

## 摘 要

本文针对国内电火花线切割加工的现状,结合目前的计算机网络技术、数据库技术、DNC 技术、数控技术和计算机通信等技术,提出了电火花线切割 DNC 加工系统,并对电火花线切割 DNC 加工系统的体系结构、功能模块以及在开发过程中所采用的关键技术原理作了详细的介绍和分析。目前,该系统业已实施并应用于生产教研实践中,实践证明,实施本系统后,线切割加工车间的资源得到优化配置和合理利用,线切割加工的自动化水平和加工效率也大大提高。本文对电火花线切割 DNC 加工系统进行了有益的探索,全文内容如下:

第一章介绍了电火花线切割加工的原理和应用范围,分析了线切割加工的现状和存在的问题,联系当前国内外 DNC 的研究现状,阐述了开发电火花线切割 DNC 加工系统的重要意义。

第二章在提出电火花线切割 DNC 加工系统的基础上,进一步介绍了系统的体系结构、特点和功能,并且分析了实现系统各个重要模块实现的关键技术。

第三章研究了电火花线切割 DNC 加工系统自动化编程系统模块的关键技术。详细分析了参数化生成加工路线的方法、钼丝轨迹的后置处理、指令代码的仿真、二维创新设计图的矢量化处理。

第四章研究了电火花线切割 DNC 加工系统通讯模块的关键技术。对异构线切割机 DNC 通信的实现和多串口通信解决方案进行了详细的讨论。

第五章对整个系统软件进行了总体设计,并对功能模块进行了划分。在此基础上,介绍了 AutoCAD 环境下利用 VC++ 和 ObjectARX 进行系统自动编程模块软件开发的过程。

第六章介绍了电火花线切割 DNC 加工系统实施的例子,浙江大学电火花线切割二维创新设计 DNC 平台的实现,进一步证明了电火花线切割 DNC 加工系统研究的使用价值。

第七章总结了论文工作的内容,指出了以后需要进一步解决的问题。

**关键词:** 电火花线切割加工 DNC 自动编程系统

## ABSTRACT

Aiming at solving the problems in domestic Wire-EDM's existing situation, the thesis puts forward an approach of the machining system of Wire-EDM DNC, on base of the technology of computer network, database, DNC, NC and computer communications at present. The thesis introduces the framework, function module of the machining system of Wire-EDM DNC and analyse the key technique & principle in developing the system. The system has been verified to be a good method of improving automation in WEDM in practice. Presently. The whole thesis as follows:

In chapter 1, the principle and application area of WEDM is introduced and the problems in existing situation are described. Based the research in existing situation, the importance of implementing the machining system of Wire-EDM DNC.

In chapter 2, the framework, characteristic and function of the machining system of Wire-EDM DNC is introduced and some key technology in realizing the system is also described.

In chapter 3, some key technique used in realizing the module of wire automation programming system is discussed. The method of building machining path automatically, disposing of track, displaying the cutting track of NC code and capturing vectograph from digital image is also researched in detail.

In chapter 4, the key technique using in developing communication module is discussed. Then an approach of hetero-structure WEDM NC system communication and multiple serial port communication is put forward.

In chapter 5, the makeup and function of the system software is designed firstly. Then the process of developing automatic programming system of WEDM is presented. The system is developed using the tool of VC++ and ObjectARX in the platform of AutoCAD. Last, some arithmetic discussed in the above paragraphs is realized.

In chapter 6, the machining system of WEDM DNC discussed above paragraphs is put in practice as the platform of 2D innovative design in WEDM of Zhejiang University. It has been verified that the system is a good method of farthest using machine tools and improve the produce efficiency distinctly.

In chapter 7, the work of the dissertation is summarized and the further research work is prospected.

**Key words:** Wire Electron Discharge Machining, direct control, distributive number control, automatic programming system

# 第一章 绪 论

## 1.1 论文研究的背景

### 1.1.1 电火花线切割加工的原理和应用范围

电火花线切割加工 (Wire EDM, 简称 WEDM) 是在电火花加工基础上于 50 年代末最早在前苏联发展起来的一种新的工艺形式, 是用线状电极 (钼丝或铜丝) 靠火花放电对工件进行切割, 故称为电火花线切割, 有时简称线切割。它已获得广泛的应用, 目前国内外的线切割机床已占电加工机床的 60% 以上。

#### 一、线切割加工的原理<sup>[1]</sup>

电火花线切割加工的基本原理是利用移动的细金属导线 (铜丝或钼丝) 作电极, 对工件进行脉冲火花放电、切割成形。

根据电极丝的运行速度, 电火花线切割机床通常分为两大类: 一类是高速走丝电火花线切割机床 (WEDM-HS), 这类机床的电极丝作高速往复运动, 一般走丝速度为 8~10m/s, 这是我国生产和使用的主要机种, 也是我国独创的电火花线切割加工模式; 另一类是低速走丝电火花线切割机床 (WEDM-LS), 这类机床的电极丝作低速单向运动, 一般走丝速度低于 0.2 m/s, 这是国外生产和使用的主要机种。

快走丝电火花线切割加工过程大致如下: 利用细钼丝作工具电极进行切割, 贮丝筒使钼丝作正反向交替移动, 加工能源由脉冲电源供给。在电极丝和工件之间注入工作液介质, 工作台在水平面两个坐标方向各自按预定的控制程序, 根据火花间隙状态作伺服进给移动, 从而合成各种曲线轨迹, 把工件切割成形。

此外, 电火花线切割机床按控制方式可分为: 靠模仿型控制、光电跟踪控制、数字程序控制等; 按加工尺寸范围可分为: 大、中、小型以及普通型与专用型等。目前国内外 95% 以上的线切割机床都已采用数控化, 而且采用不同水平的微机数控系统, 从单片机、单板机到微型计算机系统, 有的还有自动编程功能。

#### 二、线切割加工的应用范围<sup>[1][2][3]</sup>

线切割加工为新产品试制、精密零件加工特别是模具制造开辟了一条新的工艺途径, 主要应用于以下几个方面。

(1) 加工模具 适用于各种形状的冲模。调整不同的间隙补偿量, 只需一次编程就可以切割凸模、凸模固定板、凹模及卸料板等。模具配合间隙、加工精度通常都能达到要求。此外还可加工挤压模、粉末冶金模、弯曲模、塑压模等通常带锥度的模具。

(2) 加工电火花成形加工用的电极 一般穿孔加工用的电极以及带锥度型腔加工用的电

极, 以及铜钨、银钨合金之类的电极材料, 用线切割加工特别经济, 同时也适用于加工微细复杂形状的电极。

(3) 加工零件 在试制新产品时, 用线切割在坯料上直接割出零件, 例如试制切割特殊微电机硅钢片定转子铁心, 由于不需另行制造模具, 可大大缩短制造周期、降低成本。另外修改设计、变更加工程序比较方便, 加工薄件时还可多片叠在一起加工。在零件制造方面, 可用于加工品种多, 数量少的零件, 特殊难加工材料的零件, 材料试验样件, 各种型孔、特殊凸轮、样板、成型刀具。同时还可进行微细加工, 异形槽、窄槽和标准缺陷的加工等。

### 1.1.2 电火花线切割加工的现状和问题

#### 一、电火花线切割加工的现状

电火花切割加工自诞生以来, 获得了迅速的发展。特别是近十年来, 由于数控技术、脉冲电源、机床设计等方面的不断进步, 线切割机的加工功能及加工指标均比以前有较大的提高。在控制技术的进步方面, 自适应控制、自动穿丝等方面的研究使得长时间的无人操作成为可能。功能较强的自动编程软件大大提高了编程效率, 使得各种复杂图形的零件编程成为可能。

近年来随着模具工业的飞速发展。在机械、电子、电机、仪表、汽车等大中型企业中都有十几以至数十台的线切割机用来加工模具和零件。模具零件, 特别是模具关键件(模具成形部分)很大部分是由线切割完成的, 其工作量很大。特别是近年来在沿海地区的城镇涌现出一支数万人的个体线切割加工专业户, 每户都有 3~4 台以上的线切割机。但线切割机的数控系统仍然是十年前的 TP801 单板机, 它是一种 2M 主频的 8 位微机, 容量小, 抗干扰差, 人机界面差。虽然线切割加工的自动编程方法取得了很大的进步, 很多用户已使用 286 和 386 计算机, 用对话方式或图形方式进行自动编程。可是编好的程序输入却仍然使用纸带方式, 有些用户还是靠单板机键盘手工输入, 既要占用大量的机时和人工, 又很容易出错。另一方面虽然大多数工厂拥有很多台的线切割机, 而线切割机的操作时间一般仅占加工时间的 10% 以下, 却仍然是一人一机, 编程作业需专门的人员。造成大量的人力浪费。随着企业改革的深化, 很多单位要求一人操作多机<sup>[5]</sup>。

电火花线切割机的数控系统已日臻完美。但目前仍有大量线切割机是一台配一套控制系统, 自动编程程度低, 特别是早期的控制系统, 用单板机控制, 多用手工编程或计算机编程后用手工或穿孔带输入。较先进的线切割机配备了微机, 利用微机较强大的图形和数据处理功能进行绘图、编程乃至控制。即使如此, 大多也是独台使用。这对大量生产制造模具这种产品(多品种、小批量)的制造系统来说, 反应慢、效率低, 与要求高效的模具制造系统不相适应<sup>[6]</sup>。

在规模稍大的模具制造厂(车间), 需要线切割加工的零件数量很大, 往往配置了几台, 甚至几十台各种规格的线切割机。因而渐渐地出现了群控系统的概念, 但其仅仅是把自动编

程系统与线切割机简单连接,只是节省了编程和输入的工作量<sup>[6]</sup>。更先进一点的是微机跟线切割控制器之间采用了串口切换器,这样一台电脑就可以往多台数控线切割机床传输指令代码,可是美中不足的是用这种转化器每一次只能保证传给一台线切割机床,并且每传输一次就要切换开关一次,操作非常之麻烦。

总的来说,目前电火花线切割加工存在着三大问题:(1)精度和粗糙度的问题;(2)加工速度和效率问题;(3)自动化和网络问题。对于问题(1)主要跟线切割机床的加工精度有关,这是制造商关心的问题,而本文则针对后两者的问题。

## 二、电火花线切割加工中 CAD/CAM 存在的主要技术问题

当前,数控机床的改造成为许多科技工作者的研究热点。基于硬件设备很难再进行改造,故线切割数控自动编程及 CAD / CAM 等技术成为改造现有线切割数控机床的主要方式。

特别需要指出的是,在电火花线切割加工领域,国内不少企业对线切割 CAD/CAM 等计算机辅助单项应用以及数控技术已经比较成熟。如市场上流行的“Band5 WEDM 绘图式线切割自动编程系统”、“北航海尔软件公司的 CAXA 线切割软件”<sup>[65]</sup>等。

分析目前在线切割加工车间广泛应用的自动编程软件,其还存在着许多不足之处,不能满足自动化生产的需求。这些问题主要体现在:

(1) 目前仍然有一大部分线切割自动编程软件这是因为目前的绝大部分线切割自动编程系统还是运行在 DOS 系统下,而 DOS 系统有先天性的缺陷,只能访问 640K 内存<sup>[41][51][52]</sup>。

(2) 即使目前推出了许多基于 Windows 的自动编程软件,但其因为兼容性和版本的问题,常常不能兼容目前流行的操作系统。举个例子,CAXA 最新版的线切割 CAD/CAM 软件就只能在 Win98 下面运行,却无法在 Win2k 下安装运行。

(3) 不能实现加工的自动寻迹。需要用户指定加工起点、加工方向,增加了操作人员对加工的干涉,从而带来较多的不稳定因素。对于一些常用普通零件没有必有按部就班的操作。

(4) 没有矢量化功能。对于一些形状复杂的纪念章、工艺品或者模具器件很难进行自动编程,即使人工用直线和圆弧拟和编程,程序段很多,繁琐不堪<sup>[74]</sup>。

(5) 软件品种、版本五花八门,往往自成一派,兼容性极差。

相对来说,采用通用 CAD 软件作为 CAD 平台成为一种比较经济的方式。AutoDesk 公司推出的 AutoCAD 绘图软件因其强大的图形输入、编辑功能和具备开放性数据文件接口而成为广泛采用的图形输入平台。但是,这些系统中依然存在较多的问题。总得来说,国内在开发基于 AutoCAD 的数控自动化系统方面深度和广度还不够,存在着一些问题:

### (1) CAD/CAM 集成度较差

在这些系统中,CAD 与 CAM 是分离的。AutoCAD 仅用于绘图输入,自动编程过程往往用另外一个独立的应用程序进行。操作人员需要进行频繁的程序切换。完全满足不了自动化生产的需求。

### (2) 要求操作人员按照加工顺序绘图

这实际增加了对绘图人员的要求和限制。要求绘图人员必须熟悉加工过程，无法充分利用该绘图软件的绘图和编辑功能。对于一些复杂的工作，往往需要手工计算确定参数，给绘图、加工带来诸多不便。

### (3) 对于刀具轨迹无法进行自动补偿

没有考虑数控加工刀具的磨损，加工的公差、粗糙度等要求。尤其对于电火花加工来说，需要考虑刀具的半径和放电间隙的影响，需要进行刀具实际加工轨迹进行补偿。

有鉴于此，设计一套便捷的、可操作性强而且经济实用的电火花线切割加工车间自动化生产的解决方案，是非常具有现实意义的。

## 1.2 电火花线切割 DNC 加工系统研究现状

### 1.2.1 国内外 DNC 技术研究现状<sup>[4-35]</sup>

近几年，随着计算机技术、通讯技术以及数控技术自身的发展，DNC 系统从以前局限于用一台中央计算机控制多台 CNC 机床的传统 DNC 而发展成新型集成 DNC<sup>[12]</sup>（或称广义 DNC）。这种集成化 DNC 系统以计算机、通讯、数控等技术为基础，把与制造过程有关的设备与上层控制计算机集成起来，从而实现制造设备的集中控制管理以及制造设备间、制造设备与上层控制计算机间的信息交换<sup>[14]</sup>。车间级自动化制造系统是 CIMS 系统的重要组成环节，是企业实施 CIMS 的一项关键技术。DNC 作为自动化制造系统的一种模式随着 DNC 技术的发展重新引起国内外企业的重视。

#### 一、DNC 硬件接口技术研究现状

目前，国外 DNC 硬件接口技术研究主要包括以下三种形式<sup>[23]</sup>：

- (1) 基于 CNC 系统侧的插卡；
- (2) 基于 DNC 主机侧的插卡；
- (3) 介于 DNC 主机和 CNC 系统之间的 DNC 连接器；

国内对于 DNC 硬件接口技术的研究也很活跃。由于目前国内存在大量 80、90 年代进口的一些数控系统及国产的数控系统一般都只具有 RS-232 接口。此外，还存在大量的不带串口经济型数控机床<sup>[18]</sup>。针对上述国情，国内主要研究对象为后两种形式，即基于 DNC 主机侧的插卡和介于 DNC 主机与 CNC 系统之间的连接器。如深圳研发电子有限公司研制的 RS-232 扩展卡，类似于 TurboExpress Port RS-232 卡，但功能不及该卡，最多只可与 16 台 CNC 系统相联；又如重庆大学、南京航空航天大学等单位研制的以现场总线(如 CAN-BUS、BitBUS 等)作为与设备层中 CNC 系统及 I/O 口相联的纽带，均取得了一定的效果。

#### 二、DNC 软件技术的研究现状

在 DNC 技术发展之初，其功能大小主要由构成系统的硬件来决定。随着计算机通信技术及网络技术的发展，目前已出现了共识的 DNC 通信接口商品化产品，现在 DNC 的功能及特色主要由软件来扩展<sup>[19]</sup>，竞争的结果使得 DNC 功能越来越丰富、强大，使用也越来越

方便、容易。如美国 Spectrum CNC Technologies 公司的基于 Windows95、98 / NT 的 Multi-DNC 系统、DNC-Work 公司的 EXtreme DNC 软件、英国 Hitachi Seiki Systems 公司的 Networked DNC 软件、日本 FANUC 公司的 DNC2 系统、中国台湾涌佳科技有限公司的 X-DNC。其他知名的 DNC 软件公司在美国就有十多家，如 Advanced Digital Research 公司的基于 Windows 的 Advance DNC、Cim Works Gage Talker 公司的 Visual DNC、CIMNET 公司的 Shopnet DNC 等。这些 DNC 软件大部分可通过 Windows 支持的 TCP/IP 协议与 LAN 或 UNIX 系统联接，实现 CNC 程序、工艺参数等数据的上/下传递；状态采集(含故障信息、刀具信息)与显示；NC 程序编辑（含图形仿真与比较等）；生产调度与监控；单元控制与集中管理（含生产订单、设备、工具、材料、工艺文件、NC 程序、数据备份、质量等资料管理）；人员分级权限；事件日志；远程控制（含异地访问）以及与 CAD / CAPP / CAM / MIS 集成等集成功能。

国内 DNC 应用软件的研究也非常活跃，有一定影响的有：①南京航空航天大学与成都飞机公司合作研制的“成都飞机公司的 FDNC 系统”。系统生产线主要由单元控制（含工作站控制）和 DNC 接口控制二级控制器组成。单元控制器主要承担系统计划、调度、统计与查询等功能，DNC 接口控制器根据实际生产环境采用串行通信方式，主要承担 CNC 系统的控制、数据双向传送、设备加工状态与故障信息采集等功能。②重庆大学研制的“基于软插件技术的 DNC 通信系统”<sup>[13]</sup>。其软插件的含义为将异构数控设备的各通信协议转换成软件模块，从 DNC 主机中下载到与机床相联的 CAN 通讯卡内的 EPROM 中，通过 CAN 主控卡实现异构数控设备与 DNC 主机的联接与通信。该系统 1998 年 7 月通过国家教委组织的技术鉴定。③清华大学和北京机床研究所合作完成的“国家 CIMS / ERC 的 DNC 系统”。系统的通信结构为局域网型，其基础控制级互联采用星型拓扑结构，通过串行口连接到 DNC 主机，被控设备有加工中心、自动导引小车、刀具预调仪和立体仓库。该系统于 1992 年底通过国家验收。④西安交通大学和西安顺通电子工程公司合作完成的“ST-DNC1.1、2.1 和 3.1 型 DNC 系统”<sup>[16]</sup>。1.1 型为经济型 DNC 系统，用一台微机直接和多台数控机床相联；2.1 型在微机与机床之间增加了一级 DNC 智能控制台，可处理多种通信接口，如纸带阅读机口、RS-232 口和 DNC 接口等；3.1 型在 2.1 型的基础上增加了作业管理、过程监控等功能。其它还有上海交通大学研制的“基于 CORBA 的 DNC 系统”<sup>[17]</sup>、四川大学研制的“异构数控机床的 DNC”<sup>[18]</sup>等，均取得了一定的研究成果。

### 1.2.2 电火花线切割 DNC 加工系统的提出

尽管国内外 DNC 技术的研究已经相当成熟，国外昂贵的 DNC 商业软件产品也是推陈出新。而国内推广应用则不尽如人意，至今国内还没有一套能把多台异构线切割机床构成完整的电火花线切割 DNC 加工系统的解决方案。

虽然 DNC 具有诸多强大的功能模块和便利之处，但是目前讨论的 DNC 系统解决方案都是针对大型企业或是车间级的自动化加工模式，并不适用于电火花线切割加工。其存在的

弊端是：一、其经济成本太高，无法推广应用；二、许多功能不是针对线切割加工的，比如一些针对常规数控机床 DNC 通信端口对于线切割机床则无法使用，因为目前国内使用的很多线切割机床都是没有提供这方面的通信接口的，故也是难以实施。

考虑到蓬勃发展的电火花线切割加工专业工厂和车间生产要适应快的特点，制造厂商对订单的响应速度，仅仅从提高设备台数和单台生产效率是不够的，应从整体上提高电火花线切割加工的自动化水平和管理水平<sup>[6]</sup>。因此，构造具有较高自动化水平和一定的管理协调能力的电火花线切割加工系统是非常有意义的。

有鉴于此，本文独创性地提出了针对电火花线切割加工车间的 DNC 加工系统。本文提出的整个电火花线切割 DNC 加工系统，除了自带线切割 CAD/CAM 软件功能之外，还具有 DNC 系统的基本功能：通信功能、管理功能。即系统能实现线切割零件的设计、制造，同时又能对异构的线切割机床给予一定的控制和有效管理。

### 1.3 论文选题的意义

基于以上国内企业或车间电火花线切割加工中所存在问题和目前国内外 DNC 技术的研究现状，本文确立的研究内容是电火花线切割 DNC 加工系统的研究。考虑到 AutoCAD 是国内目前普及率最高的 CAD 支撑平台，对许多企事业单位来说，它仍是进行产品设计和绘图的首选工具。如果在此基础上开发一套实用的电火花线切割 DNC 加工系统的自动编程系统，其经济效益和社会效益将是巨大的，具有广阔的应用前景。开发电火花线切割 DNC 加工系统可以解决目前国内在模具工厂或是车间线切割加工应用方面的一些问题：

- (a) 该 DNC 加工系统集成生产任务分配、任务管理、图纸管理、工艺管理、NC 程序管理和机床信息管理为一体，可以实现与其他 CAD/CAM 系统的信息共享。
- (b) 该 DNC 加工系统自带了线切割 CAD/CAM 系统，可以解决目前许多企业中设计与加工相分离的问题。由于该系统的自动编程系统是基于 AutoCAD 开发的，所以可以保证产品信息的完整性与正确性，实现设计与加工的无缝集成与共享，同时可靠性和通用性比较高。
- (c) 该系统提供了交互式的矢量化功能。可以通过扫描设备将实物或者手绘图片扫描为数字图像，再将图像转化为可以直接在 AutoCAD 平台下的 DXF 文件，然后进行编程加工。
- (d) 该系统解决了与目前国内许多异构线切割机床的通信问题，实现了对线切割机床设备的有效控制和管理。

### 1.4 论文研究的内容

通过对目前我国电火花线切割加工的存在的问题和相关方面研究现状的分析，本文将利用当前成熟的计算机网络技术、通讯技术以及数控技术，对电火花线切割加工车间进行技术改造，实现电火花线切割加工车间高效率、快反应的生产目标。本文的总体结构如图 1-1 所

