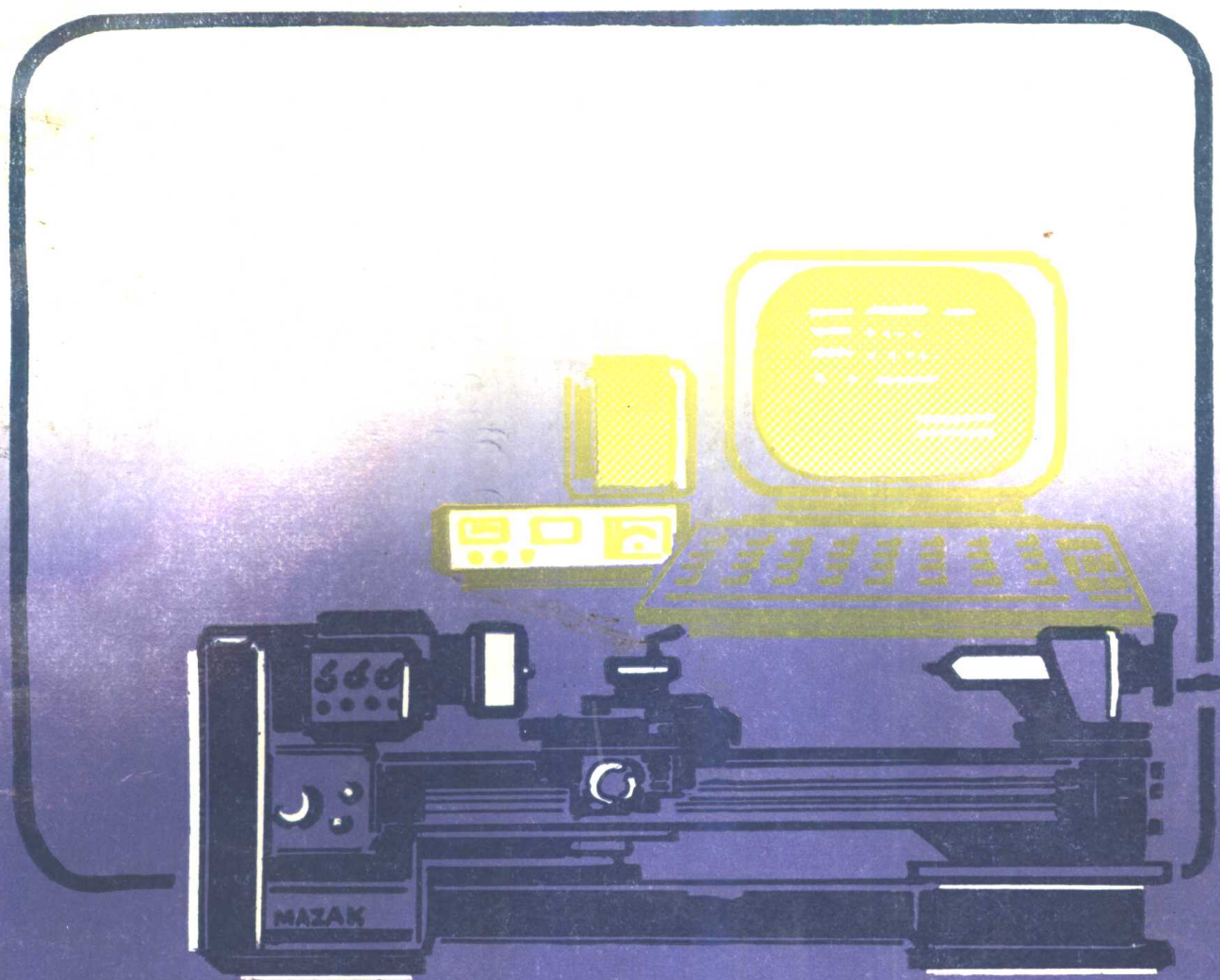




微型计算机 控制机床程序设计

胡学敏 编著

重庆大学出版社

微型计算机控制机床程序设计

胡学敏 编著

重庆大学出版社

内 容 简 介

本书详细地介绍了微型计算机控制机床的数学模型、硬件电路及软件程序。它以具体的产品和科研成果为背景，内容全面、实际、具体。

全书共分七章。主要包括：微型计算机控制机床的数学模型；AIM65单板机简介；AIM65单板机控制机床的硬件电路；微型机控制机床基本功能子程序；微型计算机控制机床主程序；特殊功能子程序；AIM65单板机群控机床等。

本书可供高等学校自动化、计算机、电子专业本科生、研究生使用，亦可供有关专业的科技工作者和操作工人参考。

微型计算机控制机床程序设计

胡学敏 编 著

责任编辑 梁 涛

*

重庆大学出版社出版发行

新华书店经销

重庆印制一厂印刷

*

开本：787×1092 1/16 印张：10.5 字数：262 千

1992年6月第1版 1992年6月第1次印刷

印数：1—5000

标准书号：ISBN 7-5624-0447-X 定价：3.18元
TP·28

前 言

随着计算机技术的迅速发展,微型机控制机床的科研成果和产品在国内相继出现。它具有精度高、通用性强、可靠性高、便于复杂图形的程序编制与加工的特点,使机床控制向智能化方向发展。

目前,在国内关于微机控制机床的书籍多限于控制方法的理论探讨,缺乏完整具体的程序,与实际应用尚有距离。读者学习了微机原理与微机控制技术等课程之后,对于如何把微机应用于实际,尚感到有些力不从心。

为普及、推广微机应用技术,本书采用美国 AIM 65 单板机来控制机床,并用剖析麻雀的方法,使广大读者对于微机控制机床有一个全面的、实际的、具体的理解,本书汇集了作者的科研成果。

