

# MACRO指令在綜合切削中心機 應用之研究

王 中 行

## 摘 要

本文主要目的在探討綜合切削中心機中之MACRO指令，在程式設計中之應用與其影響。MACRO又可稱為CUSTOM MACRO，乃是一種客戶導向的使用者編寫器，藉者此種變數指令的功能，工具機廠商或操作者得以自行編寫所需之功能；更可進一步執行其它邏輯、函數及系統變數運算。目前國內工業界、學校，均以大量使用CNC綜合切削中心機，唯大部分均不了解MACRO指令的應用，而無法充分利用切削中心機，達到三度空間之循環切削以及複雜工件簡化程式之設計。是故本文即以MACRO在切削中心機之應用，與軟體的發展為方向，並說明其與一般副程式之差異……等方面探討；而使每一位切削中心機的使用者，在程式製作上與切削加工能力上有「如虎添翼」之感受。

## 一、緒論

綜合切削中心機 ( Machining Center ) 或稱為切削中心機、綜合加工機，其觀念乃得自於 CNC 工具機與聯製加工機 ( Transfer Machine )。切削中心機可定義為擁有特殊的自動刀具更換器 ( Automatic Tool Changer )，而能於一次設定中完成面銑、鑽孔、鏜孔、及攻螺絲等多項操作之數控工具機。切削中心機於1958年間世，其距NC工具機的首次應用僅不過六年的光景，然而却使得NC工具機在機械加工的靈活應用上又向前邁進一大步，尤其是近年來電腦技術的不斷進步及硬體價格日趨下降，由切削中心機為主所構成的彈性製造系統 ( FMS ) 或彈性加工單元 ( FMC )，更逐步廣泛用於機械現場。另一方面，在CNC控制器的發展上探討，其基本功能可謂已接近完全。如G、M、F、S等編碼已標準化，但是各種控制器仍有很多不同的特徵，細加歸納，約有下列四項：

(1)在控制器上增加繪圖功能，以顯示切削路徑或模擬切削路徑及模擬工件形狀，避免錯誤發生。此項繪圖能力，且有發展為三維繪圖的趨勢。

(2)在程式製作方面，已具備客戶導向的巨集指令 ( CUSTOM MACRO )，而使得三度空間之循環切削程式製作簡化，而充分發揮切削中心機之功能。

(3)在面板顯示及工件程式輸入方面，採用教導或菜單列舉 ( Menu ) 方式，更有採用軟鍵 ( Softkey ) 方式輸入，使操作者於操作控制器時，更為便利及簡單。

(4)增加控制器的通信交通能力，使控制器能與工廠的主電腦 ( Host Computer ) 連接，將個別工具機統一作生產管理，向上發展成為DNC或FMS，而可在未來的無人化操作中使用。

本文目的主要即針對MACRO指令在切削中心機之應用，及如何利用此項功能以發展應用軟體，擴大切削中心機之功能，以達到三度空間自動循環切削立體模具，進而提升工業界之生產品質。

## 二、MACRO 指令的運用

在CNC的程式製作上，也有使用副程式的方式，此雖可省下程式製作的時間與節省控制系統記憶體空間，但是它的功能也受到一些限制；像形狀相似而幾何度量並不不同的工件，副程式之使用便無用武之地了。於是MACRO指令的運用便彌補了這一缺憾。

MACRO指令之功能是在程式製作時，不必書入真正的數值，而以變數代入，在需要使用之時，再以類似副程式呼叫方式叫出MACRO指令，並對每一個使用到的變數付與數值，此時程式便可依需要工作了。如圖 1 所示<sup>(6)</sup>。

