique-CORBA was used for implementation in the detail design.

### 二、計劃緣由與目的

以開放式架構 ( Open Architecture ) 為號召之工業控制器 ( Industrial Controller ) 在 90 年代成為各項控制系統的設計主流。這也使得自動化設備如數控機械的製造商更具競爭力，其主要原因是由於軟硬體的彈性搭配，使先進的製造及控制技術可快速且容易的實現。這項優點，也是產業界與學術研究能否緊密結合的重要契機。我國亦藉助個人電腦之功能，由工研院等單位研發 PC-based controller，其目的也是希望藉由控制器競爭力的提升，能夠推動我國在自動化及機電產業的進展。

時至今日，由於電子(含電力電子)及資訊技術的長足進步，連帶使得如 PC-based 控制器的發展產生了變化。如大型積體電路化，快速且低價的數位訊號處理器(DSP)的問世，網路功能及速度增加，及軟體技術的演變等。同時 PC 由於普及性提高且競爭日益激烈，其架構不斷變化，然而由於其主導權非工業控制，與其過度緊密結合之控制器，在 PC 改朝換代時將面臨技術更新與穩定性的考驗。

因此如何因勢利導，並善用先進的電子與資訊方面的技術，使 PC-based 控制器無論在表

面或實質上均能迎合時代的潮流，是相當重要的課題。尤其是網路技術的演進使得自動化整合時資訊結構改變，而身為自動化系統最底層之工控器，勢必也將跟上其腳步。本計劃的目的，即是研究在此趨勢下以 PC 為發展及執行平台的工業控制器結構及軟硬體相關技術。在理論上將嘗試應用離散事件系統(Discrete Event Systems)控制的技巧，設計即時迴授、順序控制、人機界面及網路通訊等模組的整合法則。實作方面將設計以數位訊號處理器為核心之即時控制模組，並應用即時多工核心、物件導向及先進網路程式設計的技巧，以達成控制軟體模組化及網路連結透明化的目標。

### 三、研究方法及成果

#### } 緒論

本論文成功地實現以 CAN Bus 為基礎之分散式伺服控制系統，其中包含 CAN Bus 建構之工控網路、自訂通訊協定、DSP 多軸運動控制卡、即時多工核心、加減速軌跡運算、以及一個視窗軟體為基礎的監控介面。相對於傳統的多軸控制，分散式控制架構具有下列優點：

- (1) 可因應不同需求彈性調整系統組態。
- (2) 系統擴充性增加。
- (3) 各單元間各自獨立，模組化設計更容易。
- (4) 提供各伺服模組的即時與同步協調控制。
- (5) 系統具有多工處理能力，更有效率。
- (6) 系統中某一單元發生錯誤時，不致造成整個系統的損壞。
- (7) 可靠度高，減低錯誤維修的困難度。
- (8) 開放式網路架構，使用者可有更多選擇。

#### } 研究方法

本篇論文提出一以 CAN Bus 為基礎之即時工控網路架構，整個系統架構如圖 1 所示。整個系統架構主要包括 PC、實驗室自行研發之 DSP TMS320F243 Servo 板及伺服馬達三個部

分。在通訊模組方面：PC 與 DSP SERVO Master 板間利用 RS232 做溝通，而各個 DSP SERVO 板間則是利用 CAN Bus 進行資料傳輸與時間的同步。在控制模組方面：各個 DSP SERVO 板，包括 Master 與 Slave，利用 PWM 模組來對馬達做控制，並且利用 Capture Units 讀回馬達實際運轉的位置。

另外我們在 DSP SERVO 板端嵌入一即時多工核心(Real-time Kernel) Taunix，此核心為整個系統工作做排程，以提高系統工作效率。以下針對硬體設備略為說明。

#### PC：

實作中，採用工業用電腦，CPU 速度為 166MHz，工作環境為 Win95。主要負責的工作為計算運動軌跡以及提供人機介面以便使用者了解系統的狀態。利用 RS232 以 baud rate 115200bits/s 的速度與 DSP Servo Master 板做溝通。

#### DSP Servo 板：

主要包括 DSP TMS320F243、FPGA、SRAM、RS232 Transceiver 及 CAN Bus Transceiver 等部分。主要負責的工作，包括對馬達的控制與各 SERVO 板間資料的傳輸。其中，資料的傳輸部分是利用 CAN Bus 以 EXTENDED FORMAT 在 baud rate 為 1Mbits/s 的速度進行傳輸。