全书共分七章。第一章详细地阐述了一种适于微机控制的实现逐点比较法的方案。第二章介绍的 AIM65 单板机具有微型机功能,其 CPU 是大家在“苹果机”中熟悉的 6502,指令简单,只有 56 条,编程容易,功能很强,在美国深受欢迎,在国内也得到很高的评价。第三章介绍了实现控制的硬件电路。第四章至第六章全面的介绍了主程序、子程序及特殊功能子程序。第七章详细地描述了实现群控的程序。全部程序及硬件电路均在实践中得到验证。读者通过学习大量的程序设计实例,既培养了汇编语言的思维能力,也促进了微机原理与实际应用的结合。

本书曾在自动化、计算机及电子等专业使用过多次,不仅适用于自动化、计算机、电子专业的学生,而且对于有关的科技工作者也是一本极好的参考书。

在编写过程中,李小林同志、广西大学电气工程系自动化专业 82 届、83 届的毕业生提供过很多帮助。广西大学教务处苏佩文同志及重庆大学出版社对于本书的出版给予有力的支持,在此表示诚挚的谢意。

由于编者水平有限,书中难免有不当之处,敬请读者批评指正。

编著者

1991. 2. 15

EAC 75/06

目 录

第一章 微型机控制机床的数学模型	(1)
1-1 逐点比较法	(1)
1-2 圆弧偏差计算	(1)
1-3 斜线偏差计算	(3)
1-4 四象限偏差计算与进给	(4)
1-5 实现逐点比较法的微型机控制方案	(7)
1-6 终点判断与程序编制	(10)
1-7 换象限修改加工指令	(12)
1-8 偏移量计算与短路自动回退	(14)
第二章 AIM65 单板机简介	(16)
2-1 AIM65单板机概述	(16)
2-2 AIM65单板机的寻址方式	(19)
2-3 AIM65单板机的指令系统	(25)
2-4 AIM65单板机的子程序概述	(38)
2-5 AIM65 单板机外设接口适配器——6522 多功能接口适配器 (VIA)	(39)
第三章 AIM65微型机控制机床的硬件电路	(44)
第四章 AIM65微型机控制机床基本功能子程序	(49)
4-1 变补子程序	(49)
4-2 加1子程序	(50)
4-3 $F+2X$ (或 $2Y$)+1 子程序	(51)
4-4 $F+X$ (或 Y)子程序	(53)
4-5 齿补子程序	(54)
4-6 输出子程序	(55)
4-7 开高频子程序	(58)
4-8 关高频子程序	(58)
4-9 取加工指令子程序	(59)
4-10 显示加工指令及序号子程序	(60)
4-11 短路自动回退子程序	(64)
4-12 插补中断 入口地址0888	(66)
第五章 AIM65 单板机控制机床主程序	(71)
第六章 特殊功能子程序	(78)
6-1 显示和键入三 B 格式子程序	(79)
6-2 偏移量计算子程序	(87)
6-3 对称工件加工的软件配制	(97)
第七章 AIM65 单板机群控机床	(108)

7-1 群控概述	(108)
7-2 双机群控主程序	(109)
7-3 双机群控子程序	(124)
附录.....	(145)
(一) 微型机控制机床主程序框图.....	(145)
(二) 微型机控制机床程序清单.....	(147)
(三) AIM65 单板机指令表	(155)
参考文献.....	(160)

第一章 微型机控制机床的数学模型

1-1 逐点比较法

首先简单介绍一下机床是如何按规定图形加工出所需工件的。机床每走一步，计算一次偏差，然后根据偏差决定下次进给，所以是三个工作节拍：偏差—判别—进给。

如要加工一段圆弧 $\widehat{AB}$ ，起点为A，终点为B，坐标原点是圆心。X轴、Y轴代表横、纵拖板的方向，圆弧半径为R。从A点开始加工，设某一时刻加工点在 M_1 ，一般说来 M_1 对圆弧 $\widehat{AB}$ 有所偏离，根据偏差可以确定下一步进给的方向，使下一步加工点尽可能向规定图形（即圆弧 $\widehat{AB}$ ）靠拢。

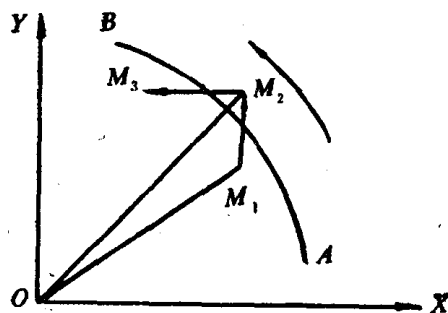


图 1-1

若用 R_{M_1} 表示 M_1 到圆心O的距离，显然 $R_{M_1} < R$ 表示加工点 M_1 在圆内，这时应控制拖板（Y拖板）向圆外进给一步到新加工点 M_2 。拖板进给由步进电机带动，进给的步长是固定的（如1微米），故新的加工点不一定正好在圆弧上。当 $R_{M_2} \geq R$ 时，表示加工点 M_2 在圆外或圆上，这时应控制拖板（X拖板）向圆内进给一步。如此不断地重复上述过程，就能加工出所需的圆弧。显而易见，这里是用折线代替圆弧。为了清楚起见，图1-1中把每步进给的步长都画得比较大，折线与圆弧的误差比较大。若步长缩小，则误差也跟着缩小。实际加工时，进给步长仅为1微米，故实际误差是很小的。

由此可见，拖板进给是步进的，每走一步都要完成以下三个工作节拍：

- (1) 判别：判别加工点对规定图形的偏离位置（例如在加工圆弧时，应判别加工点在圆内还是圆外），以决定拖板的走向。
- (2) 进给：控制X拖板或Y拖板进给一步（如1微米），以向规定图形靠拢。
- (3) 计算：对新加工点计算出反映偏离情况的偏差，以作为下一步判别的依据。

