

第3章 運動控制命令

3.1 脈衝輸出命令

MCX314 對於輸出脈衝命令有兩種模式：一種是定量脈衝輸出模式(Fixed Pulse Command Output Mode)，另一種是連續脈衝輸出模式 (Continuous Pulse Command Output Mode)。使用者能透過 MCX314 特定的暫存器來選擇脈衝輸出模式，選擇所要的命令脈波模式可以透過硬體跳線(jumper)的設定和軟體程式的選擇來達成；可選擇的命令脈波模式，大體上可分為單脈波輸出模式(PULSE/DIR)和雙脈波模式輸出模式(CW/CCW)，每種模式依驅動方向(正/負向)又分為兩種類型。如表 3-1 所示。圖 3-1~圖 3-4 為相關之脈波輸出模式示意圖。

表 3-1 選擇脈衝輸出類型

脈衝輸出類型	驅動方向	輸出脈衝波形	
		nPP / PULSE	nPM / DIR
PULSE / DIR	正向	PULSE	LOW
	負向	PULSE	HIGH
CW / CCW	正向	PULSE	LOW
	負向	LOW	PULSE

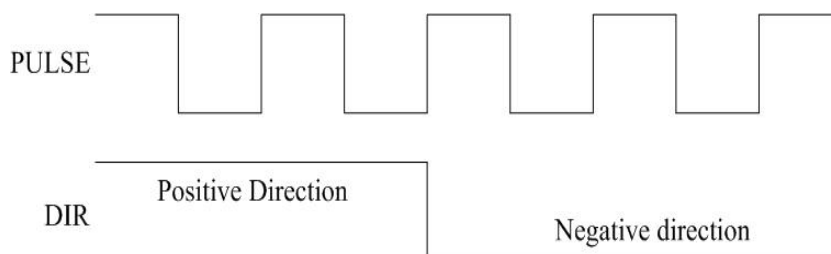


圖 3-1 脈波輸出類型一

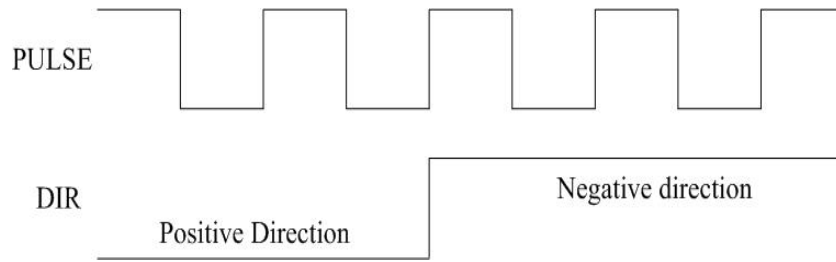


圖 3-2 脈波輸出類型二

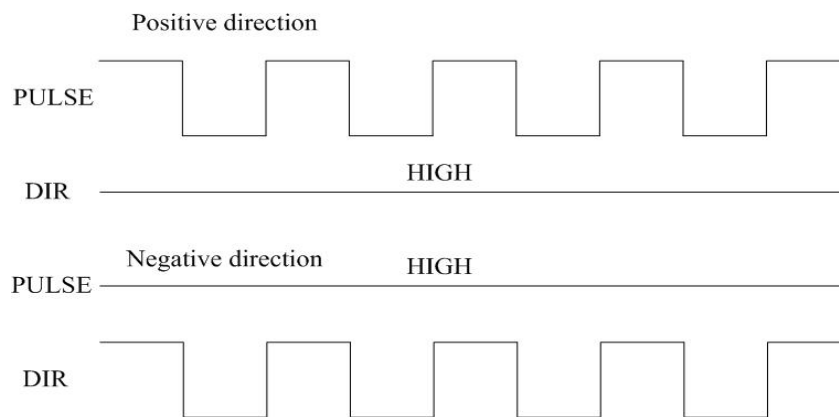


圖 3-3 脈波輸出類型三

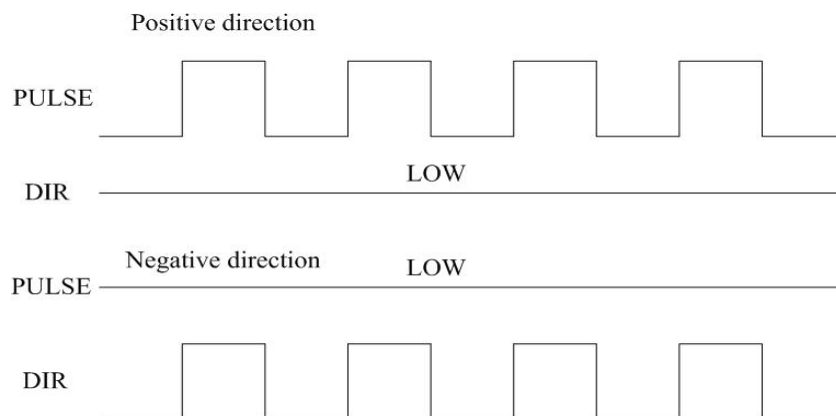


圖 3-4 脈波輸出類型四

3.1.1 定量驅動

定量驅動是依照指定之輸出脈波數進行等速或加減速驅動，將控制物移動至所指定之位置，如圖 3-5 所示。當 CPU 將脈衝數寫入 MCX314 並配置加減速比例以及速度之後，MCX314 將產生脈衝並且自動輸出。對於加減速定量驅動，當殘餘的輸出脈波數小於消耗於加速時的脈波數時，則開始減速；當指定之輸出脈波數結束時，即結束驅動。在定量驅動中，輸出脈波可以再加以變更，如圖 3-6 所示。若於加減速驅動之減速停止執行中，增加輸出脈波時會有再加速之情形，如圖 3-7 所示。減少輸出脈波則有可能產生立即停止之情形，如圖 3-8 所示。

MCX314 在定量驅動模式下，會自動計算出加減速點，並且使加速度區段的脈衝數等於或小於減速度區段的脈衝數，我們可以自由改變加減速點，或者為減速設置緩衝值 (shift pulse)，當設置緩衝值時，MCX314 會因緩衝值而提前減速，減速完成後所剩餘的脈波數，將會以初始速度輸出，直到脈波輸出完畢。如圖 3-9 所示。