#### 伺服馬達：

實作中採用的伺服馬達，為日本信濃電器株式會社所製造之 AC 伺服驅動兩軸運動平台，馬達型號為 7CB30-2SE6F，AC 伺服驅動器的型號為 DOSA010B-CB302F。其中，平台的長度為 40cm，馬達額定回轉速 3000rpm，馬達回授 ENCODER 為 8000P/R，伺服器為電壓驅動速度控制，可接受的電壓範圍在 -10V~10V 之間。

## 即時多工核心—Taunix

即時核心在微電腦控制系統上扮演的腳色，就等於一般電腦上的作業系統，其用途主要是作為人機介面、系統資源管理以及提供良好的程式發展平台等。實作中的即時多工核心系統：Taunix，是以Unix/POSIX架構為參考，即時排程法為基礎，並以C語言進行設計，以達到最高的可攜性。整套核心分為四個部分：

- 一、工作管理核心
- 二、工作間的訊息傳遞
- 三、驅動程式模組
- 四、通訊模組

## 四、結論與討論

### } 系統實作—PC 端實作

軌跡計算：軌跡的計算，利用 T 型、S 型兩種加減速，提供點對點、直線和圓弧三種軌跡的計算。

RS232 防錯措施：每次接收到資料後，檢查此次傳輸資料的總和，以及傳輸的編號。

人機介面：利用 Borland C++ Builder 3.0 設計一人機介面，以提供使用者方便操作本系統。其中包括執行點對點、直線及圓弧的軌跡運動，另外亦提供手動(JOG)調整馬達位置的功能。

### } 系統實作—DSP 端實作

時間同步：對於分散式控制系統來說，每個獨立的單元之間，都必須進行時間上的同步。本篇論文提出的系統架構中，是利用 CAN Bus 來同步各個 DSP SERVO 板，類似對時的做法，以 DSP MASTER 的時間為基準，每隔固定時間將每個 DSP SLAVE 的時間校正成與 DSP MASTER 的時間一樣，來實現各個 DSP SERVO 板之間時間的同步。

即時控制：為了讓軌跡資料的命令，能夠即時地送到各個 SERVO 板，我們在每個 DSP

SLAVE 的記憶體中，規劃出多個暫存器，用以儲存軌跡資料。由於 SERVO 板控制馬達與接收資料，是利用不相同的模組來實現，因此可以並行處理，也就是當 SERVO 板一邊在對馬達作控制的同時，一邊接收從 PC 端或 DSP MASTER 端傳來的軌跡資料，我們利用這個架構實現即時控制。

CAN Bus 防錯措施：除了執行 CAN Bus 本身提供錯誤偵測的機制外，另外我們設計 DATA FIELD 的第一個位元組為資料的編號，提供 SLAVE 端檢查資料是否有遺失及避免發生重複接收資料的錯誤。

### } 系統工作流程

初始測試階段：在測試模式下要處理的工作，包括定義出網路上的 MASTER 與 SLAVE、確認網路上有哪些單元及測試 RS232 與 CAN Bus 是否正常工作等。

手控操作模式：當 DSP MASTER 接收到 PC 端傳送 JOG 的命令後，會開始進行解碼，判定使用者所要操作的單元是否就是 MASTER。如果是，則 MASTER 即進入手控操作模式，並且解碼出使用者操作的動作，進行對馬達的控制；如果不是，則 MASTER 將手控操作模式的命令，利用傳送 CAN Bus 廣播(multicast)傳送給網路上各個 SLAVE。當 SLAVE 接收到手控模式的命令，會開始進行解碼的動作，首先判斷本身是否為使用者所要操作的單元，如果不是，則放棄此次的命令；如果是，則繼續解碼使用者所要操作的動作，並且開始對馬達進行控制。

軌跡運動模式：在軌跡運動模式下，DSP MASTER 必須即時地並行處理與 PC、SLAVE 之間的通訊及對馬達進行控制；而 DSP SLAVE 除了對馬達作控制外，還需要同時與 MASTER 進行資料雙向的傳輸。針對這些工作的處理，我們設計利用 Time Triggered Protocol 來實現。什麼是 Time

