

伺服迴路與 EPCIO

工研院機械所/蔡祥益

關鍵詞

PCL(Positioning Control Loop) 位置控制迴路

Servo Loop 伺服迴路

EPCIO 定位控制輸入輸出

PDFF 比例微調前饋控制

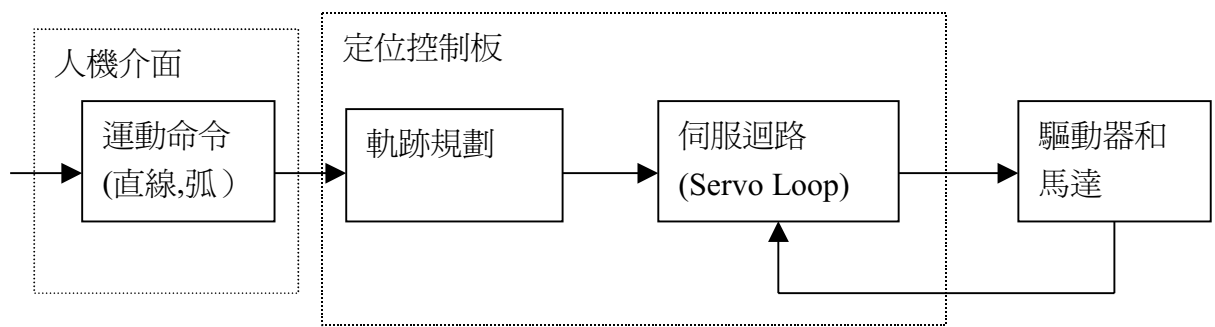
摘要：工研院機械所發展一顆運動模組 ASIC，EPCIO（英文簡稱是‘Exquisite Positioning Control Inputs and Outputs’）。在伺服迴路(servo loop)裡扮演重要角色，可以大量節省 CPU 寶貴的時間。它提供強大的 DDA(Digital Differential Analyzer)控制線路，可同時控制 6 軸，每軸運動指令 FIFO 64 筆深度，每筆運動指令補間步數:可規劃 $0 \sim (2^{10}$ -或 $2^{15})$ 步，輸出脈衝格式: A/B、Pulse/Direction、CW/CCW，輸出脈衝寬度:可規劃為 $0 \sim 4095$ 個工作時脈寬度。和 PCL(Positioning Control Loop)線路，有誤差計數器(Error Counter)長度:16 位元，比例型補償器，Gain 值範圍: $(0 \sim 127) \times (2^{-7} \sim 2^7)$ 。是實現伺服迴路的利器。

壹. 前言

一般產業機械需要用到的重要組件之一，伺服控制系統，提供位置，速度，扭力控制。其核心伺服迴路將應用 EPCIO 強大的功能和 PDFF 控制法則，來實現高性能多樣性的控制。此伺服迴路得以快速完成 1. 送出脈波形式的位置控制信號. 2. 送出電壓形式的速度控制信號. 3. 送出電壓形式的扭力控制信號。在 EPCIO 裡的 DDA 和 PCL 架構，可以在此伺服迴路中完全發揮其功能。

貳. 運動控制器簡介

一般的運動控制器動作流程如下圖一，人機介面下運動命令，如點對點，直線或圓弧給定位板。在軌跡規劃部分把運動命令切分成細點，這些細點的軌跡是由各軸運動合成的結果，所以這些細點必須轉換成各軸要走的分量，傳送到伺服迴路執行。伺服迴路就需要忠誠的控制各個馬達，在一定的速度，扭力及響應之下，使各軸依其分量到達定位。



圖一 控制器方塊圖

現在一般的馬達驅動器可以接受 1.脈波命令，2.速度命令，3.扭力命令。

一. 接受脈波命令

此時位置控制迴路是在馬達驅動器裡，外面送幾個脈波，馬達驅動器就讓馬達走幾步。這種脈波命令輸入介面有：A/B, CW/CCW, Pulse/Dir 方式。

二. 接受速度命令

此時位置控制迴路是在定位控制板上，速度控制迴路是在馬達驅動器裡。這種速度命令輸入介面是類比輸入， $\pm 10\text{V}$ 電壓。 $+10\text{V}$ – 馬達全速正轉， 0V – 靜止， -10V -馬達全速反轉。