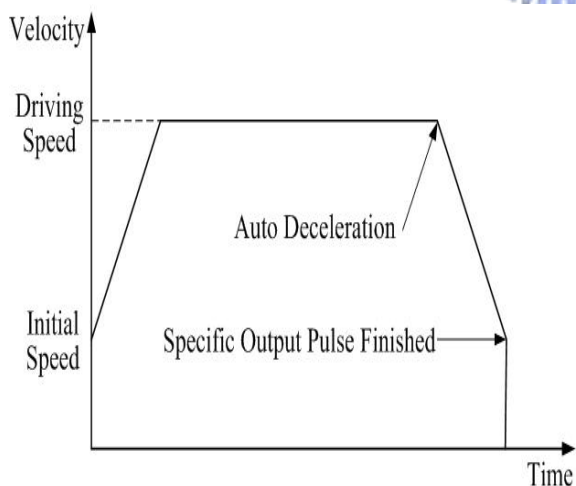


圖 3-5 定量驅動

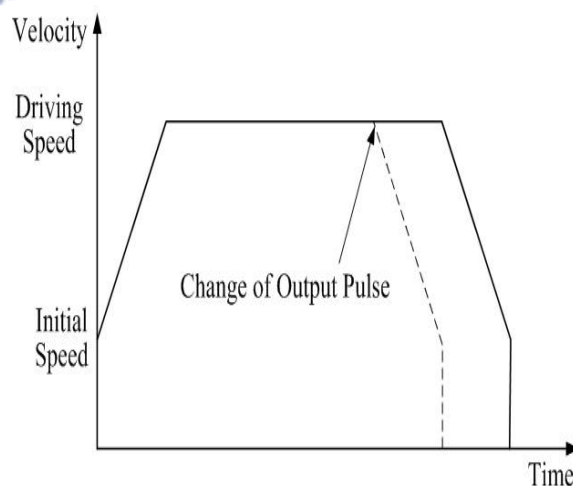


圖 3-6 定量驅動中輸出脈波數之變更

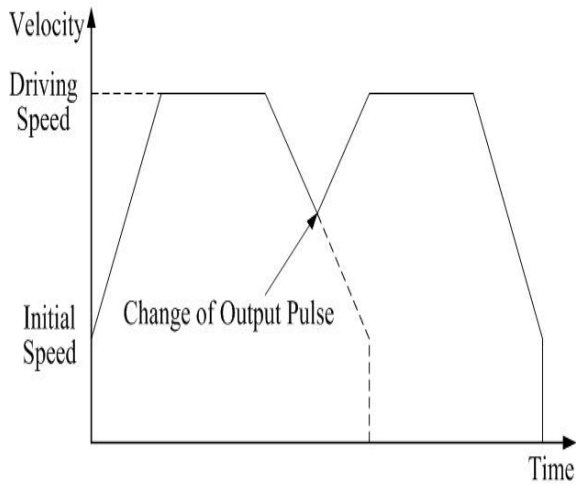


圖 3-7 減速時增加輸出脈波

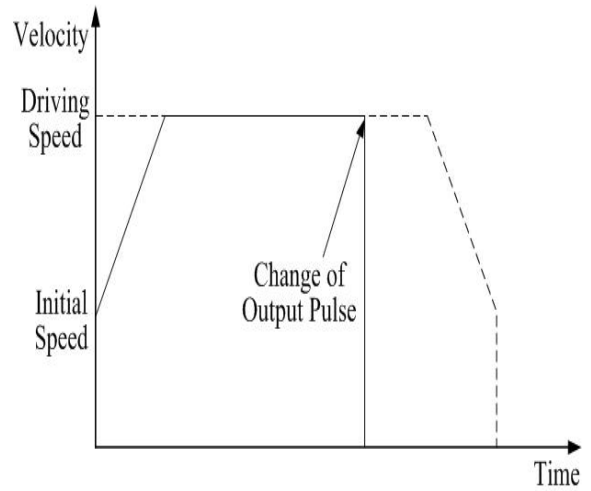


圖 3-8 變更輸出脈波為更少之脈波數

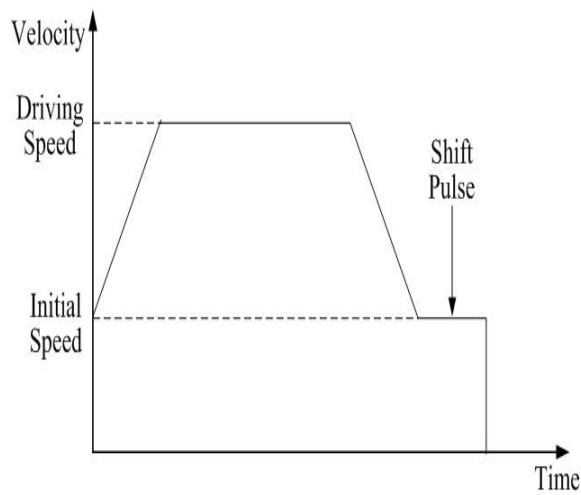


圖 3-9 定量驅動之緩衝脈衝設置

3.1.2 連續驅動

當脈衝輸出模式設置在連續驅動狀態時，MCX314 將一直以特定的速度驅動脈衝，直到接收到停止命令或者是外部中斷信號。連續驅動的主要應用於尋找原點、速度控制。如圖 3-10 所示。

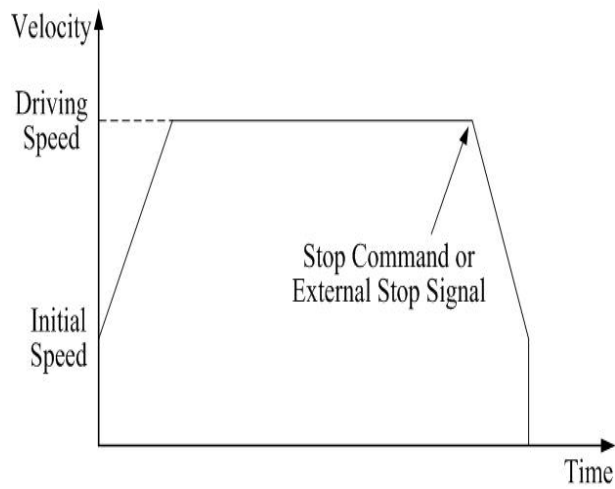


圖 3-10 連續驅動

3.1.3 等速驅動

當 MCX314 設置的驅動速度低於初始速度時，它則不執行加減速驅動，而是開始等速驅動。如果在找尋原點的過程中，希望當原點感測器或是編碼器(Encoder)的原點信號致動時(Enable)，則立即停止的話，則不建議使用加減速驅動，而是採用在低速時的等速驅動。而等速驅動功能也應用在多軸的補間運動，它能控制兩軸或三軸的合成速度，使其合成速度保持一定。如圖 3-11 所示。

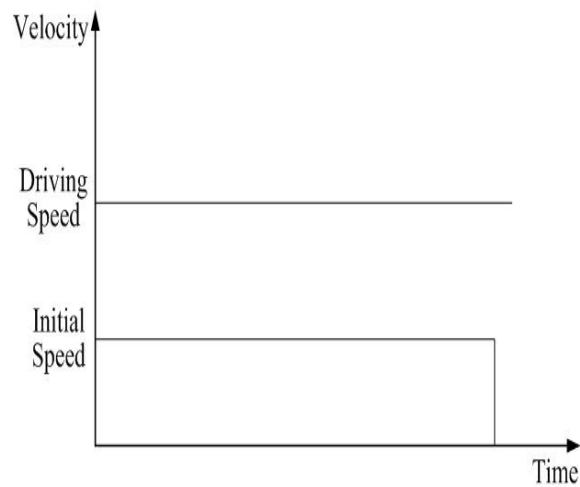


圖 3-11 等速驅動

3.2 加減速規劃

在工具機的運動控制中，為了提高運動路徑規劃的能力與降低軌跡誤差，應避免馬達的控制電流過大而進入飽和區，因此利用加減速的規劃，讓馬達的力矩輸出盡可能平滑且有效率。MCX314 將各軸之驅動脈波輸出由+/-方向之定量驅動命令，或是連續驅動命令來執行，設定運動模式或運動參數便可控制驅動速度曲線成為等速、線性加減速或 S Curve 加減速。

3.2.1 直線加減速驅動

直線加減速即是工業界常用的梯形加減速規劃。直線加減速驅動為自初始速度開始，由設置之加速度以一次線性加速的方式，將脈波速度提高至所設置之驅動速度為止；包括位移 S、初始速度 VO、驅動速度 VD、加減速時間 TA，以驅動速度 VD 做等速運動所需時間 TM，利用以上參數來規劃直線加減速。

等加速度公式：

$$VD = VO + at \quad (3-1)$$

以驅動速度運動所需時間：

$$TM = S / VD \quad (3-2)$$

加速度：

$$a = (VD - VO) / TA \quad (3-3)$$

加減速規劃如圖 3-12 所示。

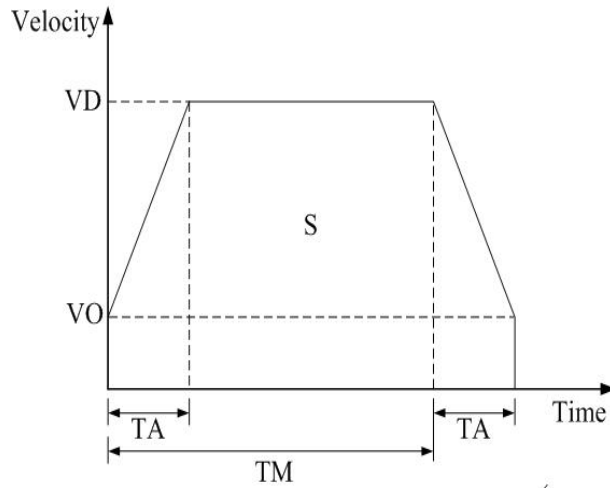


圖 3-12 直線加減速規劃

在定量驅動中，加速時會計算其消耗的脈波數，當輸出脈波數之殘餘數量小於加速脈波數時，即開始減速。如果在加速途中下達減速停止之命令，或輸出脈波數不足，無法加速至驅動速度所需的脈波數時，會於加速途中減速。如圖 3-13 所示。