Triggered Protocol 呢？簡單來說，就是在固定的時間，規劃執行特定的工作。圖 2及圖 3分別為我們設計在 CAN Bus 上以及在 RS232 上規劃出每個時間必須執行的動作。

#### } 錯誤處理

在本次實作中，我們模擬三種錯誤狀況的發生：RS232 斷線、CAN Bus 斷線及系統錯誤檢出。當系統偵測到這三種錯誤時，我們設計讓系統暫時停止運作，但是，此時馬達可能處於高速運轉的狀態，是無法說停就停的，因此，我們設計出一個讓馬達減速而且不改變其運動路徑的方法。而在系統錯誤修復後，我們設計馬達將繼續之前未完成的軌跡，但此時馬達是停止的，因此必須有一段加速的過程，讓馬達可以平緩地追上正常的運動速度，執行未完成的軌跡運動。

減速時，將暫存器的資料依序分成幾組，接著將第一組的每一筆資料分成兩等份，第二組的每一筆資料分成三等份...最後一組的每一筆資料分成 N 等份。由於軌跡資料平均的等份越多，得到新的軌跡資料值就越小，因此，如此的設計可以讓馬達運動的速度越來越慢，以達到馬達減速的目的，而且馬達運動的軌跡並不會因此而改變。在馬達的速度減慢後，讓馬達平穩的停止運動也就更為容易。

加速的方法與減速類似，不同的是將第一組的每一筆資料平均分為 N 等份，第二組的每一筆資料平均分為(N-1)等份，...最後一組的每一筆資料平均分為兩等份。執行新的軌跡資料，馬達的速度便可以平緩地追上正常的運動速度，而且運動的路徑不會因此而改變。

加減速方法示意圖如圖 4所示。

#### } 控制架構

圖 5為回授控制系統方塊圖，我們利用 PI 控制器對系統作補償，實現精密運動控制。其中 PI 控制器的參數，使用者可由 PC 端人機介

面自行調整。

#### } 結論

應用於分散式控制系統架構的控制網路中，目前以 SERCOS 發展最為成熟，是目前對於運動控制系統在軟體通訊協定上定義最為完全的工業網路。但是由於 SERCOS 利用光纖傳輸，使得價格偏高，另一方面在國內的相關資源不易取得，因此在市場上的佔有率不是很高，無法普及的應用於工業界。本論文提出利用 CAN Bus 建構之即時網路分散式伺服控制系統，不僅打破傳統多軸控制系統封閉集中式的架構，更解決以 SERCOS 為基礎分散式系統架構價格與資源取得上的問題，為控制領域上一次新的嘗試。以此為基礎，未來可將重點放在高性能運動控制軟體的研究上，建立起一套完整的 PC-Based 分散式即時多軸運動控制系統。圖 6為本系統執行一複雜的軌跡運動圖。

#### 五、計畫成果自評

項目	完成情況
與原計畫相符程度	100%
達成預期目標	90%
研究成果學術價值	新型控制器設計
研究成果應用價值	具實用性
學術期刊發表合適否	投稿中
申請專利合適否	否
主要發現或其他價值	軟體著作權

#### 六、參考文獻

- [1] Bannatyne, R., "Time triggered protocol-fault tolerant serial communications for real-time embedded systems," *Wescon/98*, pp. 86-91, 1998.
- [2] Berardinis L. A., "SERCOS lights the way for digital drivers," *Machine Design Int.*, vol. 66, no. 16, pp. 52-64, Aug., 1994.
- [3] CAN Kingdom specification Version 3.01, 1/1 1996.
- [4] CANopen draft standard 301 revision 3.0, 30/10 1996.

- [5] Hu, J. S., and In, I. T., *Real-Time Multi-Tasking Kernel Design*, Chwa Bookstore, 1995.
- [6] ISO 11898, "Controller Area Network (CAN) for high speed communication," 1993.
- [7] Monnet C., TI Application Report SPRA500 : "Digital Signal Processor Solutions: Understanding the CAN Controller on the TMS320C24x DSP Controller," Dec. 1998.
- [8] Zeltwanger H., "CANopen network and its impact on motion control application," *Powerconverters Intell. Motion*, pp. 281-288, May, 1998.
- [9] Zuberi, K. M., and Shin, K. G., "Design and implementation of efficient message scheduling for Controller Area Network," *IEEE Trans. on Computers*, Vol. 49 No. 2, pp. 182-188, 2000.