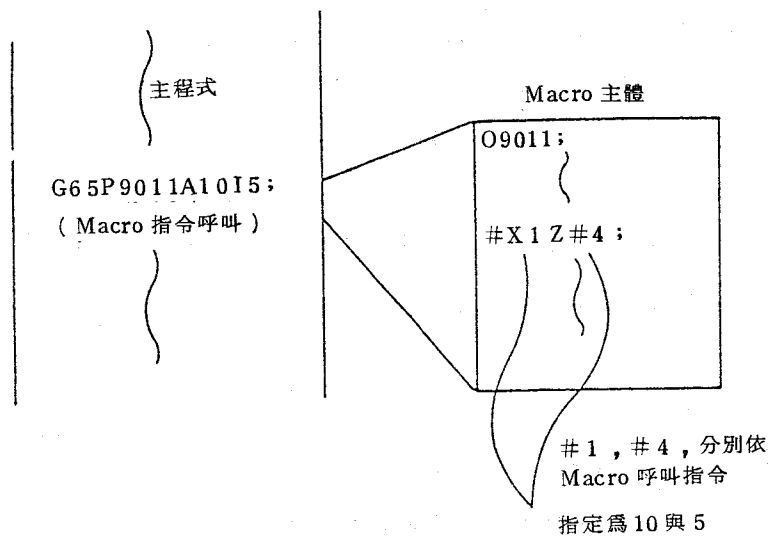


圖 1 MACRO 指令的呼叫

### 2-1 變數之表示

所有的變數之前必需加“#”以資區別，例如：# 5 代表第 5 號變數；# 109 代表第 109 號變數；#1005 代表第 1005 號變數。變數之表示亦可為下列形式：

# [ 式子 ]

例如：# [ # 100 ] 代表變數第 100 號之數值，再做為變數之號碼；# [ # 90 - 1 ] 即為第 90 號變數值減去 1 做為變數號碼。又在位址碼後之數值，亦可以用變數加以取代，即：

位址碼 # i ( 或位址碼 - # i )

例如：F # 33，若 # 33 = 1.5，此時意義即為 F 1.5；Z - # 18，若 # 18 = 20.0，則意義為 Z - 20.0。

雖然 MACRO 指令中的變數，可以用來代替位址碼中的數值，但是某些位址碼，如：程式號碼 O，順序號碼 N，選擇性單節省略符號 /，均不可以跟變數。

若變數之值未予以定義，我們稱其為「空的」，一般 # 0 多為空變數，空變數與變數值為 0，應有所差異。

## 2-2 變數的運算與流程控制

### (1) 變數的定義與替代

# i = # j ; 第 j 號變數之數值代入第 i 號變數。

### (2) 加減運算

# i = # j + # k ; 和

# i = # j - # k ; 差

# i = # j OR # k ; 邏輯和 ( 32 位元的每一位元 )

# i = # j XOR # k ; 互斥或運算 ( 32 位元的每一位元 )

### (3) 乘除運算

# i = # j \* # k ; 乘積

# i = # j / # k ; 除

# i = # j AND # k ; 邏輯積 ( 32 位元的每一位元 )

### (4) 函數運算

# i = SIN [ # j ] ; SINE ( 度為單位 )

# i = COS [ # j ] ; COSINE ( 度為單位 )

# i = TAN [ # j ] ; TANGENT ( 度為單位 )

# i = ATAN [ # j ] / [ # k ] ; ARCTANGENT ( 度為單位 )

# i = SQRT [ # j ] ; 平方根

# i = ABS [ # j ] ; 絕對值

# i = BIN [ # j ] ; 將 BCD 碼轉換為二進位數

# i = BCD [ # j ] ; 將二進位轉換為 BCD 碼

上面基本運算，可再組合成其它運算式，其優先順序為函數優先，乘法其次，加法最後。

### (5) 邏輯判斷功能

包含 EQ ( 相等 )，NE ( 不相等 )，GT ( 大於 )，LT ( 小於 )，GE ( 大於或等於 )，LE ( 小於或等於 )。此些均為數學不等式中常用之邏輯判斷功能。