示。

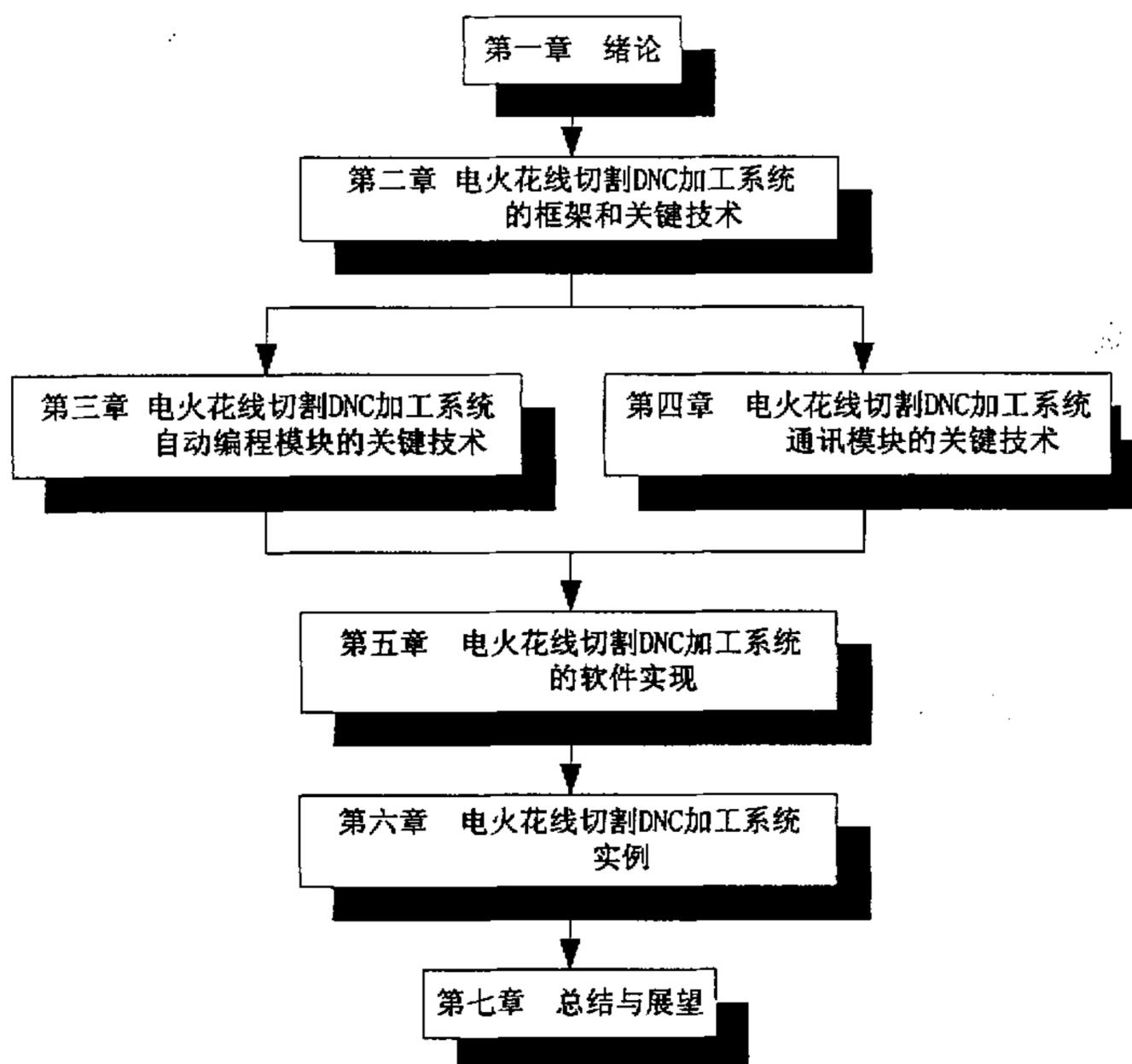


图 1-1 论文的总体结构

本文提到的电火花线切割 DNC 加工系统包括三大模块：(1) 自动化编程模块，主要解决线切割零件的加工代码生成；(2) 通信模块，主要解决 DNC 主机与异构线切割数控系统的双向信息传输问题；(3) 管理模块，解决生产过程中加工图纸、工艺、NC 程序的数据库管理。本文着重研究系统自动化编程模块和通信模块，管理模块则不作详细论述。

本论文主要包括以下内容：

第一章介绍了电火花线切割加工的原理和应用范围，分析了线切割加工的现状和问题，联系当前国内外 DNC 的研究现状，阐述了开发电火花线切割 DNC 加工系统的重要意义。

第二章提出了电火花线切割 DNC 加工系统的总体框架和模型，论述了系统的功能和原理，分析了实现系统的关键技术。

第三章研究了电火花线切割加工 DNC 系统自动化编程系统模块的关键技术。详细分析了参数化加工路线生成的方法、钼丝轨迹的后置处理、指令代码的仿真、二维创新设计图的矢量化处理。

第四章研究了电火花线切割 DNC 加工系统通讯模块实现的关键技术。对异构线切割机 DNC 通信的实现和多串口通信解决方案进行了详细的讨论。

第五章对整个系统软件进行了总体设计,并对功能模块进行了划分。在此基础上,介绍了 AutoCAD 环境下利用 VC++和 ObjectARX 进行系统自动编程模块软件开发的过程。

第六章介绍了电火花线切割 DNC 加工系统实施的例子,浙江大学电火花线切割二维创新设计 DNC 平台的实现,进一步证明了电火花线切割 DNC 加工系统研究的使用价值。

第七章总结了论文工作的内容,指出了以后需要进一步解决的问题。

作为本文所介绍的电火花线切割 DNC 加工系统实施的例子—浙江大学电火花线切割二维创新设计 DNC 平台,目前已在我校金工实习基地生产、教学实习中得到充分应用。实践证明,实施本系统后,使线切割加工车间的资源达到了优化配置和利用,提高了线切割加工的自动化水平和加工效率。本文对电火花线切割 DNC 加工系统进行了有益的探索。

## 第二章 电火花线切割 DNC 加工系统的框架和关键技术

高生产率、高质量和低费用的制造生产一直是工业界所追求的目标，电火花线切割加工更是如此。如何以最少的费用、最快的速度生产出高质量的产品，一直是人们关心的问题。本章根据前章绪论提出的问题，完成了电火花线切割 DNC 加工系统基本结构的完整设计和技术分析，为电火花线切割加工车间自动化生产提供了一种有效的解决方案。

### 2.1 电火花线切割 DNC 加工系统结构及特点

#### 2.1.1 电火花线切割 DNC 加工系统框架结构

近年来，数控机床的占有率在我国呈逐年上升趋势，在一些大型模具厂或车间已出现了一些专门的线切割数控车间。传统的一对一的简单生产作业方式，已经不能满足目前生产加工的要求。要想实现提高生产效率，必须对分散的机床设备进行有效的管理和控制，因此需要扩建设备控制层对原先单一的设备层进行管理和控制即一对多模式。但是，简单的一对多方式其实质意义上还是分散的，生产设备和人力资源得不到优化配置。由图 2-1 可知，我们只有采用多对多的生产模式，采用控制服务器对设备进行集中控制，同时使用网络数据库服务器实现与上层车间或是工厂的信息交流。另外，多对多模式，最大限度的利用了生产设备、人力资源，真正达到了大型模具厂或线切割车间高效率、快响应的生产要求。

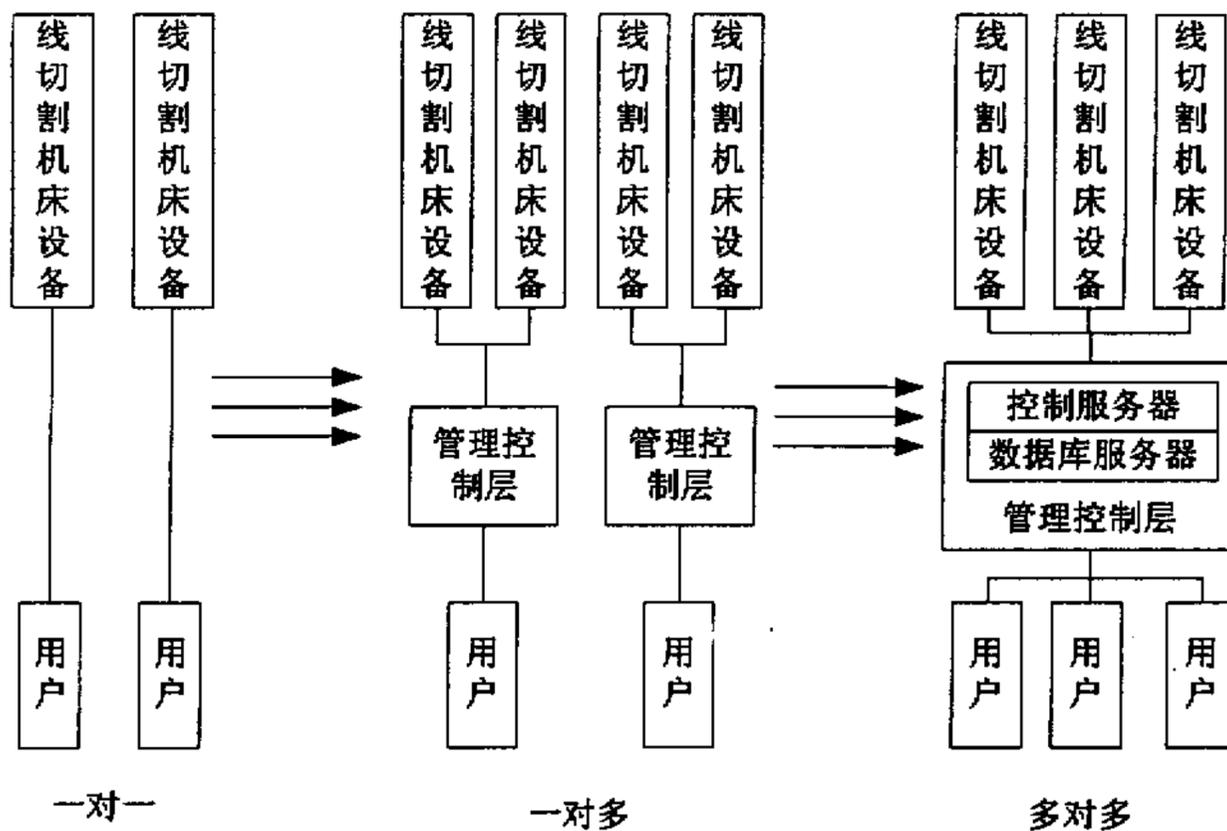


图 2-1 线切割加工模式的改进

根据多对多的模型，本文提出了由服务器层、工作站层和数控设备层构成的三级电火花线切割 DNC 加工系统。其结构主要为客户机/服务器的体系结构。服务器和 DNC 主机放在远离加工车间的控制中心，工作站放在数控线切割机床附近。服务器和 DNC 主机以及工作

站通过局域网相连，DNC 主机则通过多种方式跟数控机床实现通讯。系统网络拓扑结构如图 2-1 所示。

服务器层是整个制造系统的核心，根据功能的不同可分为 CAD/CAM 服务器、控制服务器（DNC 主机）和数据库服务器。服务器层集 CAD/CAM、DNC 控制和多种管理功能为一体，如通讯调度管理、系统信息管理、NC 程序管理、工艺信息管理、生产计划、生产调度等。服务器层向上通过集线器跟工厂局域网相连接，这样可以保证与其他车间的各系统的计算机进行通讯、资源共享，提高整个企业数控加工和加工管理的效率。

设备层考虑到目前电火花线切割机床设备的异构性，DNC 主机和从机之间采用多种通讯方式。对于装有 Windows 的微机 and 带有以太网接口的线切割机床控制器使用以太网方式进行通信；对于控制器为单片机的经济型线切割机床或是工控机的使用多串口通信；针对旧式无 RS-232 串口的单板机控制器则采用并口通信。

多个操作人员可以并发地通过各个工作站通过网络执行该系统的各项功能，这样有效的共享了数据信息和网络资源。而系统管理员也可以方便地管理、维护网络系统。

### 2.1.2 电火花线切割 DNC 加工系统的特点

该系统的主要特点是：

- (1) 集计算机网络技术、数据库技术、通讯技术、管理技术、数控技术为一体；
- (2) 分布式处理模式，CAD/CAM 服务器、数据库服务器、DNC 主机和客户机分担不同的任务；
- (3) 系统的扩充性好，可根据需要增加或减少客户机和线切割数控系统，同时对系统的其它部分几乎没有影响；
- (4) 数据库的多用户性，这对保证数据完整、安全、数据访问控制是非常有利的；
- (5) 系统的重要数据采用具有客户机 / 服务器结构的模式进行处理；
- (6) 系统中 DNC 主机除了分配任务以外，还对数据库进行管理，调用生产管理系统对图纸、工艺、NC 程序进行编辑、修改及存储等；

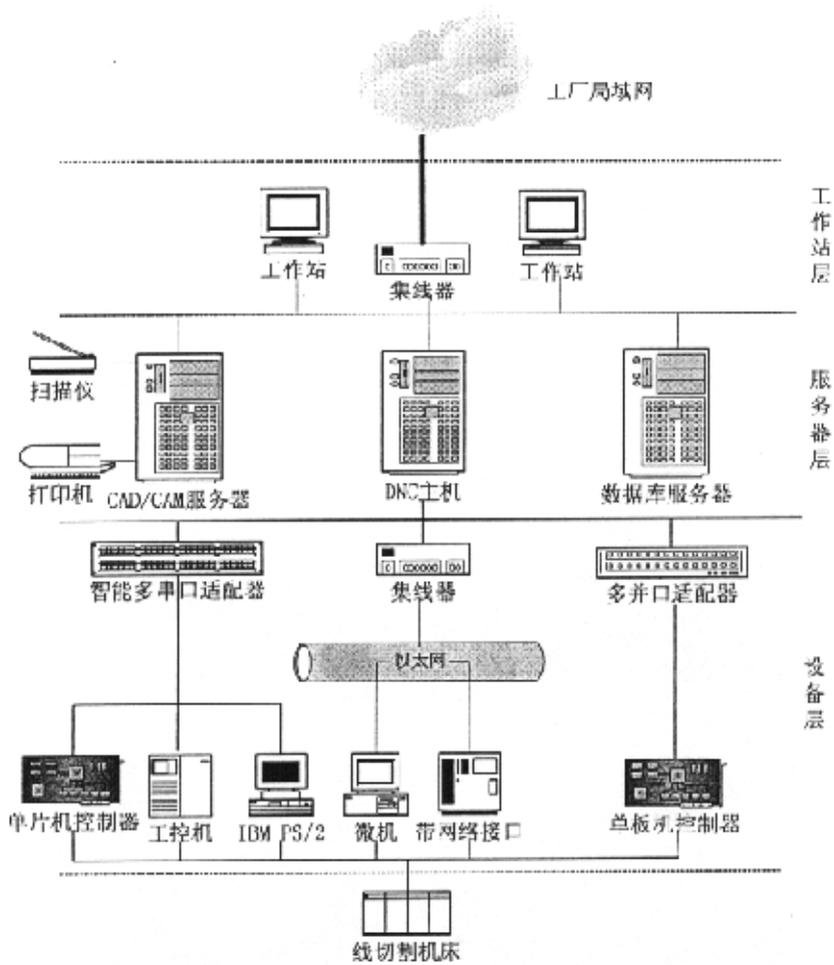


图 2-1 电火花线切割 DNC 系统结构图

## 2.2 电火花线切割 DNC 加工系统的功能

### 2.2.1 电火花线切割 DNC 加工系统各层的功能

下面分别对该系统中 CAD/CAM 服务器、数据库服务器、DNC 主机和工作站的分布式结构（如图 2-2 所示），以及系统软件的主要功能加以介绍。

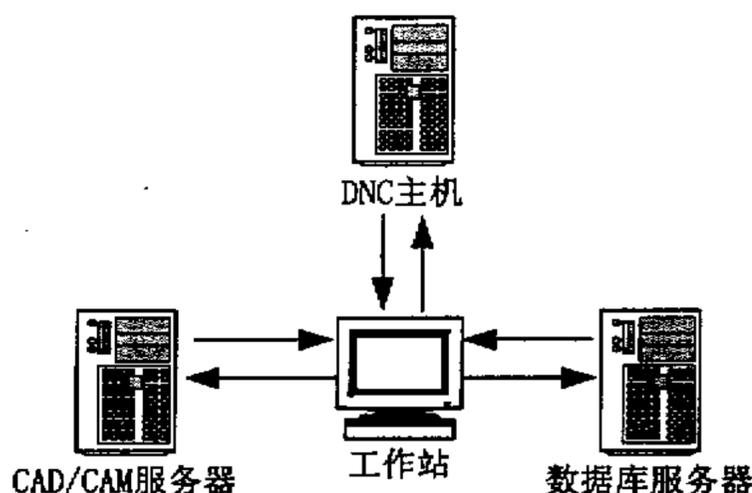


图 2-2 系统分布式结构图

**数据库服务器：**用于存放、管理线切割生产任务、图纸、工艺文件、NC 程序等数据信息。

**CAD/CAM 服务器：**实现线切割零件绘图和自动化编程。

**DNC 主机（控制服务器）：**是该系统的中间层。实现线切割生产任务、图纸、工艺文件、NC 程序从数据库服务器的下达和收集来自数控系统反馈的有关信息，实现生产任务的分配及生产任务管理、图纸管理、工艺文件管理、NC 程序管理。

**工作站：**工作站是该系统的操作层。系统的大部分操作是在工作站进行的。工作站层推荐采用无盘工作站，一方面可以大大降低投资，另一方面可以大大减少病毒入侵的机会，防止非法拷贝，保护文件，增强系统的安全性。

**从机（机床控制器）：**是该系统的媒介层，起承上启下的作用，它一方面通过网络实现接收 DNC 主机下达的任务，并向 DNC 主机反馈有关信息。另一方面，通过串行口或其它接口向线切割机床传送加工程序代码。

该系统的软件主要实现三大功能：管理功能（包括任务分配、任务管理、图纸管理、工艺管理、NC 程序管理等）、通信功能（包括 NC 文件传输、命令传输、设备跟踪和监控等）和自动编程功能（扫描识别、绘图、编码等）

### 2.2.2 电火花线切割 DNC 加工系统主从两端的功能

**DNC 主机（控制服务器）端的主要功能：**

(1) **线切割任务分配：**DNC 主机根据从机的状态进行任务分配（任务号、图纸、工艺、NC 程序）

(2) **线切割任务管理：**数据库记录的插入、查询、删除、修改、浏览、报表的打印及图纸、工艺、NC 程序的编辑、修改

(3) **线切割图纸管理：**数据库记录的插入、查询、删除、修改、浏览、报表的打印及线切割图纸的编辑、修改。

(4) **线切割工艺管理：**数据库记录的插入、查询、删除、修改、浏览、报表的打印及线切割工艺卡片的编辑、修改。

(5) 线切割 NC 程序管理：数据库记录的插入、查询、删除、修改、浏览、报表的打印及线切割 NC 程序的编辑、修改。

从机（机床控制器）的主要功能：

- (1) 监控线切割机床的加工状态，定时向 DNC 主机反馈从机的状态。
- (2) 接 DNC 主机分配的任务、图纸、工艺、NC 程序。
- (3) 向数控机床传送线切割数控程序加工代码。

整个加工系统集成线切割 CAD、CAM、通信、管理技术于一体，从而真正实现了线切割车间制造设备的集中控制管理以及制造设备与上层计算机之间的信息交换。如果 DNC 主机的存储能力足够，可以将 CAD/CAM 服务器、数据库服务器直接集成到控制器服务器即 DNC 主机内。我校金工实习中心的线切割加工车间就采用了此方案，实践证明，该系统运行稳定，使用效果良好。

## 2.3 电火花线切割 DNC 加工系统采用的关键技术

### 2.3.1 基于客户机/服务器的体系结构

本系统中，工作站与数据库服务器之间为客户机/服务器的体系结构。

近年来，分布式数据库开放体系结构中最有成效的是客户机 / 服务器体系结构。客户机 / 服务器体系结构是由客户机和服务器构成的一种网络计算机结构，客户机 / 服务器体系结构解决了客户应用与服务器数据库分布处理。应用程序的任务分别由客户机和服务器分担，一方面充分发挥客户机处理能力，另一方面减轻服务器的负担。

在本系统中，考虑到客户端访问人数不是很多，工作站与数据库服务器之间采用 MS SQL SEVER 作为后台服务器数据库；前台用户开发工具，这里采用 Visual Basic，它可通过 ODBC、DB-Library 或是 DAO 专用接口访问服务器的数据。系统采用的客户机/服务器体系结构如图 2-3 所示。

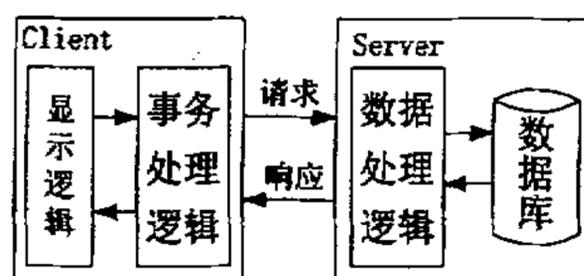


图 2-3 C/S 体系结构图

由于 DNC 主机起着通信调度的重要功能，因此从效率性上考虑，与工作站之间不能使用客户/服务器的体系结构。而是客户机直接登陆 DNC 主机，这样操作直接在 DNC 主机内进行，几乎不影响实时性的要求，经过实践操作表明此种方案是科学的。

因为篇幅有限，本文对于客户机/服务器的体系结构的具体实现不作详细介绍。

### 2.3.2 电火花线切割 CAD/CAM 技术

本系统 CAD/CAM 服务器里安装的就是线切割自动编程系统。虽然目前市场上有很多线切割自动编程系统产品可供选择,可是这些产品都是针对单机版本的,无法胜任服务器软件的要求。

AutoCAD 是目前世界上应用最广的 CAD 软件之一,在 2D 市场上占有绝对的优势。据业内人士统计,AutoCAD 的装机量 1999 年仍然稳座中国 CAD 市场的第一把交椅。如今已广泛应用于机械、建筑、电子、纺织等领域中。它实际上已成为一种微机 CAD 系统的标准,工程人员之间交流思想的公共语言。如果某种 CAD 应用软件不能与 AutoCAD 交换图形信息,或者某类图形输入输出设备不符合 AutoCAD 的硬件要求,那么它们就很难在微机市场竞争中取胜。

考虑到电火花线切割加工中,主要涉及二维图形的信息处理,因此,利用 AutoCAD 作为线切割 CAD/CAM 系统的开发平台是最适合不过的了。AutoCAD 作为系统的图形输入器,充分利用其强大的 2D 绘图、编辑、剖面线和图案绘制、尺寸标注以及方便用户的二次开发功能,可大大提高系统的开发效率及可靠性。

另外,选用 AutoCAD 作为开发平台得另一个原因是它提供了功能强大的二次开发软件包 ObjectARX。ObjectARX 编程环境提供了一个面向对象得 C++应用程序开发界面,使开发者能够使用、定做和扩展 AutoCAD。ObjectARX 库包括各种各样得工具,可以使应用程序开发者方便地利用 AutoCAD 的开放结构,这些工具可以方便应用程序对 AutoCAD 数据库结构、图形系统和本地命令直接访问。

在系统开发中,将结合 MFC 来构造 ObjectARX 应用程序,通过 AutoCAD2000 自动加载执行,在 AutoCAD2000 的系统界面中加入加工处理原型系统,本系统暂命名为 WEDM,可以通过屏幕菜单里的菜单项来调用系统的各功能。

采用 AutoCAD 作为开发平台,可谓扬长避短。自主开发一个 CAD 平台,其开发量和难度是相当大的,并且从产品推广使用方面来看也是不易的。采用 AutoCAD 作为开发平台后,使得本系统的零件 CAD 功能远远强于同类线切割加工方面的 CAD 软件。并且系统是在 AutoCAD 二次开发基础上生成的,程序运行比较稳定,可靠,适合做服务器的 CAD/CAM 系统。

本文将在第三章详细介绍基于 AutoCAD 环境下自动化编程系统的开发原理和流程。

### 2.3.3 DNC 主机与异构数控系统的通信

在本系统中,要想实现 DNC 主机对线切割机床的有效控制和管理,解决 DNC 主机与线切割机床的通信问题则是关键。使用到的技术主要有网络通信、多串口通信和异构数控系统通信。

如图 2-1 设备层之所以采用这种结构,其依据在于:

(1) 目前国外已经出现带有以太网接口的线切割机床,并且据业内人士透露国内一些机床厂家也在试制该类产品,故不久的将来的这种线切割机床将会大量出现,并且逐步替代

旧产品。

(2) 把一部分控制线切割机床的 486 以上微机进行改造, 为其配备网络适配器, 尽量通过网络协议进行通讯。(1)、(2) 此种通信使用的是以太网或 TCP/IP 协议