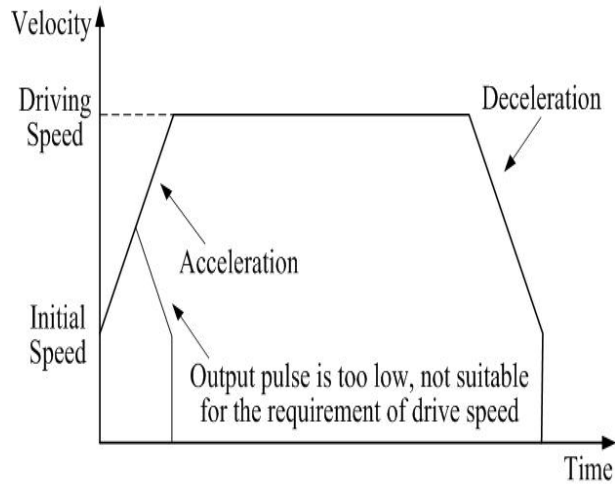


圖 3-13 直線加減速驅動

3.2.2 S-Curve 加減速驅動

S-Curve 的加減速規劃為二次速度曲線的兩段組合，假設二次曲線的加減速時間為 TA ：

前段(1)的速度方程式：

$$V(t) = Ct^2, \quad t < TA/2 \quad (3-4)$$

後段(2)的速度方程式：

$$V(t) = V - C(TA - t)^2, \quad t > TA/2 \quad (3-5)$$

由邊界條件：

$$V(0) = 0, V(TA/2) = V/2, V'(0) = 0 \rightarrow C = \frac{2V}{(TA)^2} \quad (3-6)$$

如圖 3-14 所示：

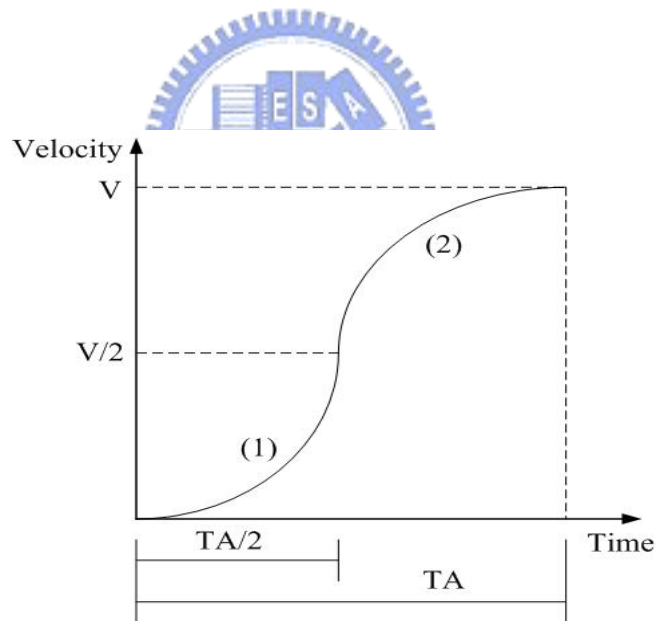


圖 3-14 S-Curve 加減速結構

將直線加減速與 S-Curve 加減速加以組合，可以得到混合的加速度曲線，其中 S-Curve 為(1)(3)段，規劃時間為 TA ；直線速度為(2)段，規劃時間為 $TA - (2 \times TS)$ ，在三段的連接點必須有相同的加速度。

前段(1)的速度方程式：

$$V(t) = C_1 t^2, \quad t < TS \quad (3-7)$$

中段(2)的速度方程式：

$$V(t) = C_2 t, \quad TS < t < TA - TS \quad (3-8)$$

後段(3)的速度方程式：

$$V(t) = V - C_1(TA - t)^2, \quad TA - TS < t < TA \quad (3-9)$$

常數 C_2 即為直線斜率：

$$C_2 = \frac{V - 2VS}{TA - 2TS} \quad (3-10)$$

由(1)(2)段連接處其加速度相等之邊界條件： $V'(TS) = C_2$ (3-11)

可求得 $C_1 = \frac{V}{2[TS^2 + (TS \times (TA - 2TS))]}$ (3-12)

如圖 3-15 所示：

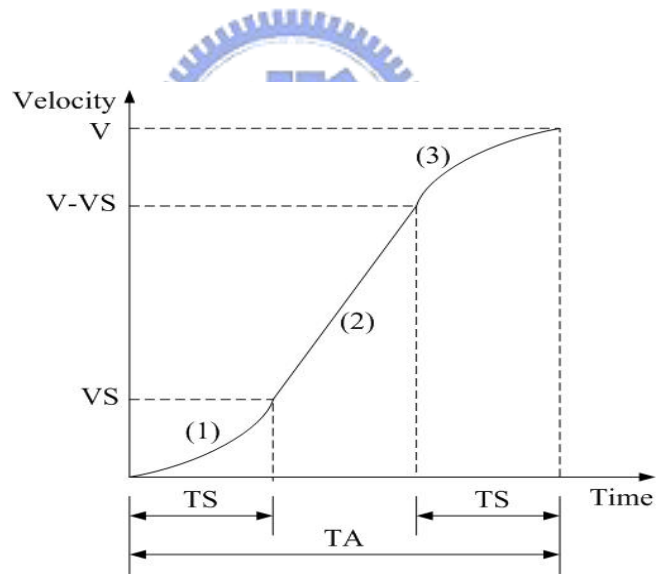


圖 3-15 混合加減速規劃

MCX314 亦利用混合加減速的規劃機制，在 S-Curve 方式加減速驅動時，是將加減速度再以一次線性加速方式做增減。所以加速度曲線並非是線性的，而是以梯形的方式重複變化。如圖 3-12 所示：S-Curve 的加速過程分為 a、b、c 三各區域，減速過程分為 e、f、g 區域。假如驅動速度小於或等於區域 a 的速度時，b 區域就會消失，成為完全的 S-Curve。

當完全的 S-Curve 在定量脈衝加減速驅動中的計算，如果定量脈衝驅動命令不足以使 S-Curve 加速到所需要的速度，則 S-Curve 的加速度將在 a 區域結束前便開始減速，因此會導致一個三角形的加速度曲線，如圖 3-13 所示：假設初始速度為 0，加速度變化率為 K，則在加速區域內 t 時刻的速度可以描述為

$$V(t) = Kt^2 \quad (3-13)$$

所以從 0 時刻至 t 時刻所輸出的脈衝 p(t) 即為速度的積分。可表示為

$$p(t) = \int V(t) dt = \frac{1}{3} Kt^3 \quad (3-14)$$

則總輸出脈衝為

$$(1/3 + 2/3 + 1 + 2/3 + 1 + 1/3) \times at^3 = 4Kt^3 \quad (3-15)$$

由(3-14)、(3-15)可以得知，加速區域內的輸出脈衝為 1/12 的總輸出脈衝。當 S-Curve 在加速中的輸出脈衝比總輸出脈衝的 1/12 大的話，MCX314 將停止加速，並開始降低加速度變化率值至零後再開始減速，加速度曲線形成三角波便為加速時的減速停止之特徵曲線，為了防止這種情形的發生，使用者需留意將最大驅動速度與與加速度設定好。

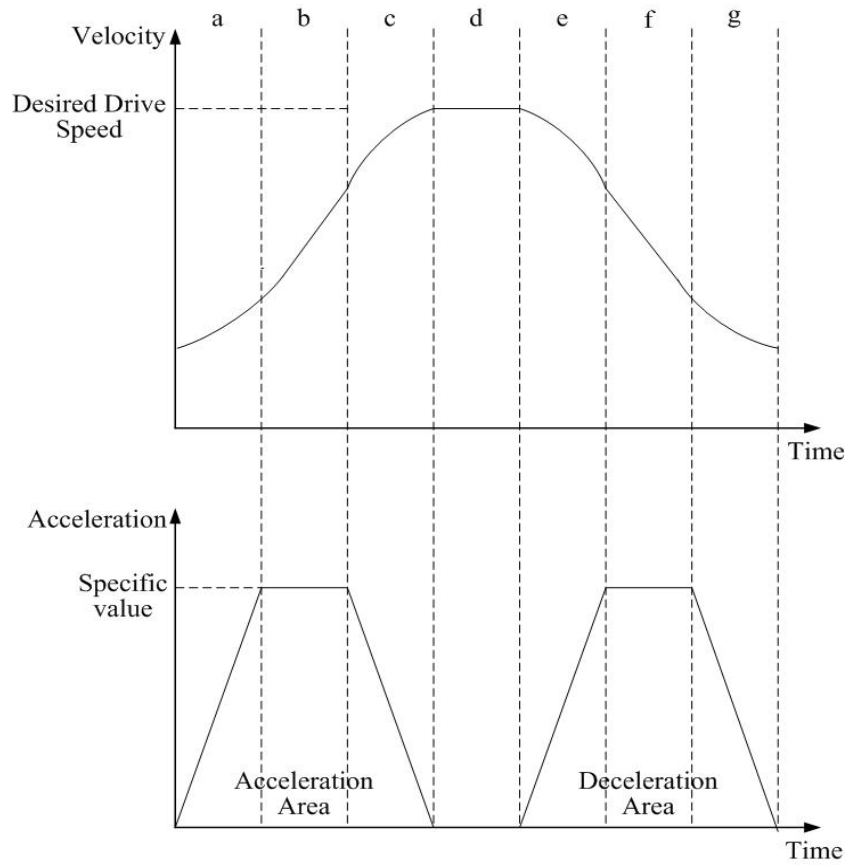


圖 3-16 S-Curve 之加減速曲線變化[1]