### (6) 流程控制指令

此類指令，提供近似高階語言的指令，可分為下列二類：

(a) IF [ 判斷式子 ] GOTO n

當判斷式子滿足時，即跳到程式中順序編號 n 的單節中執行，順序號碼亦可用變數或式子取代。

(b) WHILE [ 判斷式子 ] DO m ( m = 1 , 2 , 3 )  
 END m

當判斷式子滿足時，在 DO m 到 END m 範圍內所有單節將被執行，否則即略去。

### 2-3 變數的種類

基於不同的需要，變數通常依其性質而有區分，在MACRO指令的使用上，變數可區分為：

- (1) 區域變數 ( Local Variable )
- (2) 共通變數 ( Common Variable )
- (3) 系統變數 ( System Variable )

區域變數，是指在不同的Macro指令中，雖有相同變數號碼，但却有不同的數值，譬如在 FANUC 的控制系統中即視 # 1 至 # 33 為區域變數，是對同一主程式中之所有副程式、MACRO指令均有相同數值的變數，即變數數值可傳遞於不同的MACRO指令間，譬如，共通變數在 FANUC 系統中設定，# 100 ~ # 149 為系統電源關閉時，資料會自動消失之共通變數，# 500 ~ # 509 為電源切除後，仍能保持原先數值之共通變數。系統變數，乃系統內固定變數，系統變數會牽涉到個別系統之差異，譬如，# 1000 ~ # 1015 代表輸入訊號介面，# 1100 ~ # 1115 代表輸出訊號介面。其它有關常使用手寫程式之位址碼之系統變數，刀具補正量變數，工件位置補正量變數等分別參閱表 1、2、3、4。

表 1 刀具補正量變數

刀 具 補 正 號 碼	刀 具 補 正 值
0	# 2000
1	# 2001
2	# 2002
3	# 2003
⋮	⋮
⋮	⋮
⋮	⋮
199	# 2199
200	# 2200

※ # 2000 數值為 0

表 2 工件位置補正量變數

軸 向	工件位置補正號碼	工件位置補正量
X 軸	工件補正	# 2500
	G 59	# 2501
	}	}
	G 59	# 2506
Y 軸	工件補正	# 2600
	G 54	# 2601
	}	}
	G 59	# 2606
Z 軸	工件補正	# 2700
	G 54	# 2701
	}	}
	G 59	# 2706
第 4 軸	工件補正	# 2800
	G 54	# 2801
	}	}
	G 59	# 2806
第 5 軸	工件補正	# 2900
	G 54	# 2901
	}	}
	G 59	# 2906

表 3 位址碼之系統變數

系 統 變 數	功 能 ( 位 址 碼 )
# 4001	G 指令 ( 01 組別 )
:	:
# 4021	G 指令 ( 21 組別 )
# 4102	B 指令
# 4107	D 指令
# 4109	F 指令
# 4111	H 指令

# 4113	M指令
# 4114	N序號
# 4115	O程式代號
# 4119	S指令
# 4120	T指令

表 4 位置指令之系統變數

系 統 變 數	位 置 指 令
# 5001	X 軸單節結束座標
# 5002	Y 軸單節結束座標
# 5003	Z 軸單節結束座標
# 5004	第 4 軸單節結束座標
# 5005	第 5 軸單節結束座標
# 5041	X 軸目前座標
# 5042	Y 軸目前座標
# 5043	Z 軸目前座標
# 5044	第 4 軸目前座標
# 5045	第 5 軸目前座標
# 5061	X 軸跳躍訊號位置
# 5062	Y 軸跳躍訊號位置
# 5063	Z 軸跳躍訊號位置
# 5064	第 4 軸跳躍訊號位置
# 5065	第 5 軸跳躍訊號位置

#### 2—4 變數的引數 ( Argument ) 設定

##### (1) 第一類引數設定 ( 見表 5 )

此類引數之設定可以利用所有之英文字母，但 G、L、N、O、P 等位址除外。在引數指定時，並不一定要按英文字母順序，唯 I、J、K 必須按字母順序。下列格式均為正

確表示：

A \_\_\_\_\_ B \_\_\_\_\_ C \_\_\_\_\_ D \_\_\_\_\_ ..... Z \_\_\_\_\_  
 B \_\_\_\_\_ A \_\_\_\_\_ D \_\_\_\_\_ ..... I \_\_\_\_\_ K \_\_\_\_\_ ..... Z \_\_\_\_\_

