

# LabVIEW 環境與介面卡於控制器之設計探討

賴秋庚

曾炘文

國立勤益技術學院電機工程系

## 摘要

在圖形化控制系統的領域裡，由美商慧曇儀器公司(National Instruments, NI)所開發的 LabVIEW 系統以其獨特的撰寫方式，為大多數使用者所接受而漸露鋒頭。LabVIEW 系統以虛擬儀表(Virtual Instrument, VI)為特色來吸引大家，並且提供許多內建的子 vi，可以提供程式設計者以圖形的方式來完成所需的功能程式。本研究主題主要在於探討 LabVIEW 編號 PCI-MIO-16E-4 的 DAQ 介面卡，用於控制程式設計時的限制。文中利用一個由 OPA 電路所組成的二階系統當成控制的對象。透過模擬分析與實際實驗結果顯示，在 Windows 作業系統底下，PCI-MIO-16E-4 介面卡無法達到即時控制的功能。為改善這一個問題，本文利用加入以 C 語言所撰寫的控制程式，搭配 LabVIEW 程式所寫的監視畫面，解決 Windows 多工的影響，而達到即時控制與顯示的目的。

關鍵詞：LabVIEW，Windows，多工作業，即時控制。

## 1. 前言

本文研究探討 LabVIEW 一般型介面卡於多工作業系統底下，對於控制所需的取樣時間無法固定的問題。此種問題的解決，吾人透過 C 語言來撰寫一中斷副程式，再以良好的圖形介面(LabVIEW 程式)所撰寫的應用程式來監控受控系統的輸出，對於控制命令與參數並且可以同步的即時輸入。本研究的即時視窗型圖形控制歸類為集中式控制，監視與控制都在同一部電腦上執行，可以減少控制設備成本。

圖形化的系統應用程式(Graphical Application Program)是自動控制系統設計的主流之一。圖形化的系統應用程式包含了兩大部分：(1)人機介面：是系統與使用者直接交換資訊最重要的部分，因此其“親和性”將會是最大的考量。(2)核心程式：是系統資料擷取、分析、判斷、決策的部分，彈性與效率是其最大的考量。圖形化的系統設計，大大簡化了撰寫系統程式的程序，並提供了良好的圖形介面。程式的撰寫以堆積木或連線的方式來完成，大量的使用物件及資料流的概念，因此無論在程式維護或修改上都變的非常容易。圖形化控制程式的一個缺點，就是無法隨著受控系統的需要而調整取樣時間(Sampling time)。取樣時間對於數位控制來說格外的重要，若撇開取樣定理(Sampling Theorem)不談，而以一般控制設計為求得較好的控制結果來說，通常取樣頻率的設計為大於系統頻寬十倍，若非如此，則很難達到較好的控制效果，而且影響其控制的

準確性與穩定度。LabVIEW 系統提供 PID 控制程式的設計，可以直接利用 LabVIEW 的一般介面卡來執行數位控制。但是利用此種方式所設計執行的數位 PID 控制系統，則很容易受到 PC 因為執行多工作業的關係，而導致系統無法正確的執行所設計的取樣時間。美商慧碁儀器公司針對這一個取樣時間無法固定的缺點，發展出一套 PXI 獨立的作業系統，專門用來執行 LabVIEW 此種語言。PXI 系統是一種硬體設備，它特別應用在危險度高的地方，利用 PXI 系統雖然解決取樣時間不固定的問題，但是其需要較高的設備費用，對於簡單的控制應用來說，是非常不划算的。

目前利用 LabVIEW 語言與系統執行控制應用時，需考慮所設計的系統是單獨僅作監視用途，抑或監視與控制同步運作。以目前 Pentium CPU 的處理速度來說，要同時處理交直流馬達的內外迴路控制已經足夠，但這是僅就單純的單工控制系統來說；至於在多工作業系統底下，答案是否如此，則是本篇論文探討的重點。對於這樣的一個問題，為了使系統仍然能夠順利的執行利用所規範的監視與控制的動作，本文將監視與控制動作分開：利用 LabVIEW 系統設計監視的部分，而控制的操作則委由 C 語言所撰寫的程式來負責。兩者之間資料的傳遞則利用系統介面中其餘暫時不用的記憶體來完成。

本研究的驗證，控制器部份採用數位 PI 程式，透過根軌跡法設計所需的類比控制器參數，再予以數位化。而實作部份則由 OPA 所組成的二階系統當成控制器負載，控制目標為二階系統輸出的電壓，而二階系統數學模式為  $G(s) = 20 / (s+20)(s+2)$ 。應用 LabVIEW 系統控制的結果將與本文所探討的改善方法互相比較，藉以說明 LabVIEW 一般介面卡在執行即時控制時的能力限制，並且提出一個簡單解決取樣時間無法固定或不足的方法。