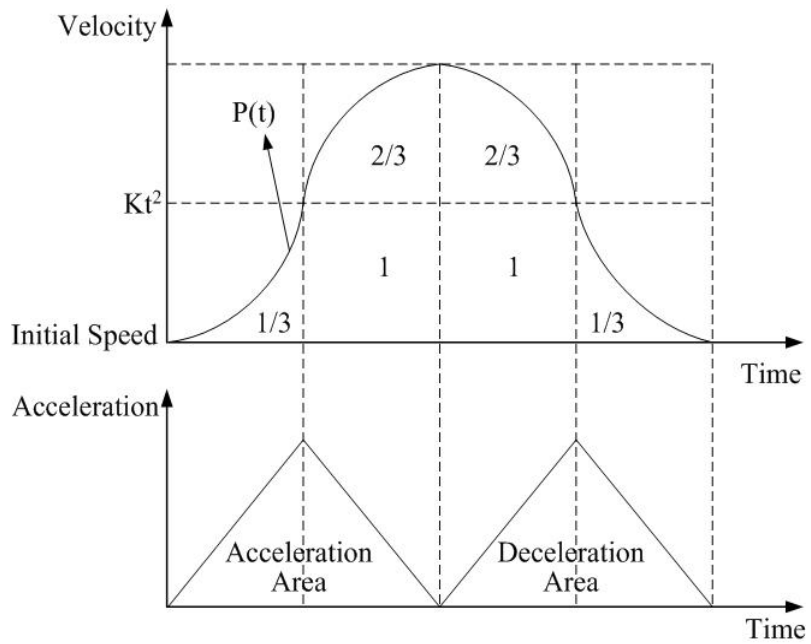


圖 3-17 S-Curve 之減速停止機制[1]

3.2.3 三種加減速規劃之比較

直線加減速規劃在運動起始點及結束點加速度是突然發生，所以 Jerk 極大，仍會造成許多振動。由二次速度曲線組合成的 S-Curve，在運動起始點及結束點時，Jerk 為一有限且連續值，改善了直線加減速 Jerk 極大的缺點，使運動的速度曲線變化平緩，可以有效降低機台的振動。倘若使用較長時間的加速度規劃，則使用完全的 S-Curve，則馬達需要產生較大的力矩以達到較大的加速度，在中段規劃直線則可以有效降低馬達的耗能，使速度的規劃更有效率。

3.3. 多軸補間運動

由第二章的介紹，可以知道 MCX314 是一個四軸運動控制的積體電路，它可以用來實現任意二軸或三軸的直線補間驅動、任意二軸的圓弧補間驅動和任意二軸或三軸的位元補間驅動。MCX314 提供了二軸、三軸的直線補間運動和二軸的圓弧補間運動；使用者僅須設置相關命令暫存器的位置並將命令送入即可實現補間運動命令。然而，位元補間運動，提供了使用者一個發揮想像力，可自行實現補間的機制；使用者可以利用插值方法，自行計算出運動路徑的資料，將其送入暫存器之後，便可實現由使用者自行規劃之任意曲線。以下就插值方法概略說明之。

3.3.1 插值方法

在數控工具機中，將各軸的參考命令訊號傳送至各相對應的控制迴路，以控制刀具或模具，依所下達之運動命令而運動，以達到精確路徑規劃的目標。而產生各軸的參考命令訊號稱之為插值(Interpolation)或命令產生器(Command generation)，常用的插值法為 DDA(Digital Differential Analyzer)的插值法則，DDA 插值方法可提供線性插值(Linear Interpolation)和圓弧插值(Circular Interpolation)，將此兩種方法搭配使用，即可勝任多種複雜的路徑規劃工作。

■ 線性插值法

線性插值的使用，是將兩點間的距離以直線來完成；若只考慮兩軸運動，其理論推導及圖示如下：其中 p_1, p_2 分別代表 x、y 軸的脈衝數， ΔT 為取樣時間。

則兩點間的距離 L 可表示為：

$$L = \sqrt{p_1^2 + p_2^2} \quad (3-16)$$

則取樣週期可表示為：

$$N = \frac{L}{V \times T} \quad (3-17)$$

x、y 的溢位參考值分別為：

$$dx = \frac{p_1}{N}, \quad dy = \frac{p_2}{N} \quad (3-18)$$

dx, dy 為每一個取樣時間中對 x、y 軸所下的命令(單位為 BLU)

依據取樣資料的架構，第 n 個插值運算參考命令可表示為

$$X_n = X_{n-1} + dx, \quad Y_n = Y_{n-1} + dy \quad (3-19)$$

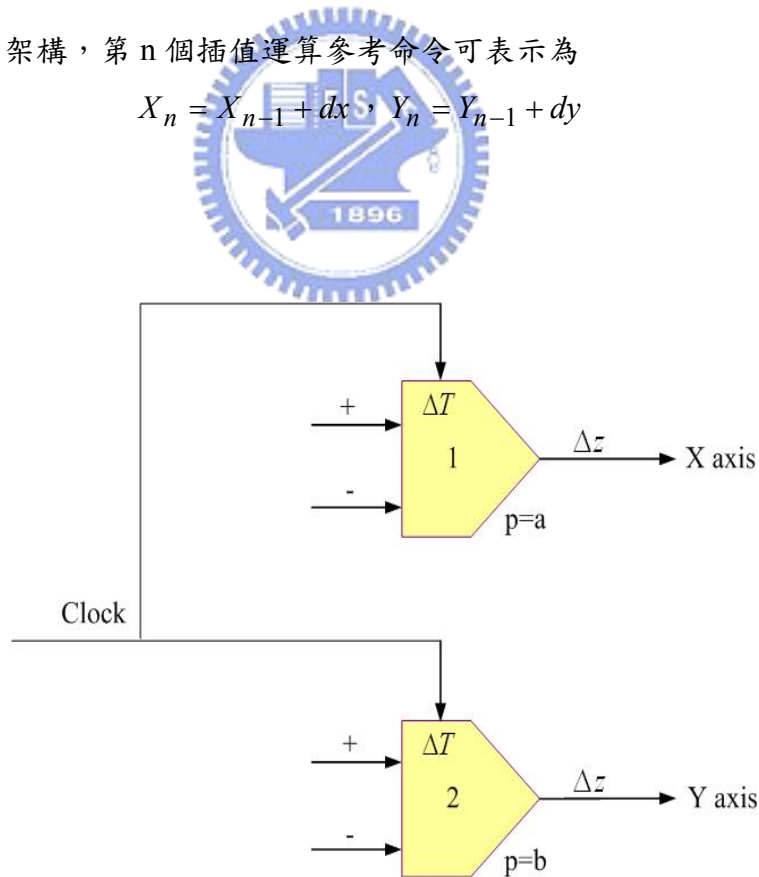


圖 3-18 DDA 直線插值示意圖[3]

■ 圓弧插值法

圓弧插值的使用，是將兩點的距離以圓弧來完成，此處的圓弧仍是以許多的直線來近似，若只考慮兩軸運動，其理論推導及圖示如下：

圓的方程式：

$$(X - R)^2 + Y^2 = R^2 \quad (3-20)$$

則 x、y 座標可表示為：

$$X = R(1 - \cos wt), Y = R \sin wt \quad (3-21)$$

x、y 的切線速度可表示為：

$$V_x = \frac{dX}{dt} = wR \sin wt, V_y = \frac{dY}{dt} = wR \cos wt \quad (3-22)$$

