



数字控制技术

数字控制(简称数控)也称数值控制(Numerical Control, NC),是利用数字化信息对机械运动及加工过程进行控制的一种方法。装有数字控制系统的机床称为数控机床。数控机床具有能加工形状复杂的零件、加工精度高、生产效率高、便于改变加工零件品种等许多特点,是实现机床自动化的重要发展方向。典型的数控机床有铣床、车床、加工中心、线切割机和焊接机等。数控技术和数控机床是实现柔性制造系统(Flexible Manufacturing System, FMS)和计算机集成制造系统(CIMS)最重要的基础技术之一。

另外,顺序控制是开环数字控制和自动化技术的重要应用领域。顺序控制是使生产过程按照规定的时序或事序工作的自动控制。由于工业控制机、第2章的数字量(开关量)输入输出接口与过程通道已具有与、或、非等逻辑运算和记忆、判断、定时、延时等功能,并能实现有触点的继电器逻辑控制、无触点的集成电路和晶体管逻辑控制,因此本章对这些内容不再赘述。

本章主要介绍数字控制基础、运动轨迹插补原理、进给速度与加减速控制、电机驱动控制与位置伺服系统和多轴运动控制技术。

3.1 数字控制基础

所谓数字控制,就是生产机械(如各种加工机床)根据数字计算机输出的数字信号,按规定的工作顺序、运动轨迹、运动距离和运动速度等规律自动地完成工作的控制方式。

3.1.1 数控技术发展概况

世界上第一台数控机床是1952年美国麻省理工学院(MIT)伺服机构实验室开发的,当时的主要动机是为了满足高精度和高效率加工复杂零件的需要。众所周知,二维和三维轮廓零件的加工是很困难的,而数控机床则很容易实现二维和三维轮廓零件的加工。早期的数控是以数字电路技术为基础实现的。随着小型和微型计算机的发展,20世纪70年代初期在数控系统中用计算机代替了控制装置,从而诞生了计算机数控(Computer Numerical Control, CNC)。表3-1给出了数控技术的发展概况。

数控系统一般由数控装置、驱动装置、可编程控制器和检测装置等构成。

表 3-1 数控技术的发展概况

特征阶段	典型应用	工艺方法	数控功能	驱动特点
研究开发 (1952—1969 年)	数控车床、钻床、铣床	简单工艺	传统数控	三轴以下步进、液压电机驱动
推广应用 (1970—1985 年)	加工中心、电加工、锻压	多种工艺方法	计算机数控、刀具自动交换、五轴联动、较好的人机界面	直流伺服电机驱动
系统化(1982 年)	柔性制造单元(FMU)、柔性制造系统(FMS)	复合设计加工	友好的人机界面	交流伺服电机驱动
高性能集成化 (1990 年至今)	计算机集成制造系统(CIMS)、无人化工厂	复合设计加工	多过程、多任务调度、模板化和复合化、智能化	高性能直线伺服电机驱动

数控装置能接收零件图纸加工要求的信息,进行插补运算,实时地向各坐标轴发出速度控制指令。驱动装置能快速响应数控装置发出的指令,驱动机床各坐标轴运动,同时能提供足够的功率和扭矩。可编程控制器发出运动的指令信号,驱动装置快速响应跟踪指令信号。检测装置检测坐标的实际值,反馈给数控装置的调节电路中的比较器,有差值就发出运动控制信号,从而实现偏差控制。数控装置包括输入装置、输出装置、控制器和插补器 4 部分,这些功能都由计算机完成。

3.1.2 数字控制原理

首先分析如何用计算机在绘图仪或数控机床上重现图 3-1 所示的平面图形,以此简要说明数字控制的基本原理。

(1) 将图 3-1 所示的曲线分割成若干段,可以是直线段,也可以是曲线段。图 3-1 中分割成 3 段,即 $\overline{ab}$ 、 $\widehat{bc}$ 和 $\widehat{cd}$,然后把 a 、 b 、 c 、 d 这 4 个点的坐标记下来并送给计算机。图形分割的原则是应保证线段所连的曲线(或折线)与原图形的误差在允许范围之内。由图 3-1 可见,显然采用 $\overline{ab}$ 、 $\widehat{bc}$ 和 $\widehat{cd}$ 比 $\overline{ab}$ 、 $\widehat{bc}$ 和 $\widehat{cd}$ 要精确得多。

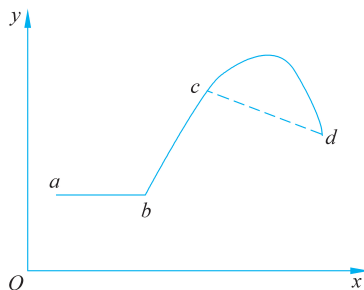


图 3-1 曲线分段

(2) 当给定 a 、 b 、 c 、 d 各点坐标 x 和 y 值之后,如何确定各坐标值之间的中间值?求得这些中间值的数值计算方法称为插补或插值。插补计算的宗旨是:通过给定的基点坐标,以一定的速度连续定出一系列中间点,而这些中间点的坐标值以一定的精度逼近给定的线段。从理论上讲,插补可用任意函数形式,但为了简化插补运算过程和加快插补速度,常用的是直线插补和二次曲线插补两种形式。所谓直线插补是指在给定的两个基点之间用一条近似直线逼近,也就是由此定出的中间点连接起来的折线近似一条直线,并不是真正的直线。所谓二次曲线插补是指在给定的

两个基点之间用一条近似曲线逼近,也就是实际的中间点的连线是一条近似曲线的折线弧。常用的二次曲线有圆弧、抛物线和双曲线等。对于图 3-1 所示的曲线,显然 ab 和 bc 段用直线插补, cd 段用圆弧插补是合理的。

(3) 根据插补运算过程中定出的各中间点,以脉冲信号的形式控制 x 、 y 方向上的步进电机,带动绘图笔、刀具等,从而绘出图形或加工出要求的轮廓。这里的每一个脉冲信号代表步进电机走一步,即绘图笔或刀具在 x 或 y 方向移动一个位置。每个脉冲信号控制移动的距离称为脉冲当量,又称为步长,常用 Δx 和 Δy 表示,并且总是取 $\Delta x = \Delta y$ 。

图 3-2 是一段用折线逼近直线的直线插补线段,其中 (x_0, y_0) 代表该线段的起点坐标值, (x_e, y_e) 代表终点坐标值,则 x 方向和 y 方向应移动的总步数 N_x 和 N_y 分别为

$$N_x = (x_e - x_0) / \Delta x, N_y = (y_e - y_0) / \Delta y$$

如果把 Δx 和 Δy 定义为坐标增量值,即 x_0, y_0, x_e, y_e 均是以脉冲当量定义的坐标值,则

$$N_x = x_e - x_0, N_y = y_e - y_0$$

所以,插补运算就是如何分配 x 和 y 方向上的脉冲数,使实际的中间点轨迹尽可能地逼近理想轨迹。实际的中间点连接线是一条由 Δx 和 Δy 的增量值组成的折线,只是由于实际的 Δx 和 Δy 的值很小,眼睛分辨不出来,看起来似乎和直线一样而已。显然, Δx 和 Δy 的增量值越小,就越逼近理想的直线段,图 3-2 中以 $\rightarrow$ 代表 Δx 或 Δy 的长度和方向。

实现直线插补和二次曲线插补的方法有很多,常见的有逐点比较法(又称富士通法或醉步法)、数字积分法(又称数字微分分析法)、数字脉冲乘法器(又称 MIT 法,由麻省理工学院首先使用)等,其中又以逐点比较法使用最广。