(3) 对于带有 RS-232C 或是 RS-422 串行接口的控制器或是工控机, 则采用 MOXA-C114 智能多串口适配器, 以此为基础可方便可靠地实现多路串行通信能同时进行。这里使用的协议, 由自己设定, 因此编写多串口通信协议是关键。

(4) 不带串口的老式单板机控制器, 可以利用纸带阅读口、纸带穿孔机通过外接式通讯适配器与 DNC 主机实行通信。本文主要采用软件的技术实现通信。

本文将在第四章详细介绍线切割机床异构通信的实现方法。

## 2.4 本章小结

本章首先搭建了电火花线切割 DNC 加工系统的三层框架结构: 服务器层、工作站层和设备层; 描述了系统的特点; 然后进一步介绍了系统的各层的功能、相互关系及其所需完成的相关操作; 最后详细分析了实现系统各个重要模块实现的关键技术, 它们分别是: 基于客户机/服务器的分布式体系结构、线切割 CAD/CAM 技术及 DNC 主机与异构数控系统的通信。

### 第三章 电火花线切割 DNC 加工系统自动编程模块的关键技术

在机械制造领域，自动化、集成化是两个重要的发展方向，两者相辅相成。自动化编程系统的集成化设计将成为重要的发展趋势，同时也将大幅度提高数控加工的效率 and 可靠性。由于计算机技术的飞速发展，特别是 CAD/CAM 技术的应用普及为计算机自动化编程提供了优越的硬件和软件条件。本章以通用的 AutoCAD 为图形化输入的平台，利用面向对象编程 (OOP) 技术，通过 ARX 与 MFC 混合编程，完成了线切割自动编程模块的开发，实现了基于图形的 CAD/CAM 软件系统的集成。系统使用标准的屏幕菜单，会 AutoCAD 操作就会使用本系统的操作。

#### 3.1 模块总体结构和功能

本系统自动编程软件模块包括六大子模块，它们是：文件管理子模块、参数化绘图子模块、零件 CAPP (工艺参数处理) 子模块、零件 CAM (自动编程) 子模块、加工模拟子模块。这些子模块的运行均在 AutoCAD 图形环境下。它们之间的有机配合构成了整个 CAD/CAM 集成系统强大的功能。其结构如下图所示：

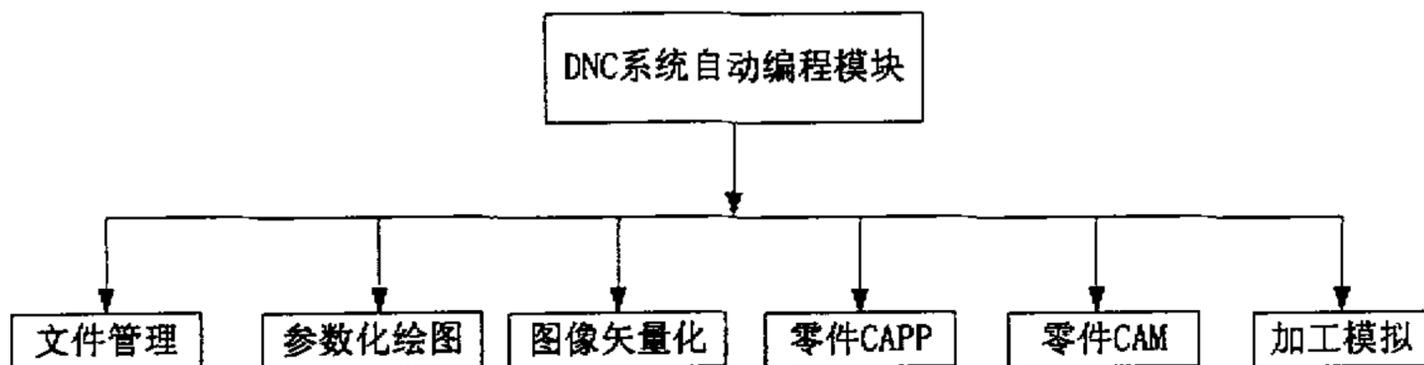


图 3-1 系统自动编程模块结构

通常 CAD/CAM 软件系统的自动编程的基本步骤主要有<sup>[3]</sup>：(1) 确定加工工艺；(2) 建立加工模型；(3) 生成刀具轨迹；(4) 产生后置代码；(5) 输出加工代码。其中步骤 (1)、(2)、(3) 一般成为前置处理，步骤 (4) 和 (5) 称为后置处理。针对电火花线切割加工来说，前置处理主要是指从图形输入模块中接受图形信息，经工艺处理及相应的数学处理产生钼丝中心轨迹。而后置处理则是指将刀具中心轨迹转换成 3B 或是 ISO 指令码的过程。

本文所研究的线切割自动编程系统，是作为整个电火花线切割加工系统中的一个核心部分，是安装在 DNC 主机或服务器中，供终端工作站访问的软件。并非如目前应用在单台微机上的自动化编程系统，因此系统对于程序执行效率、可操作性、实时性要求更加严格。很多市场上的线切割软件因为效率低下、执行速度慢、且系统极不稳定，更是无法胜任服务器应用软件的要求。因此，编写软件的工作中也不能完全照搬常规算法，程序的执行效率就显得尤为重要。

由于自动化编程技术已经发展多年，其理论体系已经比较成熟，故改进旧算法，采用新方法、增添新功能则是本章所要研究的重点内容。

本文自动编程模块的实现主要按如图 3-2 所示的开发路线进行。



图 3-2 系统实现的线索

下面几节就整个模块软件开发过程中的所涉及的一些关键技术和原理及其实现过程，进行详细介绍。

## 3.2 参数化加工路线的生成

### 3.2.1 实现的总体思路

目前很多线切割自动编程软件中，加工路线的生成一般都是按照如下步骤来编码实现的<sup>[70]</sup>：

- (1) 拾取零件轮廓；
- (2) 选择加工的方向；
- (3) 选择切割的侧边（即选择凹模或是凸模），以确定补偿量；
- (4) 确定穿丝点位置及最终切到的位置；

总的来说，这种方法只能适应一对一单台微机的操作，尤其是遇到图形实体数目多的情况下，逐段拾取实体是件非常让操作人员头痛的事情。其缺点是：人工干预太多，操作步骤颇为繁琐，不适合服务器软件高效率、操作简便的特征。

本文认为，对于常规线切割零件的加工完全没有必要按部就班的操作。有鉴于此，本系统在完成常规加工路线生成的基础上，增加了参数化生成加工路线的方法，对于那些常规零件的加工实现了真正的自动化编程，大大简化了操作步骤，明显提高了自动化水平。本文使用参数化设计的原理，通过创建一个电火花线切割工艺参数对话框的形式，如图 3-1 所示，在系统中点击参数化加工路线生成菜单项后，用户只要输入加工工艺参数，便可达到加工路线的自动生成。即使遇到特殊的零件，也可以通过手工生成加工路线来进行处理。

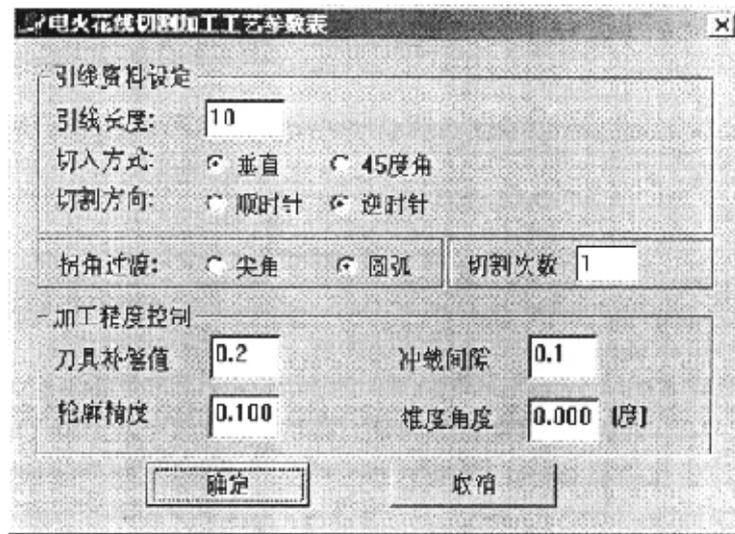


图 3-1 参数化加工路线参数输入

### 3.2.2 图形实体数据的获取

#### 一、获取实体数据的新方法

CAD/CAM 集成系统的一大特点是抛开了手工绘图和手工设计, 实现了交互式操作。整个零件的设计和参数输入都在 AutoCAD 软件环境中完成。

要实现零件图形的自动编程, 首先需要实现对输入图形信息的自动获取, 并进行相应处理。这里实际上是对 AutoCAD 图形的自动识别和拓扑重建。对于未经处理的 DWG 文件, 存在以下问题: (1) 存在大量非加工信息, 包括尺寸线、辅助线、文本等非加工信息。(2) 无序的拓扑关系, 由于 AutoCAD 的图形数据由内部数据库来组织, 组成图形的各图形实体记录完全按照生成的顺序存储, 故相互之间无拓扑关系, AutoCAD 图形文件是图形记录的无序集合。即使是生成的 DXF 数据交换文件也是如此。

许多基于 AutoCAD 的数控自动编程系统是通过读取 AutoCAD 生成的数据接口文件 DXF 文件中的实体段来获取实体数据的, 这里不作详细介绍。用这种办法需要生成、读取 DXF 文件, 操作复杂, 对于单用户操作来说, 使用效果也许并不明显。但是如果终端工作站有多个用户同时进行频繁的读操作, 系统开销太大, 大大降低软件系统的运行性能。

本文利用了 AutoCAD 二次开发函数包 ObjectARX 的强大功能, 可以直接从当前图形的图形数据库中读取信息, 从而获取零件信息更加直观和准确。从而免去了生成、读写 DXF 文件操作的繁琐步骤。并且直接针对 AutoCAD 系统生成的实体对象进行操作, 代码执行效率大大提高。

#### 二、获取图形数据的原理和算法

AutoCAD 的图形数据库的主要对象包括对象词典及符号表。AutoCAD 图形实际上是一

系列存储在数据库中的 AcDb 类的对象。该数据库的所有对象有一句柄，在特定的图形文件中该句柄是唯一的，用来标识某个对象。实体是数据库的一种具有图形特征的特殊对象，如直线、圆、多义线、椭圆及文字等。其他重要的数据对象为符号表、词典及与其他 AcDb 类对象相关的符号。所有的实体都包含在块表 (AcDbBlockTable) 中。图 3-2<sup>[45]</sup>说明了 AutoCAD 数据库的主要组织结构。符号表 (Symble Table) 和对象词典 (Object Dictionary) 是用来存储对象的容器。对象词典可以存储任何 AcDbobject 对象及其子类。

当新建一对象时，该对象被加入到数据库相应的容器对象中。实体对象被加入到块表记录中，符号表记录被加入到符号表中，所有用户自定义的非图形对象被加入到命名对象词典中。

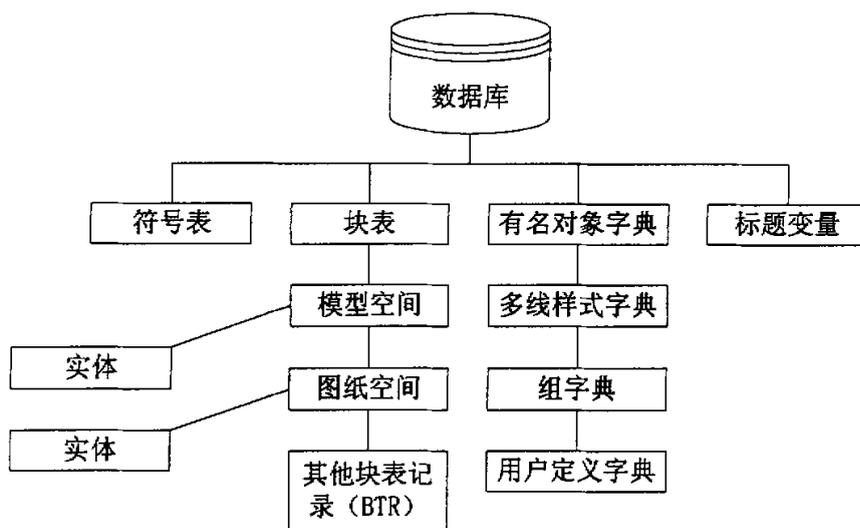


图 3-2 AutoCAD 数据库结构图

AutoCAD 数据库至少包括 9 个符号表 (包括块表及层表) 和命名的对象词典。块表包括两个记录: \*MODEL\_SPACE 及 \*PAPER\_SPACE。层表包括一个记录: 0 层。

在 AutoCAD 图形编辑过程中，可以用通用函数 acdbHostApplicationServices () ->workingDatabase () (旧版本的使用 acdbCurDwg () 函数) 对当前数据库进行数据访问。本系统就是通过此函数对当前图形数据库进行访问，从而得到组成零件的各实体的特征参数的。

对于 AcDbObject 对象可以通过三种途径访问：利用对象的实体句柄、对象的 ID 或者 C++ 的指针。三者的关系如下：

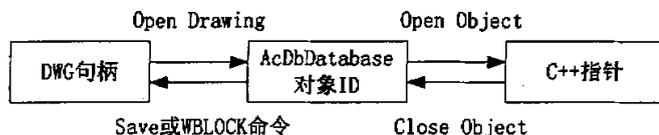


图 3-3 对象的实体句柄、对象的 ID 及指针的关系

为得到图形中的数据,需要对块表进行遍历。AcDb 类中包含的 AcDbSymbolTableIterator (遍历符号表)类及其派生类提供了相关成员函数。需要创建一个遍历符对块表记录进行遍历。

综上所述,获取当前图形中的实体信息的过程如图 3-4 所示。

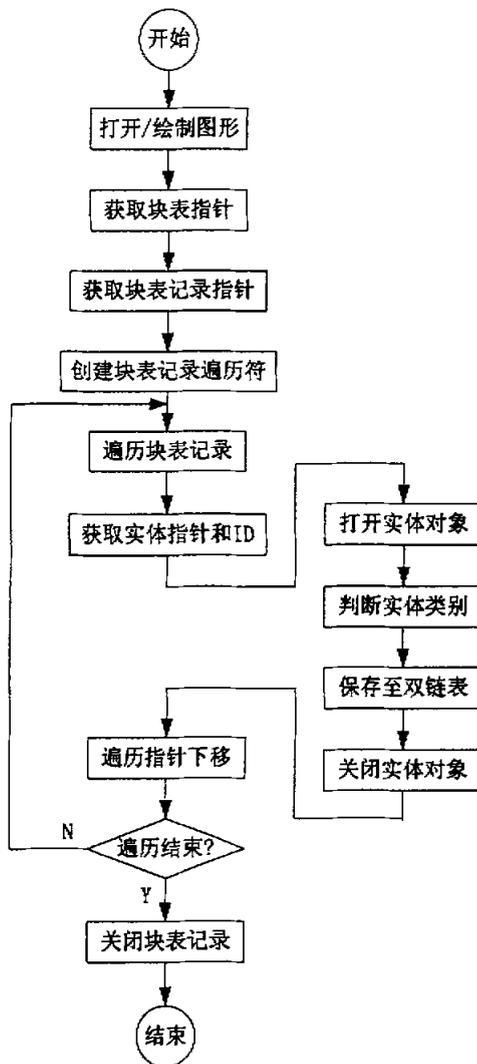


图 3-4 获取实体信息流程图

在该流程图中,遍历多义线 (AcDbPolyline) 的顶点的操作和遍历块表的操作类似。为便于加工处理,当实体为圆 (AcDbCircle) 时,将该圆分成两个半圆弧 (AcDbArc) 来存储;当为多义线 (AcDbPolyline) 时,根据该实体是否封闭,将多义线转化成首尾相接的直线或圆弧存储在双链表中,这样便于进一步处理。

读取零件组成实体特征信息后,需将其存储起来。如直线的起点和终点坐标,圆弧的圆心坐标、半径和起始角,折线或多边形的顶点坐标等。为便于后续插补数据生成加工轨迹,

采用循环双链表结构来存储数据信息。其结构将在第五章介绍。

完成加工路线生成的程序开发过程中，总共要对三个链表结构操作。这里是第一条链表。AutoCAD 绘图实体在图形数据库中按照其绘图顺序存放，与加工的顺序无关；而当图形中删除或添加某一实体时，在图形数据库中也无法体现各个实体的连接关系。因此，读取图形得到的信息是没有拓扑关系的，需要对第一条链表进行重新排序。

### 3.2.3 加工路线生成的原理和算法

#### 一、实体数据的顺序重排

本文系统根据零件轮廓线上前、后图素邻接的原则，由用户输入的加工方向和切入线方向和长度，对第一条链表上的图素进行排序，图素排序后由第二条链表进行存储。

第二条链表上的图素所确定的图形是被加工零件的轮廓，还不是电极丝中心的运动轨迹。为了得到线切割加工的数控代码，需要考虑电极丝半径和加工间隙的影响。因此，在生成加工代码前必须对第二条链表上的图素进行间隙补偿。目前，间隙补偿算法主要有两种：一种是自动圆弧过渡补偿算法（AP 法），另一种是相交过渡补偿算法（IP 法）。本系统综合采用了 IP 算法与 AP 算法，使系统能够处理线切割各种情况的间隙补偿。间隙补偿后的图素由第三条链表进行存储。

第三条链表上的因素确定的轨迹是电极丝中心的运动轨迹，对第三条链表上的数据进行处理，就可以得到数控加工代码。加工路线的生成总体过程详见图 3-5 所示。

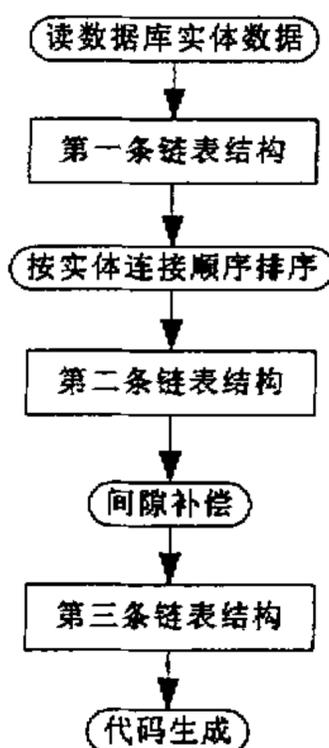


图 3-5 加工路线生成总体过程

#### 二、基本加工路线的生成

##### (1) 加工方向处理

目前很多自动编程系统只能工人拾取方向如 CAXA 软件，对于常规图形的加工完全没有必要这样；有些则是虽然有部分自动生成路线功能但是系统不能由用户来指定方向，其人机性能不强。参数化加工路线生成方法有个特点那就是：在使用中能接受用户指定的切割顺

序,即顺时针还是逆时针切割。许多相关方面的算法过于繁琐,本文通过分析矢量之间的连接关系直接根据枚举法给出简捷的处理算法。

a. 首先确定基准点 B。遍历双链表,找出距 X 坐标轴或 Y 坐标轴地距离最近地端点为基准点 B,同时用指针 pB 记录其地址。设其在链表中前后两端点分别为 A 点和 C 点。

b. 分析矢量 AB 矢量和 BC 之间的对应关系。考虑到到 AutoCAD 下一般在第一象限绘图,所以矢量 AB 只能是在第 III 或是第 IV 象限,矢量 BC 则只能是在第 I 象限或是第 II 象限如图 3-6 所示,这样就大大简化了判断过程。

(1) AB 在第 IV 象限,即:  $X_B - X_A \leq 0, Y_B - Y_A < 0$ ; BC 在第 I 象限,即:  $X_C - X_B > 0, Y_C - Y_B \geq 0$

(2) AB 在第 III 象限,即:  $X_B - X_A < 0, Y_B - Y_A \leq 0$ ; BC 在第 II 象限,即:  $X_C - X_B \leq 0, Y_C - Y_B > 0$

(3) AB 在第 III 象限,即:  $X_B - X_A < 0, Y_B - Y_A \leq 0$ ; BC 在第 I 象限,即:  $X_C - X_B > 0, Y_C - Y_B \geq 0$ 。同时,根据是否满足  $k_{III} > k_I$ ,又分 2 种情况,设为 3.1 和 3.2