## 2. 系統硬體架構

驗證本研究的系統硬體部分包括 PC 電腦、LabVIEW PCI-MIO-16E-4 DAQ 卡、仲成 16 位元介面卡、一個二階系統模擬硬體電路。在模擬的部分，吾人假設受控的系統具有轉移函數

$$G_p(s) = \frac{20}{(s+20)(s+2)} \quad (1)$$

吾人首先以 Simulink 模擬所設計的系統，接著利用 LabVIEW 程式撰寫 PID 控制器，並用 OPA 設計式(1)的二階電路，及利用 LabVIEW PCI-MIO-16E-4 DAQ 卡將此二階電路的類比輸出信號抓取到 PC 中，並進行實際的 PI 控制操作。類比電路的輸出響應可以直接在 LabVIEW 的視窗中觀察到。吾人同時利用取樣示波器抓取此類比輸出電壓，以進行實際的數據比對，並觀察因為 Windows 多工的操作對於控制的影響。LabVIEW 程式可以直接在其發展平台上執行，也可以轉成執行檔，在 Windows 底下以 .exe 的型式執行。吾人將比對這兩種控制模式對於取樣時間準確度的影響。最後再利用 C 程式語言撰寫控制程式的部分，並且應用較便宜的介面卡，進行以 LabVIEW 程式作監督的設計，並且以 C 語言撰寫控制程式的主體，據以比較何種方式可以獲取較準確的取樣時間，並且比較準其控制的效果。系統架構如圖 1 所示。

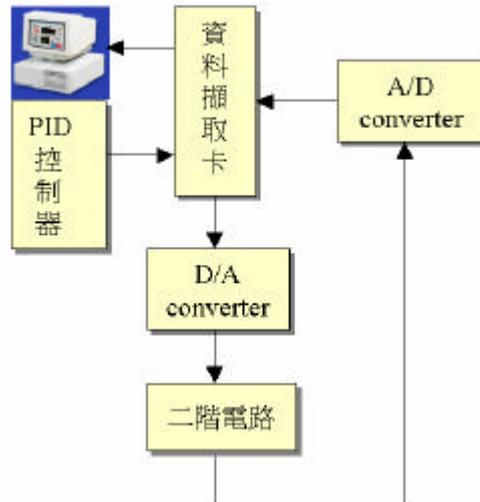


圖 1 硬體實驗系統架構圖

### 3. 資料擷取系統說明

LabVIEW 的 DAQ 擷取函式庫，包含可控制 NI DAQ 卡的 vi。用在本研究的這張卡 (PCI-MIO-16E-4) 附有許多功能，例如類比對數位(A/D)轉換、數位對類比(D/A)轉換、數位輸入輸出(Digital I/O)及計數或計時等功能。NI 的 DAQ 卡可提供不同的資料擷取和產生訊號的速度。每一個 DAQ 卡都被設計應用在特定的硬體平台的作業系統之下，而所有型號的 DAQ 卡及功能可以參考 NI 的目錄。底下為 PCI-MIO-16E-4 卡用在本研究的相關規格。

- (1) 製造廠商：美商慧基儀器公司(National Instruments)
- (2) 型號：PCI-MIO-16E-4
- (3) Analog Input 規格：
  - (a) No. of channels：8
  - (b) Sampling rate：250 kS/s
  - (c) Input Range：±0.05V ~ ±10V
  - (d) Resolution：12-bit
- (4) Analog Output 規格：
  - (a) No. of channels：2
  - (b) Output Range：±10V
  - (c) Resolution：12-bit

## 4. 數位 PID 程式介紹

### 4.1 PID 控制方塊圖

對於常用的 PID 控制器，其轉移函數可表示成

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (2)$$

其中， $K_p$ 、 $K_i$  及  $K_d$  為 PID 控制器增益參數； $U(s)$  為控制器輸出； $E(s)$  為控制器輸入； $s$  為拉氏運算子。利用 Euler's Method[2]，令

$$s = \frac{1-z^{-1}}{T_s} \quad (3)$$

並代入式(2)中，可以得到數位 PID 控制器轉移函數如式(4)所示。

$$C(z) = \frac{U(z)}{E(z)} = K_p + \frac{K_i T_s}{1-z^{-1}} + K_d \left[ \frac{1-z^{-1}}{T_s} \right] \quad (4)$$

在式(3)與(4)中， $T_s$  為取樣週期。相對應於式(4)的方塊圖如圖 2 所示。

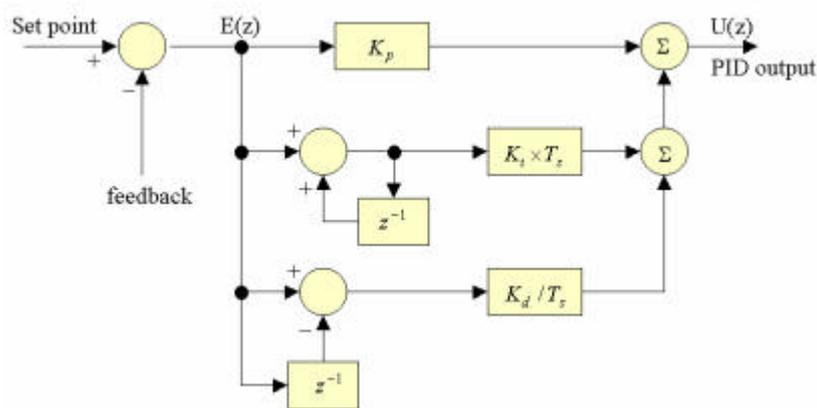


圖 2 數位 PID 控制方塊圖

#### 4.2 取樣時間的選擇與 LabVIEW 程式設定

對於數位控制器的設計來說，取樣時間的設定是重要的。根據經驗法則，取樣頻率四至十倍於系統頻寬可獲得一個與連續系統較相仿的輸出控制結果。圖 3 是本研究用來設定取樣時間的 vi。