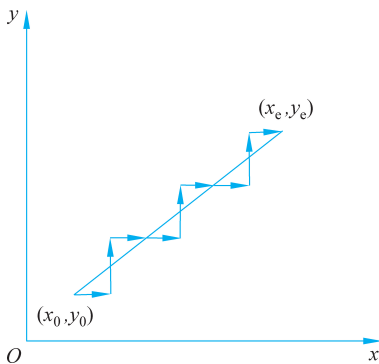


图 3-2 用折线逼近直线段

3.1.3 数字控制方式

数控系统按控制方式可以分为点位控制、直线切削控制和轮廓切削控制,这 3 种控制方式都是运动的轨迹控制。

1. 点位控制

在一个点位控制系统中,只要求控制刀具行程终点的坐标值,即工件加工点准确定位,至于刀具从一个加工点移到下一个加工点走什么路径、移动的速度、沿哪个方向趋近都无须规定,并且在移动过程中不做任何加工,只是在准确到达指定位置后才开始加工。在机床加工业中,采用这类控制的主要是孔加工机床,如钻床、镗床、冲床等。

2. 直线切削控制

直线切削控制也主要是控制行程的终点坐标值,不过还要求刀具相对于工件平行某一直角坐标轴做直线运动,且在运动过程中进行切削加工。采用这类控制的有铣床、车

床、磨床、加工中心等。

3. 轮廓切削控制

轮廓切削控制的特点是能够控制刀具沿工件轮廓曲线不断地运动,并在运动过程中将工件加工成某一形状。这种方式是借助于插补器进行的,插补器根据加工的工件轮廓向每一坐标轴分配速度指令,以获得图纸坐标点之间的中间点。这类控制用于铣床、车床、磨床、齿轮加工机床等。

在上述3种控制方式中以点位控制最简单,因为它的运动轨迹没有特殊要求,运动时又不加工,所以它的控制电路只要具有记忆(记下刀具应走的移动量和已走过的移动量)和比较(将所记忆的两个移动量进行比较,当两个数值的差为0时,刀具立即停止)的功能即可,根本不需要插补计算。和点位控制相比,由于直线切削控制进行直线加工,其控制电路要复杂一些。轮廓切削控制要控制刀具准确地完成复杂的曲线运动,所以控制电路更复杂,且需要进行一系列插补计算和判断。

3.1.4 数字控制系统

计算机数控系统主要分为开环数字控制和闭环数字控制两大类,由于它们的控制原理不同,因此其系统结构差异很大。

1. 开环数字控制

随着计算机技术的发展,开环数字控制得到了广泛的应用,例如各类数控机床、线切割机、低速小型数字绘图仪等,它们都是利用开环数字控制原理实现控制的机械加工设备或绘图设备。开环数字控制的结构如图3-3所示,这种控制结构没有反馈检测元件,工作台由步进电机驱动。步进电机接收步进电机驱动电路发来的指令脉冲,进行相应的旋转,把刀具移动到与指令脉冲相当的位置,至于刀具是否到达了指令脉冲规定的位置,则不受任何检查,因此这种控制的可靠性和精度基本上由步进电机和传动装置决定。

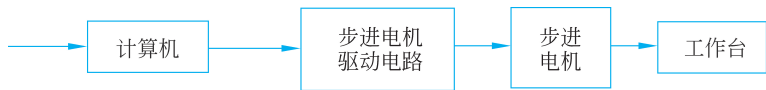


图 3-3 开环数字控制的结构

开环数字控制结构简单,具有可靠性高、成本低、易于调整和维护等特点,在经济型数控系统中应用最为广泛。由于采用了步进电机作为驱动元件,使得系统的可控性变得更加灵活,更易于实现各种插补运算和运动轨迹控制。本章主要讨论开环数字控制技术。

2. 闭环数字控制

图3-4(a)给出了闭环数字控制的结构。这种结构的执行机构多采用直流电机(小惯

量伺服电机和宽调速力矩电机)作为驱动元件,反馈测量元件采用光电编码器(码盘)、光栅、感应同步器等。该控制方式主要用于大型精密加工机床,但其结构复杂,难于调整和维护,常规的数控系统很少采用。

将测量元件从工作台移动到伺服电机的轴端,就构成了半闭环控制系统,如图 3-4(b)所示。这样构成的系统,工作台不在控制环内,克服了由于工作台的某些机械环节的特性引起的参数变动,容易获得稳定的控制特性,因此这种控制方式广泛应用于连续控制的数控机床上。

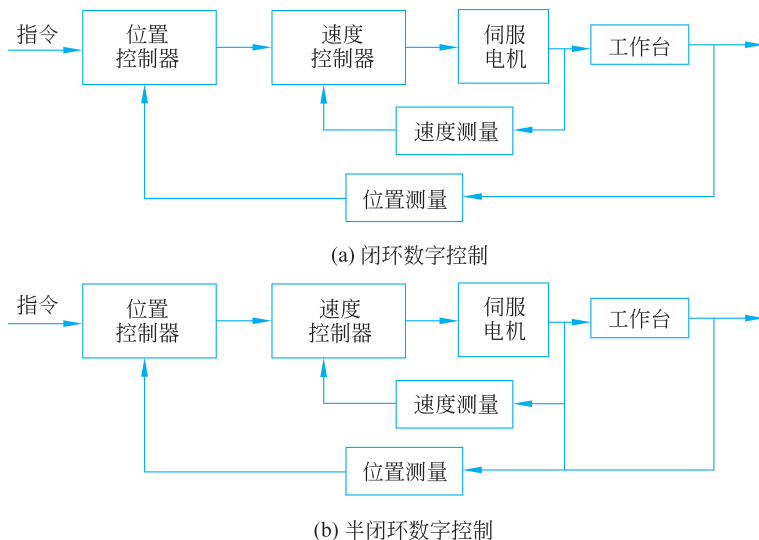


图 3-4 闭环/半闭环数字控制的结构

3.1.5 数控系统的分类

1. 传统数控系统

传统数控系统又称为硬件式数控系统,零件程序的输入、运算、插补及控制功能均由专用硬件完成,这是一种专用的封闭体系结构,其功能简单,柔性和通用性差,设计研发周期长。

2. 开放式数控系统

1) PC 嵌入数控系统结构的开放式数控系统

PC 嵌入数控系统结构的开放式数控系统是基于传统数控系统的半开放式数控系统。这类数控系统是在不改变原数控系统基本结构的基础上,在传统的非开放式的数控系统中插入一块专门开发的个人计算机模板,使得传统的数控系统带有计算机的特点。该系统借助了 PC 丰富的软硬件资源和多媒体部件,把 PC 和数控系统联系在一起,它既具有原数控系统工作可靠的特点,同时它的界面又比原来的数控系统开放,极大地增强了人机界面的功能,使数控系统的功能得以更好地体现,而且使用更加方便。



2) 数控系统嵌入 PC 结构的开放式数控系统

数控系统嵌入 PC 结构的开放式数控系统以 PC 作为系统的核心,由 PC 和开放式运动控制卡构成。所谓的开放式运动控制卡,就是一个可以单独使用的数控系统,具有很强的运动控制和 PLC 能力,它还具有开放的函数库可供用户进行自主开发,以构造自己所需要的数控系统。这类数控系统具有可靠性高、功能强、性能好、操作简单方便、开发周期短、成本低等优点,而且适合各种类型数控系统的开发,因而这种数控系统目前被广泛应用于制造业自动化控制各个领域。

3) Soft 型开放式数控系统

Soft 型开放式数控系统提供给用户最大的选择空间和灵活性。它的 CNC 软件全部装在计算机中,而硬件部分仅是计算机与伺服驱动和外部 I/O 之间的标准化通用接口。用户可以在 Windows 平台上利用开放的 CNC 内核开发所需要的各种功能,构成各种类型的高性能数控系统。Soft 型开放式数控系统具有较高的性能价格比,因而更具有生命力。

3. 网络化数控系统

网络化数控系统是近两年数控技术发展的一个新亮点。随着计算机技术、网络技术日益普遍运用,数控机床走向网络化、集成化已成为必然的趋势,互联网进入制造工厂的车间也只是时间的问题了。对于面临日益全球化竞争的现代制造工厂来说,不仅要提高数控机床的拥有率,而且拥有的数控机床必须具有联网通信功能,以保证信息流在工厂、车间的底层之间及底层与上层之间的通信。数控系统生产厂商已经推出了具有网络功能的数控系统。在这些数控系统中,除了传统的 RS-232 接口外,还备有以太网接口,为数控机床联网提供了基本条件。目前数控的网络化主要采用以太网以及现场总线的方式。随着无线技术的发展,数控系统网络在不久的将来可能会无处不在。