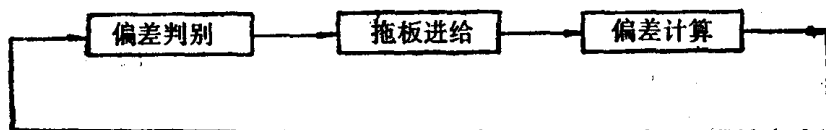


图 1-2

这种控制方案叫逐点比较法，即每进给一步，比较一下加工点与规定图形的位置，一步一步的逼近。判别是进给的依据，偏差计算则是判别的依据，所以偏差计算是很重要的。

1-2 圆弧偏差计算

(一) 加工的偏差公式

加工圆弧时，可以把加工点到圆心的距离和圆弧半径相比较来反映加工偏差。若M点为

加工点, R 为圆弧半径, 设 M 点坐标为 (x_M, y_M) , 根据勾股弦定理知

$$R_M^2 = x_M^2 + y_M^2$$

比较 R_M 与 R 的大小, 只要比较 R_M^2 和 R^2 的大小即可。取 $R_M^2 - R^2$ 作为 M 点的加工偏差, 记为 F_M , 即

$$F_M = R_M^2 - R^2 = x_M^2 + y_M^2 - R^2$$

这就是圆弧加工的偏差公式。直接按此式计算加工偏差, 对计算机是不方便的, 能否找到一个简便的方法呢?

(二) 偏差计算

如果能找出相邻两加工点偏差间的数量关系, 那么每走一步后, 新加工点的偏差可以用前一点的加工偏差来推算。这种方法叫递推法。

在圆弧起点, 加工偏差 $F=0$ 。

设在某一时刻, 加工点 $M_1(x_1, y_1)$ 位于圆外, 该点加工偏差 $F_1 = x_1^2 + y_1^2 - R^2 > 0$, 故需沿 X 轴的负方向进给1微米到 $M_2(x_2, y_2)$, 得

$$x_2 = x_1 - 1$$

$$y_2 = y_1$$

所以 M_2 点的加工偏差 $F_2 = x_2^2 + y_2^2 - R^2 = (x_1 - 1)^2 + y_1^2 - R^2 = x_1^2 + y_1^2 - R^2 - 2x_1 + 1$

$$\therefore F_2 = F_1 - 2x_1 + 1 \quad (1-1)$$

若用二-十进制补码运算, 则

$$F_2 = F_1 + 2x_1 + 1$$

式中 F_1 为正数补码, 简称正补。 $-2x_1$ 的补码仍用 $2x_1$ 表示, x_1 为负数补码, 简称负补。

结论: 偏差 F 为正补与负补之和。这就是 $F_1 > 0$ 时 F_2 与 F_1 间的递推公式。

设 M_2 点已在圆内, 即 $F_2 < 0$ (图 1-3), 则需沿 y 轴方向进给1微米到 $M_3(x_3, y_3)$, 得

$$x_3 = x_2, y_3 = y_2 + 1$$

所以 M_3 点的加工偏差

$$\begin{aligned} F_3 &= x_3^2 + y_3^2 - R^2 = x_2^2 + (y_2 + 1)^2 - R^2 \\ &= x_2^2 + y_2^2 - R^2 + 2y_2 + 1 \end{aligned}$$

$$\therefore F_3 = F_2 + 2y_2 + 1$$

若用补码运算, 则

$$F_3 = F_2 + 2y_2 + 1$$

式中 F_2 为负补, y_2 为正补。

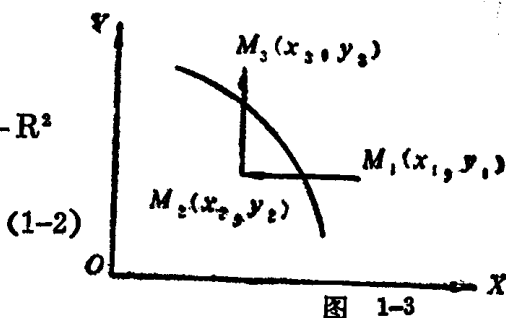


图 1-3

表 1-1 圆弧偏差计算

$F \geq 0$			$F < 0$		
进 给	计 算	补码运算	进 给	计 算	补码运算
$-\Delta x$	$F - 2x + 1 \rightarrow F$ $x - 1 \rightarrow x$ $y \rightarrow y$	$F + 2x + 1 \rightarrow F$ $x + 1 \rightarrow x$ $y \rightarrow y$	$+\Delta y$	$F + 2y + 1 \rightarrow F$ $x \rightarrow x$ $y + 1 \rightarrow y$	$F + 2y + 1 \rightarrow F$ $x \rightarrow x$ $y + 1 \rightarrow y$

注: x, y 为加工圆弧起点坐标值

以上就是 $F_2 < 0$ 时, F_1 与 F_2 间的递推公式。

结论: 偏差 F 仍为正补与负补之和。

归纳上述情况可得表1-1。

1-3 斜线偏差计算

(一) 加工偏差公式

对于斜线可取起点为坐标原点, 选横、纵拖板的方向为X、Y轴作出坐标系。如图1-4所示, 规定斜线 $\overrightarrow{OA}$ 与 $\overrightarrow{OX}$ 的夹角为 α , 某一时刻的加工点为 $M(x_M, y_M)$, 斜线起点到加工点连线 $\overrightarrow{OM}$ 与 $\overrightarrow{OX}$ 的夹角为 α_M 。比较 α 与 α_M 的值就能反映出加工偏差。

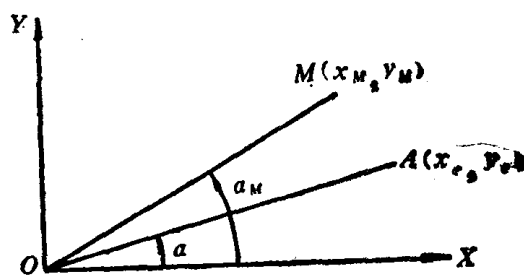


图 1-4

$\alpha_M \geq \alpha$ 表示加工点在规定斜线的左侧, 应控制拖板沿x轴方向向斜线右侧进给一步; $\alpha_M < \alpha$ 表示加工点在规定斜线的右侧, 应控制拖板沿y轴方向向斜线左侧进给一步。