表 5 第一類引數

第一類引數位址	Macro指令變數
A	# 1
B	# 2
C	# 3
D	# 7
E	# 8
F	# 9
H	# 11
I	# 4
J	# 5
K	# 6
M	# 13
Q	# 17
R	# 18
S	# 19
T	# 20
U	# 21
V	# 22
W	# 23
X	# 24
Y	# 25
Z	# 26

(2)第二類引數設定(見表6)

在第一類引數設定時，區域變數# 1至#33並不能完全利用，所以有時在應用上會感



表 6 第二類引數

第二類引數位址	Macro指令變數
A	# 1
B	# 2
C	# 3
I <sub>1</sub>	# 4
J <sub>1</sub>	# 5
K <sub>1</sub>	# 6
I <sub>2</sub>	# 7
J <sub>2</sub>	# 8
K <sub>2</sub>	# 9
I <sub>3</sub>	# 10
J <sub>3</sub>	# 11
K <sub>3</sub>	# 12
I <sub>4</sub>	# 13
J <sub>4</sub>	# 14
K <sub>4</sub>	# 15
I <sub>5</sub>	# 16
J <sub>5</sub>	# 17
K <sub>5</sub>	# 18
I <sub>6</sub>	# 19
J <sub>6</sub>	# 20
K <sub>6</sub>	# 21
I <sub>7</sub>	# 22
J <sub>7</sub>	# 23
K <sub>7</sub>	# 24
I <sub>8</sub>	# 25
J <sub>8</sub>	# 26
K <sub>8</sub>	# 27
I <sub>9</sub>	# 28
J <sub>9</sub>	# 29
K <sub>9</sub>	# 30
I <sub>10</sub>	# 31
J <sub>10</sub>	# 32
K <sub>10</sub>	# 33

不足，第二類引數設定則可將所有變數完全加以利用，其格式為：

A \_\_\_\_\_ B \_\_\_\_\_ C \_\_\_\_\_ I \_\_\_\_\_ J \_\_\_\_\_ K \_\_\_\_\_ I \_\_\_\_\_ J \_\_\_\_\_ K \_\_\_\_\_ ...

如同第一類引數，I、J、K應依順序指定，又在表6中雖有 $I_1, I_2, I_3, I_4 \dots$ 等區別，但在引數指定時，並不能將此下標代號書上，而以出現的順序指定之。

第一、二類引數可同時使用於G65指令之單節內，唯若其代表相同之變數時，以「後出現」者有效。

舉例而言：G 65 A 1.0 B 2.0 I - 3.0 I 4.0 D 5.0 P 1000 ；

此時變數# 1 = 1.0，# 2 = 2.0，# 4 = - 3.0，# 7 原設定為 4.0，但隨即為 5.0 所取代。

## 2-4 Macro 指令之呼叫

Macro 指令之呼叫，有如副程式之呼叫，但這只是止於型式上，在實際上Macro指令之呼叫，可分為四種：

- (1) 單純呼叫 ( Simple Call )
- (2) 模式呼叫 ( Modal Call )
- (3) 多重單純呼叫 ( Multiplex Call )
- (4) 多重模式呼叫 ( Multiplex Modal Call )

單純呼叫，為最簡單之一種Macro呼叫方式，其格式為：

G 65 P \_\_\_\_\_ L \_\_\_\_\_ <變數之引數設定> ；

P 為巨指令之程式代號，L 為重複執行次數，引數設定用以指定變數。

模式呼叫，為在 G 66 ~ G 67 間的任何運動指令在執行完後便呼叫 Macro 指定一次，直到 G67 取消為止。其格式為：

G 66 P \_\_\_\_\_ L \_\_\_\_\_ <變數之引數設定> ；

.....