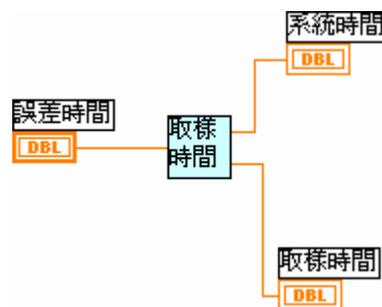


圖 3 決定取樣時間的 vi

圖 3 的取樣時間 vi 主要是控制吾人何時通知 CPU 以便做 A/D 訊號的擷取及 PID 控制的計算。於本研究探討中，我們設 1ms 當作系統的取樣時間。LabVIEW 裡面有一個 Tick count.vi，這是用來擷取系統的時間，它以 ms 作為單位。因為這是以軟體所設計的計時器，吾人以 PIII\_1.0

及 P4\_1.8 CPU 當成測試的對象，檢驗在 LabVIEW 平台上測試系統時，是否能符合所設計的操作時間。結果發現，雖然設定的平均取樣時間為 1ms，但是透過實際量測，所得到的取樣時間分別約為 3ms 與 1.7ms，與所需的設計有相當的差距。在實作驗證部分，吾人將把所設計的 LabVIEW vi 轉成執行檔，則可以降低程式執行所需的時間。

### 4.3 以 LabVIEW 方塊完成 PID 控制器的設計

底下吾人將簡介利用 LabVIEW 方塊完成數位 PID 控制器的設計流程。在圖 4 中的方塊，用以求解系統的命令(SP)與回授信號(AI)相減而得到誤差輸出。圖 4 中的另一個輸入 error\_1 代表的是上一次的控制誤差，而另一個輸出 del\_err 代表的是前一次誤差的與本次誤差的差值，作為微分器操作使用。

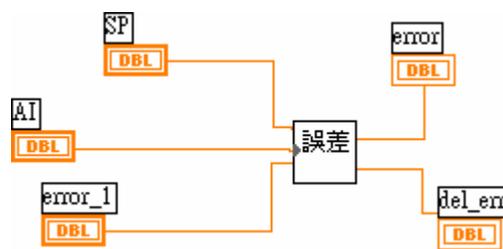


圖 4 求解誤差 vi

圖(5)為比例控制器，執行

$$P_{out}(n) = K_p \times e(n) \quad (5)$$

運算式。

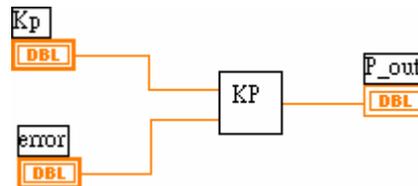


圖 5 LabVIEW 方塊執行比例運算的 vi

圖 6 為積分操作，執行

$$I_{out}(n) = [acc\_err(n-1) + e(n)] \times (K_i \times T_s) \quad (6)$$

運算，其中  $acc\_err(n-1)$  代表的是前次誤差的累積值。圖 7 為微分控制部分，執行的運算式為

$$\begin{aligned} D_{out}(n) &= del\_err \times K_d / T_s \\ &= [e(n) - e(n-1)] \times K_d / T_s \end{aligned} \quad (7)$$

利用圖 8 的 vi 可以把三個分開獨立的動作予以結合成一個完整的 PID 控制器。控制器的輸出則送到 D/A 轉換器，輸出類比電壓到受控電路。

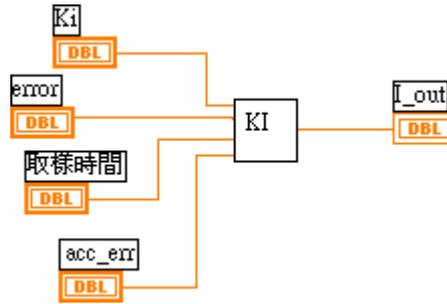


圖 6 執行積分運算的 LabVIEW vi

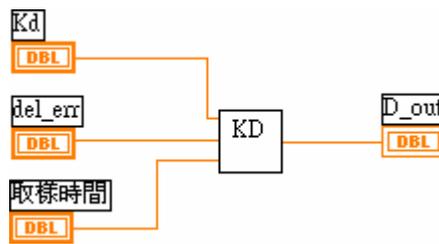


圖 7 執行微分運算的 LabVIEW vi

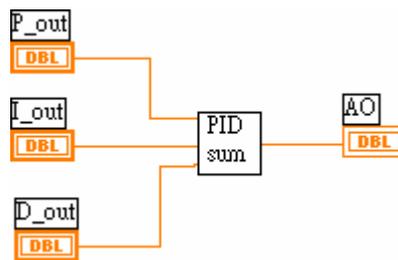


圖 8 執行綜合 PID 控制的 sum.vi

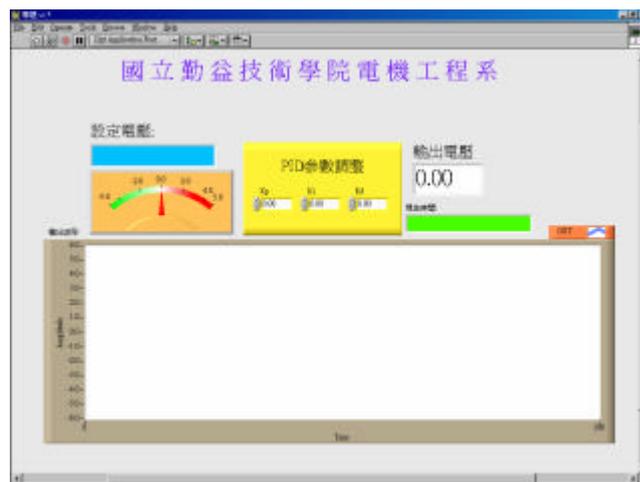


圖 9 前置控制面板

吾人另外設計一個控制面板，用以隨吾人之控制要求而可以立即的線上調整控制器參數的值。控制面板同時可以以圖形的方式設定命令電壓、以數字顯示控制變數及以曲線圖顯示控制過