x、y 的溢位參考值分別為：

$$dX = wR \sin(wt)dt = -d(R \cos wt) \equiv p_1 \quad (3-23)$$

$$dY = wR \cos(wt)dt = +d(R \sin wt) \equiv p_2 \quad (3-24)$$

x、y 溢位值分別為

$$\Delta z_1 = wR \cos wt dt, \Delta z_2 = wR \sin wt dt \quad (3-25)$$

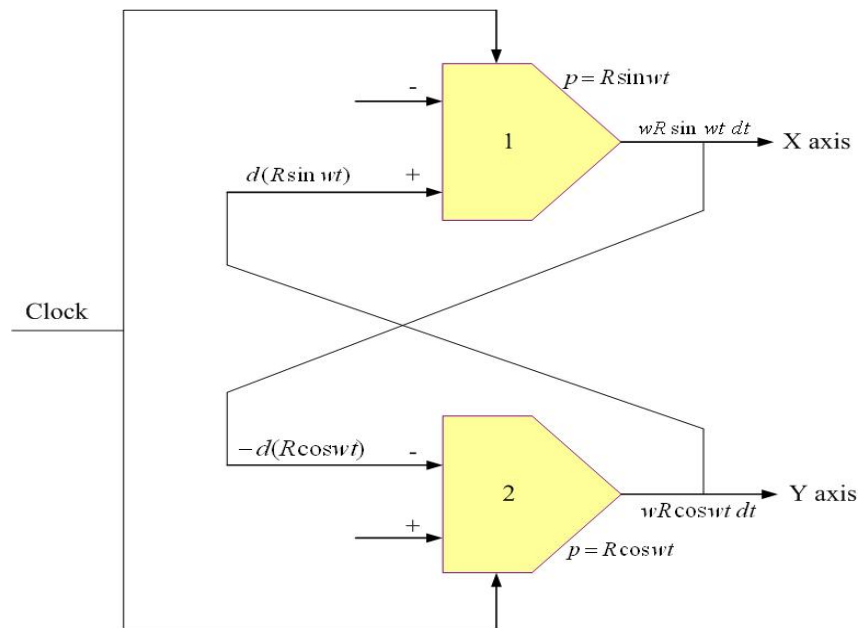


圖 3-19 DDA 圓弧插值示意圖[3]

3.3.2 二軸三軸直線補間運動

在 MCX314 中可在四軸中任意選擇二軸或三軸做直線補間運動，直線補間是將採相對座標，即設定相對於現在座標的終點座標後，而終點座標是相對於現在座標的座標之輸出脈波數的量；將相關命令暫存器設定完成後，再把二軸或三軸直線補間的命令寫入即可。通常補間運動以等速補間為主要驅動方式，關於直線補間的加減速仍然可以利用多軸同步的 T-Curve 與 S-Curve 加減速方式驅動。

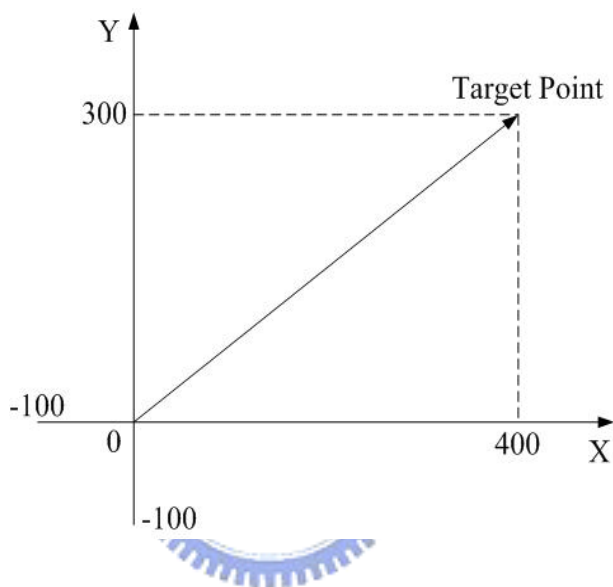


圖 3-20 二軸直線補間

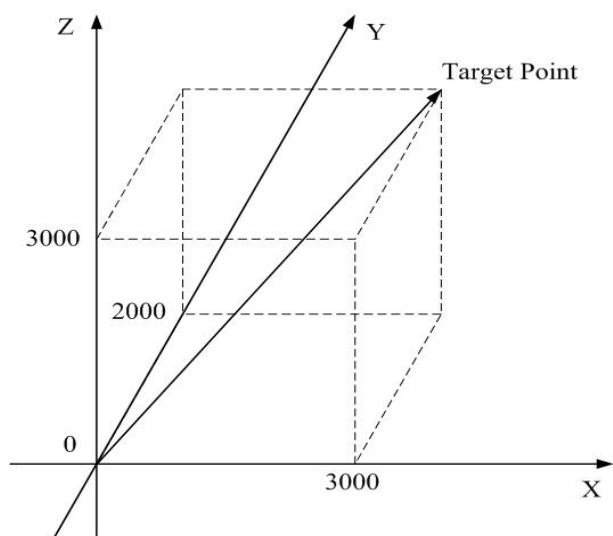


圖 3-21 三軸直線補間

3.3.3 圓弧補間運動

圓弧補間以目前位置為起點，座標為(0,0)；所設置之圓心座標係相對於起點的座標，兩者之間的直線距離即為半徑。終點座標的設置需考慮到實際的運動半徑，若終點座標設置為(0,0)代表回到起點，實際繞行了一圈，運動軌跡為一正圓。若輸入終點座標大於圓弧半徑，則補間無法完成完整的運動軌跡；補間方向可以選擇順時針方向或逆時針方向；可在四軸中任意選擇其中兩軸來進行補間。

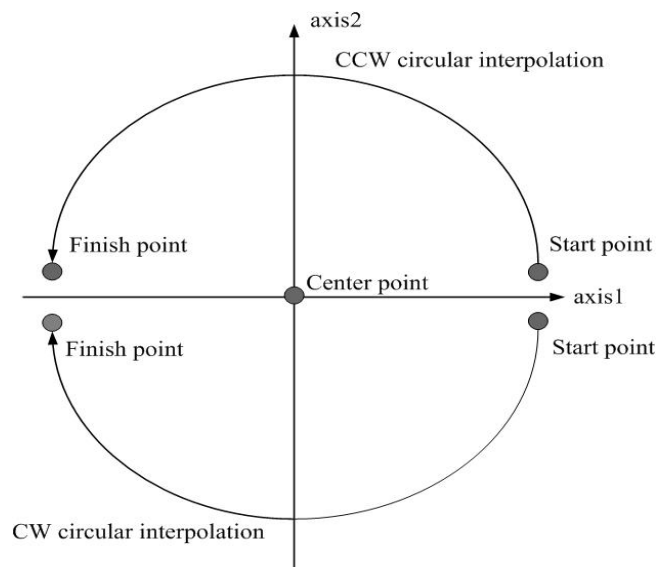


圖 3-22 圓弧補間

MCX314 規劃的圓弧補間只能以人工減速作 T-Curve 加減速驅動，而無法使用 S-Curve 加減速驅動和自動減速功能。圓弧補間的計算，以原點座標為中心可分為 0~7 象限。若運動座標在第 0 象限中，其第二軸的絕對值恆小於第一軸；若絕對值小的軸稱為短軸，則第 1、2、5、6 象限中 axis1 為短軸；同理第 0、3、4、7 象限中 axis2 為短軸。短軸在補間運動時永遠保持脈波輸出，而長軸則依補間計算結果控制脈波輸出與否。

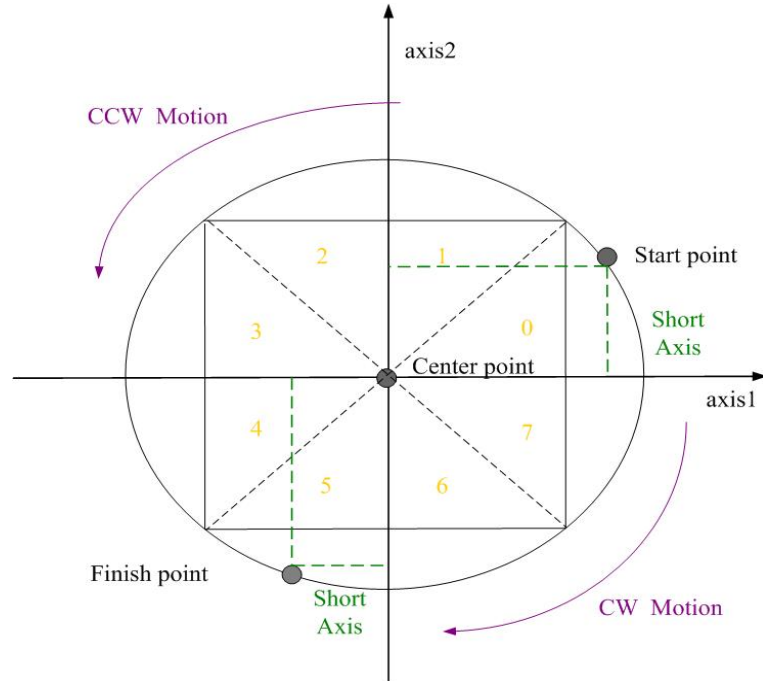


圖 3-23 圓弧補間之人工減速點計算

以下關於圓弧補間人工減速點的計算：

1. 先判斷起點與終點所在之象限。
2. 由圓心座標決定圓弧運動半徑。
3. 計算起點與終點所在象限，所輸出的主脈波數(順時針方向或逆時針方向)。主脈波輸出由短軸長度決定之。
4. 計算由(順時針方向或逆時針方向)所經過的完整象限數。並計算其輸出的主脈波數。
5. 步驟3 + 步驟4的結果即為起點到終點所輸出的脈波數。
6. 由使用者規劃之最大驅動速度，可進而求得倍率。假設倍率為M：則實際的初速度 $RSV = SV \times M$ ；實際驅動速度 $RV = V \times M$ ；實際加速度 $RA = A \times 125 \times M$ 。