三. 接受扭力命令

此時的位置控制迴路和速度控制迴路都是在定位控制板上，扭力控制迴路是在馬達驅動器裡。這種扭力命令輸入介面也是類比輸入， $\pm 10\text{V}$ 電壓。 $+10\text{V}$ – 最大正扭力， 0V – 零扭力輸出， -10V -最大負扭力。

參. 伺服迴路

因此伺服迴路裡需負責往上承接軌跡規劃的位置命令，然後轉換成脈波，速度，或扭力命令輸出到馬達驅動器。伺服迴路的程式啟動是利用 DDA Cycle 中斷，每 400us 產生一次，觸發中斷服務程式，在程式裡做伺服相關工作，離開中斷服務程式後，此中斷必須保持隨時被啟動狀態，因為再等 400us ，中斷將會再度發生。如此便會隨時不斷監督伺服器的反應。

本伺服迴路程式處理包括 1. 送出脈波形式的位置控制信號。2. 送出電壓形式的速度控制信號。3. 送出電壓形式的扭力控制信號。在 400us 裡要完成複雜的運算及控制，則有賴於硬體的支援，也就是本部門所提供的運動控制模組 ASIC- EPCIO。

本伺服控制在位置迴路採 P 控制，也就是可以利用 ASIC-EPCIO 提供的硬體 PCL 架構來完成。速度迴路採泛用型的 PDFF 控制架構。至於位置脈波命令輸出完全用 ASIC-EPCIO 提供的硬體 DDA 架構來完成。我們就這三種模式（脈波，速