程的歷史曲線圖。所設計的控制面板如圖 9 所示。

## 5 模擬分析、設計與基於 LabVIEW 系統的實驗

### 5.1 系統分析

吾人用以模擬及實測的系統具有轉移函數如式(1)所示。為方便起見，再重寫於下。

$$G_p(s) = \frac{20}{(s+20)(s+2)} \quad (1)$$

以下將就這一個系統說明控制器的設計。首先利用 MATLAB 畫出式(1)的波德響應圖，如圖 10 所示，以求得系統頻寬。

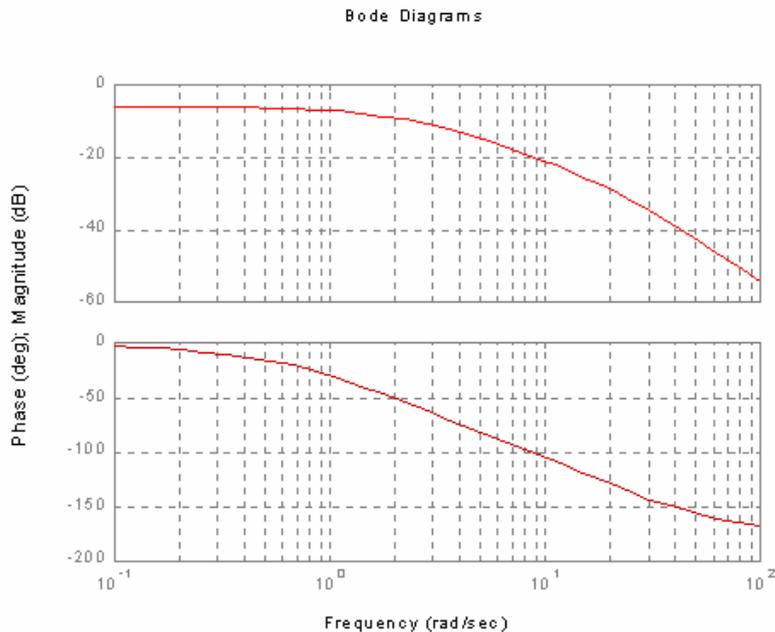


圖 10 式(1)系統波德圖

由圖 10 知道系統頻寬約 2 rad/dec，換算頻率為 0.318 Hz。數位控制器的取樣頻率設計原以採用系統十倍頻寬為原則，但在此，吾人為測試 PC 系統執行 LabVIEW 控制程式的能力，乃選定 1ms 的取樣頻率供系統測試使用。

### 5.2 PID 參數設計

PID 參數的設計採根軌跡法。Matlab 系統中函式 rlocus() 可以畫出所對應輸入系統的根軌跡；而函式 rlocfind() 則可以找出所點選位置的相對增益的值。針對式(1)系統，輔以 Matlab 函式設計所得的控制器轉移函數如式(8)所示。

$$C(s) = 10 + \frac{36}{s} + 0.0275s \quad (8)$$

為驗證所設計控制器的性能，利用 Simulink 進行系統(1)與控制器(8)的整合模擬。模擬方塊圖如圖 11 所示，模擬暫態步階響應與控制器輸出如圖 12 所示。在模擬時，考量 D/A 轉換器的輸出