3.2 运动轨迹插补原理

在数控机床上,各种曲线轮廓加工都是通过插补计算实现的,插补计算的任务就是在轮廓线的起点到终点之间再密集地计算出有限个坐标点,刀具沿着这些坐标点移动,用折线逼近要加工的曲线。进而获得理论轮廓。而确定刀具坐标的过程就称为插补。

插补方法可以分为两大类:脉冲增量插补和数据采样插补。

脉冲增量插补是控制单个脉冲输出规律的插补方法,每输出一个脉冲,移动部件都要相应地移动一定距离,这个距离就是脉冲当量,因此,脉冲增量插补也称为行程标量插补,如逐点比较法、数字积分法。该插补方法通常用于步进电机控制系统。

数据采样插补也称为数字增量插补,是在规定的时间内计算出各坐标方向的增量值、刀具所在的坐标及其他一些需要的值。这些数据严格地限制在一个插补时间内计算完毕,送给伺服系统,再由伺服系统控制移动部件运动,移动部件也必须在下一个插补时间内走完插补计算给出的行程,因此数据采样插补也称为时间标量插补。数据采样插补采用数值量控制机床运动,机床各坐标方向的运动速度与插补运算给出的数值量和插补时间有关。该插补方法用于直流伺服电机和交流伺服电机的闭环或半闭环控制系统。

数控系统中完成插补工作的装置称为插补器。下面主要介绍脉冲增量插补中的逐点比较法插补原理。

3.2.1 逐点比较法的直线插补

所谓逐点比较法插补,就是刀具每走一步都要和给定轨迹上的坐标值进行比较,看这一点在给定轨迹的上方或下方(或给定轨迹的里面或外面),从而决定下一步的进给方向。如果原来在给定轨迹的下方,下一步就向给定轨迹的上方走;如果原来在给定轨迹的里面,下一步就向给定轨迹的外面走。如此走一步比较一次,决定下一步的走向,以便逼近给定轨迹,即形成逐点比较插补。

逐点比较法是以阶梯折线逼近直线或圆弧等曲线的,它与规定的加工直线或圆弧之间的最大误差为一个脉冲当量,因此只要把脉冲当量(每走一步的距离,即步长)设置得足够小,就可达到加工精度的要求。

1. 第一象限内的直线插补

1) 偏差计算公式

根据逐点比较法插补原理,必须把每一插值点(动点)的实际位置与给定轨迹的理想位置间的误差,即偏差计算出来,根据偏差的正、负决定下一步的走向,以逼近给定轨迹,因此偏差计算是逐点比较法关键的一步。

要在第一象限加工出直线段 $\overline{OA}$,取直线段的起点为坐标原点,直线段终点坐标 (x_e, y_e) 是已知的,如图 3-5 所示。点 $m(x_m, y_m)$ 为加工点(动点),若点 m 在直线段 $\overline{OA}$ 上,则有

$$y_m/x_m = y_e/x_e \quad (3-1)$$

即

$$y_m/x_m - y_e/x_e = 0 \quad (3-2)$$

现定义直线插补的偏差判别式为

$$F_m = y_m x_e - x_m y_e \quad (3-3)$$

若 $F_m = 0$,表明点 m 在 $\overline{OA}$ 直线段上;若 $F_m > 0$,表明点 m 在 $\overline{OA}$ 直线段的上方,即点 m' 处;若 $F_m < 0$,表明点 m 在 $\overline{OA}$ 直线段的下方,即点 m'' 处。 F_m 也称为偏差符号。

由此可得第一象限直线逐点比较法插补的原理:从直线的起点(即坐标原点)出发,当 $F_m \geq 0$ 时沿 $+x$ 方向走一步,当 $F_m < 0$ 时沿 $+y$ 方向走一步;当两个方向所走的步数与终点坐标 (x_e, y_e) 相等时,发出终点到信号,停止插补。

按式(3-3)计算偏差,要做两次乘法和一次减法,比较麻烦,因此需要进一步简化。下面推导简化的偏差计算公式。

(1) 设加工点正处于 m 点,当 $F_m \geq 0$ 时,表明 m 点在 $\overline{OA}$ 上或 $\overline{OA}$ 的上方,应沿 $+x$ 方向进给一步至 $m+1$ 点,该点的坐标值为

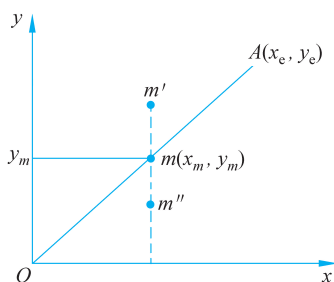


图 3-5 第一象限直线

$$\begin{aligned}x_{m+1} &= x_m + 1 \\y_{m+1} &= y_m\end{aligned}\quad (3-4)$$

该点的偏差为

$$F_{m+1} = y_{m+1}x_e - x_{m+1}y_e = y_mx_e - (x_m + 1)y_e = F_m - y_e \quad (3-5)$$

(2) 设加工点正处于 m 点, 当 $F_m < 0$ 时, 表明 m 点在 $\overline{OA}$ 的下方, 应向 $+y$ 方向进给一步至 $m+1$ 点, 该点的坐标值为

$$\begin{aligned}x_{m+1} &= x_m \\y_{m+1} &= y_m + 1\end{aligned}\quad (3-6)$$

该点的偏差为

$$F_{m+1} = y_{m+1}x_e - x_{m+1}y_e = (y_m + 1)x_e - x_my_e = F_m + x_e \quad (3-7)$$

式(3-5)和式(3-7)是简化后的偏差计算公式, 在两式中只有一次加法或减法运算, 新的加工点的偏差 F_{m+1} 都可以由前一点偏差 F_m 和终点坐标相加或相减得到。特别要注意, 加工的起点是坐标原点, 起点的偏差是已知的, 即 $F_0 = 0$ 。

2) 终点判断方法

逐点比较法的终点判断有多种方法, 下面介绍两种方法:

(1) 设置 N_x 和 N_y 两个减法计数器, 在加工开始前, 在 N_x 和 N_y 计数器中分别存入终点坐标值 x_e 和 y_e , 在 x 坐标(或 y 坐标)进给一步时, 就在 N_x 计数器(或 N_y 计数器)中减去 1, 直到这两个计数器中的数都减到 0 时, 表明到达终点。

(2) 用一个终点计数器保存 x 和 y 两个坐标进给的总步数 N_{xy} , x 或 y 坐标进给一步, N_{xy} 就减 1, 若 $N_{xy} = 0$, 就表明到达终点。

3) 插补计算过程

插补计算时, 每走一步, 都要进行以下插补计算过程: 偏差判别、坐标进给、偏差计算、终点判断。

2.4 个象限的直线插补

4 个象限直线插补的偏差符号及坐标进给方向如图 3-6 所示。

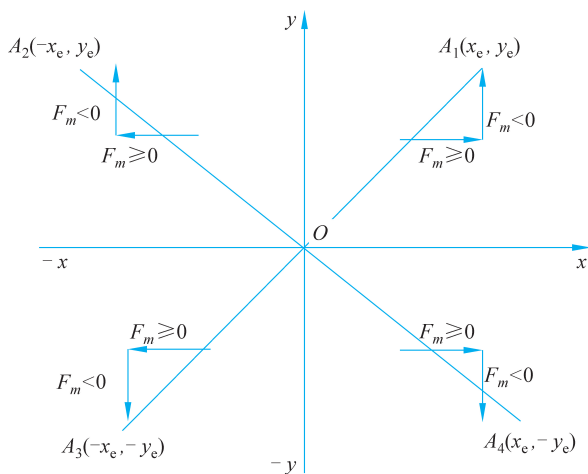


图 3-6 4 个象限直线插补的偏差符号及坐标进给方向

由图 3-6 可以推导出 4 个象限直线插补的偏差计算公式和坐标进给方向,详见表 3-2,其中 4 个象限的终点坐标值取绝对值代入计算式中的 x_e 和 y_e 。