根据三角函数知识, 角的大小可用它的正切值反映, 所以比较 α 与 α_M 的大小, 只要比较 $\text{tg}\alpha$ 和 $\text{tg}\alpha_M$ 的大小即可。

$$\text{tg}\alpha_M = y_M/x_M \quad \text{tg}\alpha = y_0/x_0$$

比较它们的大小又可化为判别它们的差值的符号, 因为

$$\text{tg}\alpha_M - \text{tg}\alpha = y_M/x_M - y_0/x_0 = \frac{x_0 y_M - y_0 x_M}{x_M x_0}$$

而 $x_M x_0 > 0$ (今后规定, 不论在哪个象限, x 、 y 坐标只取绝对值而不考虑符号), 所以 $\text{tg}\alpha_M - \text{tg}\alpha$ 与 $(x_0 y_M - y_0 x_M)$ 的符号相同, 于是可取后者为 M 点的加工偏差, 即

$$F_M = x_0 y_M - y_0 x_M$$

(二) 偏差计算

同加工圆弧一样, 直接计算偏差较麻烦, 仍用递推法计算。

若要加工斜线 $\overrightarrow{OA}$, 终点为 $A(x_0, y_0)$:

(1) M_1 在斜线左侧或在斜线上, 即

$$F_1 = x_0 y_1 - x_1 y_0 \geq 0$$

则应控制拖板沿着 $+\Delta x$ 方向进给1微米到 $M_2(x_2, y_2)$, 得

$$x_2 = x_1 + 1 \quad y_2 = y_1$$

则 M_2 点的加工偏差 F_2 为

$$F_2 = x_0 y_2 - y_0 x_2 = x_0 y_1 - y_0 (x_1 + 1) = x_0 y_1 - y_0 x_1 - y_0 = F_1 - y_0$$

若用二-十进制补码运算, 则

$$F_2 = F_1 + y_0$$

(1-3)

式中 F_1 为正补, $-y_0$ 的补码仍用 y_0 代表, y_0 为负补。

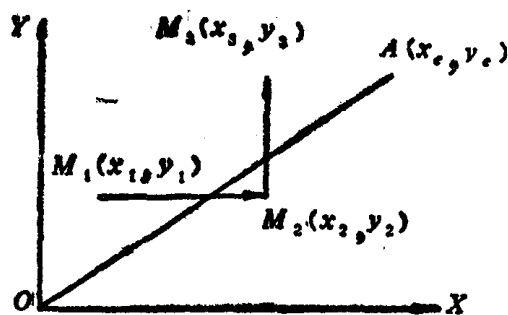


图 1-5

其最终的偏差仍为正补与负补之和。

(2) M_2 在斜线 OA 右侧, 即 $F_2 < 0$, 那么应沿 $+\Delta y$ 方向进给1微米到 $M_3(x_3, y_3)$, 得

$$x_3 = x_2 \quad y_3 = y_2 + 1$$

则 M_3 点的加工偏差 F_3 为

$$F_3 = x_3 y_3 - y_3 x_3 = x_0(y_2 + 1) - y_0 x_2 = x_0 y_2 - y_0 x_2 + x_0 = F_2 + x_0$$

若用二进制补码运算

$$F_3 = F_2 + x_0 \quad (1-4)$$

式中 F_2 为负补; x_0 为正补。

其偏差运算结果仍为正补与负补之和。

在式(1-3)及(1-4)中, 推算偏差 F 都只用到终点坐标 (x_0, y_0) , 所以在加工过程中不必计算加工点的坐标值 (x_M, y_M) 。圆弧偏差 F 的推算需要不断计算新加工点的坐标值, 这是有所不同的。

表 1-2 直线偏差计算

$F \geq 0$			$F < 0$		
进 给	计 算	补码运算	进 给	计 算	补码运算
$+\Delta x$	$F - y \rightarrow F$	$F + y \rightarrow F$	$+\Delta y$	$F + x \rightarrow F$	$F + x \rightarrow F$

1-4 四象限偏差计算与进给

必须指出, 表1-1仅适用于第一象限的圆弧, 表1-2仅适用于第一象限的斜线, 例如对于顺时针走向的第三象限的圆弧(图1-6)则有不同的公式。

若 M_1 在圆外, 则

$$F_1 = x_1^2 + y_1^2 - R^2 \geq 0$$

需沿 $+\Delta y$ 方向进给1微米至 M_2 , 得

$$x_2 = x_1 \quad y_2 = y_1 - 1$$

这里 $y_2 = y_1 - 1$ 是因为规定 x, y 都取坐标绝对值。则

$$\begin{aligned} F_2 &= x_2^2 + y_2^2 - R^2 = x_1^2 + (y_1 - 1)^2 - R^2 \\ &= x_1^2 + y_1^2 - R^2 - 2y_1 + 1 \end{aligned}$$

所以 $F_2 = F_1 - 2y_1 + 1$

若令 $y = [-y_1]_{\text{补}}$, $F = [F_1]_{\text{补}}$ 则

$$F = F + 2y + 1$$

设 M_2 已在圆内, 则 $F_2 < 0$, 需沿 $-\Delta x$ 方向进给1微米至 M_3 , 得 $x_3 = x_2 + 1, y_3 = y_2$

$$\begin{aligned} F_3 &= x_3^2 + y_3^2 - R^2 = (x_2 + 1)^2 + y_2^2 - R^2 \\ &= x_2^2 + y_2^2 - R^2 + 2x_2 + 1 \end{aligned}$$