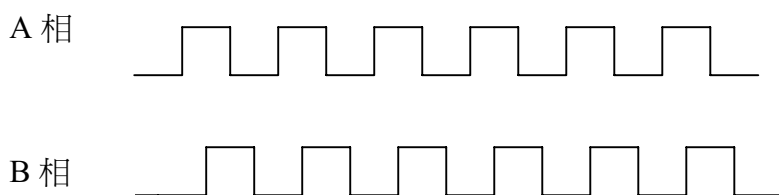
度，及扭力) 分別敘述如下：

肆. 脈波信號輸出模式

一. 脈波輸入格式介紹

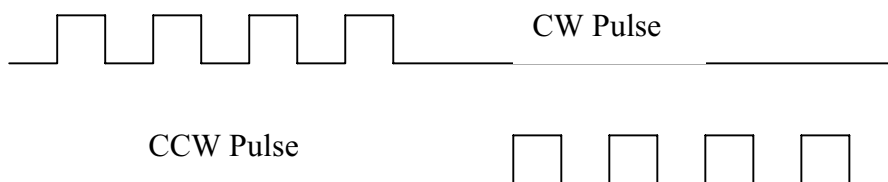
一般的脈波輸入有三種方式即 A/B, CW/CCW, Pulse/Dir。

A/B 方式：如圖二，A 相領先 B 相，則馬達正轉，否則 B 相領先 A 相，則馬達正轉。其中 A 相與 B 相的相差為 90 度。



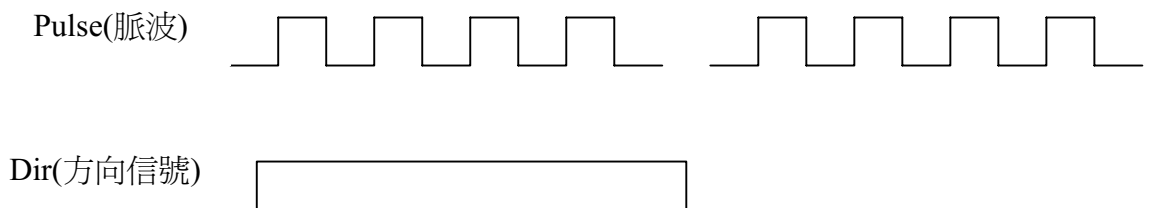
圖二

CW/CCW 方式：如下圖三，當有 CW 脈波時則馬達正轉，有 CCW 脈波時則馬達反轉。



圖三

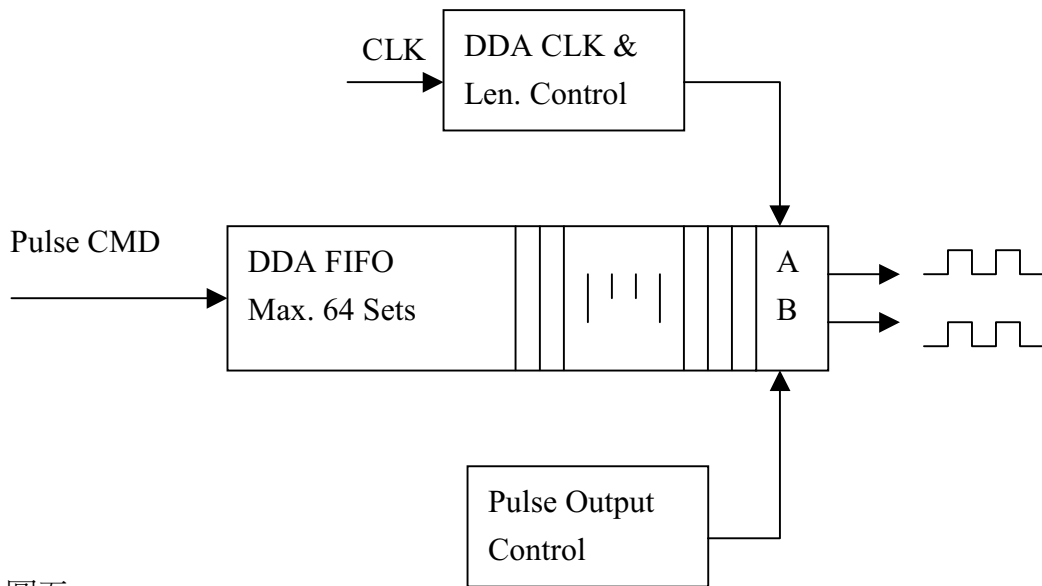
Pulse/Dir 方式：如圖四當有脈波輸入及方向脈波為高電位，則馬達正轉，當有脈波輸入及方向脈波為低電位，則馬達反轉。



圖四

二. EPCIO 提供的 DDA 功能

DDA 就是 Digital Differential Analyzer，是在一段時間內如 1ms（此段期間叫 DDA Cycle），把數量為 m 如 1024 的 Pulses 平均送出去，也就是使脈波與脈波の間隔相等。EPCIO 提供六組 DDA 功能，可以同時控制六軸。每組的方塊圖如圖五：



圖五

- ◆ 每組的 DDA 都內含 64 筆放 Pulse Command 的 Buffer 稱 FIFO，所以最多可存放 64 筆運動命令。採用先進先出法則。
- ◆ Pulse Command 脈波數可以是從 $1 \sim 2^{15}$ 個脈波(可規劃的)
- ◆ 若 DDA 被啓動，則每一 DDA Cycle 時間消耗一筆運動命令。
- ◆ 當運動命令被消耗完或是，剩下規劃最小中斷的筆數時，則可以產生中斷通知系統，即時補充運動命令，使運動軸繼續移動，不至於半途停頓。因此有最長的時間 63 ms (DDA Cycle \times 64, 如 $1\text{ms} \times 64=64\text{ms}$ ，其中 1 ms 處理伺服迴路相關程式)，EPCIO 硬體可獨自處理 DDA 機制，系統的 CPU 可以做其他的工作。由此可知 EPCIO 提供強大功能，省下許多 CPU 時間。
- ◆ Pulse Output Control 提供 Pulse 輸出介面規劃，如輸出端電氣信號極性可以正負對調，A/B 兩端也可以對調，這些功能使輸出端配線具彈性和匹配性，維護容易。
- ◆ 有一些馬達驅動器所能接受輸入的脈波寬度也不一致，DDA 模組也提供可以規劃的脈波輸出寬度。如 DDA CLK & Len. Control 方塊圖所支援。
- ◆ 爲了因應不同系統應用，DDA Cycle 時間也可以規劃的。如下所述
- ◆ DDA Cycle Time 計算法則

$$\text{DDA cycle time} = T * (D+1) * 2^L$$

--請參閱 EPCIO 使用手冊部分資料

Where T = System clock period

D = DDA clock Divider ($0 \leq D < 4096$) (12 bits)

E = DDA pulse Extension number ($1 \leq E < 2048$) (11 bits)

L = DDA Length ($10 \leq L \leq 15$)

T : 系統時脈週期，如 40Mhz, $T=25$ ns。

D : 除頻值是 D ($0 \leq D < 4096$, 12 bits)，DDA 模組的 DDA CLK 是由系統時脈經過除頻(除頻值是 D)所提供，再觸發 DDA 模組。