.....

G 67 ；

多重單純呼叫，此乃Macro指令中再呼叫Macro指令之方式，唯限制至多僅可為四個層次。

多重模式呼叫，係指第一Macro指令在第二Macro指令的運動命令完後，再做一次。

舉例言之：

N 011 G 66 P 1000 X 5000 ; 程式執行之順序號碼為：  
 N 012 Z 10000                      N 011 → N 012 → N 1001 →  
 N 013 G 66 P 2000 Z 6000 ; N 1002 → N 013 → N 014 →  
 N 014 Z 15000 ;                      N 2001 → N 1001 → N 1002  
 N 015 G 67 ;                          → N 2002 → N 1001 → N 1002  
 N 016 G 67 ;                          → N 2003 → N 015 (取消  
 N 017 Z - 25000 ;                      P 2000 的多重模式呼叫) →  
 .....                                  N 016 (取消 P 1000 的多重模  
 O 1000 ;                                  式呼叫) → N 017 。  
 N 1001 G 01 X # 24 ;  
 N 1002 M 99 ;  
 O 2000 ;  
 N 2001 G 01 Z # 26 ;  
 N 2002 Z # 26 + 1000 ;  
 N 2003 M 99 ;

## 2—5 Macro 呼叫其一般副程式呼叫之比較

- (1) Macro 指令呼叫，G 65，可設定變數的引數，一般副程式呼叫，M98，不能設定變數的引數。
- (2) G65的指令單元不做任何實際運動，而直接跳到所指定之Macro指令去執行。M98之指令單元要先執行完其它動作，才理會指令中的M，P，L等碼。
- (3) G65可以更改區域變數的數值，M98不能。例如：#1在Macro指令呼叫前是一回事，在呼叫後又是另一回事，各不相干，但在M98的副程式呼叫，#1仍為#1。

## 三、應用程式

配合Macro指令，可發展應用各類複雜圖形之切削方式，在此僅就一些應用程式加以說明：

- (1)沿一直線等分點之鑽孔

O 9200 ;

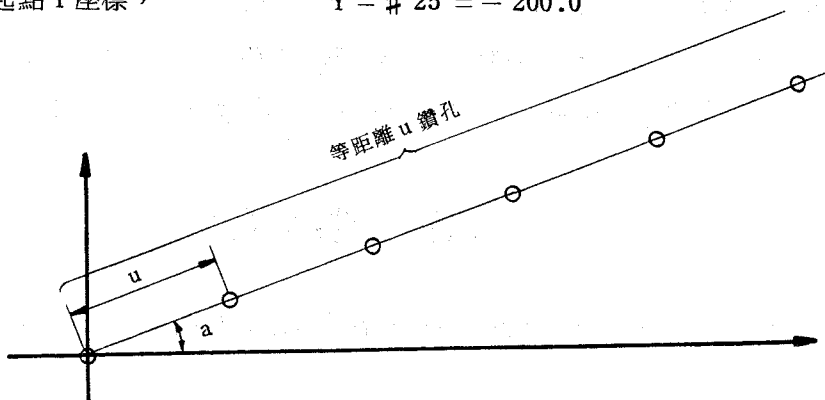
```

G 81 G 99 X 0. Y 0. Z - 150. R - 200. F 1000 L 0 ;
G 65 P 9401 X 40. Y - 200. A 30 U 10. H 5 ;
M 02 ;
O 9401 ;
# 3 = # 4003 ;
# 12 = 3000 ;
# 13 = 03 ;
S # 12 M # 13 ;
X # 24 Y # 25 ;
WHILE [ # 11 GT 1 ] DO 1 ;
G 91 X [ # 21 * COS [ # 1 ] ] Y [ # 21 * SIN [ # 1 ] ] ;
# 11 = # 11 - 1 ;
END 1 ;
G # 3 ;
M 99 ;

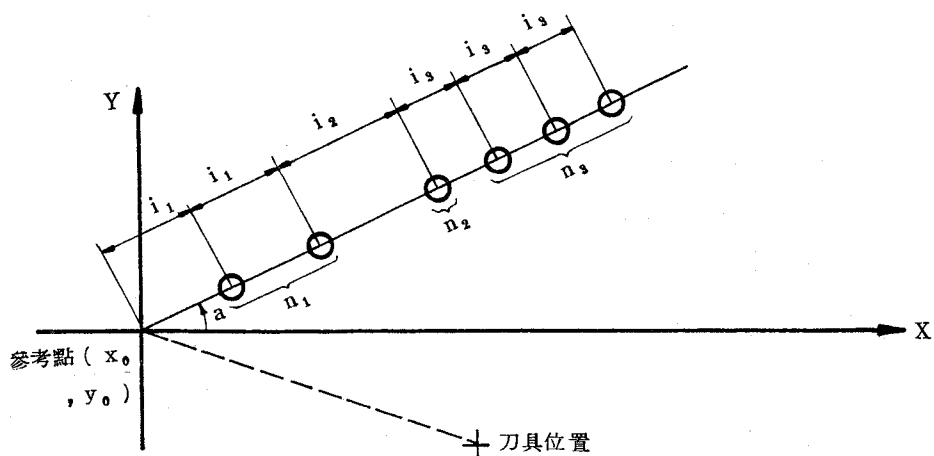
```