表 3-2 直线插补的进给方向及偏差计算公式

$F_m \geq 0$			$F_m < 0$		
所在象限	进给方向	偏差计算公式	所在象限	进给方向	偏差计算公式
一、四	$+x$	$F_{m+1} = F_m - y_e$	一、二	$+y$	$F_{m+1} = F_m + x_e$
二、三	$-x$		三、四	$-y$	

3. 直线插补计算的程序实现

1) 数据的输入及存放

在计算机的内存中开辟 6 个单元 XE、YE、NXY、FM、XOY 和 ZF,分别存放终点横坐标 x_e 、终点纵坐标 y_e 、总步数 N_{xy} 、加工点偏差 F_m 、直线所在象限值 and 步进方向标志。这里 $N_{xy} = N_x + N_y$,XOY 单元里的值 1、2、3、4 分别表示第一、第二、第三、第四象限,XOY 单元的值可由终点坐标 (x_e, y_e) 的正负号确定, F_m 的初值为 $F_0 = 0$,ZF=1,2,3,4 分别代表 $+x$ 、 $-x$ 、 $+y$ 、 $-y$ 步进的方向。

2) 直线插补计算的程序

图 3-7 为直线插补计算的程序流程图。该程序按照插补计算过程的 4 个步骤(即偏

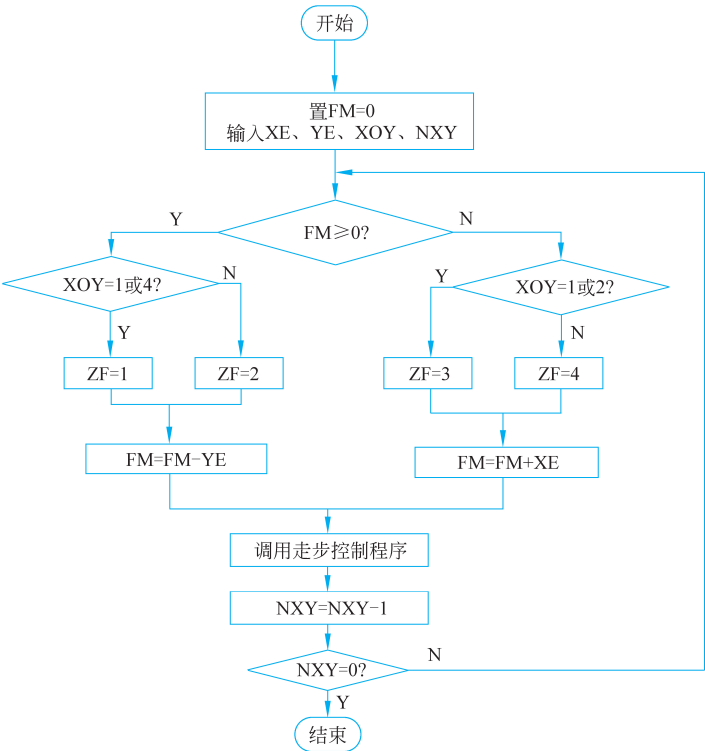


图 3-7 直线插补计算的程序流程图

差判别、坐标进给、偏差计算、终点判断)实现插补计算。偏差判别、偏差计算、终点判断是逻辑运算和算术运算,容易编写程序,而坐标进给通常是给步进电机发送走步脉冲,通过步进电机带动机床工作台或刀具移动。

【例 3-1】 设要加工第一象限直线 $\overline{OA}$,起点为 $O(0,0)$,终点为 $A(6,4)$,试进行插补计算并作出走步轨迹图。

【解】 坐标进给的总步数 $N_{xy}=|6-0|+|4-0|=10$, $x_e=6$, $y_e=4$, $F_0=0$, $XOY=1$ 。插补计算过程如表 3-3 所示。

表 3-3 插补计算过程

步数	偏差判别	坐标进给	偏差计算	终点判断
0(起点)			$F_0=0$	$N_{xy}=10$
1	$F_0=0$	$+x$	$F_1=F_0-y_e=-4$	$N_{xy}=9$
2	$F_1<0$	$+y$	$F_2=F_1+x_e=2$	$N_{xy}=8$
3	$F_2>0$	$+x$	$F_3=F_2-y_e=-2$	$N_{xy}=7$
4	$F_3<0$	$+y$	$F_4=F_3+x_e=4$	$N_{xy}=6$
5	$F_4>0$	$+x$	$F_5=F_4-y_e=0$	$N_{xy}=5$
6	$F_5=0$	$+x$	$F_6=F_5-y_e=-4$	$N_{xy}=4$
7	$F_6<0$	$+y$	$F_7=F_6+x_e=2$	$N_{xy}=3$
8	$F_7>0$	$+x$	$F_8=F_7-y_e=-2$	$N_{xy}=2$
9	$F_8<0$	$+y$	$F_9=F_8+x_e=4$	$N_{xy}=1$
10	$F_9>0$	$+x$	$F_{10}=F_9-y_e=0$	$N_{xy}=0$

直线插补的走步轨迹图如图 3-8 所示。

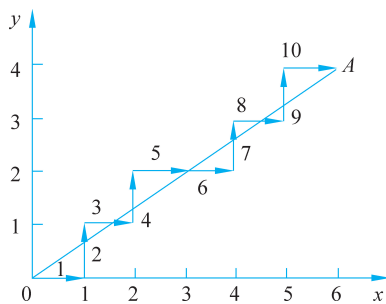


图 3-8 直线插补的走步轨迹图

3.2.2 逐点比较法的圆弧插补

1. 第一象限内的圆弧插补

1) 偏差计算公式

设要加工逆圆弧 $\widehat{AB}$,如图 3-9 所示,圆弧的圆心在坐标原点,并已知圆弧的起点为

$A(x_0, y_0)$, 终点为 $B(x_e, y_e)$, 圆弧半径为 R 。令瞬时加工点为 $m(x_m, y_m)$, 它与圆心的距离为 R_m , 显然, 可以比较 R_m 和 R 来反映加工偏差。比较 R_m 和 R , 实际上是比较它们的平方值。

由第一象限逆圆弧 $\widehat{AB}$ 可知

$$R_m^2 = x_m^2 + y_m^2, \quad R_0^2 = x_0^2 + y_0^2 \quad (3-8)$$

因此, 可定义偏差判别式为

$$F_m = R_m^2 - R^2 = x_m^2 + y_m^2 - R^2 \quad (3-9)$$

若 $F_m = 0$, 表明加工点 m 在圆弧上; 若 $F_m > 0$, 表明加工点在圆弧外; 若 $F_m < 0$, 表明加工点在圆弧内。

由此可得第一象限逆圆弧逐点比较插补的原理: 从圆弧的起点出发, 当 $F_m \geq 0$, 为了逼近圆弧, 下一步向 $-x$ 方向进给一步, 并计算新的偏差; 若 $F_m < 0$, 为了逼近圆弧, 下一步向 $+y$ 方向进给一步, 并计算新的偏差。如此一步步计算和一步步进给, 并在到达终点后停止计算, 就可插补出图 3-9 所示的第一象限逆圆弧。

为了简化偏差判别式(3-9)的计算, 下面推导出简化的偏差计算的递推公式。

(1) 设加工点正处于 $m(x_m, y_m)$ 点, 当 $F_m \geq 0$ 时, 应沿 $-x$ 方向进给一步至 $m+1$ 点, 其坐标值为

$$\begin{aligned} x_{m+1} &= x_m - 1 \\ y_{m+1} &= y_m \end{aligned} \quad (3-10)$$