七、圖表

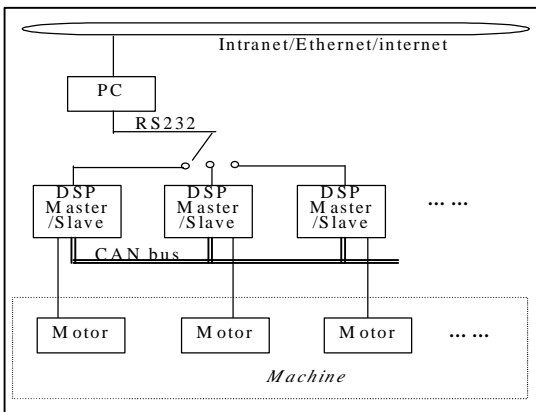


圖 1 所提之硬體架構

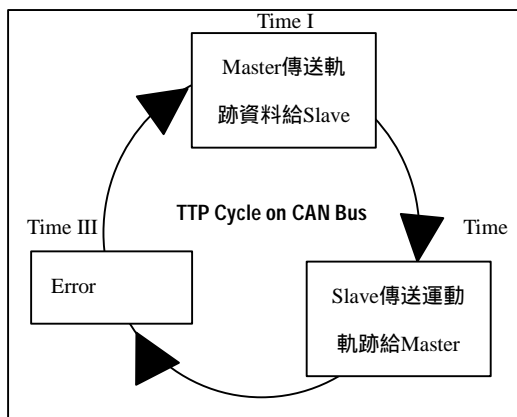


圖 2 CAN bus TTP cycle

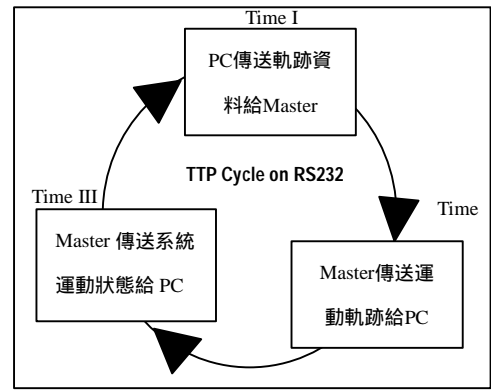


圖 3 RS232 TTP cycle

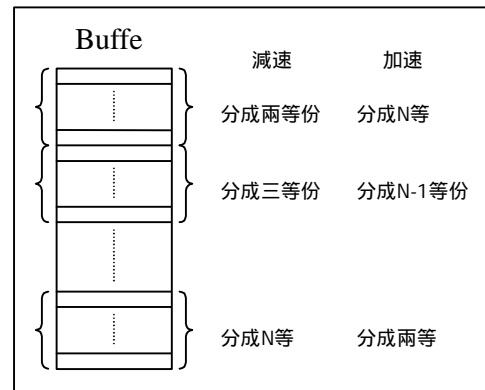


圖 4 錯誤狀況之加減速方法

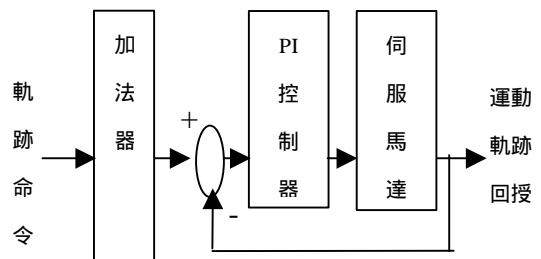


圖 5 控制架構

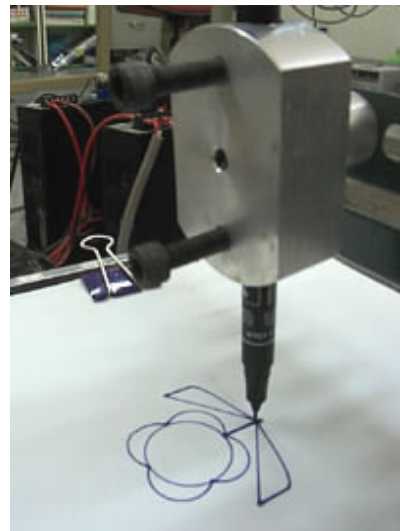


圖 6 系統執行一複雜的軌跡運動