說明：

- |    |                 |                    |
|----|-----------------|--------------------|
| 變數 | A = 直線與 X 軸之角度， | A = # 1 = 30       |
|    | U = 鑽孔之間隔距離，    | U = # 21 = 10.     |
|    | H = 孔之數目，       | H = # 11 = 5       |
|    | X = 起點 X 座標，    | X = # 24 = 40.0    |
|    | Y = 起點 Y 座標，    | Y = # 25 = - 200.0 |



(2)沿一直線不等間隔之鑽孔



G 65 P 9203 AaI<sub>1</sub> Kn<sub>1</sub> I<sub>2</sub>Kn<sub>2</sub> ……;

O 9203 ;

# 2 = # 101 ;

# 3 = # 102 ;

# 5 = 4 ;

# 8 = 0 ;

WHILE [ # 5 LE 31 ] DO 1 ;

IF [ # [ # 5 ] EQ 0 ] GOTO 9001 ;

DO 2 ;

# 8 = # 8 + # [ # 5 ] ;

# 101 = # 2 + # 8 \* COS [ # 1 ] ;

# 102 = # 3 + # 8 \* SIN [ # 1 ] ;

X # 101 Y # 102 ;

# 100 = # 100 + 1 ;

# [ # 5 + 2 ] = # [ # 5 + 2 ] - 1 ;

IF [ # [ # 5 + 2 ] LE 0 ] GOTO 9002 ;

END 2 ;

N 9002 # 5 = # 5 + 3 ;

END 1 ;

N 9001 # 101 = # 2 ;

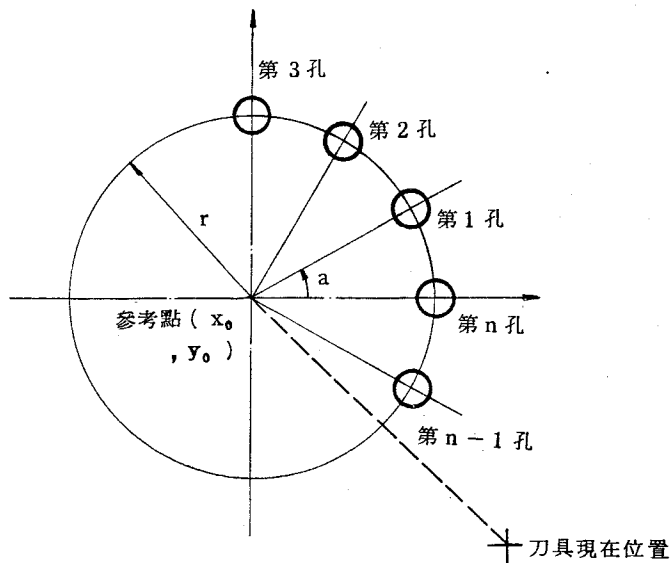
# 102 = # 3 ;