新的加工点的偏差为

$$\begin{aligned} F_{m+1} &= x_{m+1}^2 + y_{m+1}^2 - R^2 = (x_m - 1)^2 + y_m^2 - R^2 \\ &= F_m - 2x_m + 1 \end{aligned} \quad (3-11)$$

(2) 设加工点正处于 $m(x_m, y_m)$ 点, 当 $F_m < 0$ 时, 应沿 $+y$ 方向进给一步至 $m+1$ 点, 其坐标值为

$$\begin{aligned} x_{m+1} &= x_m \\ y_{m+1} &= y_m + 1 \end{aligned} \quad (3-12)$$

新的加工点偏差为

$$\begin{aligned} F_{m+1} &= x_{m+1}^2 + y_{m+1}^2 - R^2 = x_m^2 + (y_m + 1)^2 - R^2 \\ &= F_m + 2y_m + 1 \end{aligned} \quad (3-13)$$

由式(3-11)和式(3-13)可知, 只要知道前一点的偏差和坐标值, 就可求出新一点的偏差。因为加工点从圆弧的起点开始, 故起点的偏差 $F_0 = 0$ 。

2) 终点判断方法

圆弧插补的终点判断方法和直线插补相同。可将 x 方向的走步步数 $N_x = |x_e - x_0|$ 和 y 方向的走步步数 $N_y = |y_e - y_0|$ 的总和 N_{xy} 作为一个计数器, 每走一步, 从 N_{xy} 中减 1, 当 $N_{xy} = 0$ 时发出终点到信号。

3) 插补计算过程

圆弧插补计算过程比直线插补计算过程多一个环节, 即要计算加工点瞬时坐标(动

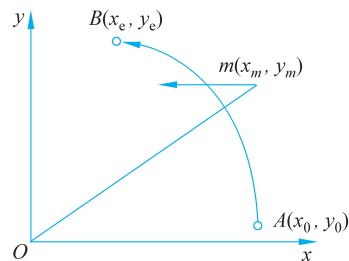


图 3-9 第一象限逆圆弧

点坐标)值,其计算公式为式(3-10)和式(3-12)。因此圆弧插补计算过程分为 5 个步骤,即偏差判别、坐标进给、偏差计算、坐标计算、终点判断。

2. 4 个象限的圆弧插补

其他各象限中逆、顺圆弧都可以参照第一象限得出各自的偏差计算公式及进给方向。前面介绍了第一象限逆圆弧的插补计算,为了导出其他各象限的插补计算,下面先来推导第一象限顺圆弧的偏差计算公式。

1) 第一象限顺圆弧的插补计算

设第一象限顺圆弧 $\widehat{CD}$,圆弧的圆心在坐标原点,并已知起点 $C(x_0, y_0)$,终点 $D(x_e, y_e)$,如图 3-10 所示。设加工点现处于 $m(x_m, y_m)$ 点,若 $F_m \geq 0$,则沿 $-y$ 方向进给一步,到 $m+1$ 点,新加工点坐标将是 $(x_m, y_m - 1)$,可求出新的偏差为

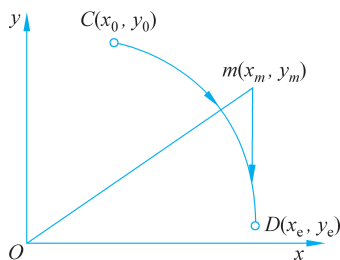


图 3-10 第一象限顺圆弧

$$F_{m+1} = F_m - 2y_m + 1 \quad (3-14)$$

若 $F_m < 0$,则沿 $+x$ 方向进给一步至 $m+1$ 点,新加工点的坐标将是 $(x_m + 1, y_m)$,同样可求出新的偏差为

$$F_{m+1} = F_m + 2x_m + 1 \quad (3-15)$$

2) 4 个象限的圆弧插补

式(3-11)、式(3-13)~式(3-15)给出了第一象限逆、顺圆弧的插补计算公式,其他象限的圆弧插补可与第一象限的情况相比较而得出,因为其他象限的所有圆弧总是与第一象限中的逆圆弧或顺圆弧互为对称,如图 3-11 所示。在图 3-11 中,用 SR 和 NR 分别表示顺圆弧和逆圆弧,所以可用 SR_1 、 SR_2 、 SR_3 、 SR_4 和 NR_1 、 NR_2 、 NR_3 、 NR_4 8 种圆弧分别表示第一至第四象限的顺圆弧和逆圆弧。

如图 3-11 所示, SR_1 与 NR_2 对称于 $+y$ 轴, SR_3 与 NR_4 对称于 $-y$ 轴, NR_1 与 SR_4 对称于 $+x$ 轴, NR_3 与 SR_2 对称于 $-x$ 轴, SR_1 与 NR_4 对称于 $+x$ 轴, SR_3 与 NR_2 对称于 $-x$ 轴, SR_2 与 NR_1 对称于 $+y$ 轴, SR_4 与 NR_3 对称于 $-y$ 轴。4 个象限中的圆弧插补的进给方向、偏差计算公式及坐标计算公式如表 3-4 所示。这里,同图 3-6 所示的直线插补一样,将 4 个象限的起点坐标值、终点坐标值和加工点坐标值取绝对值后,再代入偏差计算和坐标计算公式。

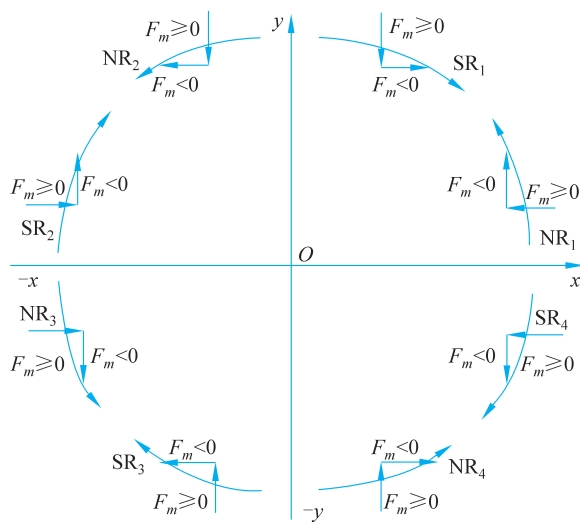


图 3-11 4 个象限圆弧插补的对称关系

表 3-4 圆弧插补的进给方向、偏差计算公式及坐标计算公式

偏差	圆弧种类	进给方向	偏差计算公式	坐标计算公式
$F_m \geq 0$	SR_1, NR_2	$-y$	$F_{m+1} = F_m - 2y_m + 1$	$x_{m+1} = x_m$
	SR_3, NR_4	$+y$		$y_{m+1} = y_m - 1$
	NR_1, SR_4	$-x$	$F_{m+1} = F_m - 2x_m + 1$	$x_{m+1} = x_m - 1$
	NR_3, SR_2	$+x$		$y_{m+1} = y_m$
$F_m < 0$	SR_1, NR_4	$+x$	$F_{m+1} = F_m + 2x_m + 1$	$x_{m+1} = x_m + 1$
	SR_3, NR_2	$-x$		$y_{m+1} = y_m$
	NR_1, SR_2	$+y$	$F_{m+1} = F_m + 2y_m + 1$	$x_{m+1} = x_m$
	NR_3, SR_4	$-y$		$y_{m+1} = y_m + 1$

3. 圆弧插补计算的程序实现

1) 数据的输入及存放

在计算机的内存中开辟 8 个单元 X0、Y0、NX_Y、FM、RNS、XM、YM 和 ZF,分别存放起点的横坐标 x_0 、起点的纵坐标 y_0 、总步数 N_{xy} 、加工点偏差 F_m 、圆弧种类值 RNS、 x_m 、 y_m 和走步方向标志。这里 $N_{xy} = |x_e - x_0| + |y_e - y_0|$,RNS 等于 1,2,3,4 和 5,6,7,8,分别代表 SR_1 、 SR_2 、 SR_3 、 SR_4 和 NR_1 、 NR_2 、 NR_3 、 NR_4 ,RNS 的值可由起点和终点的坐标的正负符号确定, F_m 的初值为 $F_0 = 0$, x_m 和 y_m 的初值为 x_0 和 y_0 ,ZF=1,2,3,4 分别表示 $+x$ 、 $-x$ 、 $+y$ 、 $-y$ 走步方向。