限制在  $\pm 10V$ ，因此模擬系統加入了飽和方塊，以使模擬所得的結果符合於實際的系統。

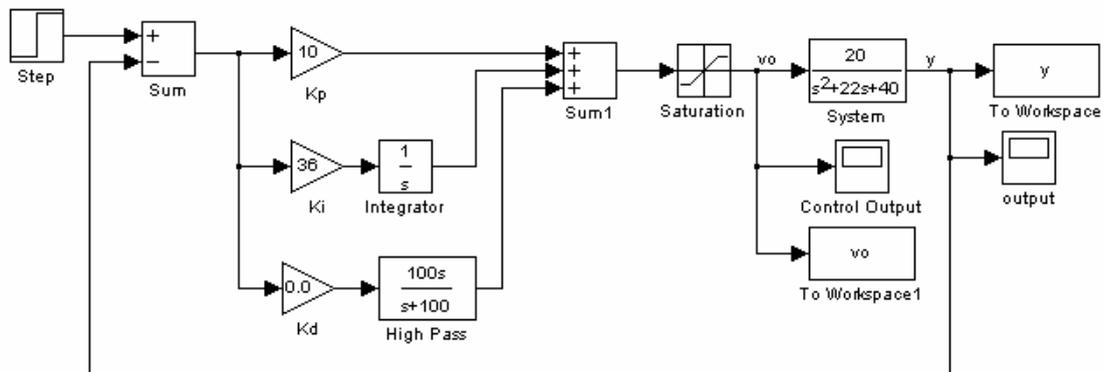
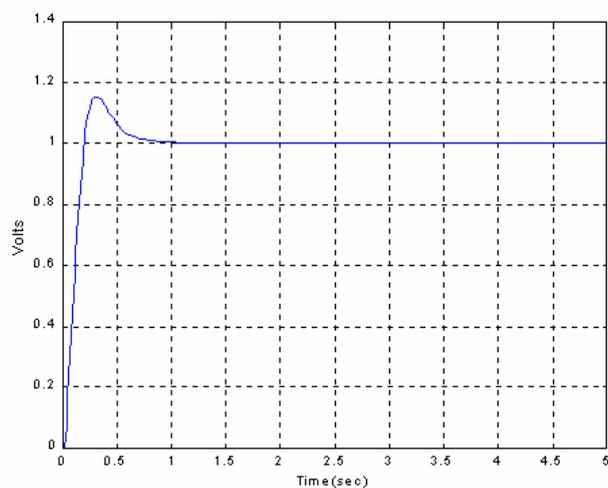
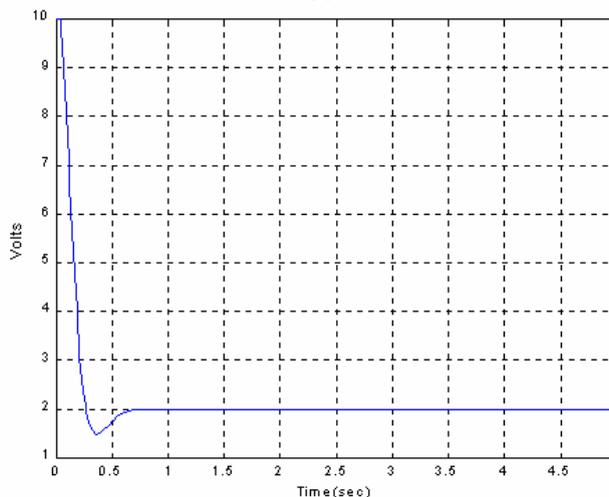


圖 11 Simulink 模擬方塊圖



(a)



(b)

圖 12 (a)系統輸出響應、(b)控制器輸出

### 5.3 LabVIEW 介面卡控制實驗與問題改善

吾人利用 OPA 設計一個二階電路，具有轉移函數  $G(s) = 20 / (s + 20)(s + 2)$ ，並且利用 LabVIEW 系統的方塊規劃取樣時間。系統初始值為 0 伏特，接著再把命令電壓設為 1 伏特。圖 13 與 14 分別為經由 LabVIEW vi 所顯示出的系統輸出歷史曲線圖。圖 15、16 與 17 為利用示波器量測的系統輸出，控制器輸出與用以測試取樣週期的系統測試信號。由圖 17 中可看出系統的取樣週期不固定，導致控制器的輸出相當不穩定。但是因為所設定的取樣頻率仍遠高於所需的十倍系統頻寬，因此系統的輸出仍在合理可以接受的範圍。