(4) AB 在第 IV 象限,即:  $X_B - X_A \leq 0, Y_B - Y_A < 0$ ; BC 在第 II 象限,即:  $X_C - X_B \leq 0, Y_C - Y_B > 0$ 。同时,根据是否满足  $k_{IV} > k_{II}$  又分 2 种情况,设为 4.1 和 4.2

c. 根据用户输入的加工方向:顺时针或是逆时针,结合 b 过程的判断,确定在遍历链表节点数据的方向(左或是右对应加工方向)。

如图 3-7 给出了判断流程。

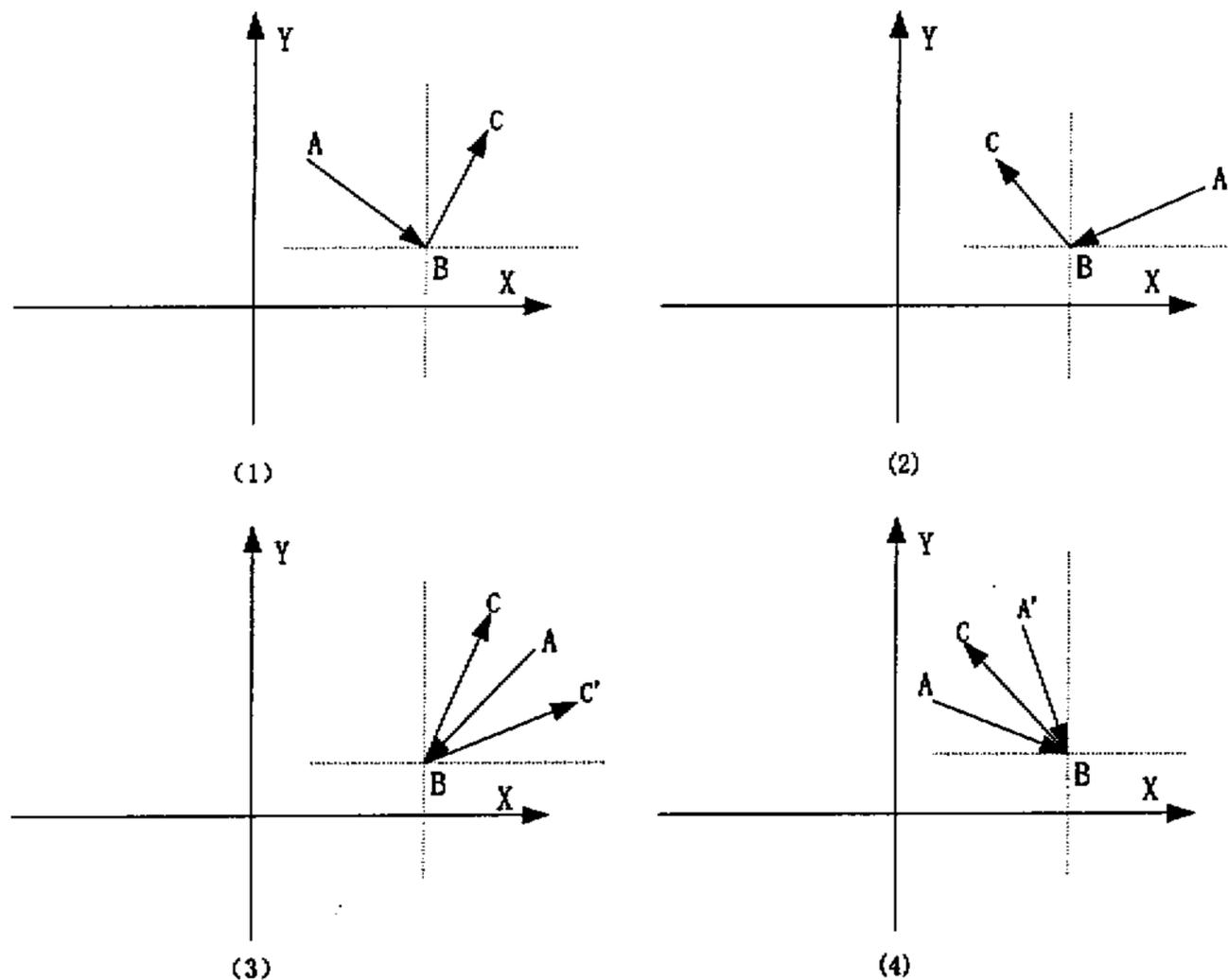


图 3-6 矢量 AB、BC 连接关系

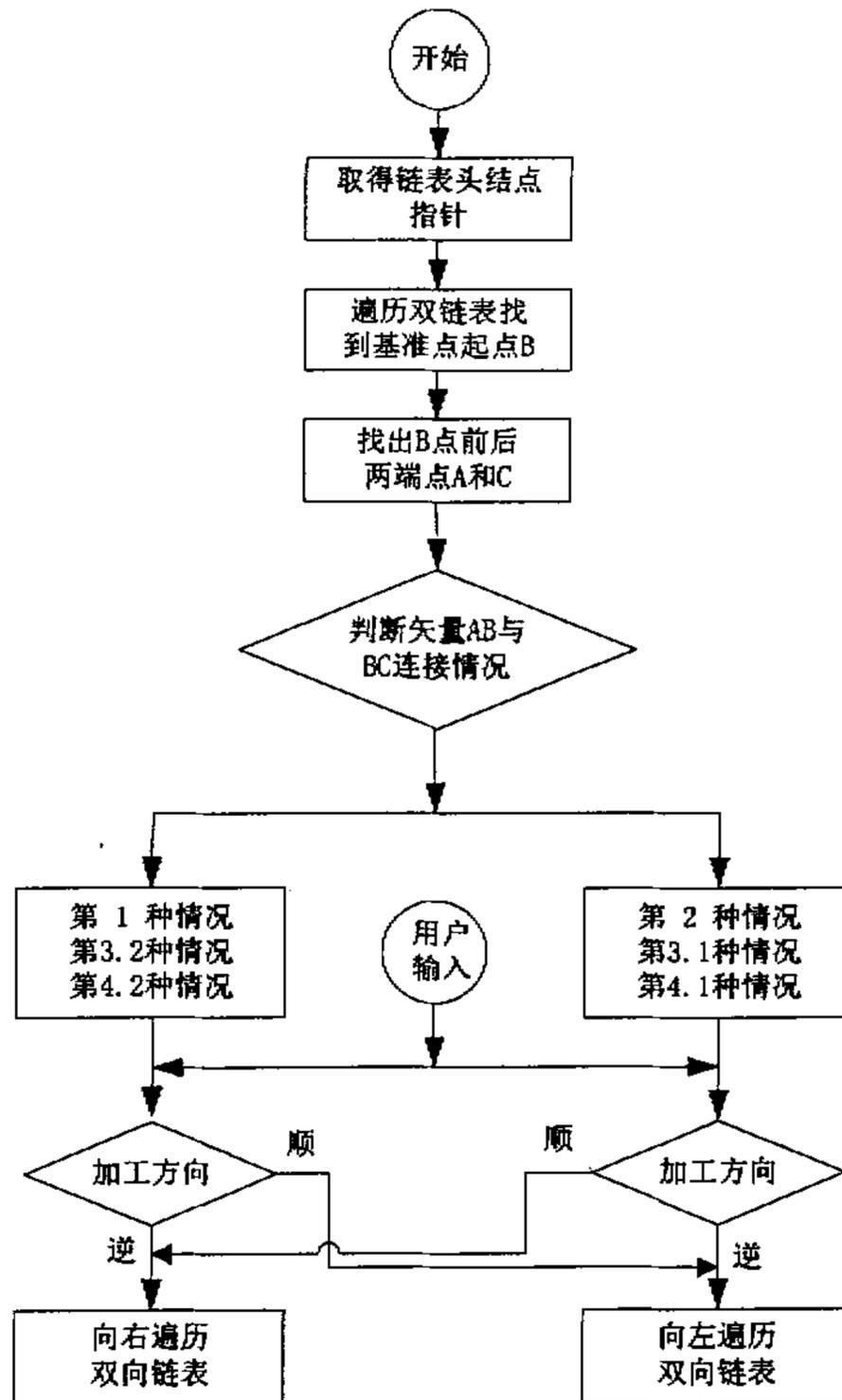


图 3-7 判断加工方向流程图

此方法简便易行，简化了程序的编制，只有加减运算，运行速度快，提高了程序执行效率。

### (2) 切入点的选定

切入点是指一个封闭轮廓中首先被切割的节点。选择起割点一般首先考虑选在线段的交叉点上，或是选在精度要求不高的表面上，一般不宜选在切点或是圆滑表面上。有些自动化编程系统中，直接以上文的基准点 B 或实体的交点处为起始点，如果基点 B 点刚好是两条线段的交叉点上，则没有问题。但是当 B 点是线段跟圆弧的交点时，因为拟合的时候圆弧切点有误差，当切割完了后容易出现疤痕，所以不宜选基点 B 为切入点。

本文采取遍历双链表结构，按加工方向搜索链表结构，判断实体的属性，看是否有线段，如有则以线段的中点作为切入点。在此过程中一直判断，当前指针所指的点是否是 B 点，如是，说明没有找到线段，则程序结束，系统弹出对话框显示：图形中没有线段，本图形宜采用人工加工路线生成方法进行操作，程序的流程图如图 3-8 所示。

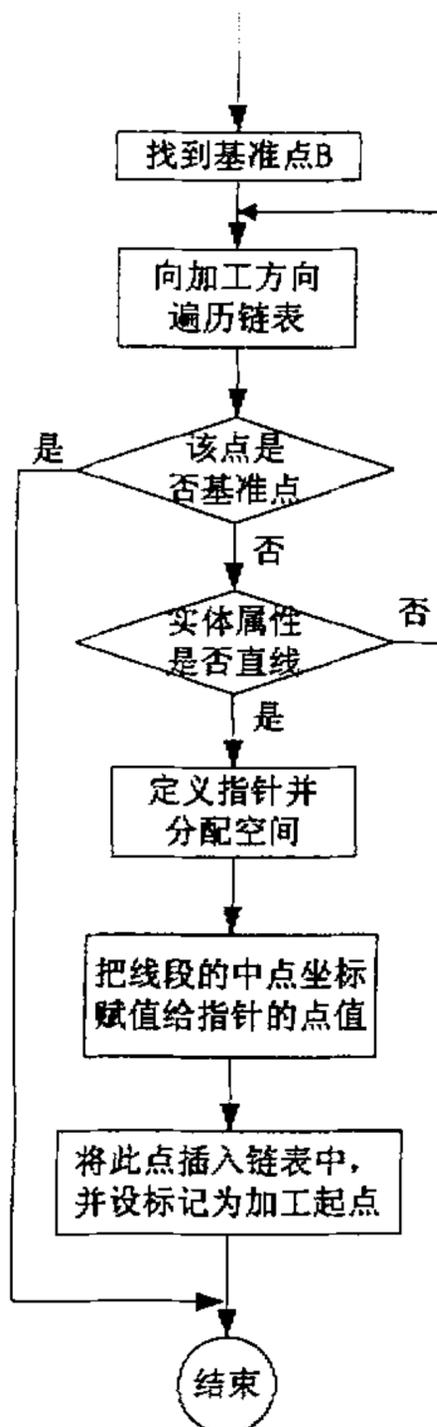


图 3-8 切入点的选定算法

### 3.2.4 电极丝轨迹的生成

在实际的加工中必须将加工对象的凹凸性、钼丝直径、放电间隙等工艺参数考虑进来，对基本加工路线进行调整，这样才能保证加工工件的精度。对于一个凸模类的工件来说，刀具的实际加工轨迹位于给定图形的外侧。

如果线切割加工零件钼丝半径为  $r_{\text{钼}}$ ，放电间隙（单边）为  $\delta_{\text{电}}$ ，则钼丝的偏移补偿量为  $f=r_{\text{钼}}+\delta_{\text{电}}$ 。图中实线为零件的轮廓，虚线为钼丝刀具圆心的实际运动轨迹。由此可见，在自动编程中考虑刀具的直径和放电间隙，从而确定刀具轨迹补偿量，对前面建立的基本加工路线进行相应的调整，是保证加工精度和质量很关键的一个步骤。

目标工件的轮廓一般由直线和圆弧组成。其它的曲线可由直线和圆弧拟和得到。工件轮廓可由以下三种连接方式中的一种或几种连接组成：(1) 直线与直线连接；(2) 直线与圆弧连接；(3) 圆弧与圆弧连接。在 AutoCAD 软件中，椭圆在 DXF 图形文件中是以多义线的

形式存储的，即拟合而成。因此在进行刀具实际加工轨迹的调整时，我们只需要对直线和圆弧的情况进行处理。

对圆弧只涉及半径  $R$  的增大或减小，即

$$R=R \pm f \quad (3.3.1)$$

对于直线，我们需将其在法线方向上平移  $f$ 。

由直线的两个端点  $P_S (X_S, Y_S)$ 、 $P_E (X_E, Y_E)$ ，可得直线  $L: Ax+By+C=0$  的参数  $A$ 、 $B$ 、 $C$  分别为

$$A=Y_E-Y_S, B=X_S-X_E, C=X_E Y_S-Y_E X_S \quad (3.3.2),$$

平移  $f$  后的直线方程为  $L: A' x+B' y+C' =0$ ，由两平行线间的距离公式

$$d=\frac{|C-C'|}{\sqrt{A^2+B^2}}, \text{ 可得 } f=\frac{|C-C'|}{\sqrt{A^2+B^2}}, \Rightarrow C'=C \pm f\sqrt{A^2+B^2} \quad (3.3.3)$$

关于正负号的选取，需根据是左平移还是右平移来确定。取  $P(x, y) \in L, P'(x', y') \in L'$  若是左平移，当  $y=y'$  时，有  $x'-x < 0$ ，即

$$x'-x=\frac{C-C'}{A}=\frac{\pm f\sqrt{A^2+B^2}}{A} < 0 \quad (3.3.4)$$

故当  $A=y_e-y_s > 0$ ，即位于第 I, II 象限时，取  $C' = C + f\sqrt{A^2+B^2}$ ，否则  $C' = C - f\sqrt{A^2+B^2}$ 。同理可以推出右平移时正负号的选取情况。

调整刀具实际加工轨迹时，需要考虑工件的凸模、凹模特性及加工方向的影响。

对于凸模加工，刀具的实际轨迹位于给定图形的外侧；对于凹模具的实际轨迹位于给定图形的内侧。加工方向决定直线平移的方向及圆弧半径的变化。将与加工方向相同的圆弧称为顺圆，反之为逆圆。

各种情况下的刀具轨迹调整见表 3-1。

表 3-1 刀具轨迹形成圆弧、直线调整情况表

		I II 象限 矢量直线	III IV 象限 矢量直线	X 轴 平行线	Y 轴 平行线	顺圆	逆圆
凸模	顺时针	左平移	右平移	$\Delta x > 0, \text{上}$ $\Delta x < 0, \text{下}$	$\Delta y > 0, \text{左}$ $\Delta y < 0, \text{右}$	$R+f$	$R-f$
	逆时针	右平移	左平移	$\Delta x > 0, \text{下}$ $\Delta x < 0, \text{上}$	$\Delta y > 0, \text{右}$ $\Delta y < 0, \text{左}$	$R+f$	$R-f$
凹模	顺时针	右平移	左平移	$\Delta x > 0, \text{下}$ $\Delta x < 0, \text{上}$	$\Delta y > 0, \text{右}$ $\Delta y < 0, \text{左}$	$R+f$	$R-f$
	逆时针	左平移	右平移	$\Delta x > 0, \text{上}$ $\Delta x < 0, \text{下}$	$\Delta y > 0, \text{左}$ $\Delta y < 0, \text{右}$	$R+f$	$R-f$

根据表 3-1 调整双链表中各个实体的特征参数，便完成了刀具轨迹的生成任务。

### 3.2.5 添加过度圆弧

在实际加工中，目标工件轮廓可能不是光滑轮廓，在考虑了刀具半径与放电间隙的情况下仍会出现拐角、尖角等情况。这些尖角容易造成断丝、短路、塌角缺陷等问题。另外，还出现空程现象，浪费加工时间。如图 3-7 所示，当按照第三节中的方法求得钼丝的实际轨迹之后，可以看到 ABC 为尖角连接。而实际上根据电火花线切割腐蚀加工的特点，可以知道，钼丝只需要沿圆弧 AB 路径运动即可满足加工需要，而不必走到 C 点。因此当钼丝运动到 A 点时，只要添加过渡圆弧就可以了。否则两曲线连接夹角越小，钼丝的中心位移量就比实际轮廓尺寸大得越多，造成加工的浪费。

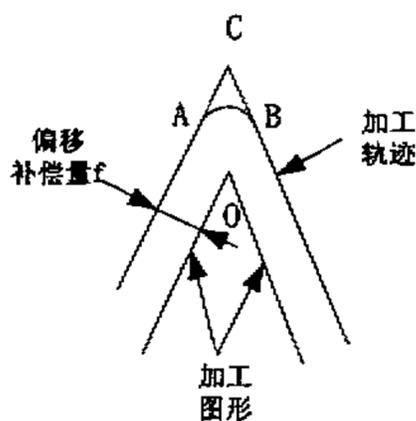


图 3-7 拐角过渡方式

于是采用加入过渡圆弧的办法对刀具轨迹进行进一步调整。取过渡圆弧的半径为  $f$ ，这样使整个加工轨迹圆滑过渡，有利于保护钼丝，节省加工时间，提高加工效率。

根据加工曲线的连接情况，其过渡方式有以下几种情况：

(1) 伸长型；(2) 缩短型；(3) 插入型

当为伸长型和插入型过渡转接时，需要添加半径为  $f$  的过渡圆弧，而当转接为缩短型时，为保证轮廓加工精度，可以不作处理。

可以同样根据矢量方法来判断转接为哪种方式。

对于添加过渡圆弧，要求一个与原有两相交曲线相切的圆弧。编程时可以采用枚举判断的方法来得到其圆心和两个交点坐标。这样就不需要判断该连接属于伸长型连接的哪种情况。枚举判断法需要事先确定所有枚举的情况和结束条件，当满足条件时，枚举结束，保存结果。其判断流程如下：

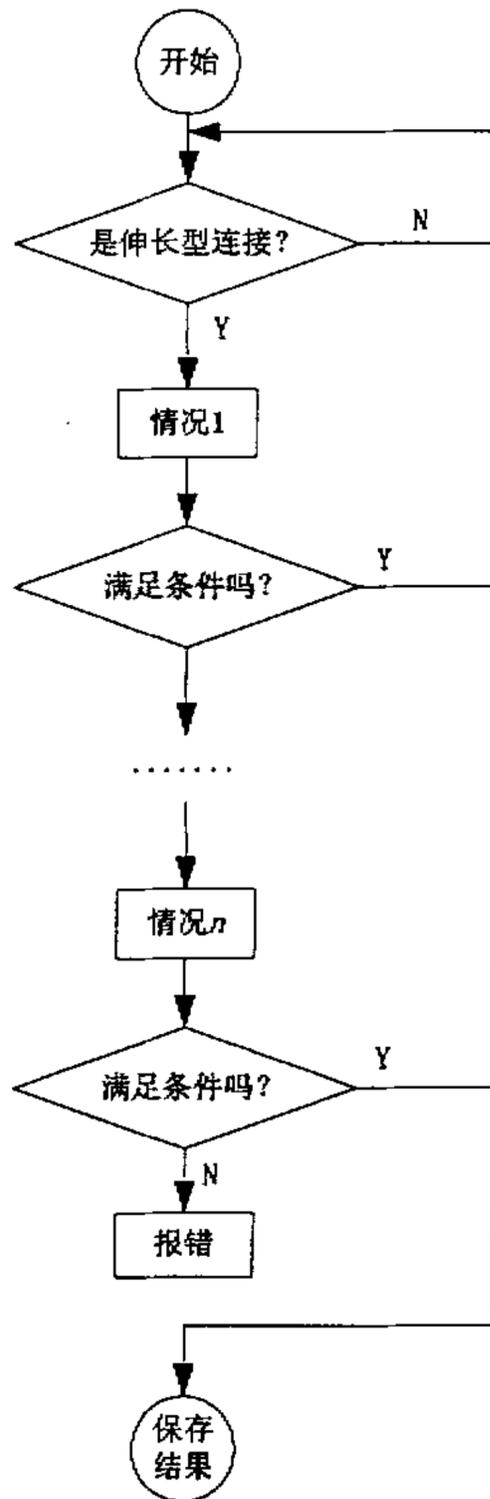


图 3-8 枚举法流程图

用这种办法，可以简化编程思路，容易实现。

添加必要的过渡圆弧之后，刀具实际运动轨迹得到最后确定。

### 3.3 后置处理

后置处理就是结合特定机床把系统生成的轨迹转化成机床代码指令，生成的指令可以直接输入数控机床用于加工，这是自动化编程系统的最终目的。

几十年来，全世界出现了许多系列的线切割机床，其相应的加工指令也有了国际 ISO 和 EIA 标准。然而，国产机床广泛采用的是 3B 格式的加工指令。许多自动编程系统仅仅支持某一种代码格式，兼容性比较差。为了与国际接轨，ISO 代码进行数控编程是线切割编程和控制的发展趋势。考虑到生成程序的通用性，本软件针对不同的机床，可以设置不同的

机床参数和特定的数控代码程序格式，同时还可以对生成的机床代码的正确性进行校验。

本节以 3B 指令为例，讨论了数控代码指令化的过程，它采用 ObjectARX 语言读取实体组码数据来转化成 3B 加工代码，实践证明其精确、实用、效率高。

### 3.2.1 3B 代码的格式<sup>[71]</sup>

3B 格式如下表所示：

表 3-2 3B 程序格式

B	X	B	Y	B	J	G	Z
分隔符	X 坐标值	Y 坐标值	分隔符	分隔符	计数长度	计数方向	插补线型

其中：X、Y、J 均取绝对值，单位为微米。其含义直线和圆弧各有不同：

#### (1) 图素为直线时的格式含义：

X、Y—直线终点横、纵坐标（直线起点为坐标原点）；

G—计数方向，当直线起点靠近 X、Y 轴时分别取 G<sub>x</sub>、G<sub>y</sub>；

J—计数长度，直线在计数方向的坐标轴上投影的长度；

Z—加工指令，依据直线终点所在象限不同而定义，依次有 L1、L2、L3、L4。

#### (2) 图素为圆弧时的格式含义：

X、Y—直线终点横、纵坐标（圆心为坐标原点）

G—计数方向，当圆弧终点靠近 X、Y 轴时分别取 G<sub>y</sub>、G<sub>x</sub>；

J—计数长度，圆弧在计数方向的坐标轴上投影的长度之和；

Z—加工指令，当被加工圆弧为逆弧时，依据圆弧终点所在象限不同而定义，依次有 NR1、NR2、NR3、NR4；当被加工圆弧为顺弧时，同理分别有 SR1、SR2、SR3、SR4。

计数方向 G 和加工指令 Z 的选择以及圆弧插补线形表和计算长度计算表格很多教科书都有介绍，这里不再展开讨论。

### 3.2.2 程序设计方法<sup>[49]</sup>

程序首先建立一个由用户自定义的 NC 文件（该文件名以 3B 为扩展名），然后取得加工起点的指针，保存用户在弹出对话框的输入参数：加工方向（顺时针、逆时针）、刀具补偿值、冲裁间隙等。接着根据点的实体信息（直线或是圆弧）按照前面的方法进行不同的处理。程序流程如下图 3-14 所示：

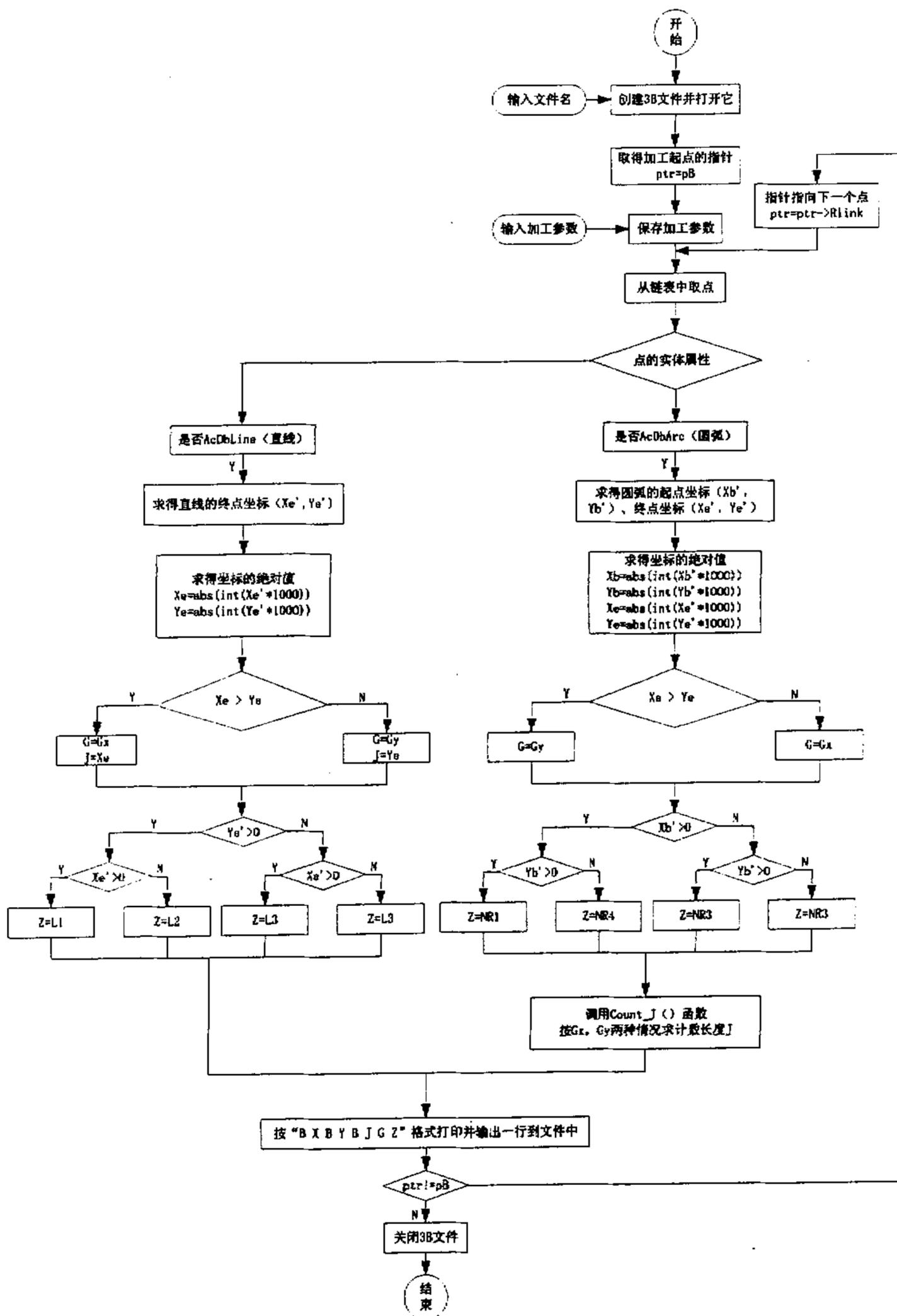


图 3-14 代码转换程序流程图

程序的具体实现将在第五章详细介绍。

### 3.3 数控加工指令集的仿真

为了防止在加工路线的处理过程中出现异常问题,造成加工出现错误或缺陷,有必要在实际加工前,对目标工件进行加工模拟,即利用插补计算所得数据,用绘图方式在计算机屏幕上模拟实际加工。有利于及时发现错误,避免产生不必要的损失。