E : 是脈波寬度係數 ($1 \leq E < 2048$, 11 bits)，

輸出脈波寬度 = $25\text{ns} \times E$ 。

L : DDA Length ($10 \leq L \leq 15$)，也就是 Bits 數。 2^L 是表示每筆運動命令最大的輸出脈波量。如 $L=12$ ，則每筆運動命令的輸出脈波量從 1 ~ 4095。

◆ 最大一筆運動命令的輸出脈波量的規劃

$$\begin{aligned} \text{Maximum Command} &= 2^L && \text{if } D > E \\ &= 2^L / x && \text{if } D \leq E \end{aligned}$$

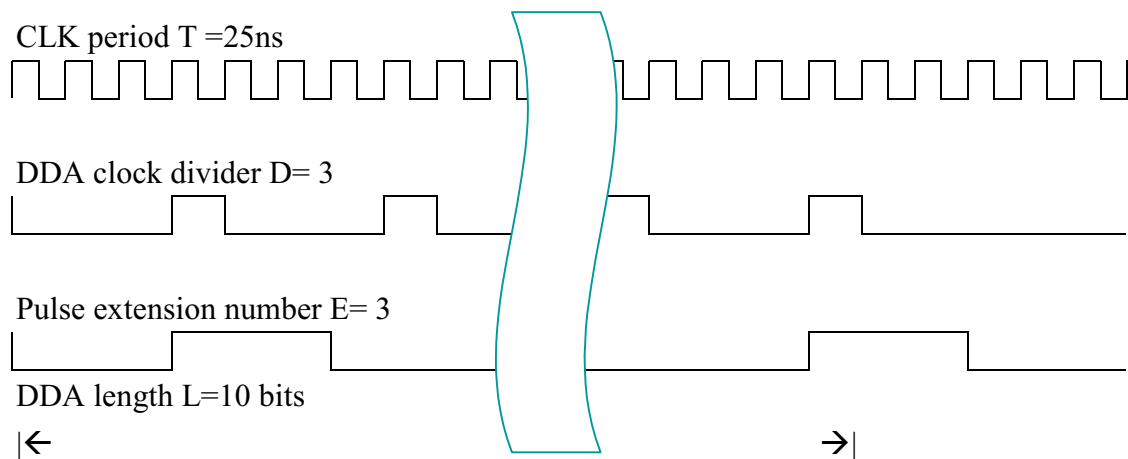
Where $(E+1)/(D+1) \leq x < 1 + (E+1)/(D+1)$

/* x 為 $(E+1)/(D+1)$ 之高斯值



範例： For 40MHz system clock, DDA clock divider set to 3, DDA pulse extension number set to 3, DDA length set to 10 bits. Then, what's the DDA cycle time?
What's the maximum pulse command?

圖六是：系統時脈 CLK，T=25 ns。除以 3 後的 DDA CLK。及輸出脈波（x3 的寬度）。



$$\begin{aligned}
 \text{DDA Cycle Time} &= T \cdot (D+1) \cdot 2^L \\
 &= 25\text{ns} \cdot (3+1) \cdot 2^{10} \\
 &= 0.1024 \text{ ms} \\
 \text{Max Command} &= 2^L / x \\
 &= 2^{10} \\
 &= 1024 \text{ Pulse/DDA}
 \end{aligned}$$

圖六

三. 實現

因此伺服迴路裡，若在脈波命令輸出模式，可以利用最小筆數中斷機制，產生中斷，執行伺服迴路，只需負責把整批運動指令放入 FIFO 即可。便可減少 CPU 時間，不必每個 DDA Cycle 中斷驅動伺服迴路。

範例：設 DDA Cycle Time = 1ms, (因為系統要求在 1 ms 能送出 $1 \sim 2^{15}$ pulses)
Servo Loop 寫入 FIFO 資料及處理相關控制約需 2 ms 時間，
軌跡規劃必需花費 10 ms 可以產生 1 ~ 50 筆運動命令，
請問至少要用到幾個 FIFO，及最小運動命令筆數中斷規劃，才能使運動順暢不致停頓。