所以 $F_3 = F_2 + 2x_2 + 1$

若令 $F = [F_2]_{\text{补}}$, $x = [x_2]_{\text{补}}$, 则

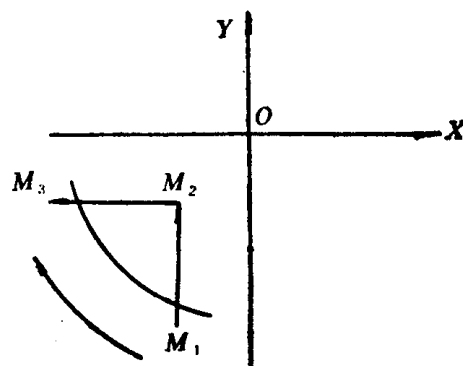


图 1-6

$$F_2 = F + 2x + 1$$

归纳为表1-3

表 1-3 顺圆第三象限偏差公式

$F \geq 0$			$F < 0$		
进 给	计 算	补码运算	进 给	计 算	补码运算
$+\Delta y$	$F - 2y + 1 \rightarrow F$ $x \rightarrow x$ $y - 1 \rightarrow y$	$F + 2y + 1 \rightarrow F$ $x \rightarrow x$ $y + 1 \rightarrow y$	$-\Delta x$	$F + 2x + 1 \rightarrow F$ $x + 1 \rightarrow x$ $y \rightarrow y$	$F + 2x + 1 \rightarrow F$ $x + 1 \rightarrow x$ $y \rightarrow y$

显然，此表与表1-1（逆圆第一象限）不同。事实上，圆弧共有顺圆一、二、三、四象限和逆圆一、二、三、四象限8种加工指令；斜线也有一、二、三、四象限4种加工指令。这12种情况分别用 SR_1 、 SR_2 、 SR_3 、 SR_4 、 NR_1 、 NR_2 、 NR_3 、 NR_4 、 L_1 、 L_2 、 L_3 和 L_4 表示。把各种情况下的偏差计算公式归纳于表1-4及图1-7中。

由此可以看出，对于不同的线型（直线或是圆弧）、不同的象限、不同的加工方向（顺圆

表 1-4 加 工 运 算 表

线 型	加 工 指 令	$F \geq 0$		$F < 0$	
		进给坐标	偏差运算	进给坐标	偏差运算
圆 弧	SR_1 , SR_3 NR_2 , NR_4	Δy	$F + 2y + 1 \rightarrow F$ $y + 1 \rightarrow y$	Δx	$F + 2x + 1 \rightarrow F$ $x + 1 \rightarrow x$
	SR_2 , SR_4 NR_1 , NR_3	Δx	$F + 2x + 1 \rightarrow F$ $x + 1 \rightarrow x$	Δy	$F + 2y + 1 \rightarrow F$ $y + 1 \rightarrow y$
直 线	L_1 , L_3	Δx	$F + y \rightarrow F$	Δy	$F + x \rightarrow F$
	L_2 , L_4	Δy	$F + x \rightarrow F$	Δx	$F + y \rightarrow F$

表 1-5 变 补 坐 标 b 值

象 限	加工方向	变补坐标	象 限	加工方向	变补坐标
I	顺 圆	补 y $b=1$	I	逆 圆	补 x $b=0$
II	顺 圆	补 x $b=0$	II	逆 圆	补 y $b=1$
III	顺 圆	补 y $b=1$	III	逆 圆	补 x $b=0$
IV	顺 圆	补 x $b=0$	IV	逆 圆	补 y $b=1$

还是逆圆)，偏差方向不同，进给的坐标轴不同，其方向也不一定相同。

从上述偏差计算公式中可以看出：进给的方向若使该坐标的绝对值减小，则在偏差计算公式中就要出现减法运算，使用补码运算时，需将该坐标送去变补。所以在不同的象限、不同的加工方向中，当进给方向使坐标的绝对值减小时，应将该轴坐标送去变补。如在逆圆一、三象限中，X的进给方向使X坐标的绝对值减小，应将x变补；在顺圆一、三象限中，应将y送去变补（简称补y）。各象限在不同的加工方向时变补坐标详见表1-5。