由 $RSV = RV + RA \times TA$ ，可從使用者輸入的初速度(SV)、速度(V)、加速度(A)可以計算出加速段時間TA和斜線加速區的面積。

7. 假設減速段與加速段驅動脈波數相等，則可以簡單推算出：

人工減速點 = 起點到終點所輸出的脈波數 - 加速段的驅動脈波數

3.3.4 位元補間運動

MCX314所提供的位元補間模式(Bit Pattern Interpolation)，其實是一種用來實現補間的機制。在此模式之下，MCX314可以設定二軸或三軸之的每個軸所提供六個暫存器，構成可容納三組補間數據的陣列。每組數據由兩個16位元數據所組成，分別送入該軸的正負方向暫存器中。在同一段時間週期內產生一個正向脈衝，則正向暫存器內數據為1，負向暫存器內數據為0；同理，產生一個負向脈衝，則正向暫存器內數據為0，負向暫存器內數據為1；若在同一段時間週期內不產生脈衝，則兩方向的暫存器皆為0。位元補間資料數據的產生方式，沒有任何形式的限制，可以利用此靈活性而加以自由設計。由CNC的觀點，以DDA的插值法則便可輕易產生位元補間的數據資料。將DDA所計算出的所有0和1的資料按順序排列，每16筆轉換成16進位後送至暫存器。當每一個16進位的命令送入暫存器後，堆疊計數器(Stacking Counter, SC)的值便增加1，堆疊計數器是一個2-bit的計數器，其值的範圍從0到3；當堆疊計數器的值為3時，數據無法再堆疊，便開始送出堆疊的數據，每送出一組數據堆疊計數器便減少1，當堆疊計數器的值從2減為1時，MCX314會有產生中斷的機制，使數據繼續堆疊，如此反覆進行，直至數據傳送完。

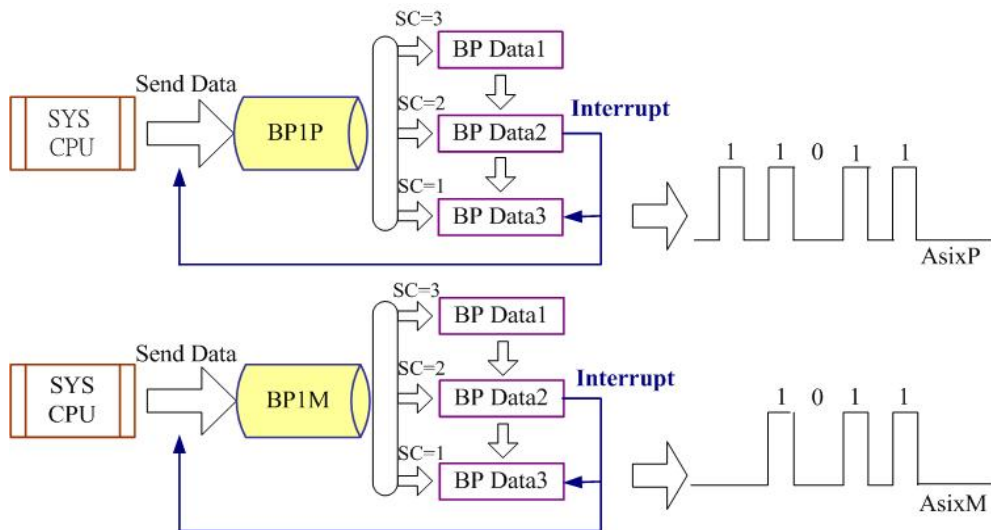


圖 3-24 位元補間資料輸出[1]

以雙軸位元補間運動為例，設定位元補間的資料是以直線插值 DDA 的方式產生，(位元補間資料產生方式可以自由設計，不限於直線插值法)，便能以多段的直線來近似任意曲線。將 DDA 所產生的 2 位元的位元補間資料依序組合，轉換成 16 位元送至各軸之正負方向的暫存器中，數據資料堆疊後，即可呼叫補間驅動函數，開始進行位元補間運動，程式流程如圖 4-5 所示。

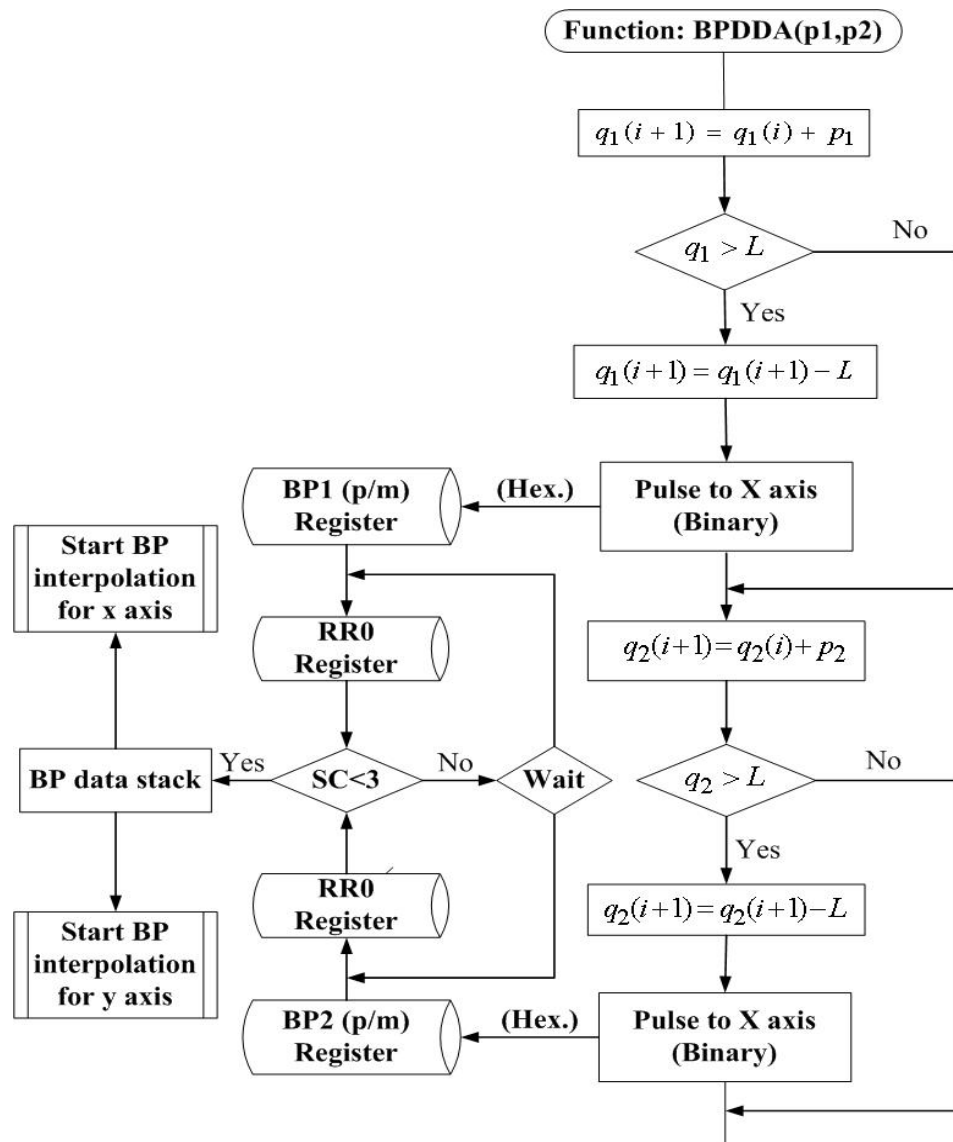


圖 3-25 位元補間程式流程圖

倘若設定雙軸位元補間在第一象限運動，而分別給定 X 軸 30 個 pulse，Y 軸 40 個 pulse，接著利用 DDA 的直線插值法可以求得運動路徑的資料。運動路徑 $L = \sqrt{P_1^2 + P_2^2}$ ；其中 P_1, P_2 分別為 X、Y 軸的脈衝數。如圖 4-7 所示。位元補間資料是將 16 個二進位數字(0 或 1)，每 16 依序組合成一組 16 位元的命令，其中 Z1、Z2 代表 X、Y 軸的脈衝分布；X、Y 分別表示 Z1、Z2 的累加值；而 Hex1、Hex2 正是代表每 16 個單位時序的脈波分佈所轉換成之 16 位元的命令。如表 3-2 所示。