Ans: 最小運動命令筆數中斷是 3，才來得及把運動命令放入 FIFO，因為 Servo

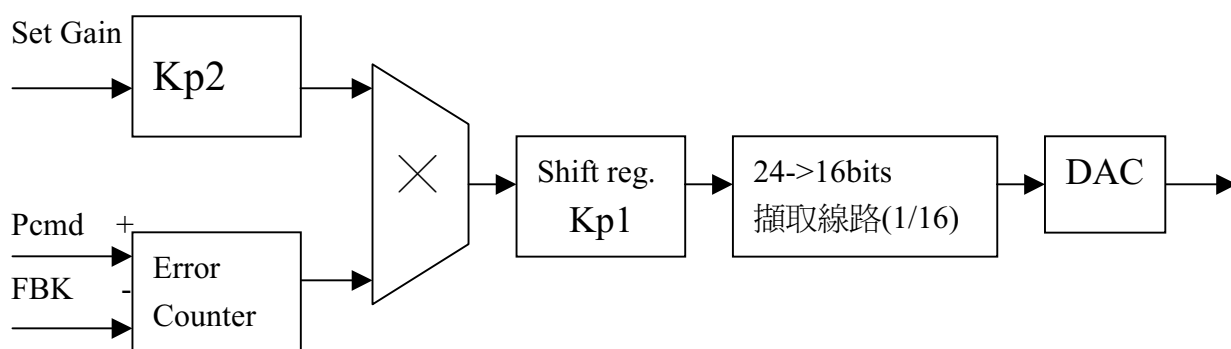
Loop 佔用 2 ms 時間。最小一次放入 FIFO 運動命令筆數 = $(10\text{ms} + 2\text{ms}) / 1\text{ms} = 12$ 筆。但是若只放 12 筆，則 CPU 也只能處理運動控制，人機介面動彈不得。因此最小一次放入 FIFO 運動命令筆數至少大於 12。若設其值為 50 筆， $1\text{ms} \times 50 = 50\text{ms}$ 。 $50\text{ms} - 10\text{ms} - 2\text{ms} = 38\text{ms}$ ，則整個系統處理人機介面可以有 38 ms 時間，餘 12 ms 為用在運動控制上。

伍. 速度命令輸出模式

一. 速度命令輸入介面

一般馬達驅動器的速度命令輸入介面是採用類比， $\pm 10\text{V}$ 電壓，正電壓正轉，負電壓則反轉。

二. EPCIO 提供的 PCL 功能



圖七

EPCIO 提供的 PCL 功能方塊圖如圖七。

- ◆ 在伺服迴路的速度命令輸出模式，採用 EPCIO 硬體來實現，即 PCL 線路，此迴路用 P 控制。
- ◆ P Gain 值算法是 $P(\text{Total Gain}) = Kp2 \times Kp1 \times 1/16$ 。
- ◆ Pcmd 是要移動多少位置的脈波量，FBK 是從馬達迴授來的 Encoder 脈波量，相減後放入 16-bit Error counter。Kp2 是 8-bit gain value (-128~127)，兩者相乘得 24 bits 放入 Shift Reg.。
 - Shift Reg. 可做左移或右移 0~7 Bits(也就是乘 Kp1，2 的倍數)，相當是乘或除以 $1 \sim 2^7$ 。
 - EPCIO 的 DAC 輸出是 16-bit，故需要將 24 bits 刪除 8 bits，得 16 bits。

其刪除方式就是刪掉最小 4 bit 及最大 4 bit 位元，湊取其中第 4 bit 到第 20 bit，擷取得 16 bits，最後送出類比 +/- 10 V 電壓，到馬達驅動器。也就是因為有 24 bits 轉到 16 bits 的擷取線路，所以要乘以 1/16。

◆ Kp1 的作用是調整 PCL 迴路輸出值。設 $Kp2 * Error\ Counter = X$ 。

- 若 $Kp1=1$ ，則輸出值(X)會因乘(1/16)，而導致控制解析度增大，X 值變小，要恢原來的值就需設 ($Kp1=16$)。
- 因此能預估系統的所有 X 值皆小於 2^{16} 值，則設 $Kp1=16$ 較妥，可以保持原來的迴路輸出值，不失真。
- 若 $2^{16} < X < 2^{20}$ ，則設 $Kp1=1$ ，爲了裁減成 16 bits，適用於 DAC 輸出。
- 若 $2^{16} < X < 2^{24}$ ，則設 $Kp1=1/128$ 較妥。也是爲了裁減成 16 bits，適用於 DAC 輸出。但其控制迴路值解析度很大。
- Kp1 的另一項功能是，若 Kp2 較大時，Kp1 可以動態提供 PCL 迴路微幅增益調整。也就是每個 DDA 可以調 Kp1 值，達到動態控制目的。