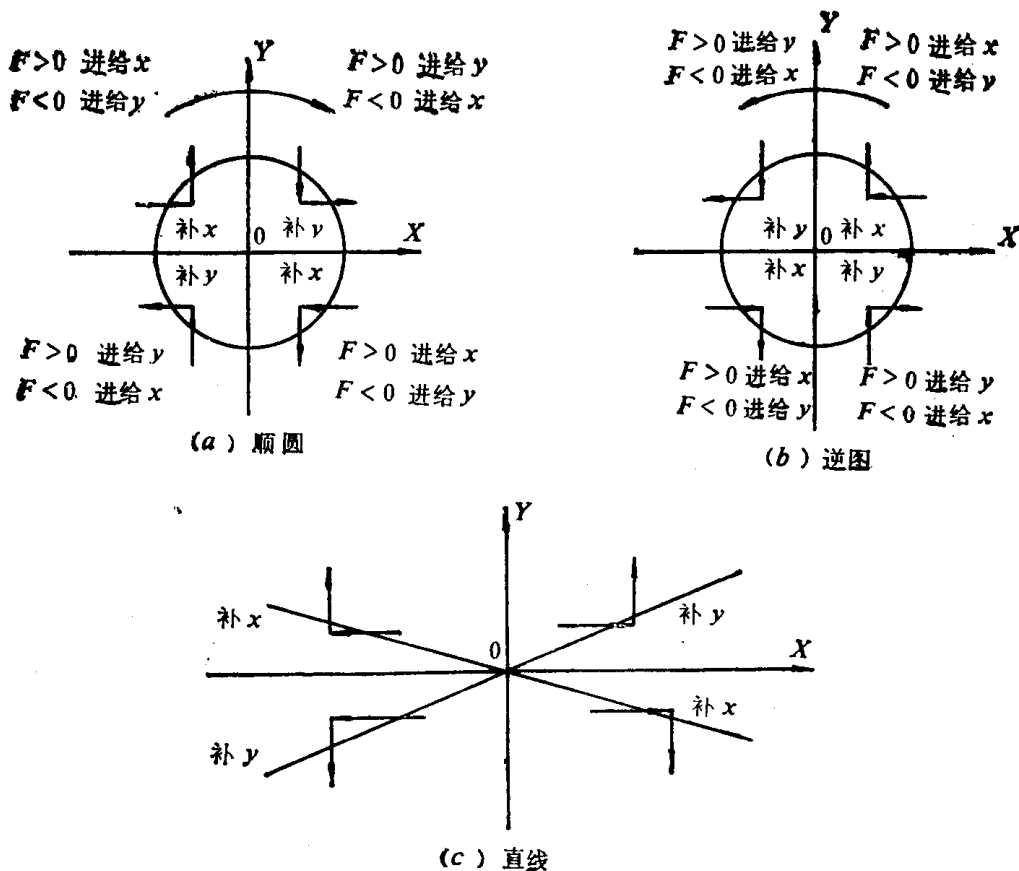


图 1-7

直线加工的四象限进给方向，均使坐标的绝对值增大，故无变补问题。但是为了与圆^取一致的计算公式，仍考虑变补问题，定为与顺圆一致。把X坐标变补定为 $b=0$ ，Y坐标变补定为 $b=1$ ，那么可以看出 b 是一个既代表象限又对加工方向有代表性的量。

从图1-7还可以看出：

①每换象限，变补坐标要改变。

②每换象限，有一轴的运动方向要改变，要改变的轴恰恰是换象限前不进行变补的那个坐标轴。

用 L 表示代表线型的量， $L=0$ ，代表加工圆弧； $L=1$ ，代表加工直线。

用 F 表示代表偏差方向的量，若 $F>0$ ，用 $CC=0$ 表示；若 $F<0$ 则 $CC=1$ 。

再定义一个决定进给方向及计算公式的 Q 值， Q 为进给坐标。 $Q=0$ ，进给x； $Q=1$ 进给y。

1-5 实现逐点比较法的微型机控制方案

(一) 决定Q值的逻辑式

在加工过程中每走一步，机床的计算装置便进行一次计算，以判别加工点现在的位置是否偏离加工曲线，然后按偏差决定下次进给方向。因此关键的问题在于如何反映偏差，怎样进行偏差计算，如何让微型机自己判别而后自动进给。

根据图1-7及规定的Q、CC、L、b，可得出逻辑式

$$Q = CC \oplus L \oplus b$$

其中

$$\begin{cases} Q=0 & \text{进给}x, \\ Q=1 & \text{进给}y, \end{cases} \quad \begin{cases} CC=0 & F \geq 0 \text{时} \\ CC=1 & F \leq 0 \text{时} \end{cases}$$

$$\begin{cases} L=1 & \text{加工直线} \\ L=0 & \text{加工圆弧} \end{cases} \quad \begin{cases} b=0 & x \text{坐标变补} \\ b=1 & y \text{坐标变补} \end{cases}$$

上式是根据实际加工情况（12种加工指令）的真值表归纳出的逻辑式。

表1-6 真 值 表

线 型 L	加工指令	变 补 坐 标	$F \geq 0 (CC=0)$		$F < 0 (CC=1)$	
			运 算	进 给	运 算	进 给
圆 弧 L=0	SR ₁ , SR ₂ , NR ₁ , NR ₂	补 y b=1	$F+2y+1 \rightarrow F$ $y+1 \rightarrow y$	Δy Q=1	$F+2x+1 \rightarrow F$ $x+1 \rightarrow x$	Δx Q=0
	SR ₃ , SR ₄ , NR ₃ , NR ₄	补 x b=0	$F+2x+1 \rightarrow F$ $x+1 \rightarrow x$	Δx Q=0	$F+2y+1 \rightarrow F$ $y+1 \rightarrow y$	Δy Q=1
直 线 L=1	L ₁ , L ₃	补 y b=1	$F+y \rightarrow F$	Δx Q=0	$F+x \rightarrow F$	Δy Q=1
	L ₂ , L ₄	补 x b=0	$F+x \rightarrow F$	Δy Q=1	$F+y \rightarrow F$	Δx Q=0

其中L与b在加工指令字Z中给出，偏差CC在每次进给后计算， Δx 与 Δy 的进给方向也在加工指令字中给出，于是计算机就可自动进给了。

加工指令字Z由一字节组成，从低位到高位，每位的涵义如下：

加工指令 Z

			b	L	O	D _y	D _x
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀

以Z称呼这个内存单元字。

1. D₀位 表示x进给方向

D₀=0=D_x(或d_x) x正向进给

D₀=1=D_x(或d_x) x负向进给

2. D_1 位 表示y进给方向
 $D_1=0=D_y$ (或 d_y) y正向进给
 $D_1=1=D_y$ (或 d_y) y负向进给
3. D_2 位 顺逆圆加工
 $D_2=0=O$ 顺圆加工
 $D_2=1=O$ 逆圆加工
4. D_3 位 走圆走线判别
 $D_3=1=L$ 直线加工
 $D_3=0=L$ 圆弧加工
5. D_4 位 变补坐标
 $D_4=0=b$ 变补x坐标
 $D_4=1=b$ 变补y坐标

如 NR_{11} , $b=0$ $L=0$ $O=1$ $D_y=0$ $D_x=1$ 。所以加工指令为#05

(二) 计算偏差CC

对于逐点比较法的插补运算公式,已导出这样的结论:无论在哪个象限、哪一种线型、哪一种偏离情况,任何一个公式的偏差总是正数补码和负数补码的求和运算。其最终结果是一个正补码还是一个负数的补码,是判别偏差的可靠依据,即F的符号:

$$\begin{aligned} F \geq 0 & \quad CC=0 \\ F < 0 & \quad CC=1 \end{aligned}$$

偏差计算公式:

$$\begin{aligned} \text{圆弧} \quad & F+2x+1 \rightarrow F \\ & F+2y+1 \rightarrow F \\ \text{直线} \quad & F+x \rightarrow F \\ & F+y \rightarrow F \end{aligned}$$