M 99 ;

說明：

- # 1 為斜角  $a$ 。
- # 2 參考點 X 座標值存入。
- # 3 參考點 Y 座標值存入。
- # 4 第 1 組孔距 ( $i_1$ )。
- # 5 某組之孔之計數。
- # 6 第 1 組之孔數 ( $n_1$ )。
- # 7 第 2 組孔距 ( $i_2$ )。
- # 8 由參考點到現在位置孔之距離。
- # 9 第 2 組之孔數 ( $n_2$ )。
- # 100 孔之計數。
- # 101 參考點 X 座標值。
- # 102 參考點 Y 座標值。

(3) 圓周等分點之鑽孔



G 65 P 9207 R r A a H h ;

O 9207 ;

```

# 30 = # 101 ;
# 31 = # 102 ;
# 32 = 1 ;
WHILE [ # 32 LE ABS [ # 11 ] ] DO 1 ;
# 33 = # 1 + 360 * [ # 32 - 1 ] / # 11 ;
# 101 = # 30 + # 18 * COS [ # 33 ] ;
# 102 = # 31 + # 18 * SIN [ # 33 ] ;
X # 101 Y # 102 ;
# 100 = # 100 + 1 ;
# 32 = # 32 + 1 ;
END 1 ;
# 101 = # 30 ;
# 102 = # 31 ;
M 99 ;

```

說明：

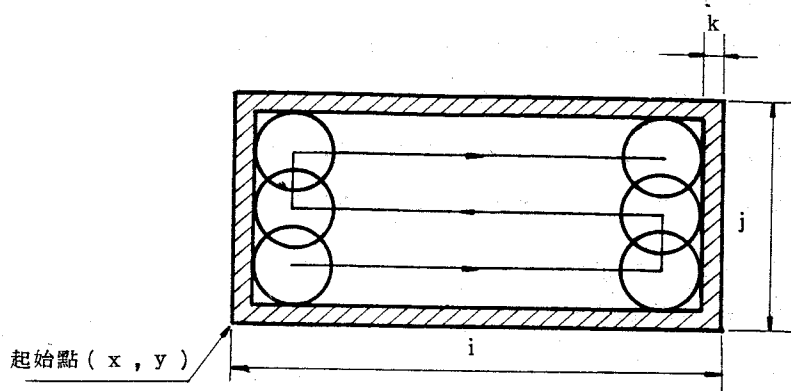
- # 1 為開始點角度 a。
- # 11 孔數 h。
- # 18 圓之半徑 r。
- # 30 X 軸參考點。
- # 31 Y 軸參考點。
- # 32 孔數之累積計算。
- # 33 孔位之角度值。
- # 100 孔數之計算。
- # 101 X 軸座標值。
- # 102 Y 軸座標值。

#### (4)槽型銑削

```

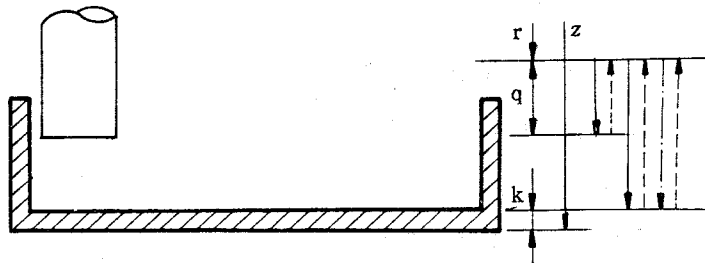
G 65 P 9802 X x Y y Z z R r Q q I i J j K k T t D d F f E e ;
O 9802 ;
# 27 = # [ 2000 + # 7 ] ;
# 28 = # 6 + # 27 ;