本系统的加工模拟主要完成以下功能:

1. 模拟刀具的实际加工轨迹。
2. 显示加工方向(顺时针或逆时针)和加工起始点,有利于用户调整钼丝与工件的方位关系。本系统的加工起点的自动选择原则为:对于封闭曲线,首先选择距离 X 轴最近的点,如果有多个点满足条件,则选择其中距离 Y 轴最近的点。对于不封闭曲线,上述原则同样使用,只是选择起点时只需要比较起始点和结束点。
3. 对用户所输入的加工零件的轮廓最大尺寸进行检验,如果插补数据超过从机 RAM 的最大存储能力,则需分成多次加工。
4. 对用户所设置的参数进行检验,如果因参数不当而导致插补、加工无法进行,则及时给出提示。

### 3.3.1 加工模拟的原理

1. 从双链表中提取加工实体(已经按加工顺序排序),检验其是否有无法插补的实体,如果有则报错。
2. 采用相应的插补算法,对各实体进行插补运算,并将插补所得数据(步进电机的送给指令)保存在创建的一个临时数据文件中。当实际加工结束后,自动将其删掉。
3. 从数据文件中读取数据,仿照步进电机的加工原理,将加工过程在屏幕上显示出来,并对加工起点和加工方向进行突出显示。

由于屏幕的大小有限,在仿真时除了可以由用户设定仿真比例外,还可以在程序中根据屏幕和工件实际尺寸选择合适的模拟比例。

### 3.3.2 程序算法流程图<sup>[48]</sup>

在模拟显示过程中,走丝路线分为 2 种:直线、圆弧。因为用 3B 指令编程,所产生的图形都可以用直线和圆弧来表示,因此在进行设计过程中,可以只考虑这两种情况。

读 3B 指令文件时,把指令各字符分离出来,分别送到数组中去,具体做法如下:判断字符,在第一个字符 B 之后,将其数值赋给 X 轴;在第二个 B 之后,将其数值赋给 Y 轴;在第三个 B 之后,将其值赋给 J (J 为步长)。接着判断是画直线还是圆弧,是画直线还是要画斜线,画圆弧是画顺圆还是要画逆圆等。详细的程序流程图见图 3-15 所示。

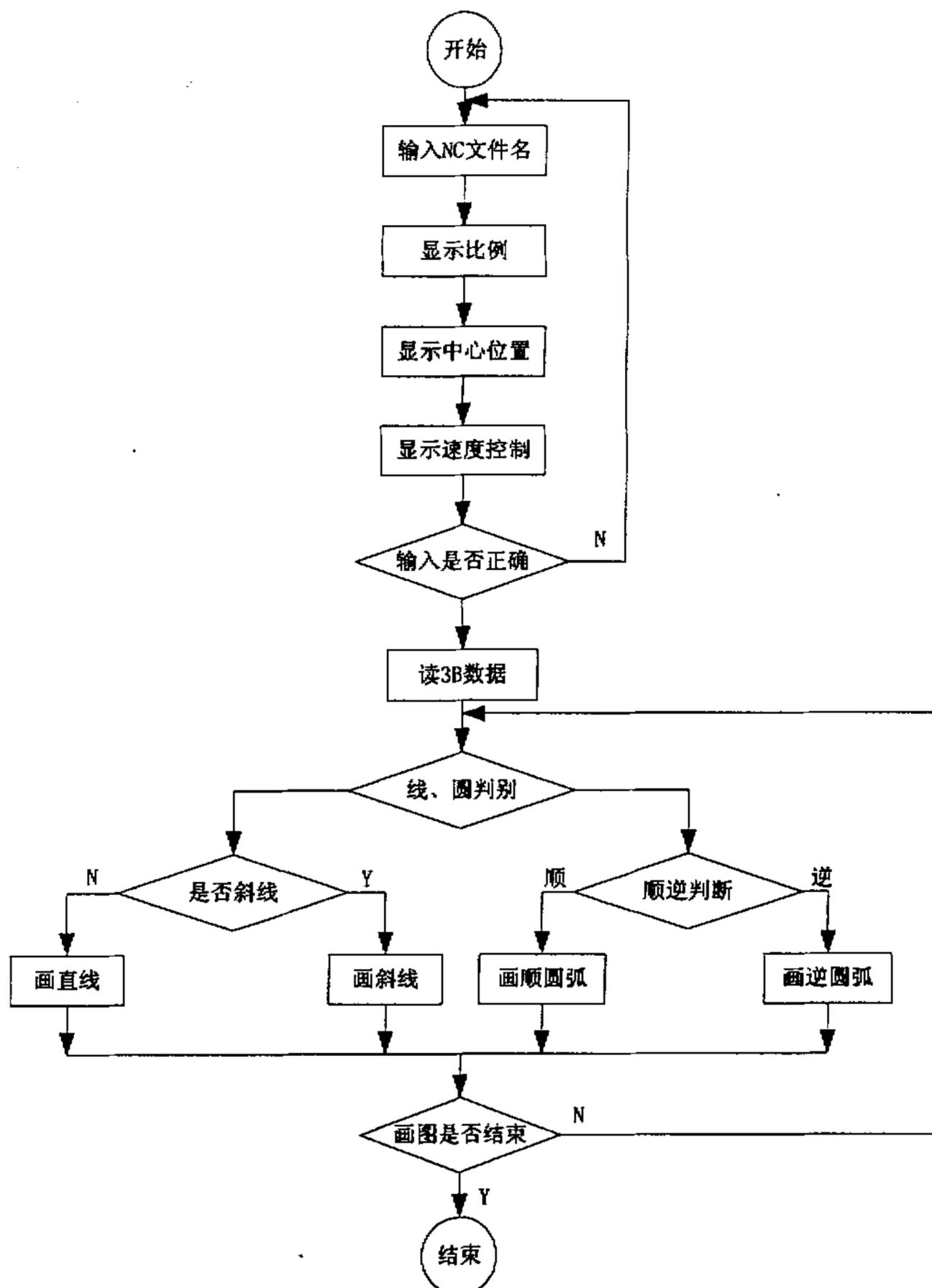


图 3-15 程序流程图

### 3.4 数字图象的矢量化处理<sup>[56][74]</sup>

在当今生产以多品种、小批量和创新设计为特征的环境下，仿形加工作为一种传统的加工方法，表现出了其特有的灵活性和敏捷性。利用线切割加工进行仿形加工将广泛应用于模具、轻工和工艺品制造领域。本文对线切割仿形加工的方法进行了深入研究，并提出了一个基于图像矢量化线切割仿形加工的方法。这一方法首先通过扫描设备将实物或手绘图片扫描为数字图像，再通过矢量化将图像转化为图形，进而用普通的线切割自动编程平台进行

编程、加工。

### 3.4.1 基于图像矢量化的一线切割仿形加工的设计思想

纪念章、旅游工艺美术品、玩具、首饰、眼镜等行业都要用到复杂图形的冲压落料模具，而这些行业的模具相当一部分都是用线切割加工的。但是目前普通的自动化编程系统，都很难对那些从设计的美观性随意勾画轮廓的零件进行自动编程。因为这些图形尺寸标注不严格，没有直线、圆弧起点、终点、圆心等的确切值，甚至根本没有尺寸标注，此外这类取形基本上都是由多段不规则曲线组成，即使人工用直线和圆弧程序段很多，繁琐不堪。因此，开发根据零件的样品或设计图样，借助于扫描仪自动输入零件的轮廓，经计算机处理后，自动生成零件的加工程序的编程系统，使这类零件的程序编制更加快捷、轻松，已成为一种必然的市场需求。

尽管，目前关于矢量化算法研究已经广泛，并且相关的矢量化功能也是比较强。但是如果仅仅为线切割加工车间去配一套上万元或是几十万元的矢量化软件，这是不现实的。目前，流行的线切割自动编程软件中很少带有矢量化功能的，即使有，如新版的CAXA软件，可是使用起来的效果不佳，有待于以后的进一步完善。因此，研究矢量化算法把它应用到线切割加工中去，是非常有意义的。本文结合线切割加工的特点，提出了一种具有针对性的矢量化功能解决方法。

基于图像矢量化的一线切割二维创新设计零件加工的设计思想是利用计算机强大的图像处理能力和扫描设备的实物扫描能力，对实物或者手绘图片进行线切割仿形加工。其基本的工作流程如图3-16所示。

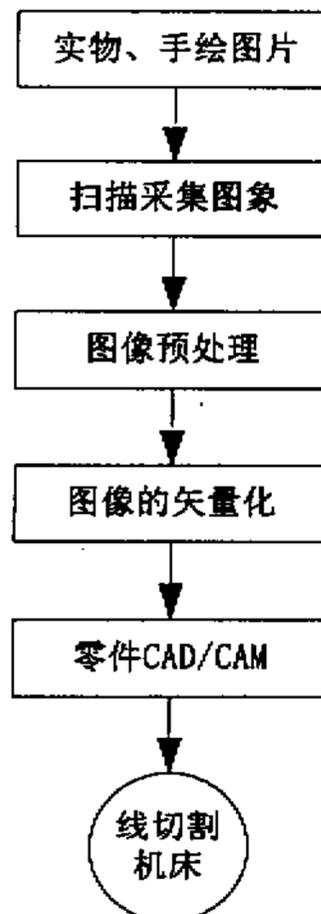


图 3-16 工作流程图

在这一过程中扫描采集图像是将所要加工的产品模型扫描为计算机可以存储的数字图像。图像预处理是将图像转化为方便下一步处理的图像格式。主要包括图像灰度化和细化。图像的矢量化是将图像中模型轮廓提取出来并转化为 DXF 格式的标准图形交互文件。最后将矢量化后的 DXF 文件提交已有的线切割数控系统进行自动编程、加工。

### 3.4.2 基于图像矢量化的线切割仿形加工的关键技术

#### 一、图像的预处理

利用图形扫描仪，将所需编程零件的图样或样板放在扫描仪上，驱动扫描仪，将零件的信息以图像的形式输入到计算机。如果放在扫描仪上的是黑白图纸，那么得到的图像是二值图像；如果是样板或背景与图形对比不明显的图纸，则输入的图像是灰度图像；如果是灰度图像，则首先要将灰度图像转化为二值图像。

图像的预处理是重要的一步，它处理的好坏将直接影响到后面矢量化处理的质量。在这一步将对图像进行灰度化和对轮廓进行细化。灰度化的算法比较简单，在此不再论述。轮廓的细化是比较重要且复杂的工作，我们采用了一种简单而有效的算法。

在图 3-17 中我们看到细化过程中可以根据某点的相邻点的情况来判断该点是否可以删除：

- (1) 的中心点不能删除，需要的是骨架，所以中心点不能删除；
- (2) 不能删除，原因同上；
- (3) 可以删除；
- (4) 不能删除，删除后原来相连的部分将断开；
- (5) 可以删除；
- (6) 不能删除，这是直线的端点；
- (7) 不能删除，因为孤立点的骨架就是其本身。

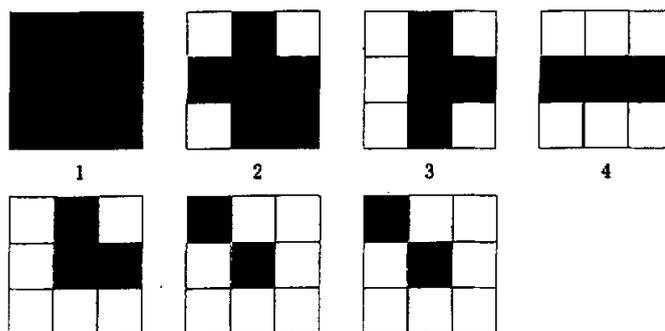


图 3-17 某点的相邻点情况

总结可以得到如下的判据：

- (1) 内部点不能删除;
- (2) 孤立点不能删除;
- (3) 直线端点不能删除;
- (4) 假设 P 点是边界点, 删除 P 后连通份量不增加, 则 P 点可以删除。

根据以上的判据, 可以做一张判断表。表中共有 256 个元素, 元素的类型为 BOOL 型, 这样可以表示某点相邻 8 个点的所有情况。根据 8 个相邻点的情况查表, 若表中元素为 1, 则可以删除, 否则保留。图 3-18、图 3-19 是采用这种算法细化的手绘扫描图前后比较。

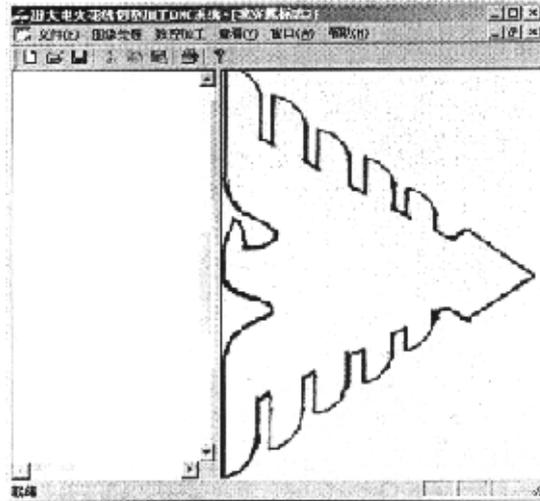


图 3-18 细化之前的图

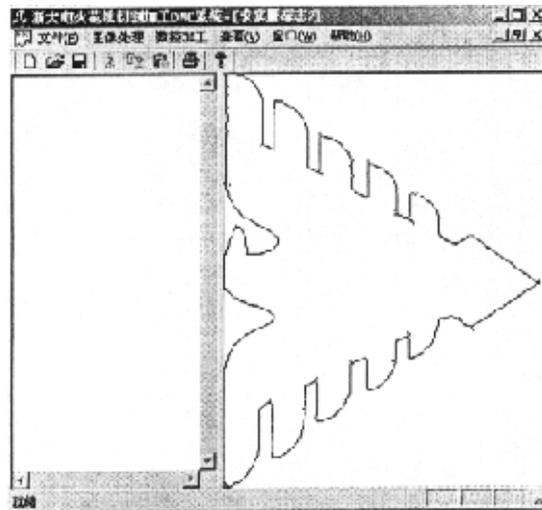


图 3-19 细化之后的图

## 二、图像轮廓的矢量化

图像轮廓的矢量化是将图像的轮廓信息提取出来, 转化为标准的图形交换文件—DXF 文件。这里考虑到线切割加工的特点, 我们采用直线拟合轮廓线。其基本思想是: 在轮廓为直线的地方用一条直线表示, 在轮廓为曲线时用若干条直线表示。实验证明, 只要直线的条数足够多, 并且大于电火花线切割机的员小步长, 这种算法的精度是可以达到要求的。

$(i-1, j-1)$	$(i-1, j)$	$(i-1, j+1)$
$(i, j-1)$	$(i, j)$	$(i, j+1)$
$(i+1, j-1)$	$(i+1, j)$	$(i+1, j+1)$

图 3-20 点  $(i, j)$  周围的 8 个点坐标关系

对于一个点  $(i, j)$  其周围 8 个点的坐标关系如图 3-20 所示。采用直线拟合的算法如下: 假设点  $(i, j)$  的上一个轮廓点是  $(i+1, j-1)$ , 本次计算从  $(i, j-1)$  开始寻找下一个轮廓点, 顺序依次为  $(i, j-1)$ ,  $(i-1, j-1)$ ,  $(i-1, j)$ ,  $(i-1, j+1)$ ,  $(i, j+1)$ ,  $(i+1, j+1)$ 。如果其中有点  $P$  是轮廓点, 再判断  $P$  是否与  $(i+1, j-1)$  和  $(i, j)$  在同一条直线上, 如果是 (如点  $(i-1, j+1)$ ) 则以  $(i-1, j+1)$  为中心点,  $(i, j)$  为上一轮廓点重复上面的步骤。如果不在同一直线则记录上一直线到 DXF 文件中, 并以  $P$  点为中心点,  $(i, j)$  为上一轮廓点重复上面过程。如果所有 7 个点没有点在轮廓线上, 则此条轮廓线扫描结束。图 3-21 为这个算法的流程图。

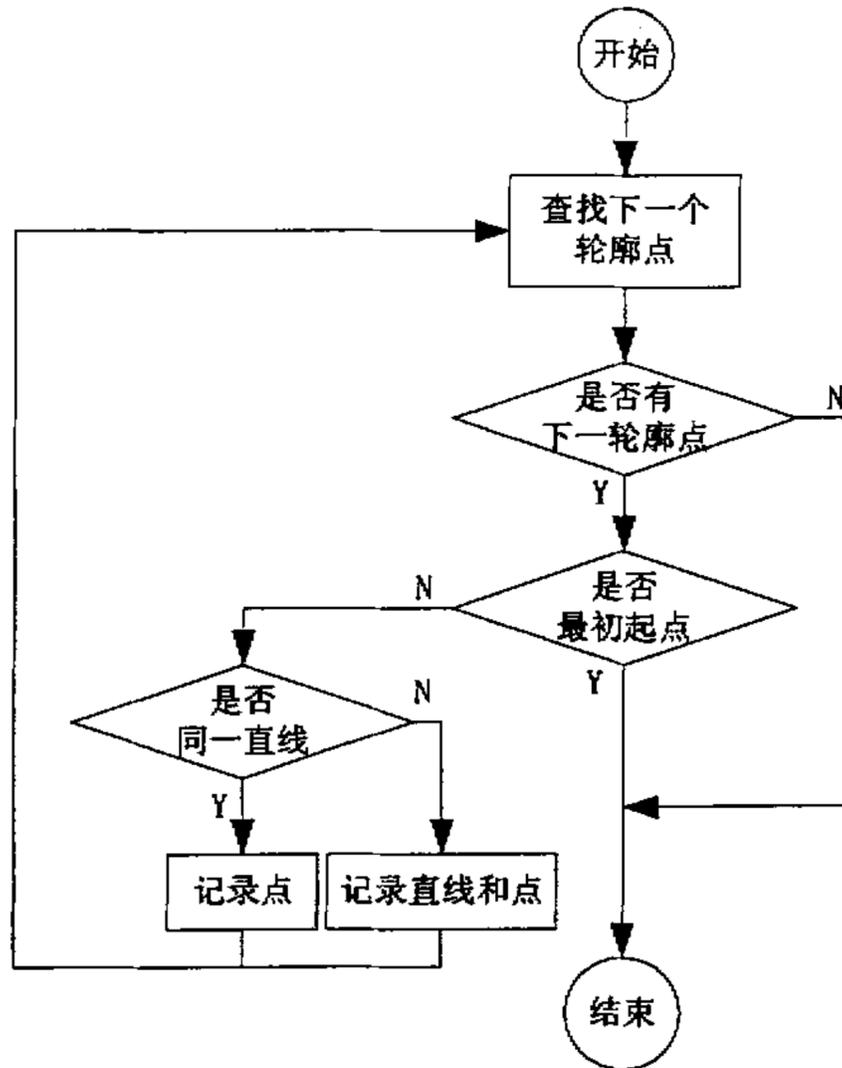


图 3-21 矢量化算法流程图

通过矢量化后的图形文件(DXF 文件)就可以拿到通用的 CAD 或者其他线切割加工自动编程软件中加以编辑和修改, 再进行加工。图 3-22 是图 3-18 经过矢量化以后在 AutoCAD 中的显示。

细化和矢量化的算法实现将在第五章详细介绍。

### 3.5 本章小结

本章研究了电火花线切割 DNC 加工系统自动化编程系统模块的关键技术。详细分析了参数化生成加工路线的方法、钼丝轨迹的后置处理、指令代码的仿真、二维创新设计图的矢量化处理。

1. 对于常规线切割零件的加工完全没有必要按部就班的操作, 用户只要输入加工工艺参数, 便可达到加工路线的自动生成。即使遇到特殊的零件, 也可以通过手工生成加工路线来进行处理。

2. 在 AutoCAD 平台上可以直接从当前图形的图形数据库中读取信息, 从而获取零件信息更加直观和准确, 从而免去了生成、读写 DXF 文件操作的繁琐步骤。

3. 对那些从设计的美观性随意勾画轮廓的零件可以借助于扫描仪自动输入零件的轮廓, 经计算机预处理和矢量化处理, 再调用自动编程系统生成零件的加工程序。其中矢量化算法的好坏直接影响到系统的处理能力, 所以非常关键。

## 第四章 电火花线切割 DNC 加工系统通讯模块的关键技术

### 4.1 数控线切割 DNC 通信的解决方案

DNC 主机与数控系统的联网目前主要有计算机局域网 (MAP 或 TCP/IP)、现场总线和点对点串行通讯三种方式。本文针对目前存在的几种线切割机床的通信接口展开讨论。

#### 4.1.1 带网络接口的线切割数控系统

目前国外已经出现带有以太网接口的线切割机床,并且据业内人士透露国内一些机床厂家也在试制该产品,故不久的将来的这种线切割机床将会大量出现,并且随着网络制造的发展趋势,此类产品将逐步替代旧产品。

另外,把一部分控制线切割机床的 486 以上微机进行改造,为其配备网络适配器,可以通过局域网进行通讯。

此种网络通信传输介质主要有:双绞线、同轴电缆和光纤三种。网络拓扑结构主要有星型、环型、总线型和树型四种。网络通讯协议标准有 MAP/TOP、TCP/IP、以及 DECNET 和 Ethernet 等。

#### 4.1.2 带串口的线切割数控系统<sup>[32]</sup>

对于带有 RS-232C 或是 RS-422 串行接口的线切割控制器、微机或是工控机,宜采用点对点串行通讯类型。如果有多台线切割数控系统,则采用 MOXA-C114 智能多串口适配器,以此为基础可方便可靠地实现多路串行通信能同时进行。这里使用的协议,由自己设定,因此编写多串口通信协议是关键。

点对点型接口联接常用 RS-232C 和 20mA 电流环、RS-422 和 RS-485。其结构简单、传输可靠,在拓扑上为星形结构,其网络协议简单,通讯速率一般在 110-9600 bit/s 之间。总之,这种通讯联接技术应用较早,技术上比较成熟,系统组成也相对简单,易于实现。可以说它是目前大家讨论较多而又经济可靠的一种方案。

在整个 DNC 系统中,管理控制层由一台高档微机、RS-232C 通信扩展板及有关外围设备组成,用来完成线切割零件程序的管理和传送等功能;数控设备层由数控机床和相应通信接口组成。可以根据具体情况,以 8MHz 8031 单片机为基础开发一个智能 DNC 控制台。该智能控制台具有双向传送 NC 指令且具有系统状态采集与远程控制等功能,除了自动完成 NC 程序上传、下调工作外,还有程序缓冲、查询上位机文件库的目录,下传任务单、上传加工情况、传送呼叫等,能较好满足边加工边传递 NC 程序的要求,同时解决了远加工区机床操作人员与机房管理人员联系不方便的问题。

### 4.1.3 不带串口的单板机线切割数控系统<sup>[32]</sup>

不带串口的老式单板机控制器，可以利用其自带的电报头和光电头接口或是纸带穿孔机，通过外接式通讯适配器与 DNC 主机实行通信。

对国内仅有纸带读入机磁带放音机的经济型数控线切割机床只能配作下传型 DNC 接口板；而未配 RS-232 串行通信口，具有并行输入、输出口数控系统（如 FANUC 7M）可以按需要配作下传型 DNC 接口板、双向型接口板以及智能型控制台；具有 RS-232C 通信口的数控系统（如 FANUC 6M）可直接通过 RS-232 实现数控程序上、下传功能，当然可按需要开发成智能控制台进行通信；具有较强 DNC 功能的数控系统可直接通过 RS-232C 实现智能通信功能。