从数值运算角度来看,其计算结果可能有下列3种情况:

- ① 小于寄存器容许存放数值的范围,即最高位没有进位。
- ② 大于寄存器容许存放数值的范围,即最高位有一次进位。
- ③ 再一次大于寄存器容许存放数值的范围,即最高位有二次进位。

由于采用补码运算,故②、③两种情况是正常情况,而不是溢出。理由如下:

在计算机中只有+1与-1相加才有进位,即

$$[-1]_{\text{补}} = [-1]_{\text{反}} + 1 = 11111110 + 1 = 11111111$$

$$[-1]_{\text{补}} + 1 = 11111111 + 1 = \boxed{1} 00000000$$

如果有进位表示正负抵消,由于 $F + \frac{2x}{2y} + 1 \rightarrow F$,故可以有两次正负抵消情况,可能有两次进位。用最高位的进位寄存器 F_c 记录进位状态,并规定: $F_c=0$,表示没有进位或二次进位; $F_c=1$,表示一次进位。

必须根据原偏差F的符号CC与运算后的进位 F_c 共同判断运算后的偏差符号。

1. 圆弧计算偏差CC公式

$$CC \oplus F_c \rightarrow CC$$

下面简单给以论证

(1) 原偏差 $CC=0$

① 当 $F_c=1$ 即一次进位时, 新偏差

$$CC=CC \oplus F_c=0 \oplus 1=1$$

即新偏差为负, 这是由于原偏差的正值被负值抵消的缘故。

② 当 $F_c=0$ 即无进位时, 新偏差

$$CC=CC \oplus F_c=0 \oplus 0=0$$

即新偏差为正, 原偏差的正值未被负值抵消, 仍保持正值。

③ 当 $F_c=0$ 有二次进位时, 新偏差

$$CC=CC \oplus F_c=0 \oplus 0=0$$

即新偏差为正, 原偏差的正值在运算过程中发生了两次正负抵消, 最后还原为正值。

(2) 原偏差 $CC=1$

① $F_c=1$ 即一次进位时, 新偏差

$$CC=F_c \oplus CC=1 \oplus 1=0$$

即新偏差为正, 这是由于原偏差的负值被正值抵消所致。

② $F_c=0$ 无进位时, 新的偏差

$$CC=F_c \oplus CC=0 \oplus 1=1$$

即新偏差为负, 这是由于原偏差的负值并未被抵消, 所以保持原来的符号。

③ $F_c=0$ 二次进位, 新的偏差

$$CC=F_c \oplus CC=0 \oplus 1=1$$

即新偏差为负, 原偏差的负值发生了两次正负抵消, 运算的最后结果仍为负值。

2. 直线计算偏差 CC 公式

$$CC=F_c \oplus L \oplus T_c$$

其中 T_c 为 x 、 y 坐标进位寄存器, 当 $x=y=0$ 送去变补时

$$[00000000]_{\text{补码}}=[00000000]_{\text{反码}}+1$$

$$=11111111+1=1\ 00000000$$

此时有进位, 即 $T_c=1$ 。

下面简单地加以论证:

(1) 平行于 x 、 y 坐标轴的直线

在插补运算中, 加工平行于 x 轴或 y 轴的直线时, 终点坐标 x_e 、 y_e 用“零”来表示, 因此在公式 $F+x \rightarrow F$ 或 $F+y \rightarrow F$ 中永远不会有进位, 即 $F_c=0$, 而 $L=1$; 当坐标值 $x=y=0$ 变补时 $T_c=1$, 故

$$CC=F_c \oplus L \oplus T_c=0 \oplus 1 \oplus 1=0$$

平行于 x 轴或 y 轴的直线加工, 在四象限中 CC 永为零。而公式 $Q=CC \oplus L \oplus b$ 中的 b 值不同, 保证了在四象限中分别平行于 x 轴或 y 轴的直线的进给不同。如第一象限, $b=1$, 从而 $Q=0$, 保证进给 x , 进行平行于 x 轴直线的加工。

(2) 斜线加工

由于斜线的终点坐标 $x \neq 0$, $y \neq 0$, 故变补时不会有进位, 即 $T_c=0$, $L=1$ 。下面仅就

第一象限的斜线加工验证上述逻辑式:

设加工斜线 $\overrightarrow{OA}$, 其路径为 $O \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4$.

(a) 在原点 O : 偏差值 $F=0$, 在偏差运算 $F+x$ (或 y) $\rightarrow F$ 中不会有进位, $F_c=0$; $CC=F_c \oplus L \oplus T_c=0 \oplus 1 \oplus 0=1$,

(b) 在 $O \rightarrow M_1$ 段: 原偏差 $F=0$, 进给 x 后 $F+y \rightarrow F$, 无进位, 故 $F_c=0$, 新偏差