2) 圆弧插补计算的程序

图 3-12 为圆弧插补计算的程序流程图,该程序按照插补计算的 5 个步骤(即偏差判

别、坐标进给、偏差计算、坐标计算、终点判断)实现插补计算。

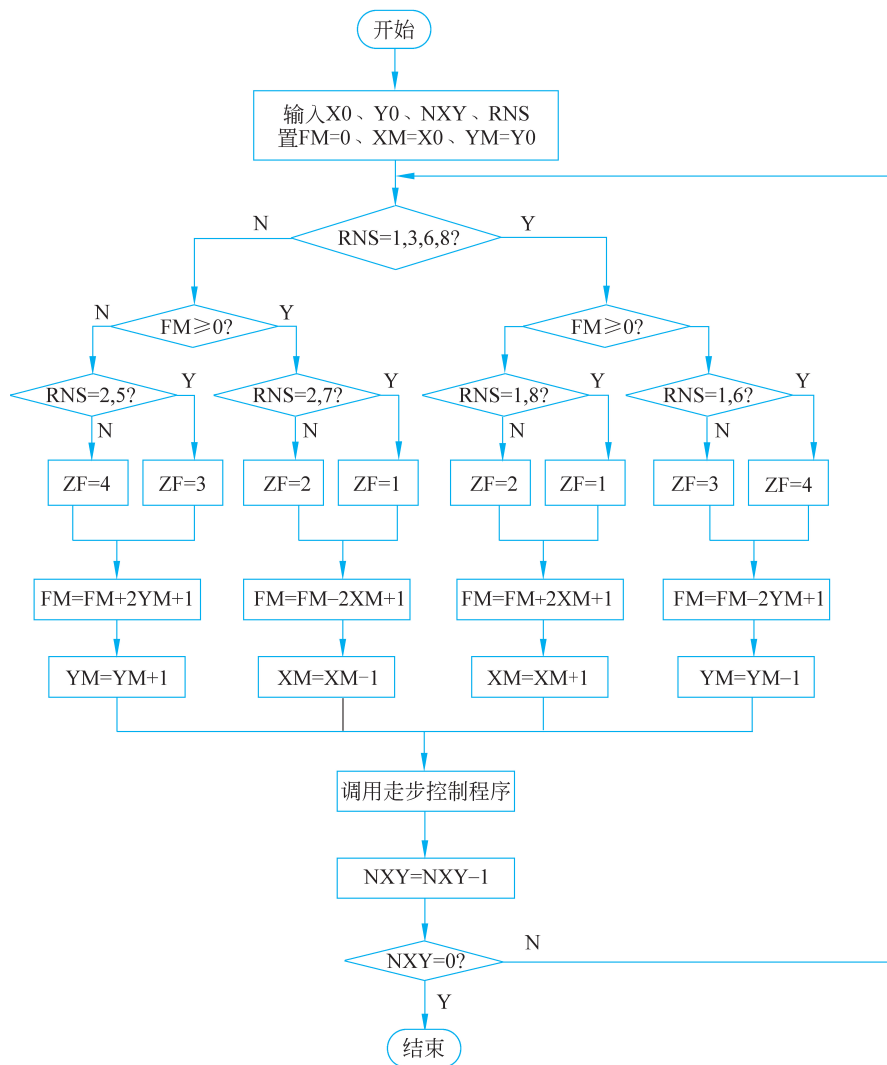


图 3-12 4 象限圆弧插补程序流程图

【例 3-2】 设要加工第一象限逆圆弧 $\widehat{AB}$, 已知起点为 $A(4,0)$, 终点为 $B(0,4)$, 试进行插补计算并作出走步轨迹图。

【解】 圆弧插补计算过程如表 3-5 所示。

表 3-5 圆弧插补计算过程

步数	偏差判别	坐标进给	偏差计算	坐标计算	终点判断
0(起点)			$F_0 = 0$	$x_0 = 4, y_0 = 0$	$N_{xy} = 8$
1	$F_0 = 0$	$-x$	$F_1 = F_0 - 2x_0 + 1 = -7$	$x_1 = x_0 - 1 = 3, y_1 = 0$	$N_{xy} = 7$
2	$F_1 < 0$	$+y$	$F_2 = F_1 + 2y_1 + 1 = -6$	$x_2 = 3, y_2 = y_1 + 1 = 1$	$N_{xy} = 6$

续表

步数	偏差判别	坐标进给	偏差计算	坐标计算	终点判断
3	$F_2 < 0$	+y	$F_3 = F_2 + 2y_2 + 1 = -3$	$x_3 = 3, y_3 = y_2 + 1 = 2$	$N_{xy} = 5$
4	$F_3 < 0$	+y	$F_4 = F_3 + 2y_3 + 1 = 2$	$x_4 = 3, y_4 = y_3 + 1 = 3$	$N_{xy} = 4$
5	$F_4 > 0$	-x	$F_5 = F_4 - 2x_4 + 1 = -3$	$x_5 = x_4 - 1 = 2, y_5 = 3$	$N_{xy} = 3$
6	$F_5 < 0$	+y	$F_6 = F_5 + 2y_5 + 1 = 4$	$x_6 = 2, y_6 = y_5 + 1 = 4$	$N_{xy} = 2$
7	$F_6 > 0$	-x	$F_7 = F_6 - 2x_6 + 1 = 1$	$x_7 = x_6 - 1 = 1, y_7 = 4$	$N_{xy} = 1$
8	$F_7 > 0$	-x	$F_8 = F_7 - 2x_7 + 1 = 0$	$x_8 = x_7 - 1 = 0, y_8 = 4$	$N_{xy} = 0$

根据表 3-5 绘制出的走步轨迹图如图 3-13 所示。

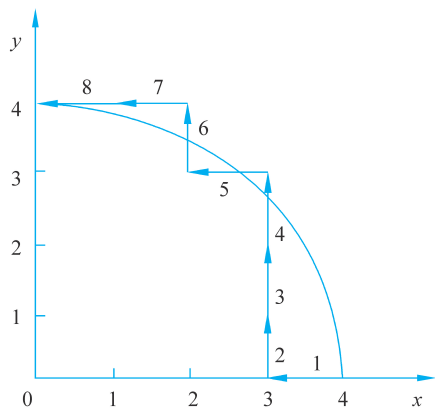


图 3-13 圆弧插补走步轨迹图

3.2.3 数字积分法插补

数字积分法也称数字微分分析器(Digital Differential Analyzer, DDA)法,是根据数学中的积分几何概念,将函数的积分运算变成变量的求和运算,求积分的过程是用数的累加近似的,如果脉冲当量足够小,则用求和运算代替积分运算,由此引起的误差可以控制在容许的范围内。用数字积分法可以实现一次、二次及高次曲线插补,脉冲分配均匀,容易实现多坐标联动控制,因此应用广泛。

1. 数字积分法原理

设函数曲线 $y = f(t)$, $t_0 \leq t \leq t_n$, 曲线 y 、横轴 t 在区间 $[t_0, t_n]$ 上所包围的面积为

$$S = \int_{t_0}^{t_n} y dt \tag{3-16}$$

当将 $t_0 \sim t_n$ 划分成 n 个间隔 Δt 的区间时,可用数的累加求此面积:

$$S = \lim_{n \rightarrow \infty} \sum_{i=1}^n y_i \Delta t \tag{3-17}$$

于是,求此面积的近似矩形公式为

$$S \approx \sum_{i=1}^n y_i \Delta t \quad (3-18)$$

当 Δt 取得足够小时,就可求出符合所需精度的面积。数字积分法就是基于此原理而实现的一种插补算法。

对于一个脉冲当量,若 Δt 取为一个最小单位时间(一个脉冲周期时间),即

$$\Delta t = 1 \quad (3-19)$$

则

$$S \approx \sum_{i=1}^n y_i \quad (3-20)$$

2. 数字积分法的直线插补

数字积分法的工作原理可用加工图 3-14 所示的第一象限中的线段 AB 加以说明,起点为 $A(x_0, y_0)$,终点为 $B(x_e, y_e)$ 。

线段 AB 的方程为

$$y = \frac{y_e - y_0}{x_e - x_0}(x - x_0) + y_0 \quad (3-21)$$

对式(3-21)微分得

$$\frac{dy}{dx} = \frac{dy/dt}{dx/dt} = \frac{y_e - y_0}{x_e - x_0} \quad (3-22)$$