◆ 範例

DAC 總輸出是 $= Error\ Reg. * Kp2 * Kp1 * 1/16$

總輸出 DAC Reg. 是 16 bits

P1. 設 $Error\ Reg.=2512$, $Kp2=16$, $Kp1$ (左移 3 位)=8

Ans:則輸出是 $=2512*16*8*(1/16)=20096$

P2. 設 $Error\ Reg.=2512$, $Kp2=16$, $Kp1$ (左移 4 位)=16

Ans:則輸出是 $=2512*16*16*(1/16)=32767$

原值是 40192, 因爲輸出是 16 bits 長

P3. 設 $Error\ Reg.=2512$, $Kp2=16$, $Kp1$ (設爲 0)=1

Ans:則輸出是 $=2512*16*1*(1/16)=2512$

P4. 設 $Error\ Reg.=2512$, $Kp2=16$, $Kp1$ (右移 2 位)=1/4

Ans:則輸出是 $=2512*16*(1/4)*(1/16)=628$

三. 實現

伺服迴路裡，若在速度命令輸出模式，其輸出以電壓形式送出。我們用 PCL 提供的功能，只需負責把運動命令放入 FIFO 即可，在每個 DDA Cycle，從 FIFO 取出 送到 Pcmd，經過 PCL 到 DAC，輸出電壓到馬達驅動器。這些都是 EPCIO 硬體線路完成，不需用到 CPU 時間，只需事先規劃 Kp1, Kp2 即可。再者利用最小筆數中斷機制，產生中斷，執行伺服迴路，負責把整批運動指令放入 FIFO。因此有最長的時間 63 ms ($DDA\ Cycle \times 64$, 如 $1ms \times 64=64ms$ ，其中 1 ms

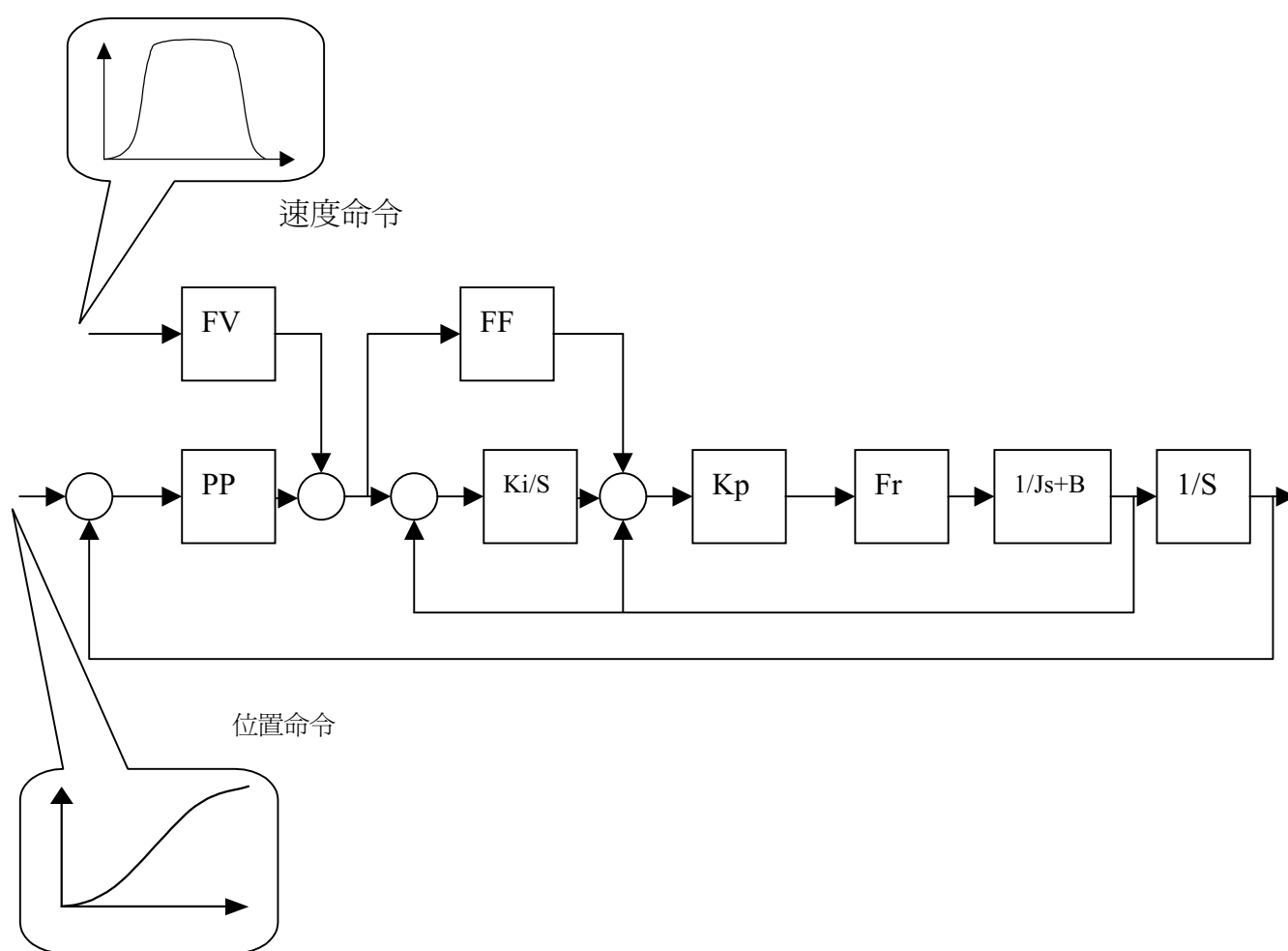
處理伺服迴路相關程式)，EPCIO 硬體可獨自處理 DDA 機制，系統的 CPU 可以做其他的工作。若要實現動態控制，則每個 DDA Cycle 需維護 Kp1。