$$CC=F_c \oplus L \oplus T_c=0 \oplus 1 \oplus 0=1$$

即新偏差值为负, 这是符合实际情况的。

(c) $M_1 \rightarrow M_2$ 段: 从 M_1 点进给 y 后至 M_2 点, 偏差值的方向有变化, 故有进位, $F_c=1$, 新偏差

$$CC=F_c \oplus L \oplus T_c=1 \oplus 1 \oplus 0=0$$

新偏差为正, 符合实际情况。

(d) $M_2 \rightarrow M_3$ 段: 从 M_2 进给 x 后, 偏差值又有变化, 发生二次进位, 即 $F_c=0$, 于是新偏差

$$CC=F_c \oplus L \oplus T_c=0 \oplus 1 \oplus 0=1$$

说明新偏差为负是正确的。

(e) $M_3 \rightarrow M_4$, 类推……

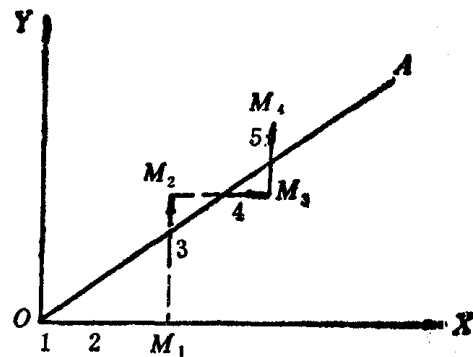


图 1-8

1-6 终点判断与程序编制

(一) 终点判断

按逐点比较法一步一步进行下去, 直至无穷大。实际加工时, 需要在加工点到达终点时能自动结束。因此, 在逐点比较法中还要增加一个“终点判断”的工作节拍。

进给及偏差计算完后, 不是立即进行下一步的偏差判别, 而应先判断是否到达了加工终点, 若已到达终点则加工结束。可以通过控制某个拖板从起点到终点的进给总长度来进行终点判断, 可设置一个计数器 J , 拖板进给与计数长度方向一致, 则 $J-1 \rightarrow J$, 直至 J 为 0, 表示加工至终点。

选取 x 拖板进给总长度来进行计数的称为计 x , 用 $G=0$ 表示; 选取 y 拖板进给总长度来进行计数的称为计 y , 用 $G=1$ 表示。

(二) 计数方向的选取

选取计数方向的原则是使加工误差尽可能最小。加工斜线时, 应对进给距离比较长的一个拖板进行计数长度的控制, 当 J 减至 0 时, 保证进给距离比较短的一个拖板已进给完毕。相反, 若选取对进给距离比较短的一个拖板长度进行计数长度, J 减至 0 时进给距离比较长的一个拖板尚未进给完毕, 如此时结束加工, 必定造成误差。

(1) 若斜线加工终点为 $A(x_0, y_0)$, 当 $y_0 > x_0$ 时计数方向取 y , 即 $G=1$ 或称 G_y , 当 $y_0 < x_0$ 时计数方向取 x , 即 $G=0$ 或称 G_x 。

斜线位于阴影区内, 计数方向为 G_y , 反之为 G_x 。若斜线正好在 45° 线上, 计数方向可任

意选取。

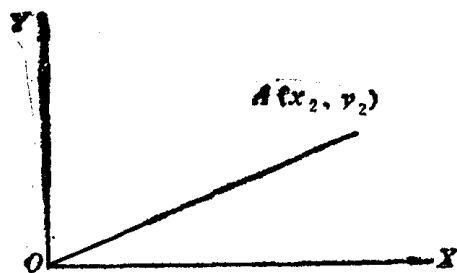


图 1-9

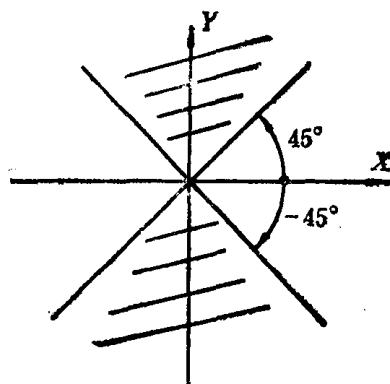


图 1-10

(2) 圆弧计数方向的选取，视加工到圆弧终点附近的进给情况而定。终点靠近y轴，在终点附近圆弧的趋向平行于x轴，最后一步进给的拖板必定是x拖板，故需选取 $G=G_x$ 才不会漏走。同理，当圆弧终点靠近x轴时，在终点附近圆弧趋向平行于y轴，应取 $G=G_y$ 。以 45° 为界确定圆弧的计数方向，必须看终点坐标 (x_e, y_e) 。若 $x_e < y_e$ ，即终点在阴影区内，计数方向为 G_x ，反之则为 G_y 。圆弧与斜线的情况正好相反。

在计算机中，若 $G \oplus Q = 0$ ，表示计数方向与进给方向一致，此时 $[J]_{补} + 1 \rightarrow [J]_{补}$ (补码运算)。直至坐标进位寄存器 $T_c = 1$ ，表示计数长度减为0，加工完毕。

(三) 计数长度的确定

计数长度应取某拖板从起点到终点移动的总距离。当计数方向确定后，就是被加工曲线在该方向上投影长度的总和。对于圆弧而言，它可能跨越几个象限，计数方向确定后，计数长度应取圆弧各段在该方向上投影的总和。

(四) 程序编制

控制台是按照人的“命令”去控制机床加工的。因此，必须事先将人们的意图，用机器所能接受的语言“编排好”命令并告诉控制台，这个工作叫程序编制。

在编排命令时，为使机器能够接受，必须符合一定的格式，本控制台程序格式如下

B	x	B	y	B	J	G	Z
	x坐标值		y坐标值		计数长度	计数方向	加工指令

对此格式作几点说明：

- ① B 为分隔符，因为x、y、J均为数码，需用B将它们分隔开来。
- ② 加工圆弧时，坐标原点取在圆心，x、y为起点坐标值。
- ③ 加工斜线时，坐标原点取在起点，x、y为终点坐标值。斜线加工偏差仅由斜线与X轴正向的夹角或终点坐标x、y之比来决定。故可允许将x、y缩小相同倍数(其比值保持不变)。在加工一个工件的过程中，X、Y坐标轴的方向应始终保持不变，即为x拖板和y拖板的运动方向。加工曲线不同时，取不同的坐标原点，只是坐标的平移。
- ④ x、y、J数值均以微米为单位。当x或y为0时，可以不写。对于控制长度J，应写足6位数，如 $J=1970$ 微米，应写为001970。
- ⑤ 加工指令共有12种。对于平行于坐标轴的线，为了区别于一般的斜线称之为直线。根据进给情况，直线加工指令的选取由图1-11规定。且此时程序中应取 $x=y=0$ 。圆弧可能