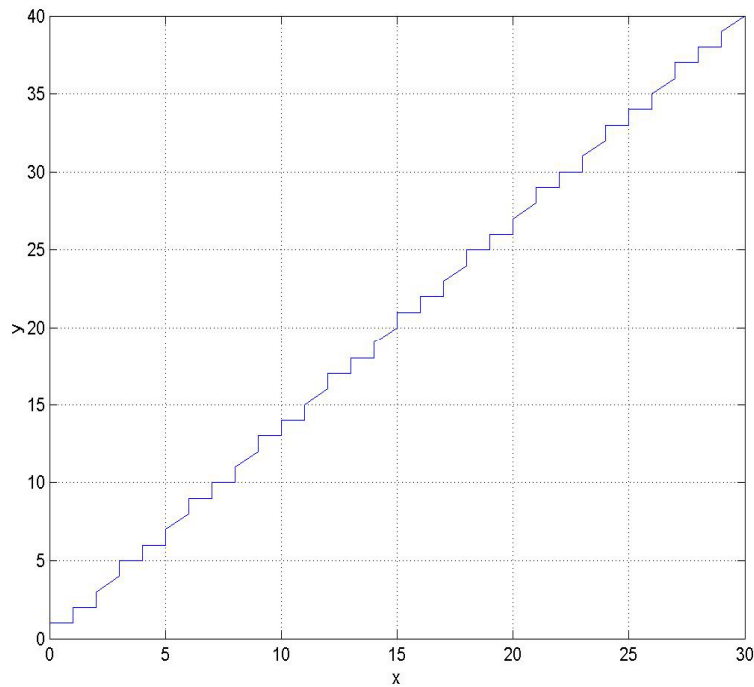


圖 3-26 利用直線插值法產生位元補間資料

表 3-2 位元補間資料的產生與轉換

Index	Z1	Hex1	Z2	Hex2	X	Y	Index	Z1	Hex1	Z2	Hex2	X	Y	
1	0	0x 6b5a	0	0x 7bde	0	0	26	0	0x 5ad6	0	0x def7	15	20	
2	1		1		1	1	1	27		1		1	16	21
3	0		1		1	2	2	28		0		1	16	22
4	1		1		2	3	3	29		1		1	17	23
5	1		1		3	4	4	30		1		1	18	24
6	0		0		3	4	3	31		0		0	18	25
7	1		1		4	5	4	32		1		1	19	25
8	0		1		4	6	4	33		0		1	19	26
9	1		1		5	7	5	34		1		1	20	27
10	1		1		6	8	6	35		1		1	21	28
11	0		0		6	8	6	36		0		0	21	29
12	1		1		7	9	7	37		1		1	22	29
13	0		1		7	10	7	38		0		1	22	30
14	1		1		8	11	8	39		1		1	23	31
15	1		1		9	12	9	40		1		1	24	32
16	0	0	9	12	9	41	0	0	24	32				
17	1	0x b5ad	1	0x bdf7	10	13	42	1	0x 3	1	0x 3	25	33	
18	0		1		10	14	10	43		0		1	25	34
19	1		1		11	15	11	44		1		1	26	35
20	1		1		12	16	12	45		1		1	27	36
21	0		0		12	16	12	46		0		0	27	36
22	1		1		13	17	13	47		1		1	28	37
23	0		1		13	18	13	48		0		1	28	38
24	1		1		14	19	14	49		1		1	29	39
25	1		1		15	20	15	50		1		1	30	40

3.3.5 連續補間

連續補間，顧名思義是以不停止驅動的方式連續進行補間的運動。可將直線補間和圓弧補間互相搭配成一系列的補間過程，進而規劃成特定的運動軌跡。連續補間實現的方式有兩種方法：第一種為Polling的方式，利用MCX314中RR0暫存器的D9位元中(1、0)數值的改變，以控制下一個補間節點資料和補間驅動指令的寫入與否。第二種為Interrupt的方式，利用MCX314中WR5暫存器的D14位元，來設置中斷功能是否為有效，之後依MCX314中RR0暫存器的D9位元中(1、0)數值的改變，MCX314的INTN輸出訊號將會變成Low位準，進而向CPU發出一個中斷處理程序，在每下一筆的補間節點資料和補間驅動指令皆在此中斷的副程式中實作。

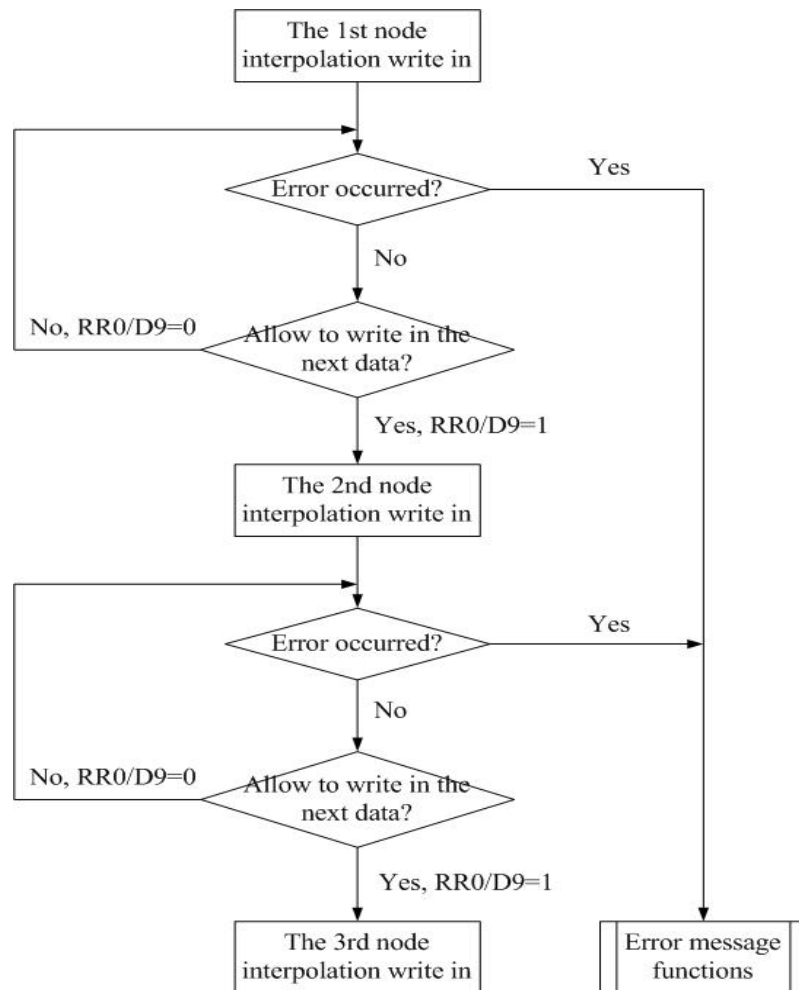


圖 3-27 連續補間程式流程圖

3.4. 原點返回運動

返回原點運動(Home Searching)係為系統啟動時或加工機械產生異常錯誤需要重新啟動時，必須返回原先定義之操作原點。為描述原點返回運動，有以下常見的信號定義：ORG，為 Origin 的簡寫，主要用途在於指示該軸是否在原點；EL，為 End Limit 的簡寫，此信號用於保護機械裝置，避免機械衝出所設計之範圍。EZ，為 Encoder 的 Z 相訊號，大多用於尋找原點的參考訊號。用以偵測原點訊號或左右極限的感測器，則選擇光遮斷器，架設在馬達平台的側邊，並與控制卡所外接的端子台相接，當馬達平台移動時，觸發了原點或極限訊號，訊號傳回 MCX314 之後再透過 PC 的程式控制。當例外處理發生時(圖 3-28~圖 3-31 中 Action2、Action3 的圖示，當原點返回之運動方向與實際機台運動方向相反時，則無法回到原點，而會運動至左/右極限而停止)，這時便需檢查運動方向與實際機台運動方向的相對關係。當機台在左右極限上時，需先除能極限訊號在使機台運動，以移離左右極限。表 3-3 為規劃 4 種常用的原點返回模式：