圖 13 系統輸出 0 伏特



圖 14 系統輸出 1 伏特

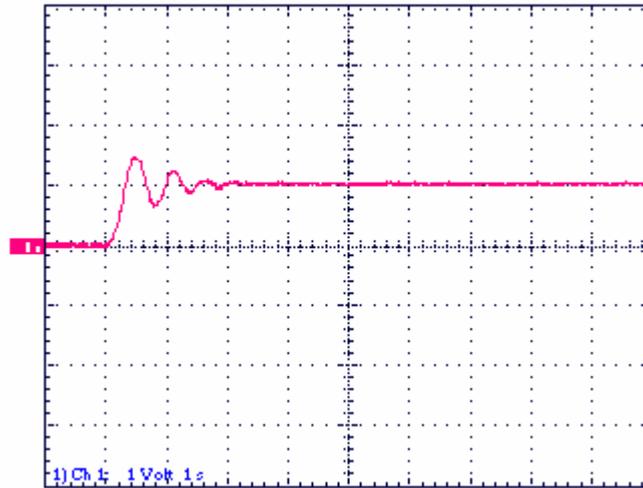


圖 15 二階系統輸出步階暫態響應

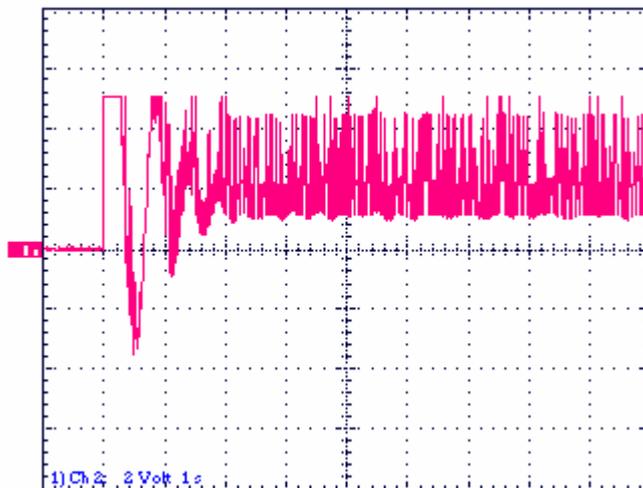


圖 16 PID 控制器輸出步階暫態響應

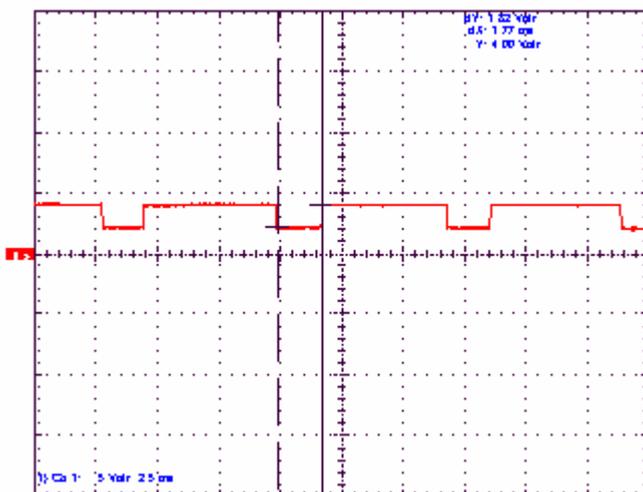


圖 17 LabVIEW 系統控制器輸出週期測試響應波形

由圖 17 看來每次取樣的時間並不固定，整體平均取樣時間較所設定為長。但經過變成執行檔且在 Windows 下的 DOS 執行時，系統的取樣時間就可以縮短到符合原先 1ms 的設計。圖 18 為系統輸出，而圖 19 控制器的暫態響應輸出，它比較接近模擬的控制器暫態響應。比較圖 15 與圖 18 的系統輸出步階暫態響應，發現幾乎是完全一樣，但以控制的品質來說，變成執行檔的 PI 控制器是比較好的。圖 20 執行檔 PI 穩態輸出來看取樣週期，與圖 17 比較，雖然取樣週期比未做成執行檔的短很多，但仍是不很固定，這是因為在 Windows 底下多工的關係，只是因為系統改為執行檔，不必在 LabVIEW 平台上執行，系統所需的時間較充裕，而能比較快速的執行程式。