陸. 扭力命令輸出模式

一. 扭力命令輸入介面

一般馬達驅動器的扭力命令輸入是採用類比， $\pm 10V$ 電壓， $+10V$ 為最大正扭力輸入， 0 零扭力輸入， $-10V$ 為最大負扭力輸入。

二. 控制架構



圖八

- ◆ 在此模式之下，伺服迴路裡的位置迴路採用 P 控制加前饋控制，速度迴路用 PDFF 架構。

◆ 由上面控制架構圖八：

$1/Js+B$ ：為馬達轉移函數， J ：轉動慣量， B ：黏滯摩擦係數。

在速度迴路用的 PDFF 法則，其好處是：

* 若 $FF=0$ 時，則呈現為 PDF 或 ID 控制器，具有類似微分器功用及特性。

可以增加頻寬，減少 rise time and settle time，減少 overshoot。

* 若 $FF=1$ 時，則呈現為 PI 控制器的功用及特性。

可以增快 rise time，減少頻寬，有利於高頻雜訊抑制功能，

有較好的穩態追蹤性能，穩態誤差可趨近於零。

因此可以藉著調 FF gain 適應自己的控制器。

FV：位置迴路前饋增益，具有增加位置追蹤性。

PP：位置迴路比例增益，增益愈大，反應越快，穩定性越不佳，容易引起震盪。

KI：速度迴路積分增益，增益愈大可以減少穩態誤，容易產生 Ring 現象，其值小於 1。

Fr：轉動慣量調變增益，一般值為一，用於微調受控體。

Kp：速度迴路比例增益，增益愈大，反應越快，穩定性越不佳，容易引起震盪。

◆ 調增益是要依經驗法則來進行，下列步驟供參考。

1.先設定 $FV=0, Fr=1, KI$ 盡量小 (0.1)。

2.先調外迴路，即 PP。然後再調 KP。調在 PP, KP 最大且不會引發震盪。

3.然後再調 KI, FF, FV 於所要的規格。

四. 實現

伺服迴路裡的扭力命令輸出模式，都是用軟體實現，因此其 CPU 所付出的時間較多，代價高。其中速度迴路的 KI 項是累積的，因此需設最大截止數，避免使驅動器被進入飽和區而誤動作，其截止量設在飽和量的 % 3 ~ 10 左右。在整個伺服迴路裡，計算時也需注意不能產生溢流現象 (Overflow)，以保證運作正確。

結論

本文主要敘述由 EPCIO 提供的功能，如 DDA 和 PCL 線路所提供的功能機制，在伺服迴路裡，處理位置，速度，扭力命令輸出，完整的實現 EPCIO 功能，發揮到淋漓盡致。提升整體系統的效率。

參考文獻

[1].DSP 伺服控制法則與參數調整研究期末報告，蔡明祺，May, 1999。

[2].EPCIO ASIC 應用設計說明手冊。工研院機械所，May, 1999。

[3].B.C. Kuo, Automatic Control Systems, Prentice-Hall,1991.