总之必须对不同类型的数控机床，按不同的接口及用户需要开发 DNC 接口板或 DNC 智能控制台，并按不同的通信协议开发通信软件。

## 4.2 异构线切割机床 DNC 通讯的实现<sup>[57]-[62]</sup>

电火花线切割加工因加工效率高、能满足一般曲线曲面的加工要求，再加上设备投资相对便宜等而被广为应用。到目前为止，线切割机床控制系统主要有工控机、PC 机、单板机、单片机等几种档次。其中，单板机和单片机控制器在性能和价格上相差不大，比较而言，用单板机的控制系统多为 80 年代中期产品，虽然工作稳定，但体积较大、硬件较复杂、维修不便。而在近几年渐由以单片机为主控单元的系统取而代之。数控技术一方面在向高精度、高效率、高柔性和智能化发展，同时也在低成本自动化方面取得了很大的成就。事实上，单片机控制系统就是经济型数控线切割的代表。因此，解决 DNC 主机与主流数控系统之间的 DNC 通信问题就显得尤为重要。

### 4.2.1 DNC 主机与单片机控制器的通信

目前国内广泛使用的快走丝线切割单片机控制系统主要由高频电源、步进电机驱动电源、单片机控制器等三部分组成，许多单片机控制器由 8031 构成，显示均为 8 位 LED 数码管，键盘通常为 24 键，能满足监控、编程输入(3B)、加工进给、正割、逆割、单步、单段、等快走丝线切割机床的一放需求。本文以 8031 单片机为例，旨在提出一种 DNC 主机与单片机系统的解决方案，确保不提高成本，实现多台线切割机床的 DNC 通信。

#### 一、多机通信原理

多机通信中，要保证主机与所选从机实现可靠通信，必须保证通信接口具有识别功能。8031 串行口控制寄存器 SCON 中的 SM2 就是满足这一通信要求的多机控制位。

8031 单片机的串行通信有四种方式，即方式 0、方式 1、方式 2、方式 3，其中方式 3

是波特率可变的 9 位数据异步通信方式，是多机通信的常用方式。在此方式中，传送一帧数据共 11 位，1 位起始位，8 位数据位，1 位可编程为“1”或“0”的第九位数据和 1 位停止位。其中第 9 位数据可作奇偶校验位，也可以“地址帧”或“数据帧”的识别标志位。在方式 3 中，接受完一帧数据后，当同时满足以下两个条件时：

- (1) RI=0;
- (2) SM2=0 或接收到的第 9 位数据位“1”

标志位 RI 被置位，产生接受中断。否则接收到的信息作废，RI 不被置位。利用以上两个条件，可以方便的区分通信数据是“地址帧”还是“数据帧”。PC 机的串行通信由其接口芯片 INS8250 完成，不具备多机通信功能。但可以灵活使用 8250，用软件实现上述功能。8250 可以发送几种字长，其中一帧最长为 11 位，与 8031 单片机发送的帧格式相比，差别仅在第 9 位数据，即 PC 机的 8250 发送的第 9 位数据是奇偶校验位，而不是相应的地址 / 数据标志位。但是，我们可以通过对 8250 的线路控制寄存器 (LCR) 进行编程设置，使它的数据传送格式与 8031 保持一致，使 8250 的奇偶校验位形成正确的地址/数据标志值。从而使 DNC 主机与 8031 单片机实现多机通信。

由于本系统不仅是简单的地址数据的传输，有时需要 DNC 主机发送操作指令给从机，从机或进行某种加工操作，和准备接受数据，或向 DNC 主机发送状态信息，这时，上述功能就不能满足需要。需要在通信协议中增加命令字。为避免命令和数据混淆，这里采用设置标志位的方法来实现可靠通信。

## 二、通信协议

在通信中，命令和数据采用相同的发送格式。为避免产生将命令作为数据的误操作，采用图 4-1 所示的通信方式：即命令紧跟在地址命令之后。

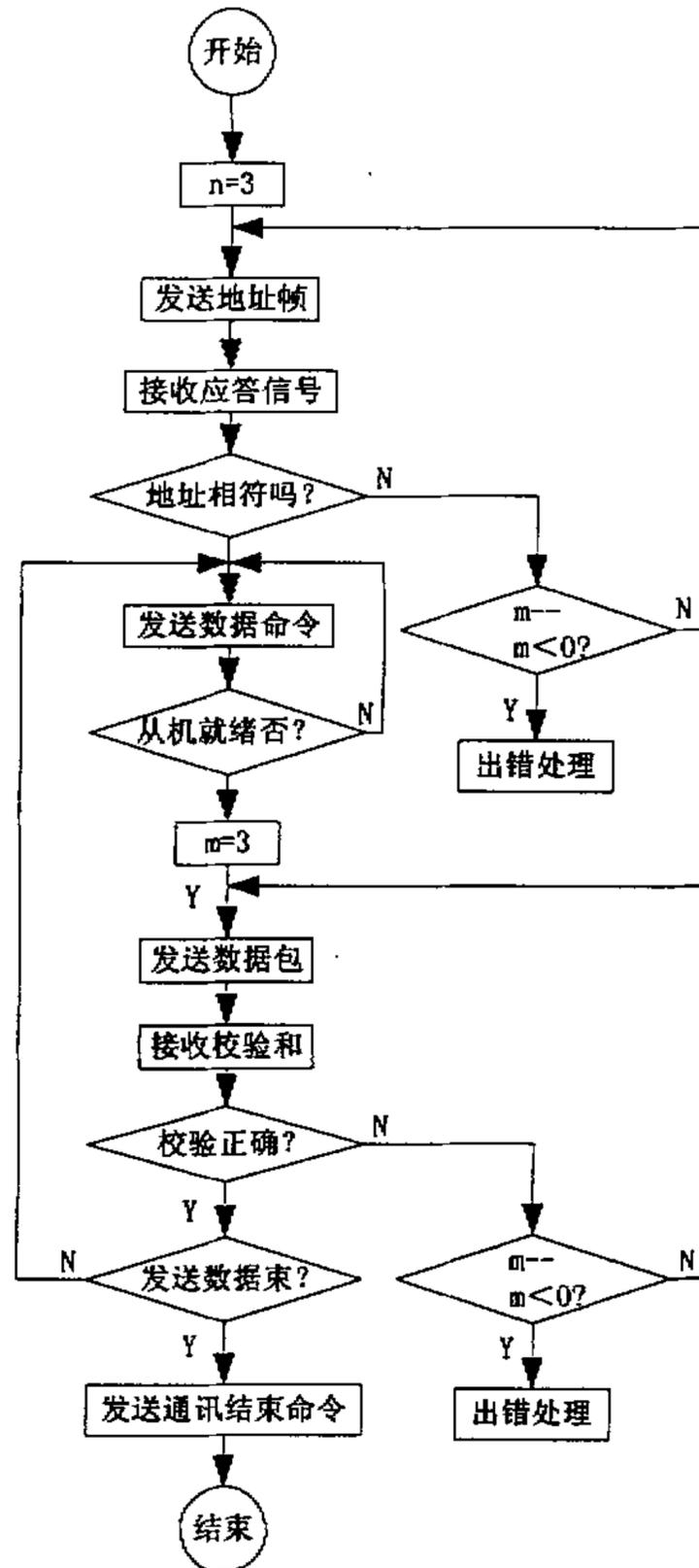


图 4-1 主机通信流程图

在单片机中断服务程序中设置一 FLAG 标志位，用来判断接收到的是命令还是加工插补数据。当找到从机后，该从机将 SM2 清零，并将该标志位置“1”。当该标志位为“1”时，将接收到的信息作为命令处理；只有当接收到的命令为接收数据时，将该标志位置“0”。当该标志位为“0”时，认为接收到的信息为加工数据，将其保存至扩展的 RAM 中。当数据传输结束后，将 SM2 置“1”，使单片机免受主机对其他从机操作的干扰。这样就解决了多机通信中地址、命令和数据的识别问题。主机和从机的通信协议如图 4-2 所示。

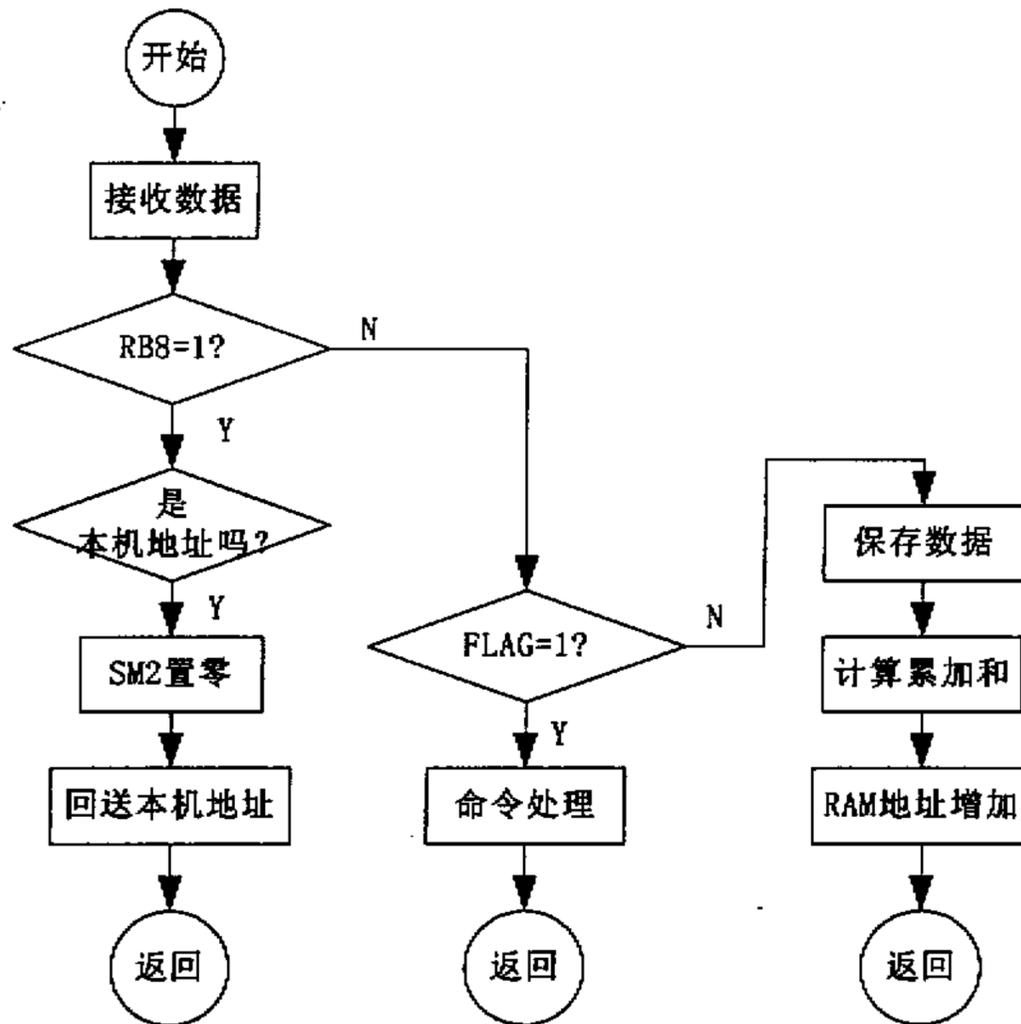


图 4-2 从机串行口中断服务程序流程图

在主机的出错处理中，向从机发送复位命令，然后返回。

从机软件在开始时作好相应的初始化工作。在从机串行口中断服务程序流程中的命令处理部分，按照不同的命令执行相应操作。只有在接收数据命令中对标志位 FLAG 进行操作，将其清零，为接收数据作好准备。在接收数据程序中，在每接收一个数据包之前，将 RAM 地址保存起来，当出错时，将该地址重新赋给当前地址指针，以接收重发的数据。多次接收均出错时，接收主机发来的复位命令，将当前地址置为 0000H。

根据以上协议，就可以实现较为可靠的主从多机通信。

### 三、通信程序设计

#### (1) DNC 主机通讯程序设计

本系统中 DNC 主机的操作系统为 Windows2000，采用的开发工具是 Visual C++。虽然微软提供了一个具有串行通信功能的 MSCOMM 控件，运用起来十分方便，但 MSCOMM 控件提供的 11 位信息帧中，第 9 位只能作为奇偶校验位使用，MSCOMM 控件设备没有提供设置第 9 位的功能，因此利用 MSCOMM 控件难以实现多机通信。

Windows 是一个与设备无关的、基于消息驱动操作系统，不提用户用户对硬件进行操作，而是由系统自动进行处理。Windows API 提供了一系列标准的串行通讯函数，用户可以利用这些函数来完成串口通信。由于这类函数比较多，且其所带的参数页比较复杂，因而使用很不方便。本文使用使用 CSerialPort 类封装 Windows 提供的 API 函数。这个类除了初始化串口，读写等基本功能外，还可以扩展的功能有，设置并检验校验码，使用多线程，对

串口进行轮巡等。MSComm32 控件也可以做到对串口进行轮巡，但要为每一个串口添加一个控件。MSComm32 控件的源码是不公开的，所有的修改必须在它的外面，以至于代码的重用性比较差。

串行通信通常有 2 种方法：查询法和事件驱动法。为提高微机 CPU 的利用率，本文在单片机发送数据和 DNC 主机接收数据过程中采用事件驱动法。为了捕获通讯时所发生的事件，本文利用 MFC 类库提供的全局函数 AfxBeginThread 创建了一个线程以监视串口有无数据，若串口有数据，则发送消息 WM\_COMM\_RXCHAR 给应用程序，应用程序根据消息响应处理函数如 OnCommunication，运行相应的接收数据处理程序，以完成工作。下面结合 CSerialPort 类对串口初始化和创建监视线程，程序部分源代码如下：

```

CSerialPort  m_Ports[4];//定义对象数组，4 表示串口个数
    for (int i = 0; i < 4; i++)
    {
        if (m_Ports[i].InitPort(this,1,9600,'n',8,1))
            //缺省设置：串口 i 的波特率 9600，无奇偶校验位，8 位数据位，2 位停止位，共 11 位
            //一帧

            m_Ports[i].StartMonitoring();
        else
        {
            // port not found
            m_Edit[i].SetWindowText("NOT FOUND");
            m_Edit[i].EnableWindow(FALSE);
            m_ListBox[i].EnableWindow(FALSE);
        }
    }

BOOL CSerialPort::StartMonitoring()//创建监视线程的类成员函数
{
    if (!(m_Thread = AfxBeginThread(CommThread, this)))
        return FALSE;
    TRACE("Thread started\n");
    return TRUE;
}

```

## (2) 单片机通讯程序设计

单片机下通讯程序一般使用 MCS-51 汇编语言编写，相关具体实现方法很多地方都有介

绍，这里不再赘述。需要特别注意的是：从机收发数据一定要遵从双方的握手协议，否则就无法实现正常通信。

#### 4.2.2 DNC 主机与单板机控制器的通信

目前在线切割控制器方面，国内以单板机作为控制器仍然较多，老一代控制器还在使用。把微机上自动编程编制的加工程序用并口输出，再人工用键盘输入单板机控制器，这样做不仅费时，并且容易出错。分析老一代线切割控制器，它们都带有电报机头纸带信号输入接口，本文介绍利用微机并口与单板机控制器的电报机头接口进行简单连接，并编制少量的程序，实现微机向单板机控制器传送加工数据的方法。从而解决了DNC通讯中兼容老式线切割机床的问题。

##### 一、硬件电路原理

单板机线切割控制器，都有电报机头纸带输入接口，纸带输入的数据一般以五单位的3B格式居多。五单位的+12V电报机头纸带输入接口的接口信号线如图4-3。其中， $I_1 \sim I_5$ 为数据信号线，“地”为数据信号地线，+12V与DP电报机头线圈驱动，当DP为“0”低电平时，线圈驱动，驱动齿轮进一步，再读入下一个数据。

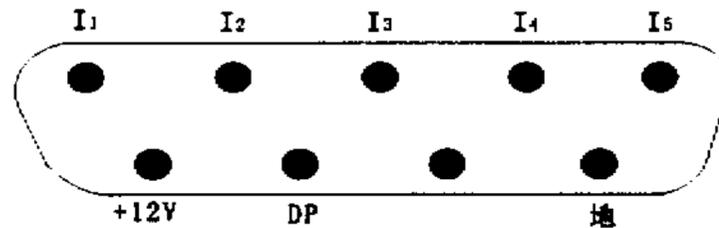


图4-3 电报机头纸带输入接口

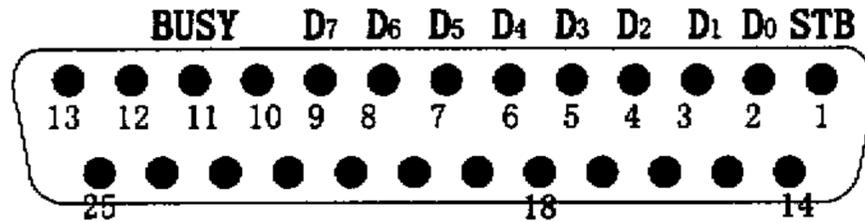


图4-4 微机并口适配器输出接口

微机并口输出接口为25芯插口，如图4-4 1脚为并口送出的选通信号STB线，STB信号是通知外设，数据已准备好，请求外设取走数据；2~9脚为接收或发送数据的数据D0~D9信号线；11脚为PC机读入外设状态的忙信号BUSY线，它的高电平+5V表示外设正在工作或没作好接收数据的准备，低电平0V表示外设已作好接收数据，请求微机发送数据。

以上接口电路中，微机并口信号的电平为TTL电平，即低电平为0V，高电平为+5V；单板机控制器的电报机头输入接口中驱动信号DP低电平0V为有效，悬浮时无效，所以DP信号可作为与外部通讯的联络线，为了使DP信号线能与TTL电平连接，我们在DP与+12V、DP与地之间分别接入10K $\Omega$ 电阻，把DP信号线电平降成TTL电平后，就能直接与打印接口的BUSY信号线相接。单板机控制器输入接口的数据信号线 $I_1 \sim I_5$ 分别接至微机并口的D0~D4信号线上。信号线连接如图4-5

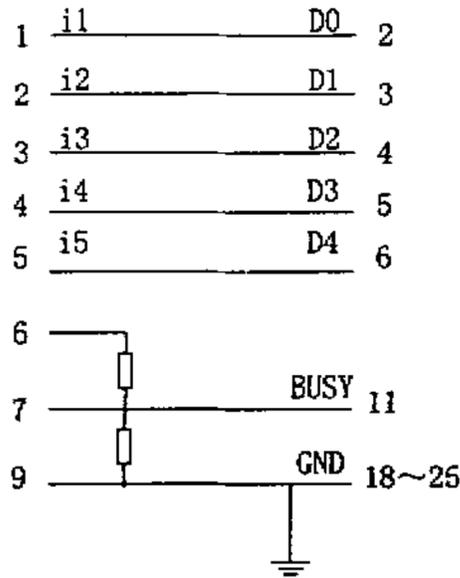


图4-5 信号线的连接

## 二、软件设计

标准的并口共有36根信号线，但是我们只用到最主要的几种信号：STB、D0~D7、BUSY，其他信号是为了更好地控制和监视外设地工作。并口与外设进行数据交换的信号可分三类：数据信号、状态信号和控制信号，信号对应的端口是：

(1) 数据端口。数据端口为可读写带输出锁存的8位数据，高电平为1，低电平为0，它的I/O地址为378H；

(2) 状态端口。状态端口有5位只读状态数据，在一个8位的字节数据上，如下表，其中D7即为BUSY信号，当外设输入高电平时D7为0，低电平为1，它的I/O地址为379H；

BUSY	ACK	PE	SLCT	ERROR			
D7	D6	D5	D4	D3	D2	D1	D0

(3) 控制端口。控制端口有可读写锁存的5位数据，在一个8位的字节数据上，安排在D4~D0，如下表，其中D0即为选通信号，D0输出时为高电平，1为低电平，它的I/O地址为37AH。

			IRQ	SLCTIN	INIT	AFDXT	STROBE
D7	D6	D5	D4	D3	D2	D1	D0

以下以3B格式的程序为例说明数据传送过程。程序首先读入一行文本的3B格式数据，并把数据转化为纸带格式的3B代码。再读入并口状态地址为379H的一个字节数据，判断最高位即BUSY位的状态，如果BUSY有一个正脉冲信号，表示控制器已准备好读入下一个数据。程序然后在数据口378H上送出一个数据，再在控制口37AH上送出一个数据，准备好选通信号STB，通知外设取走数据。一个数据送完，进入下一个数据传送，直至一行数据送完；一行数据送完，则进入下一行数据传送，直至整个数据送完，程序结束。程序流程见图4-6。

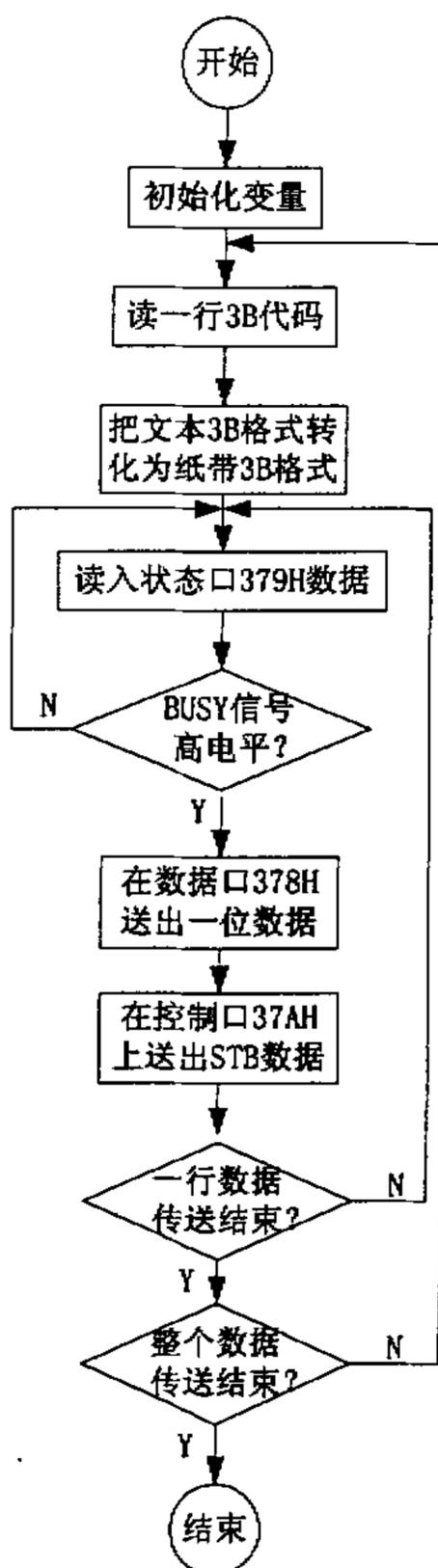


图4-6 传送代码的程序流程

以上程序在传送数据过程中，送出的STB选通信号，是为了和目前部分单片机线切割控制器配有专用程序传送口的线切割控制器而设计的。

#### 4.2.4 DNC 主机与工控机的通讯

目前较为先进的电火花线切割机床，例如苏州沙迪克三光公司的 DK7740 机床都配备了工控机。系统通过插在系统主机的 ISA 插槽中的 BKDC2.2 和 BKDC2.3 两块接口卡，实现对具体执行传感器的采样及部件驱动与控制，以实现对所需特定零件的切割加工。