将 x, y 化为对时间 t 的参数方程,则有

$$\begin{cases} \frac{dx}{dt} = K(x_e - x_0) \\ \frac{dy}{dt} = K(y_e - y_0) \end{cases} \quad (3-23)$$

式(3-23)中, K 为比例系数。

对式(3-23)中每个方程的两边进行积分,得

$$\begin{cases} x(t) = \int_{t_0}^t K(x_e - x_0) dt + x_0 \\ y(t) = \int_{t_0}^t K(y_e - y_0) dt + y_0 \end{cases} \quad (3-24)$$

式(3-24)中, t_0 为线段起点 A 的时间, t_n 为线段终点 B 的时间, $t_0 \leq t \leq t_n$ 。于是,有

$$\begin{cases} x(t_0) = x_0, & x(t_n) = x_e \\ y(t_0) = y_0, & y(t_n) = y_e \end{cases} \quad (3-25)$$

因为积分是从线段起点 A 开始的,所以坐标增量就是线段终点坐标 B,即

$$\begin{cases} \int_{t_0}^{t_n} K(x_e - x_0) dt = x_e - x_0 \\ \int_{t_0}^{t_n} K(y_e - y_0) dt = y_e - y_0 \end{cases} \quad (3-26)$$

若将 $t_0 \sim t_n$ 平均分成 n 份,每一份为一个插补周期 Δt ,则当 Δt 足够小($dt \approx \Delta t$)时,在 $t = t_k = k \cdot \Delta t$ ($k = 1, 2, \dots, n$) 时式(3-24)的积分可用累加代替,即

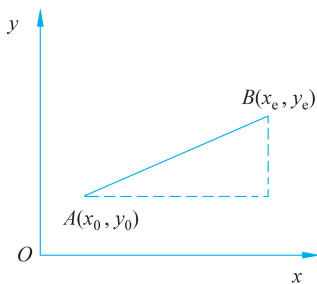


图 3-14 数字积分法直线插补

$$\begin{cases} x(t_k) = \sum_{i=1}^k K(x_e - x_0)\Delta t + x_0 \\ y(t_k) = \sum_{i=1}^k K(y_e - y_0)\Delta t + y_0 \end{cases} \quad (3-27)$$

式(3-27)表明,若数字积分用一个累加器进行,且累加器的溢出脉冲由计数器计数,则积分完成后计数器中所存的数就是坐标差 $(x_e - x_0)$ 和 $(y_e - y_0)$ 。

从式(3-27)可以看出,可以用两个累加器完成直线的插补计算,它们分别对 $(x_e - x_0)$ 或 $(y_e - y_0)$ 进行累加,累加器每溢出一个脉冲,就表明相应的坐标值变化了一个脉冲当量,即 $\Delta t = 1$ 。若将累加器的溢出脉冲送到数控系统进给部分的执行机构,每当累加器溢出一个脉冲就在相应的坐标进给一步,数控系统进给运动的轨迹就是接近 AB 的一条直线。当积分到终点 B 时, x 轴和 y 轴所走的总步数就恰好等于坐标差 $(x_e - x_0)$ 和 $(y_e - y_0)$ 。

当累加 n 次后,加工动点就到达了终点,插补结束,此时

$$\begin{cases} x(t_n) = \sum_{i=1}^n K(x_e - x_0)\Delta t + x_0 = nK(x_e - x_0) + x_0 = x_e \\ y(t_n) = \sum_{i=1}^n K(y_e - y_0)\Delta t + y_0 = nK(y_e - y_0) + y_0 = y_e \end{cases} \quad (3-28)$$

使式(3-28)成立的条件为

$$n = \frac{1}{K} \quad (3-29)$$

K 的数值与累加器的容量有关,它的值应为

$$K = \frac{1}{2^N} \quad (3-30)$$

式(3-30)中, N 为累加器的位数。于是,累加次数为

$$n = 2^N \quad (3-31)$$

式(3-31)表明,若累加器的位数为 N ,则整个插补过程要进行 2^N 次累加才能到达线段的终点 B 。

图3-15是数字积分直线插补器框图,它由被积函数寄存器 J_x 、 J_y ,累加器 J_{Rx} 、 J_{Ry} 及全加器 J_Σ 组成。

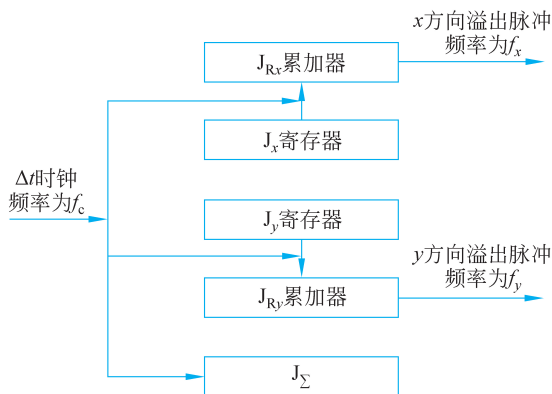


图 3-15 数字积分直线插补器框图

首先把 $(x_e - x_0)$ 、 $(y_e - y_0)$ 的值初始化,即把以毫米为单位的数值除以脉冲当量,变成脉冲数字量。然后把 $(x_e - x_0)$ 的值输入 J_x 寄存器,把 $(y_e - y_0)$ 的值输入 J_y 寄存器。每出现一个时钟脉冲, J_x 中的内容要进入 J_{Rx} 累加器中累加, J_y 中的内容要进入 J_{Ry} 累加器中累加,在从A点到B点的加工过程中,要给出 n 个时钟脉冲, J_{Rx} 、 J_{Ry} 累加器要进行 n 次累加,在 J_{Rx} 溢出端要溢出 $(x_e - x_0)$ 个脉冲,在 J_{Ry} 溢出端要溢出 $(y_e - y_0)$ 个脉冲。对每一个时钟脉冲,全加器 J_Σ 都要做加1运算,当 $J_\Sigma = n$ 时,加工结束。

由于累加器的累加过程类似于积分过程,因此称这种插补方法为数字积分法, J_x 、 J_y 中的数也称被积函数。

从上述数字积分直线插补器的工作过程可以看出,数字积分直线插补器必须满足下述要求才能正确工作:

(1) 输入的 n 个时钟脉冲必须使 J_{Rx} 端溢出 $(x_e - x_0)$ 个脉冲,使 J_{Ry} 溢出 $(y_e - y_0)$ 个脉冲。

(2) J_{Rx} 溢出脉冲的速度与 J_{Ry} 溢出脉冲的速度之比必须等于 $(x_e - x_0)/(y_e - y_0)$,才能保证加工轨迹与线段AB吻合。

上述的 n 值是累加器 J_{Rx} 、 J_{Ry} 和寄存器 J_x 、 J_y 有效位的最大值。若寄存器为16位, $(x_e - x_0)$ 的数值(二进制)的字长为8位,有效位是8位, $(y_e - y_0)$ 的字长为6位,有效位也是8位,则 $n = 2^8$ 。有效位的位长是两个坐标方向的坐标值大者的二进制位长,且各寄存器都相同。寄存器 J_x 、 J_y 的位长通常与累加器 J_{Rx} 、 J_{Ry} 的位长相等,它所用的有效位也与累加器的有效位相等。由于被加工线段AB的长短不同,所以累加器的有效位长也随之变化。加工线段的最大长度是它在各坐标轴上投影值中的大者,等于寄存器的最大容量。例如,对16位的累加器和寄存器,它的最大容量是 $2^{16} - 1$,最多能输出65 535个脉冲,若脉冲当量为0.001mm,则 $(x_e - x_0)$ 或 $(y_e - y_0)$ 中的大者不能大于65.535mm。