```



```

# 29 = # 5 - 2 * # 28 ;
# 30 = 2 * # 27 * # 23 / 100 ;
# 31 = FUP [ # 29 / # 30 ] ;
# 32 = 29 / # 31 ;
# 10 = # 24 + # 28 ;
# 11 = # 25 + # 28 ;
# 12 = # 24 + # 4 - # 28 ;
# 13 = # 26 + # 6 ;
G 00 X # 10 Y # 11 ;
Z # 18 ;
# 14 = # 18 ;
DO 1 ;
# 14 = # 14 - # 17 ;
IF [ # 14 GE # 13 ] GOTO 1 ;
# 14 = # 13 ;
N 1 G 01 Z # 14 F # 8 ;
X # 12 F # 9 ;
# 15 = 1 ;
WHILE [ # 15 LE # 31 ] DO 2 ;
    
```





```

Y [ # 11 + # 15 * # 32 ] ;
IF [ # 15 AND 1 EQ 0 ] GOTO 2 ;
X # 10 ;
GOTO 3 ;
N 2 X # 12 ;
N 3 # 15 = # 15 + 1 ;
END 2 ;
G 00 Z # 18 ;
X # 10 Y # 11 ;
IF [ # 14 LE # 13 ] GOTO 4 ;
G 01 Z [ # 14 + 1 ] F [ 8 * # 8 ]
END 1 ;
N 4 M 99 ;

```

說明：

- x , y : 槽型底部之起點絕對座標 X , Y 值。
- z , r : 圖面上 Z 點與 R 點之絕對座標值。
- g : 一次切削深度 ( 正值 ) 。
- i , j : X , Y 之區域長度。
- k : 精銑預留量。
- t : 重複銑切寬度。
- d : 刀具補正編號。
- f : X Y 平面之進給率。
- e : Z 方向進給率。

## 四、結 論

(1) 切削中心若配合上觸發型探針 ( Touch Probe ) , 則 Macro 指令的應用即更廣泛了 , 可進一步執行胚料基準的設定、工作台基準孔之對正、孔距測量、孔中心線對應平面之平行度、兩孔中心線之直角度等尺寸或形狀有關之檢測 , 祇要直接呼叫 Macro 程式即可執行。

- (2) Macro指令多用於切削形狀一定，尺寸變化之工件，甚而更複雜者，如立體橢圓、立體螺槳者亦可達到，唯對任意曲線方程式之切削描述，仍有待發展。
- (3) 在自動程式語言（例如：APT、FAPT等）所發展之MACRO指令，將更簡捷，相信日後使用上必能日益推廣。
- (4) 系統變數對MACRO本自功能之影響極大，日後如欲發展智慧型切削中心時，則此方面亦值得做進一步之探討。

## 參考資料

1. R. S. Pressman & J. E. Williams, Numerical Control and Computer Aided Manufacturing, New York: Wiley, 1983.
2. Y. Koren, Computer Control of Manufacturing Systems, New York: McGraw-Hill, 1983.
3. M. P. Groover & E. W. Zimmers, Computer Aided Design and Manufacturing, Englewood Cliffs, NJ: Prentice Hall, 1984.
4. T. G. Gunn, Computer Applications in Manufacturing, New York: Industrial Press, 1981.
5. A. D. Roberts & R. C. Prentice, Programming for Numerical Control Machines, New York: McGraw-Hill, 1978.
6. K. Hitomi, Manufacturing Systems Engineering, 中央, 1984.
7. Y. C. Pao, Elements of Computer Aided Design & Manufacturing, 東南, 1985.
8. J. J. Childs, Principles of Numerical Control, New York: The Industrial Press, 1967.
9. P. Bezier, Numerical Control—Mathematics and Applications, New York: Wiley, 1972.
10. FANUC System Operator's Manual
  - . FAPT TURN/MILL
  - . 10/11/12 Operator Manual