实际生产中常用的 CAD/CAM 软件，如 Pro/E、UGII、Mastercam 等，一般都是在 Windows 下运行的，而我们使用的数控系统是以 DOS 为平台的。为了满足生产的需要，要

想实现 DNC 主机与工控机的通讯，必须要能够实现 Windows 与 DOS 之间的数据交换。虽然 BKDC 系统自带计算机与工控机之间的传输程序，但是无法与本系统集成，而且该程序只能在 Windows95 下传输，无法满足现有生产的要求。故本文通过对 Windows 平台下与 DOS 平台下的串口通讯编程的研究，有效的解决了这一问题。在 DOS 下使用的编程环境是 TC2.0，Windows 下使用的编程环境为 Visual C++6.0。

### 一、DOS 平台下的 C 语言通信程序设计

BKDC 数控系统部分是基于 DOS 平台的，使用 TC2.0 作为开发工具，实现 DNC 的串口通讯包含以下几个基本函数：

`int comm_open (unsigned speed)` 一用来打开串行通讯口，设置通讯的波特率的函数，以便为通讯做准备。

`void comm_close (void)` 一关闭所有的串行通讯口函数，结束串口通讯。

`void comm_flush (void)` 一清空数据缓冲区函数，使输入、输出指针指向缓冲区开始，并设置输入、输出计数器为零：

`void comm_avail ()` 一读取字符个数函数，读取当前输入、输出缓冲区字符的个数。初始化时该值为 0。

`void comm_putc ()` 一串口输出函数，往串口输出字符。

`void comm_getc ()` 一串口输入函数，从串口读取字符；

`void comm_dtr ()` 一数据准备就绪函数，数据发送前，发出数据传输就绪信号。这是数据传输的握手信号。发出该信号之前、系统等待 500ms。

`void dobaud()` 一波特率设置函数。用来设置数据传输速度的函数。该函数由 `comm_open` 调用。

有了这几个函数，就能够完成基本的串口输入输出功能。此外，由于与主机相连的数控系统不止一个，而 Windows 系统的实时性又不太好，为了防止数控系统在工作过程中等待数据输入而造成的加工停顿，使用循环缓冲技术，有效地解决了这一问题。

循环缓冲区主要用在数控系统接受数据部分。现定义了一个 `buffer[n]` 数组，其大小为 32k ( $n=32k$ )。同时还定义了两个指向 `buffer` 数组元素的指针：`inptr`，`outptr`。由程序可知，当读取端口数据放入数组中 (`inptr`) 以及从数组中取走数据 (`outptr`) 时，这两个指针的值一直不停的累加，直到等于 `&buffer+32k` 时再从 `&buffer` 开始，这样不停地循环往复，数组 `buffer[n]` 就是所谓的循环缓冲区。在循环过程中，`inptr` 总是在 `outptr` 前面，并且两者之间的最大差值为  $n-1$ 。这就是说，数控机床在加工过程中，只要该通讯处理程序在相邻的两次传递数据的时间间隔内，数控系统尚未处理完 32k 的数据（数控程序），那么该通讯程序就能够满足数控加上的需要，不会因为数据的传输延时而影响数控机床的加工。如果有更多的数控机床需要进行控制，只要修改相应的循环缓冲区的大小就可以实现。

下面简单介绍一下数控系统中的串口中断服务程序。程序流程如图 4-7 所示。

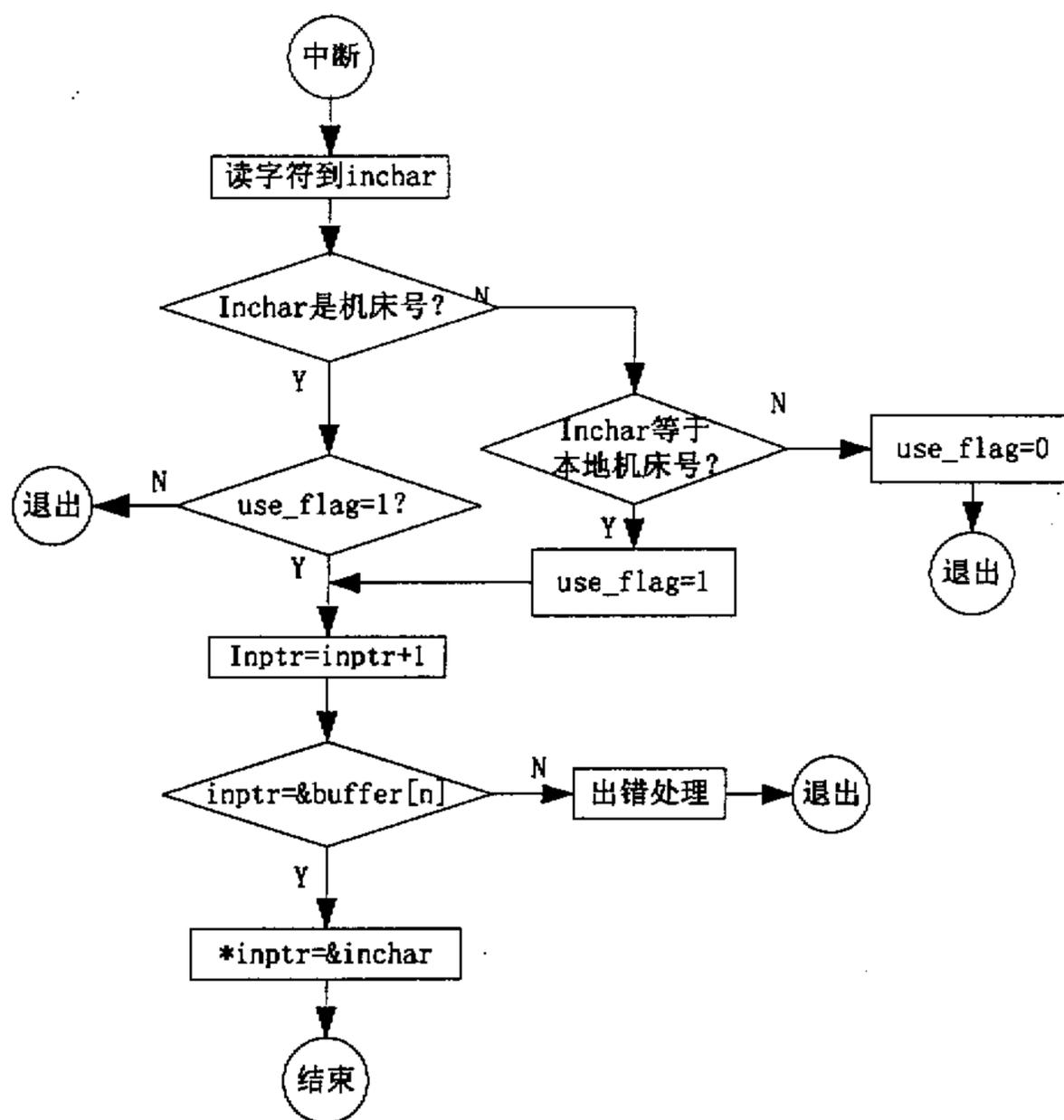


图 4-7 串口中断程序流程图

程序主要执行以下几步：

- (1) 首先从串口读字符到 inchar，判断该值是否大于 128，如大于 128 则为控制信号或是机床号。若是机床号，则判断是否本机床号。
- (2) 接着判断机床的使用标志 flag，如果是，则所有传入的数据都在该系统中处理。
- (3) 判断 Inptr 是否等于 &buffer[n]，如相等，则将 Inptr 指针（指向存入数据的当前位置）后移一位，并且将 Inptr 指针指向 inchar 的地址。

## 二、Windows 下的 DNC 通讯程序开发

DNC 主机的通讯程序使用 Visual C++ 开发，具体实现方法大致跟前面差不多，这里不再详细讨论。需要注意的是 DNC 主机与各类数控系统之间采用的通信协议必须一一对应，这样才能保证稳定、可靠的通信传输。

### 4.3 多串口通信解决方案

普通微机一般只带 2 个串口，无法满足 DNC 主机对多台带有 RS-232C 串口的线切割机

床的群控管理。本文选用了 MOXA (摩莎) 公司出品的产品—C104 智能多串口卡, 它由一块控制卡、一条 DB25 到 DB9 的电缆线和若干模块组成, 并提供了强大的软件包 Pcomm, 用于开发串行通讯程序。本文选用的 C104 多串口卡的 UART 模块带 8 个 RS232/RS422 串口, 基本能满足与 4 台 BDKC 线切割数控系统、2 台 MCS-51 系列单片机控制器或是其他老式的微机之间的通信。

MOXA 多串口控制卡的安装与设置比较简单, 详见《MOXA 用户手册》。安装内容包括 MOXA 的设备驱动程序和 Pcomm。

Pcomm 是 MOXA 提供的串行通信函数库, 功能强大, 支持多重编程语言 (VC, VB, Delphi 等), 本文则使用 VC 来开发通信程序, 它的整个体系结构如图 4-8 所示:

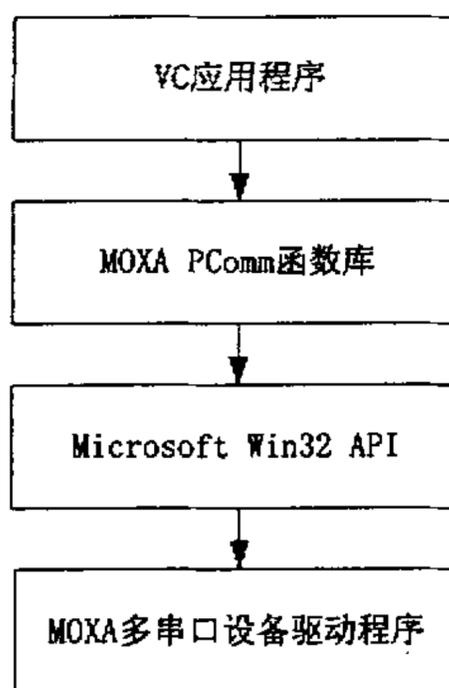


图 4-8 多串口通信软件的体系结构

多串口通信程序的通信协议在 4.2.1 已经论述的比较详细, 这里不再一一细述。需要注意的是: 开发通信程序时, 应把 “Pcomm.h” 文件包含到程序中去, 把 Pcomm.lib 输出库文件添加到程序的连接库列表中去。只有这样才能使用由 Pcomm.dll 文件输入的函数来进行串口操作。程序中主要应用到以下几个函数来实现 DNC 主机同下位机之间的数控文件传输及命令传输。

- (1) sio\_open 函数: 为发送或接收数据打开一串口;
- (2) sio\_close 函数: 关闭串口并停止发送或接收数据;
- (3) sio\_ioctl 函数: 配置通信参数;
- (4) sio\_baud 函数: 设置波特率;
- (5) sio\_getch 函数: 从驱动器的输入缓冲区读一个字符;
- (6) sio\_read 函数: 从驱动器的输入缓冲区读数据;
- (7) sio\_putch 函数: 写一个字符到驱动器的输出缓冲区;
- (8) sio\_write 函数: 向驱动器的输出缓冲区写数据;
- (9) sio\_FtASCIIRx 函数: 使用 ASCII 协议接收一文件;
- (10) sio\_FtASCIITx 函数: 使用 ASCII 协议发送一文件;

#### 4.4 本章小结

本章研究了电火花线切割 DNC 加工系统通讯模块的关键技术。对异构线切割机 DNC 通信的实现和多串口通信解决方案进行了详细的讨论。

1. 带网络接口的线切割数控系统可以采用 TCP/IP 或以太网协议与 DNC 主机进行通信。
2. 带串口的线切割数控系统可以用多串口控制器把多个数控系统跟 DNC 主机连接, 约定好双方之间的通讯协议, 编写上位机及下位机的通讯软件, 完成主机与单片机控制器的通信。
3. 不带串口的单板机线切割数控系统可以利用其自带的电报头和光电头接口或是纸带穿孔机, 通过外接式通讯适配器或者进行简单的连接实现与 DNC 主机的通信。

## 第五章 电火花线切割 DNC 加工系统的软件实现

### 5.1 软件的总体设计

#### 5.1.1 系统软件的功能及设计

在电火花线切割 DNC 加工系统软件中，要完成的主要功能有：

1. 电火花线切割 NC 代码的生成。用户登陆到 CAD/CAM 服务器，进行线切割零件的设计、自动化编程；同时可以访问位于数据库服务器上的基于专家系统的电火花线切割工艺数据库，输入加工条件，取得工艺参数后，最后输出加工代码。

2. DNC 主机完成与线切割数控系统的双向通信以及接受上层数据库服务器的制造信息。DNC 主机实现生产任务、图纸、工艺文件、NC 程序从数据库服务器的下达和收集来自线切割数控系统反馈的有关信息，实现生产任务的分配及生产任务管理、图纸管理、工艺文件管理、NC 程序管理。并且能对部分线切割数控系统实行监控和跟踪。

系统软件的功能模块结构组成如图 5-1 所示。

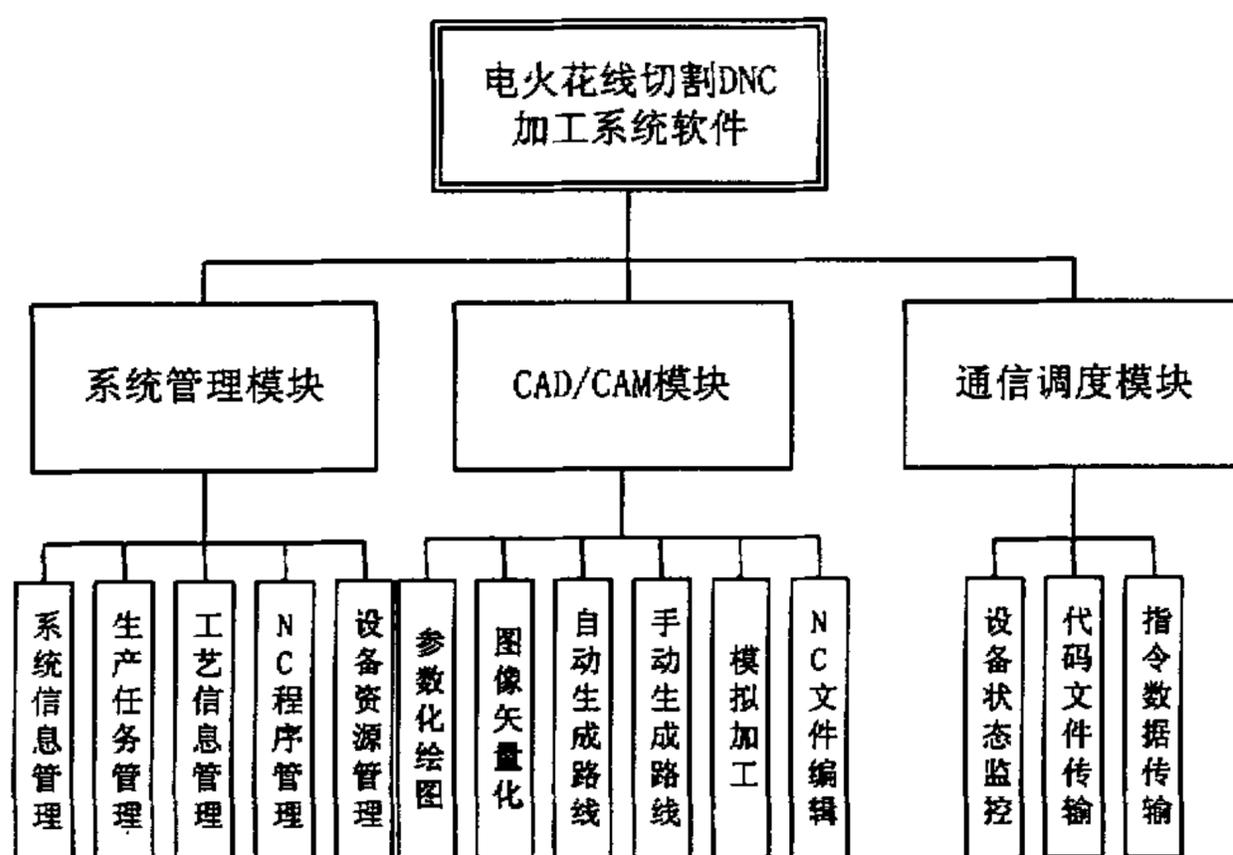


图 5-1 系统软件功能框图

根据软件工程及实际应用的需要，在软件设计过程中应考虑以下几个方面：

(1) 软件具有较好的鲁棒性。能保证可靠的工作，而不是经常发生异常事件，这是工程应用中必须考虑的问题。

(2) 软件的界面。操作人员的计算机应用知识方面的限制，决定了软件应该易学、易用。设计简单、直观、美观的人机界面就非常重要。

(3) 软件易维护性。随着系统功能的扩展和改进，在软件设计阶段应该考虑到源代码

的可读性及易扩展性，本文使用了面相对象的设计方法，使得软件维护起来比较方便。

### 5.1.2 软件的应用平台和开发工具

本系统软件的自动编程模块是在 AutoCAD 平台下，用微软公司的 Visual C++和二次开发工具 ObjectARX 进行开发。

本系统的通信模块用 Windows2000 Sever 作为系统平台，用 Visual C++6.0 开发的。

本系统的系统管理模块是采用 MS SQL SEVER2000 作为后台数据库，Visual Basic 作为前台开发工具的。

### 5.1.3 系统的关系模型图

系统各个模块之间的关系如图 5-2 所示。

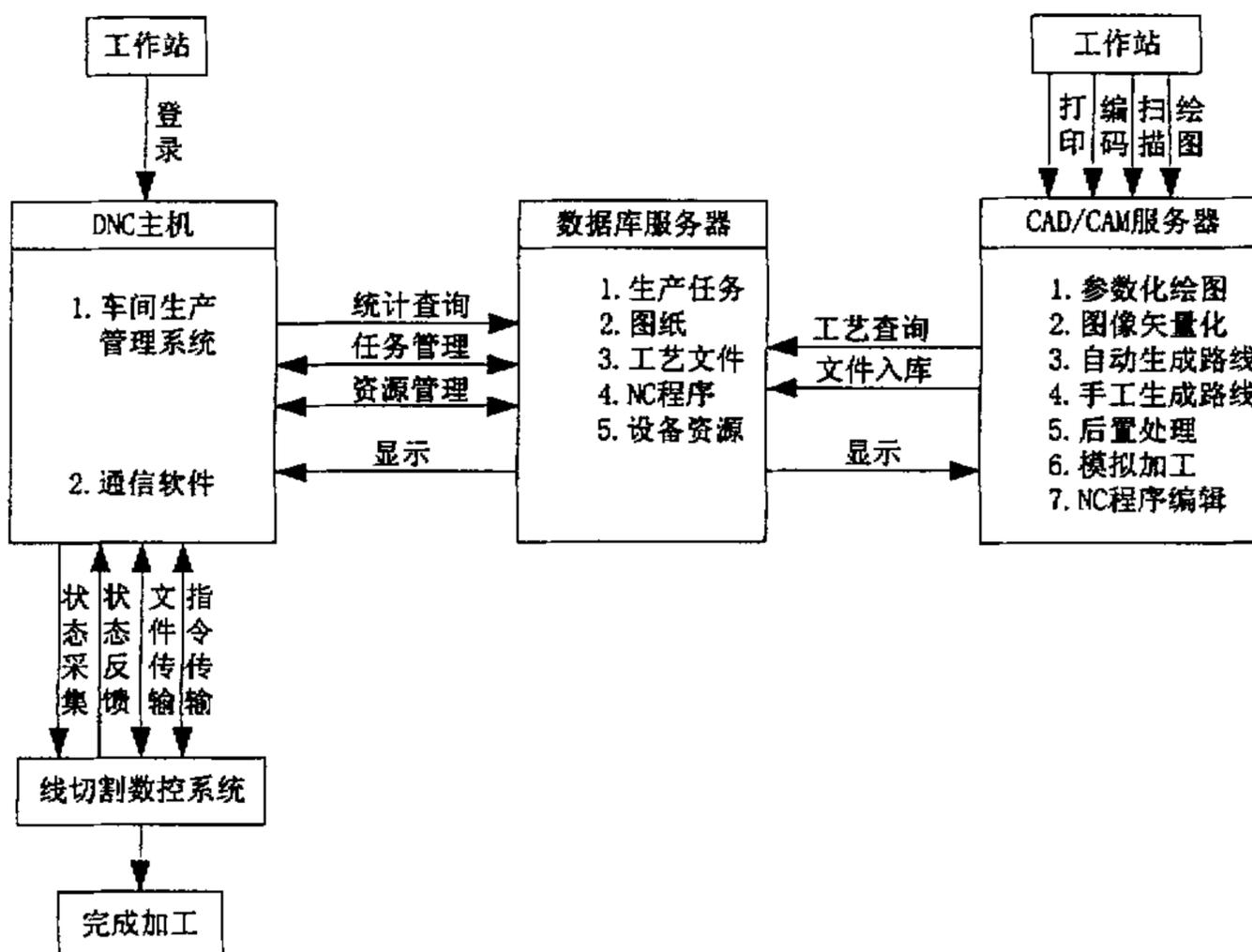


图 5-2 系统的关系模型

## 5.2 自动编程模块软件的实现

### 5.2.1 基于 ObjectARX 的 AutoCAD2000 开发技术<sup>[44][45][46]</sup>

#### 一、ObjectARX 概述

##### (1) AutoCAD 2000 开发系统概述

ObjectARX 是 Autodesk 公司针对 AutoCAD 平台上的二次开发而推出的一个开发软件包, 它支持面向对象编程, 同时也向下兼容 ADS C。ObjectARX 的早期版本称为 ARX, 是 AutoCAD R13 后推出的一个全新的面向对象的开发环境, 是 AutoCAD 第一次提供面向对象的二次开发工具。ADS C 已经可以在 AutoCAD 上开发较大规模综合应用程序, 然而计算机技术的发展不仅要求 CAD 的交互性、综合性, 同时对自动化、智能化提出了更高的要求。ARX 技术就是在这种需求下诞生的。

ObjectARX 开发技术在 AutoCAD R14 版有了很大增强, 用户可以使用 C++ 来开发 AutoCAD 应用程序, 扩充 AutoCAD 的类及协议, 以及建立新的和 AutoCAD 的内部命令在统一层次上作用的命令。

目前, 针对 AutoCAD 2000 的开发已发展为 ObjectARX 3.0 版本, 即 ObjectARX for AutoCAD 2000, 也称 ObjectARX 2000。与 2.0 版本比较, ObjectARX 2000 针对 AutoCAD 2000 的多文档界面 (MDI) 提供了相应的编程手段, 并且还提供了功能强大的 MFC ARX 类, 从而极大地方便了基于图形界面地应用开发。

ARX 应用程序实质上是一个动态链接库 (DLL) 文件, 它和 AutoCAD 共享地址空间, 并且直接和 AutoCAD 进行通讯。

## (2) ObjectARX 程序特点

ARX 程序本质上是地道的 Windows DLL 程序, 而 AutoCAD 本身则是一个典型的 Windows 程序。ARX 与 AutoCAD、Windows 之间均采用 Windows 消息传递机制直接通信, 其程序的调用、运行是通过消息机制实现的。其通信如图 5-3 所示:



图 5-3 ARX 与 AutoCAD、Windows 通信

ARX 应用程序是一种可共享 AutoCAD 地址空间和建立对 AutoCAD 直接函数调用的动态链接库 (DLL), 因而 ARX 应用程序与 AutoCAD 间的通讯更加直接, 运行效率更高。它具有直接访问和控制 AutoCAD 的能力: 可以监控和处理 AutoCAD 的各种事件, 具有事件驱动特点; 可以定义 AutoCAD 命令 (和其部命令在同一层次), 定义 AutoCAD 数据库对象等。