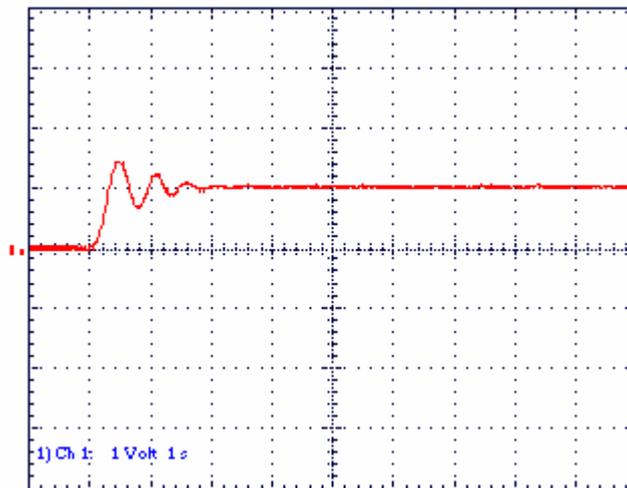


圖 18 執行檔系統輸出步階暫態響應

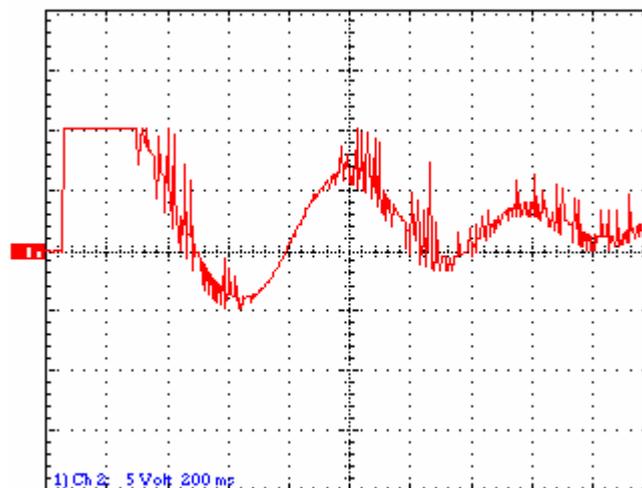


圖 19 執行檔 PID 輸出步階暫態響應

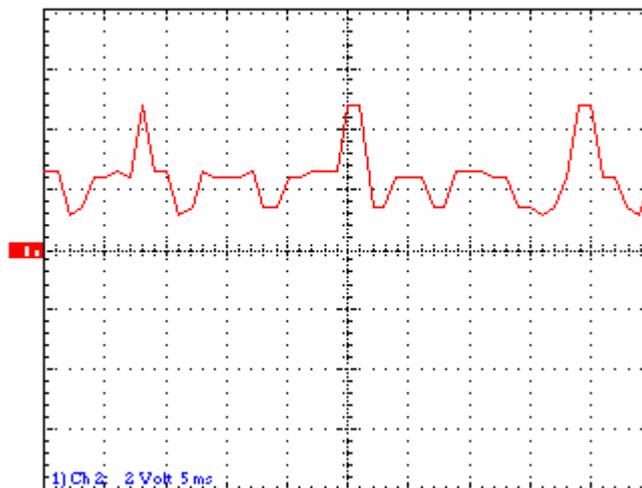


圖 20 執行檔控制器輸出態響應

## 6、基於非 NI 控制介面卡的 LabVIEW 平台控制系統設計

利用 LabVIEW 系統，當系統介面卡不具有即時控制功能，而完全利用軟體的規劃來進行數位控制器的設計時，很明顯的系統的性能將受到電腦本身 CPU 運作頻率及控制器系統是否已經轉成執行檔有關。而對於具有 250K/s 轉換速率的 A/D 轉換器介面卡，在執行即時控制的考量時，事實上是發揮不了它的作用的。為了克服 Windows 多工操作對於系統的影響及保有 LabVIEW 系統虛擬儀器畫面監控的功能；同時不必依賴 LabVIEW 介面卡而仍然能夠執行所設計的控制器系統，我們將 PI 控制程式的部分，利用 C 程式來撰寫，並且採用較 LabVIEW 介面卡便宜的一般市售介面卡來擷取類比資料，並且當成硬體的觸發來源。吾人以 C 撰寫 IRQ5 中斷控制副程式，並設定取樣時間為 1 ms。為順利的將類比轉數位的結果送至 LabVIEW 軟體系統，同時又可以利用 LabVIEW 的虛擬儀表功能傳送控制命令、PI 參數值至控制器以執行即時的控制設定，以及將控制結果傳回至 LabVIEW 監控畫面。吾人搜尋在 PC 內部的諸多已經存在的介面系統，包括印表機、網路卡、COM1/COM2 等。當這一些系統暫時不使用時，分配到他們的 I/O 位址就可用來作為傳遞 LabVIEW 系統與硬體觸發電路使用。而 LabVIEW 可隨時調整設定的目標(Set Point)，也可調整  $K_p$ 、 $K_i$  等參數，這一些動作皆是由所搜尋到的介面位址當暫存器，以便於互傳參數資料。介面卡的暫存器用來當成資料傳遞使用必須是可讀/可寫，並且還必須是暫存入的資料不會受到更改。為了找出可用的 I/O 位址，吾人利用 LabVIEW 寫一段 I/O 測試程式，獲得用來作為這一個測試控制系統的暫時參數儲存相對位址如表 1 所示。

接著進行所分析方式的 LabVIEW 系統監視與 PID 控制器設計。因為 LabVIEW 系統必須進行與表 1 所述介面位址的資料傳輸，因此，重新設計前置面板如圖 21 所示。在控制初期，控制器設定為 0 伏特時，顯示如圖 21 所示；當更改控制設定為 1 伏特時，顯示畫面如圖 22 所示。利用顯示歷史曲線的方式並不易獲得實際暫態響應的情況，底下接著利用示波器擷取控制電路輸出，以分析控制效果。

表 1 用於暫時存放參數的介面卡位址

控制參數名稱：	介面名稱：	暫存器位址：
比例參數( $K_p$ )：	印表機	0x37A
積分參數( $K_i$ )：	網路卡	0xE827
微分參數( $K_d$ )：	網路卡	0xE800
設定目標(Set Point)：	印表機	0x378
回授資料(Feedback Data)：	網路卡	0xE818

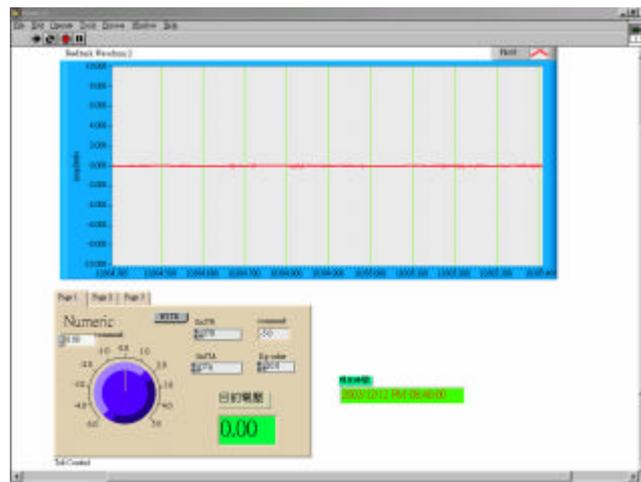


圖 21 系統一開始歸 0 伏特輸出

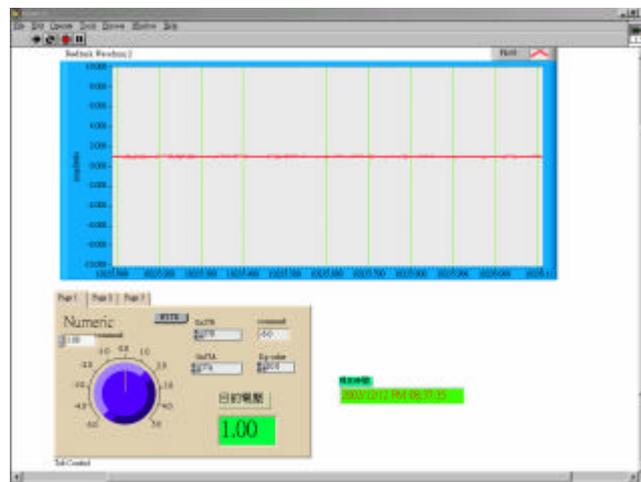


圖 22 系統輸出 1 伏特

採用所分析方式設計的系統輸出的暫態響應如圖 23 所示，因為控制器的取樣時間高出受控電路頻寬許多，因此在電路輸出響應上，與圖 15 及 18 並沒有明顯的不同。而圖 24 的控制器輸出則與圖 16 和 19 則有較明顯的不同，在信號的品質上改善了許多。圖 25 為 PI 控制器穩態控制輸出，由圖中可以明顯的看出控制器穩定的每隔 1ms 執行一次的控制命令。