若 $(x_e - x_0) = 101\text{B}$ 、 $(y_e - y_0) = 10\text{B}$,则它们占用寄存器和累加器中的3位,有效位为3位, $n = 2^3 = 8$,当累加器作8次累加后,在 J_{Rx} 端必定溢出5个脉冲,在 J_{Ry} 端溢出2个脉冲。若3位初始值都为0的二进制数每次都作加1计算,则加到第8次时,必然进位到第4位,而前3位仍然是0。由于101B是1的5倍,010B是1的2倍,它们分别与初始值为0的数累加8次后必然从第4位分别溢出5个1和2个1,因此,对于 $(x_e - x_0)$ 和 $(y_e - y_0)$ 的任何数值,在经过 n 次累加后,从高位溢出1的数量必然等于它们本身的数值。

x 方向溢出的频率 f_x 等于时钟脉冲频率 f_c 与 $\frac{n}{x_e - x_0}$ 之积:

$$f_x = \frac{f_c n}{x_e - x_0} \quad (3-32)$$

y 方向的溢出频率 f_y 等于时钟脉冲频率 f_c 与 $\frac{n}{y_e - y_0}$ 之积:

$$f_y = \frac{f_c n}{y_e - y_0} \quad (3-33)$$

x 、 y 方向的溢出速度分别为 $v_x = 1/f_x$ 、 $v_y = 1/f_y$,则

$$\frac{v_y}{v_x} = \frac{y_e - y_0}{x_e - x_0} \quad (3-34)$$

由式(3-34)可以看出,累加器溢出脉冲的速度与其中的数值成正比。

在累加器累加之前,需把 J_x 、 J_y 中的被积函数 $(x_e - x_0)$ 、 $(y_e - y_0)$ 左移,进行规格化处理,由于被积函数的位长常常小于 J_x 、 J_y 寄存器的位长,只有有效位处于累加器的最左端(高位)时,才能及时溢出脉冲, n 次累加后才能完成插补工作。例如,若 J_x 、 J_y 、 J_{Rx} 、 J_{Ry} 是 8 位, $(x_e - x_0) = 101\text{B}$, $(y_e - y_0) = 10\text{B}$, J_{Rx} 、 J_{Ry} 中的有效位为 3 位,若没有作左移规格化处理,则 J_x 、 J_y 中的数据情况如图 3-16(a)所示,这样的数据需进行 2^N (N 为寄存器位数)累加才能溢出正确的脉冲数。若 $N = 8$,需累加 $2^8 = 256$ 次。在这 256 次累加中, $(2^8 - 2^3) = 248$ 次累加溢出为 0,只有最后的 2^3 次累加才能溢出需要的脉冲,这显然是不行的。左移规格化处理是把有效位移到被积函数寄存器的最左端,使它们占据高位。如图 3-16(b)所示,由于被积函数处在高位端,因而累加 n 次后就能完成积分插补计算。左移的位数 M 等于寄存器位数 N 减去被积函数有效位数 N_1 ,即 $M = N - N_1$ 。对其他象限的直线加工,可通过对电动机转向的控制实现。

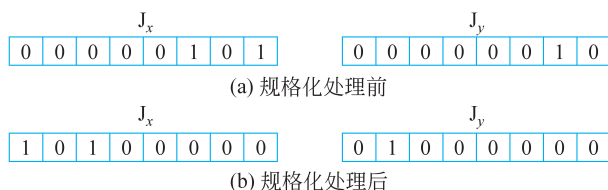


图 3-16 左移规格化处理前后 J_x 、 J_y 中的数据

3. 数字积分法的圆弧插补

假设在 xy 平面内加工第一象限逆圆弧 $\widehat{AB}$,如图 3-17 所示,圆心为 O' , A 点相对于圆心 O' 的坐标为 I_A 、 J_A 。 P_i 点为圆上任意一点,相对于 O' 的坐标为 I_i 、 J_i , t 为 P_i 点相对于横坐标轴的旋转角度。圆的方程为

$$I^2 + J^2 = R^2 \quad (3-35)$$

参数方程为

$$\begin{cases} I_i = R \cos t \\ J_i = R \sin t \end{cases} \quad (3-36)$$

在 P_i 点刀具沿各坐标方向的速度分量为

$$\begin{aligned} v_x &= \frac{dI_i}{dt} = -R \sin t = -J_i \\ v_y &= \frac{dJ_i}{dt} = R \cos t = I_i \end{aligned} \quad (3-37)$$

由式(3-37)可以看出:圆弧上任意一点在 x 方向的速度分量的绝对值 $|v_x|$ 等于该点在 y 方向的瞬时坐标值 J_i ,在 y 方向的速度分量 v_y 等于该点在 x 方向的瞬时坐标值 I_i , v_x 、 v_y 是随 P_i 点的位置变化而变化的。

$$\begin{aligned} dI_i &= -J_i dt \\ dJ_i &= I_i dt \end{aligned} \quad (3-38)$$

用累加和代替积分,得

$$\begin{aligned} I_i &= \sum_{i=1}^m -J_i \Delta t \\ J_i &= \sum_{i=1}^m I_i \Delta t \end{aligned} \quad (3-39)$$

式(3-39)中, m 为从 A 点到 P_i 点的累加次数。

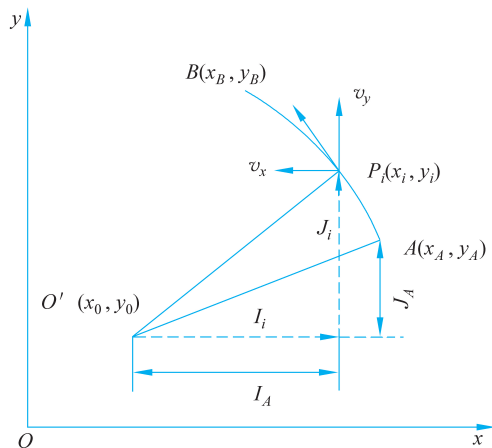


图 3-17 数字积分法圆弧插补

图 3-18 是圆弧插补器原理。在插补运算开始前,在 J_x 寄存器中装入圆弧起点 A 相对于圆心 O' 的 y 向增量值 J_A ,而在 J_y 寄存器中装入 I_A 值。 J_{Rx} 、 J_{Ry} 中清零。插补开始时,使时钟脉冲有效,对应每一个时钟脉冲, J_x 、 J_y 中的被积函数都要对应进入 J_{Rx} 、 J_{Ry} 中累加。当累加器 J_{Rx} 有脉冲溢出时,要在 J_y 寄存器中减 1;而当累加器 J_{Ry} 中有脉冲溢出时,需在寄存器 J_x 中加 1。这是因为 x 方向溢出脉冲使 x 方向步进电机反转一步,在 x 方向减小一个脉冲当量值,使 I_i 比 I_{i-1} 小 1,因而, J_x 中的值要减 1。同理,当 y 方向的步进电机正向走一步时,应使 J_y 加 1。

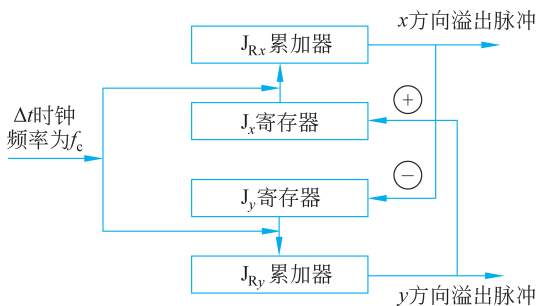


图 3-18 圆弧插补器原理