表 3-3 原點返回模式說明

設定模式	狀態	說明	相關信號
1	Action 1	當移動至原點 ORG 的訊號位置時，即減速停止。	ORG
	Action 2 / 3	例外處理。	
2	Action1	當移動至原點 ORG 的訊號位置時，即減速停止，然以等速後退，再以慢速前進到達原點停止。	ORG
	Action 2 / 3	例外處理。	
3	Action 1	當移動至原點 ORG 的訊號位置時，即減速至初始速度然後偵測至 EZ 信號後停止。	ORG、EZ
	Action 2 / 3	例外處理。	
4	Action 1	當移動至原點 ORG 的訊號位置時，即減速停止，待偵測至 EZ 信號後停止，以等速後退，再以慢速前進到達原點停止。	ORG、EZ
	Action 2 / 3	例外處理。	

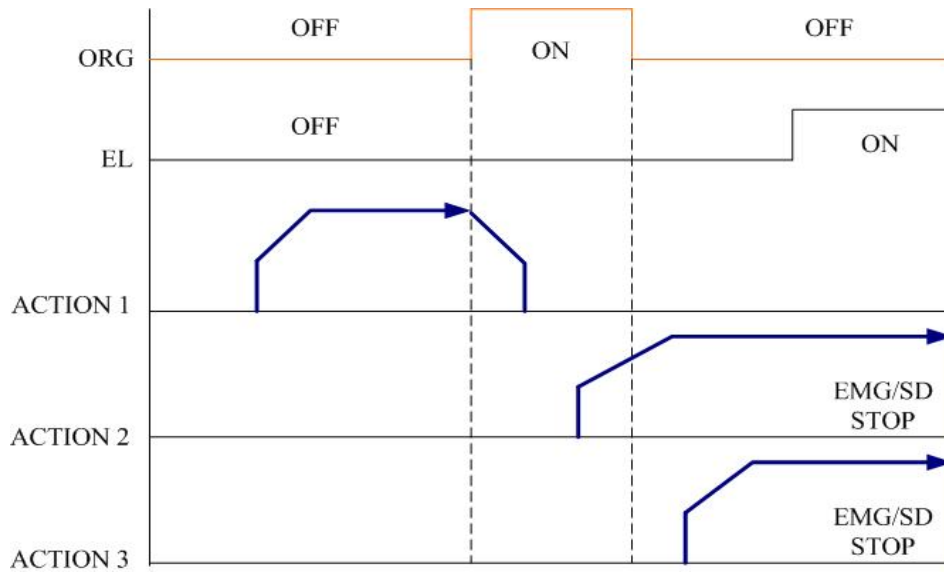


圖 3-28 原點返回模式 1

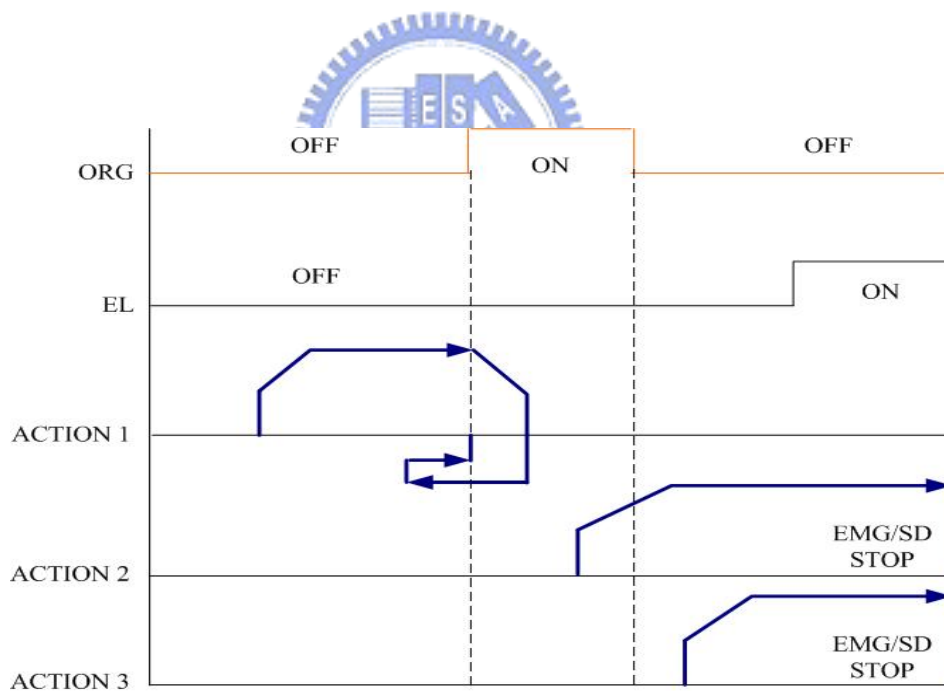


圖 3-29 原點返回模式 2

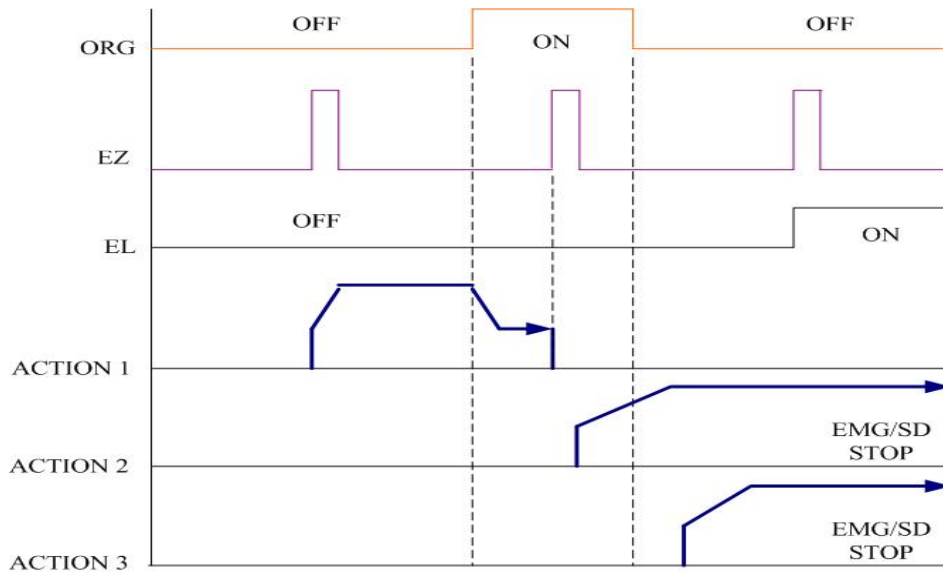


圖 3-30 原點返回模式 3

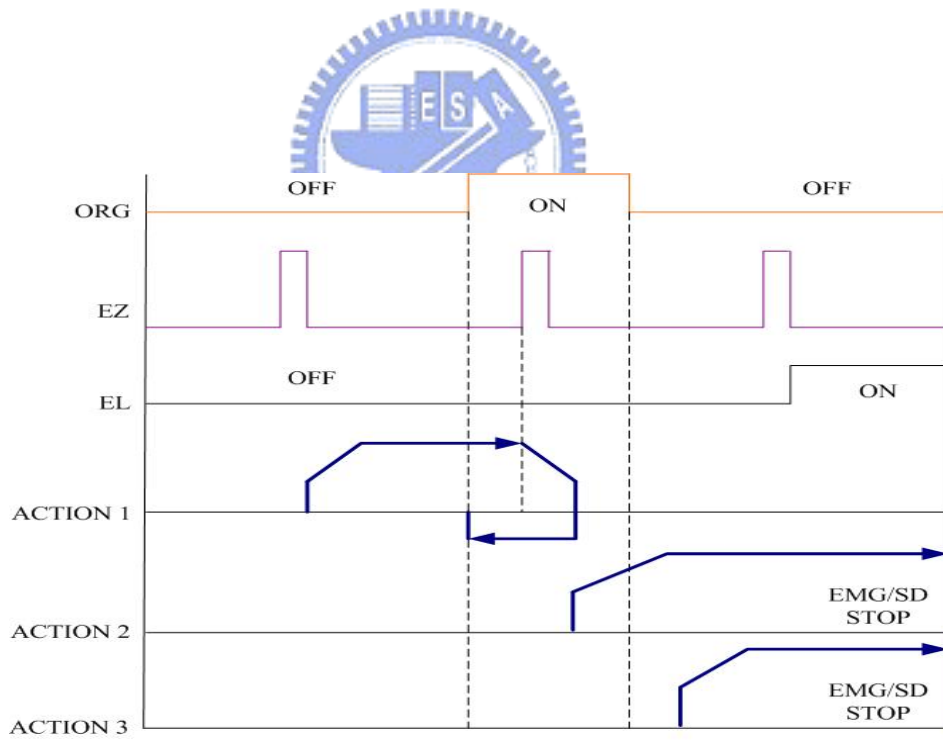


圖 3-31 原點返回模式 4