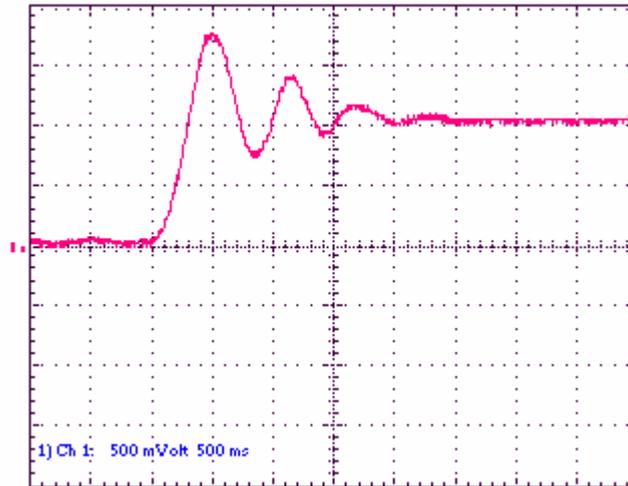


圖 23 受控系統步階響應

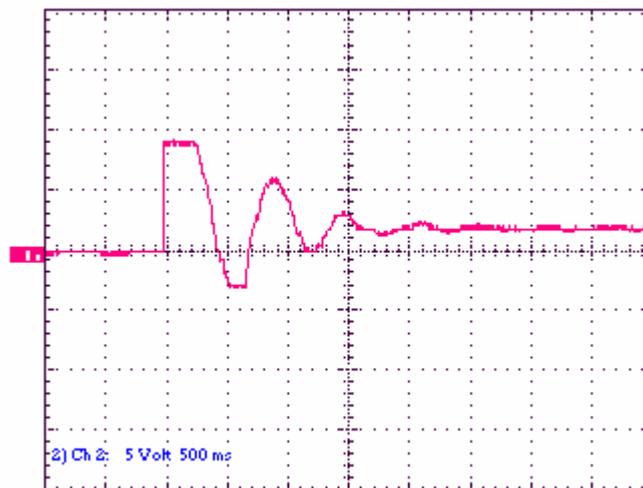


圖 24 控制器輸出暫態響應

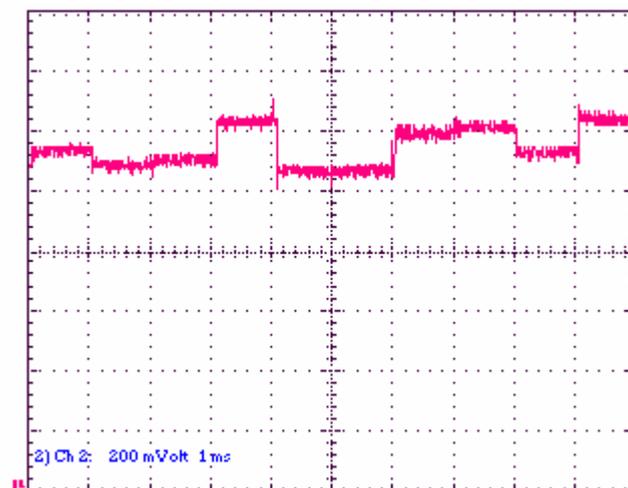


圖 25 控制器輸出穩態響應

## 7、結論

本論文探討 LabVIEW 介面卡 PCI-MIO-16E-4 用在即時控制的效能，藉由直接在 LabVIEW 平台上執行 PI 控制，以及將 LabVIEW 程式轉成執行檔.exe 的方式來執行。發現直接在 LabVIEW 平台上執行 PI 控制時，系統的取樣時間將因 windows 的多工作業及 CPU 操作的頻率快慢而無法固定。當將 LabVIEW 程式轉成執行檔時，系統的執行可以大量的節省時間，也較能符合即時控制的要求。接著我們省去不用 LabVIEW 介面卡，直接利用一般市售介面卡以執行 A/D 及 D/A 的功能，並且利用 IRQ5 執行數位控制的中斷，LabVIEW 系統則負責監視及命令參數更改與傳遞的功能；同時利用暫時不用的其餘介面系統位址當成暫存器。結果顯示所得到的控制效果相當理想，而且也可以達成 LabVIEW 虛擬儀表的功能。

## 8、參考文獻

- [1] <http://www.fly.to/labview>.
- [2] G. F. Franklin, J. D. Powell and M. Workman, "Digital Control of Dynamic Systems", Addison Wesley, 3<sup>th</sup> edition, 1998.
- [3] 鄧人豪、李錦昌、詹舜宇, "電力品質記錄器之研發、製作與測試", 電力電子雙月刊 2001/Aug pp.71~85。
- [4] 儲昭律、王智昱、蕭子健, LabVIEW 進階篇二版, 高立圖書有限公司。
- [5] 林啟萬、林鑫志、蕭子健, "LabVIEW 在系統上的應用弱視者電子擴影系統", 機電整合月刊 2001/May pp.76~79。
- [6] 蔡季霖、蕭子健, "LabVIEW 與微電腦 I/O 介面實驗", 機電整合月刊 2001/July pp.62~65。