ARX 程序运行速度快, 访问 AutoCAD 数据库速度快, 执行 ARX 程序所需系统开销最小。

ARX 提供了面向对象编程的技术。在 90 年代, 面向对象的编程技术 (OOP) 成为软件开发的主流技术。OOP 技术具有可封装性、可继承性、多态性等特点。

ObjectARX 提供了以 C++ 为基础的面向对象的开发环境和应用程序接口, 能真正快速地访问 AutoCAD 图形数据库。用其开发的工程 CAD 软件具有模块性好、独立性强、链接简单、使用方便。内部代码高效实现以及代码可重用性强等特点, 并且支持 VC++ MFC 基

本类库，能够简洁高效地实现许多复杂功能。

## 二、ObjectARX 基本编程技术

### (1) ObjectARX 类库简介

ObjectARX 编程环境主要有 AcRx、AcEd、AcDb、AcRt、AcGi、AcGe 和 ADS 类库组成。

AcRx 库提供动态链接库 (DLL) 初始化及链接所需的系统级的类和 C++ 的指令集，用于 DLL 应用程序的初始化、连接及实时类的注册和标识。它的基类是 AcRxObject。AcRx 库同时提供了大量的宏定义，用于派生新的 ARX 类。

AcEd 库提供用于定义和注册新的内部命令。所定义的新的内部命令与 AutoCAD 内部命令具有相同的运行模式，与 AutoCAD 本身内部命令具有相同的命令堆栈 (AcEdCommandStack)。AcEditorReactor 是此类中一个很重要的类，能监测 AutoCAD 编辑器的运行状态，并根据特定的事件运行相应的程序。

AcDb 库提供了直接访问 AutoCAD 数据库数据结构的类，AutoCAD 数据库中包含了各种构成 AutoCAD 图形的图形对象 (实体) 及非图形对象 (层、线形、字体风格等) 的信息。利用该类库提供的类，可以对已经存在的 AutoCAD 实体及对象进行查询等操作，并可创建新的数据库对象。它主要包含 9 个命名唯一的符号表对象、对象词典和系统变量。AcDb 库是 ARX 最重要的类库。

AcGi 提供了许多图形界面工具用来绘制和显示 AutoCAD 的实体，还可以通过重载 worldDraw () 函数来定义新的实体。

AcGe 通用几何库可以被 AcDb 类所引用，并提供诸如向量、点及转换等用于普通二维或三维的几何操作，同时也提供了简单的几何对象，如点、曲线及曲面。它包括两个子集：二维操作类和三维操作类。

ADS 库用于创建 AutoCAD 应用程序的 C 语言库。通常用来执行实体选择、选择集操作，以及数据获取等。

### (2) 应用程序结构

ObjectARX 应用程序是共享 AutoCAD 地址空间并对 AutoCAD 进行直接调用的动态链接库，其调用不同于普通的 C 函数，AutoCAD 调用 ARX 应用程序的入口是靠 acrxEntryPoint () 函数建立的，而不是靠 main () 函数。

acrxEntryPoint () 函数不仅作为一个 ObjectARX 应用程序与 AutoCAD 通信的入口点，而且 ObjectARX 应用程序可以通过返回状态码与 AutoCAD 顺序通信。所有激活函数的请求均通过 acedDefun () 函数来定义，该函数由 acrxEntryPoint () 函数生成。如果利用 ObjectARX 或 acedDefun () 函数定义一新的命令，AutoCAD 会立即执行与该命令相关的函数。

acrxEntryPoint () 函数的格式如下：

```
extern "C" AcRx::AppRetCode //返回给 AutoCAD 的状态码
```

```

acrxEntryPoint (AcRx::AppMsgCode msg, //ARX 内核向应用程序发的消息
void* pkt //回调数据信息指针
)
{
switch (msg) {
case AcRx::kInitAppMsg:
acrxDynamicLinker->unlockApplication(pkt);
acrxDynamicLinker->registerAppMDIAware(pkt);
InitApplication();
break;
case AcRx::kUnloadAppMsg:
UnloadApplication();
break;
}
return AcRx::kRetOK;
}

```

在 `acrxEntryPoint()` 函数定义中，用 `switch` 语句对 AutoCAD 发送的消息进行处理，如上所示。

### 5.2.2 MFC 与 ObjectARX 混合编程的实现<sup>[44][45][46]</sup>

Visual C++ 是当前最流行的编程语言之一。MFC 基础类库提供了丰富的界面资源，如对话框、菜单、WINDOWS 控件等。在使用 ObjectARX 对 AutoCAD 进行二次开发时，使用 MFC 类库，可以充分利用 VC++ 环境可视化功能以及资源编辑功能，对自己所做的界面进行可视化的合理布局调整，做出的界面美观形象，从而消除了 ADS 编程工具中 DCL 语言编写时对话框控件的布局依靠计算坐标点定位、回调函数编写的繁琐。调试必须在 AutoCAD 下等不便。

使用 MFC 进行 ARX 编程可以采用 MFC 类库动态链接方式或静态连接方式。使用动态链接方式的 ARX 程序比使用静态链接方式要小，并且运行速度快，对内存的要求也不高。因此，系统采用 MFC 动态链接方式。

#### 1. 工程类型设置

创建 VC 程序时，选择工程类型为 ObjectARX 2000 AppWizard，然后选择 Regular DLL using Shared MFC DLL”。在 Project Settings 对话框中的 General 一栏，在 Microsoft Foundation Classes 下拉列表框中选择 “Use MFC in a Shared DLL” 然后删掉自动生成的文件，并且把实现资源管理的程序添加进来。

#### 2. 资源管理

设计使用 MFC 类库并与 AutoCAD 或其他应用程序共享资源的 ARX 时, 资源管理是很重要的一项内容。资源管理不善, 将会导致该程序与 AutoCAD 或其他程序产生资源冲突, 致使系统崩溃。必须将定义的使用 CdynaLinkLibrary 模块的状态插入 MFC 对资源操作时检查的链表中。为避免产生资源管理混乱, 采用在外部管理资源的办法。

- 1) 在进行任何会引起 MFC 寻找用户资源的操作之前调用 AfxSetResourceHandle () 函数, 把用户资源设置成用户默认资源。
- 2) 在把系统资源用作用户资源之前, 调用 AfxGetResourceHandle () 函数, 获得当前系统资源, 并保存。
- 3) 在完成用户资源的调用后, 应当立即重新设置成原来的资源句柄。

在需要 AutoCAD 资源的用户对话框内调用 AutoCAD API 函数(或激活 AutoCAD 命令)时, 必须在执行这些调用之前, 将资源设置成 AutoCAD 的资源。然后再重新装入程序的资源。可以使用 acedGetAcadResourceInstance () 函数来获得 AutoCAD 的资源句柄。

为简化资源管理, 设计了一个临时资源重载类 CtemporaryResourceOverride 类来管理。它的结构如下:

```
class CtemporaryResourceOverride
{
public:
    CtemporaryResourceOverride (HINSTANCE hInstNew);
    CtemporaryResourceOverride ();
    Virtual ~CtemporaryResourceOverride ();
    Static void SetDefaultResource (HINSTANCE hInstNew);
private:
    void CommonConstruction (HINSTANCE);
    static HINSTANCE m_hInstanceDefault;
    HINSTANCE m_hInstanceOld;
};
```

该类中静态变量 m\_hInstanceDefault 用来存储程序的句柄。当创建该类的一个对象时, 构造函数 CtemporaryResourceOverride () 将系统资源设置为用户程序的资源; 当删除该对象时, 类的析构函数 ~CtemporaryResourceOverride () 将系统资源恢复为初始值。这样, 当某个函数中需要将用户资源设置为默认值时, 只需要创建该类的一个对象。

在使用 MFC 的 ARX 程序中, 当接受到 AutoCAD 发送的 AcRx:: kInitAppMsg 消息时, 需要将本程序的 DLL 插入资源链表中; 当接受到 AutoCAD 发送的 AcRx:: kUnloadAppMsg 消息时, 将该 DLL 从资源链表中清除。

### 5.2.3 自动编程模块软件开发的实现过程

## 一、创建 ARX 应用程序框架

用 ObjectARX 2000 AppWizard 应用程序设计向导可以方便、快捷地创建 ARX 应用程序框架。操作步骤为（前提是安装了 Visual C++和 ObjectARX2000）：

(1) 进入 VC6.0 集成开发环境，选菜单 File/New 选项。在 New 对话框中选 Projects 选项卡，选择创建 ObjectARX 2000 AppWizard。在 Project name: 输入框中输入项目文件名：WEDM，单击 OK 按钮。

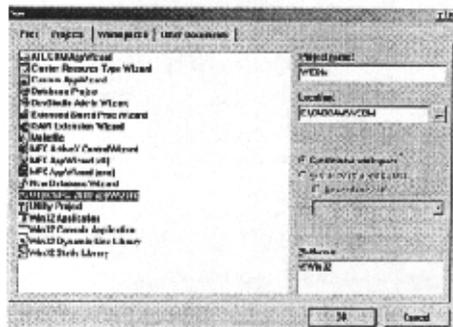


图 5-4 New 对话框

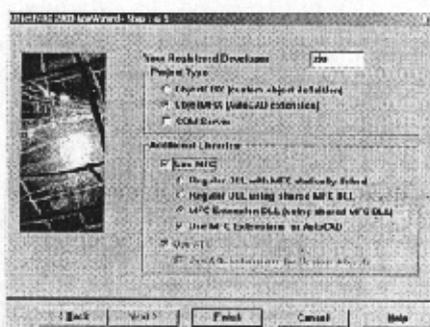


图 5-5 ObjectARX 2000 AppWizard 对话框

(2) 在 ObjectARX 2000 AppWizard 对话框中选中“Using MFC”复选框，然后按 Finish 结束。

(3) 在 Tools 菜单下选择 Options，当出现 Options 对话框后，选择 Directories 选项卡，增加 ObjectARX 2000\inc 目录和 ObjectARX 2000\Lib 目录的位置。因为程序编译连接的时候需要用到库文件。

这样应用程序的框架结构就生成了，以后只要在此基础上添加功能模块就可以了。

## 二、设计主体程序

ObjectARX 程序具有面向对象的编程特点。本系统以 ObjectARX 2000 AppWizard 应用程序设计向导生成的框架为基础，主要编制了以下类或结构来实现程序的功能：

### 1) CParaDlg 工艺参数设置类

该类对应于参数处理对话框，定义了各个工艺参数，包括：切入线长度、切入方式（垂直或是 45 度角）切割方向、拐角过渡以及切割次数，刀具补偿等。

### 2) Comm 通信类

该类定义了与通信有关的一些函数，如串行口初始化、发送地址、发送单个数据、成批发送数据、接受数据等。

### 3) CInterpolation 插补类

该类提供了各种曲线的插补方法：直线插补、圆弧插补、抛物线插补、椭圆插补、双曲线插补等。

### 4) CCommDlg 通讯对话框类

实现通信进度显示和通信状态

5) CSimulation 加工模拟类

实现加工过程在屏幕上的模拟运行。

6) Cmonitor 监视类

实现加工过程的实时跟踪，显示加工进度和加工状态。查询各机床的加工状态，并在屏幕上显示。

7) 各种曲线类

实现自定义实体曲线如抛物线、双曲线等的构造、显示和在数据库的驻留。

8) Node 节点结构

此结构是构成双链表的基本节点单元。程序执行过程中，凡是涉及到对实体数据进行操作的都使用到了该节点结构。其中计算基准点 B，加工起点 S 时，要用到指向该节点结构的指针。其结构如下所示：

```
typedef struct node * dlink; //定义双链表指针
struct node
{
    char entity [20]; //实体名，本程序中主要涉及到三类实体：直线、圆弧和圆
    char direction; //加工方向，顺时针或者逆时针
    struct point
    {
        double x, y;
    } pStart, pEnd, pCenter; //起点、终点、圆心（圆弧）
    double k, A, B, C; //k 为直线的斜率，A、B、C 为直线方程的参数
    //考虑刀具补偿半径计算钹丝实际轨迹所用
    double radus, startAngle, endAngle; //半径、起始角、终止角
    dlink Llink, Rlink; //链表左、右指针
}
```

由于采用了 OOP 编程技术，程序清晰而具有条理，提高了程序的可读性，有利于对程序进行进一步完善开发。

下面以 3.2.2 一节为例子，给出算法对应的详细的 c 源代码实现。

```
*****保存为 3B 指令文件*****
//把 3B 代码写到 test.3b 文件中
int save3b ()
{
    struct node *ptr;
    char *G,*Z;
```

```
double X,Y,c,Xe,Ye,Xb,Yb;//(Xe,Ye)终点坐标,(Xb,Yb)起点坐标
int k,J,XX,YY;//3B 代码中的数值
ptr=pB;

//向文件写入
ofstream out("test.3b");
if(!out){
    cout<<"Cannot open file!";
    return 1;
}
    int count=0;
do{

//如果是直线
if(strcmp(ptr->entity,"AcDbLine")==0)
{
    X=ptr->Rlink->pt.x-ptr->pt.x;
    Y=ptr->Rlink->pt.y-ptr->pt.y;
    if(abs(X)>=abs(Y))
    {
        G="Gx";
        J=abs(X)*1000;
    }
    else
    {
        G="Gy";
        J=abs(Y)*1000;
    }
    XX=abs(X)*1000;
    YY=abs(Y)*1000;

    if(Y>0)
    {
        if(X>0)
```

```

        Z="L1";
    else
        Z="L2";
    }
    else if(Y<=0)
    {
        if(X>0)
            Z="L4";
        else
            Z="L3";
    }
}
//如果是圆弧
if(strcmp(ptr->entity,"AcDbArc")==0)
{
    Xb=ptr->pt.x- ptr->pCenter.x;
    Yb=ptr->pt.y- ptr->pCenter.y;
    Xe=ptr->Rlink->pt.x-ptr->pCenter.x;
    Ye=ptr->Rlink->pt.y-ptr->pCenter.y;
    XX=abs(Xb)*1000;
    YY=abs(Yb)*1000;
    if(abs(Ye)>abs(Xe))
    {G="Gx";k=1;}
    else
    {G="Gy";k=-1;}

    if(Xb>0)
        if(Yb>0)
            Z="NR1";
        else
            Z="NR4";
    else
        if(Yb>0)
            Z="NR2";
        else

```

```

        Z="NR3";

        count_j(Xb*1000,Yb*1000,Xe*1000,Ye*1000,k,ptr->radius*1000,10000);
        J=abs(J);
    }

    out<<'B'<<XX<<' '<<'B'<<YY<<' '<<'B'<<J<<' '<<'G'<<' '<<Z<<"\n";
    count++;
    ptr=ptr->Rlink;
}
while(ptr!=pB);
    out.close();
    return 0;
}

```

其中 count\_j () 为圆弧计算长度，限于篇幅关系，不再列出。

### 三、程序运行

程序编译完后执行，系统要求连接 AutoCAD。连接后进入 AutoCAD 环境，在命令提示行下输入“ARX”命令，接着输入“1”装载应用程序。输入应用程序名字，本系统名字为 zjuWEDM，即可。这样用户就可以通过屏幕菜单和 AutoCAD 系统的命令来进行自动编程了。

#### 5.2.4 细化和矢量化算法的实现

##### 一、细化算法代码的详细实现如下：

```

BOOL CDib::Thinning(HDIB hDib)
{
    LONG          x,y;
    int           num;
    BOOL         Finished;
    int           nw,n,ne,w,e,sw,s,se;
    static int erasetable[256]={} //这里是个判断表,

    WaitCursorBegin();
    HDIB hNewDib = NULL;
    // we only convolute 2bpp DIB, so first convert DIB to 24bpp
    WORD wBitCount = DIBBitCount(hDib);
    HANDLE h;
    PLOGPALETTE plogPal;

```

```
h = GlobalAlloc(GHND, sizeof(DWORD) + sizeof(PALETTEENTRY)*256);
if (! h)
    return FALSE;
plogPal = (PLOGPALETTE)GlobalLock(h);
if (! plogPal)
    return FALSE;

plogPal->palVersion=PALVERSION;
plogPal->palNumEntries=256;
plogPal->palPalEntry[0].peGreen= 0;
plogPal->palPalEntry[0].peBlue= 0;
plogPal->palPalEntry[0].peRed= 0;
plogPal->palPalEntry[0].peFlags=NULL;

plogPal->palPalEntry[1].peGreen=255;
plogPal->palPalEntry[1].peBlue=255;
plogPal->palPalEntry[1].peRed=255;
plogPal->palPalEntry[1].peFlags=NULL;
HPALETTE hpl=CreatePalette(plogPal);

HPALETTE palold=NULL;
if (wBitCount != 8)
{
    if(wBitCount!=24)
    {
        CPalette* Cpal=GetPalette();
        palold=(HPALETTE)Cpal->GetSafeHandle();
    }
    hNewDib = ConvertDIBFormat(hDib,8, hpl);
}
else
{
    hNewDib = ConvertDIBFormat(hDib,8, hpl);
}
if (! hNewDib)
{
```

```

    WaitCursorEnd();
    return FALSE;
}
// new DIB attributes
WORD wDIBWidth = (WORD)DIBWidth(hNewDib);
WORD wDIBHeight = (WORD)DIBHeight(hNewDib);
WORD wBytesPerLine = (WORD)BytesPerLine(hNewDib);
DWORD dwImageSize = wBytesPerLine * wDIBHeight;
// Allocate and lock memory for filtered image data
HGLOBAL hFilteredBits = GlobalAlloc(GHND, dwImageSize);
if (!hFilteredBits)
{
    WaitCursorEnd();
    return FALSE;
}
LPBYTE lpPtr = (LPBYTE)GlobalLock(hFilteredBits);
// get bits address in DIB
LPBYTE lpDIB = (LPBYTE)GlobalLock(hNewDib);
LPBYTE lpDIBits = FindDIBBits(lpDIB);
memcpy(lpPtr, lpDIBits, dwImageSize);
int judge=1;
Finished=FALSE;
while(!Finished){
    Finished=TRUE;
    for (y=1; y<wDIBHeight-1; y++){
        lpPtr=(unsigned char *)lpDIBits+(dwImageSize-wBytesPerLine-y*wBytesPerLine);
        x=1;
        while(x<wDIBWidth-1){
            if(*(lpPtr+x)==0){
                w=(unsigned char)*(lpPtr+x-1);
                e=(unsigned char)*(lpPtr+x+1);
                if (w==judge|| e==judge){
                    nw=(unsigned char)*(lpPtr+x+wBytesPerLine-1);
                    n=(unsigned char)*(lpPtr+x+wBytesPerLine);
                    ne=(unsigned char)*(lpPtr+x+wBytesPerLine+1);

```

```

        sw=(unsigned char)*(lpPtr+x-wBytesPerLine-1);
        s=(unsigned char)*(lpPtr+x-wBytesPerLine);
        se=(unsigned char)*(lpPtr+x-wBytesPerLine+1);

        num=nw/judge+n/judge*2+ne/judge*4+w/judge*8+e/judge*16+sw/judge*32+s/judge*64+se
        /judge*128;

        if(erasetable[num]==1){
            *(lpPtr+x)=(BYTE)judge;
            Finished=FALSE;
            x++;
        }
    }
    x++;
}

for (x=1;x<wDIBWidth-1;x++){
    y=1;
    while(y<wDIBHeight-1){
        lpPtr=(unsigned char *)lpDIBBits+(dwImageSize-wBytesPerLine-y*wBytesPerLine);
        if(*(lpPtr+x)==0){
            n=(unsigned char)*(lpPtr+x+wBytesPerLine);
            s=(unsigned char)*(lpPtr+x-wBytesPerLine);
            if( (n==judge)|| (s==judge)){
                nw=(unsigned char)*(lpPtr+x+wBytesPerLine-1);
                ne=(unsigned char)*(lpPtr+x+wBytesPerLine+1);
                w=(unsigned char)*(lpPtr+x-1);
                e=(unsigned char)*(lpPtr+x+1);
                sw=(unsigned char)*(lpPtr+x-wBytesPerLine-1);
                se=(unsigned char)*(lpPtr+x-wBytesPerLine+1);

                num=nw/judge+n/judge*2+ne/judge*4+w/judge*8+e/judge*16+sw/judge*32+s/judge*64+se
                /judge*128;

                if(erasetable[num]==1){

```

```
        *(lpPtr+x)=(BYTE)judge;
        Finished=FALSE;
        y++;
    }
}
}
y++;
}
}
}
// a filtered image is available in lpDestImage
// copy it to DIB bits
// cleanup temp buffers
GlobalUnlock(hFilteredBits);
GlobalFree(hFilteredBits);
GlobalUnlock(hNewDib);
// rebuild hDib
HDIB hTmp = NULL;
if (wBitCount != 8)
{
    if(wBitCount ==24)
        hTmp = ConvertDIBFormat(hNewDib, wBitCount, NULL);
    else
        hTmp = ConvertDIBFormat(hNewDib, wBitCount, palold);
}
else
    hTmp = CopyHandle(hNewDib);
GlobalUnlock(h);
GlobalFree(h);
GlobalFree(hNewDib);
DWORD dwSize = GlobalSize(hTmp);
memcpy((LPBYTE)GlobalLock(hDib), (LPBYTE)GlobalLock(hTmp), dwSize);
GlobalUnlock(hTmp);
GlobalUnlock(hDib);
WaitCursorEnd();
```

```
return TRUE;  
}
```

### 5.3 本章小结

本章对整个系统软件进行了总体设计，并对功能模块进行了划分。在此基础上，介绍了 AutoCAD 环境下利用 VC++ 和 ObjectARX 进行系统自动编程模块软件开发的过程。

## 第六章 电火花线切割 DNC 加工系统系统实例

本文提出的电火花线切割 DNC 加工系统为电火花线切割加工车间自动化生产提供了一种有效的解决方案。狭义上说，它是一个针对线切割加工车间的系统；广义上来说，它更是一种解决问题的方案。如何因地制宜，合理利用现有的生产条件，最大限度的提升线切割加工工厂或车间生产能力，这才是本文提出本系统的根本出发点。

本系统具有实用性、灵活性、易实施性和应用的广泛性等特点，可以根据电火花线切割加工工厂或车间自身的不同设备状况和人力资源进行优化配置，达到提升生产能力的目的。只要将本系统稍加改变，就可以灵活地应用到生产、教学实践中去。

目前已经投入使用并且效果良好的浙江大学电火花线切割二维创新设计 DNC 平台，就是本文提出的电火花线切割 DNC 加工系统一个很好的实施案例。实践进一步证明电火花线切割 DNC 加工系统具有较强的实际应用价值。

### 6.1 系统实施方案

#### 6.1.1 系统实施前金工实习基地车间实习情况

系统实施前的基本情况：

软件：AutoCAD2000

数控系统自带软件

硬件：4 台 BKDC 电火花线切割数控机床和若干台经济型线切割机床

1 台高档微机和十几台普通微机

打印机、扫描仪各一台

系统实施前，由于每次实习中，学生只能编写一些简单的程序交由实习老师操作进行加工，可能一堂课下来，大部分学生就仅仅学会编了一个线切割数控程序，生产、教学效率极其低下，而且大量设备闲置。

#### 6.1.2 浙江大学电火花线切割二维创新设计 DNC 平台架构

在充分利用现有生产条件和人员的基础上，我们对系统进行了改造，成功的设计了浙江大学电火花线切割二维创新设计 DNC 平台，如图 6-1 所示。

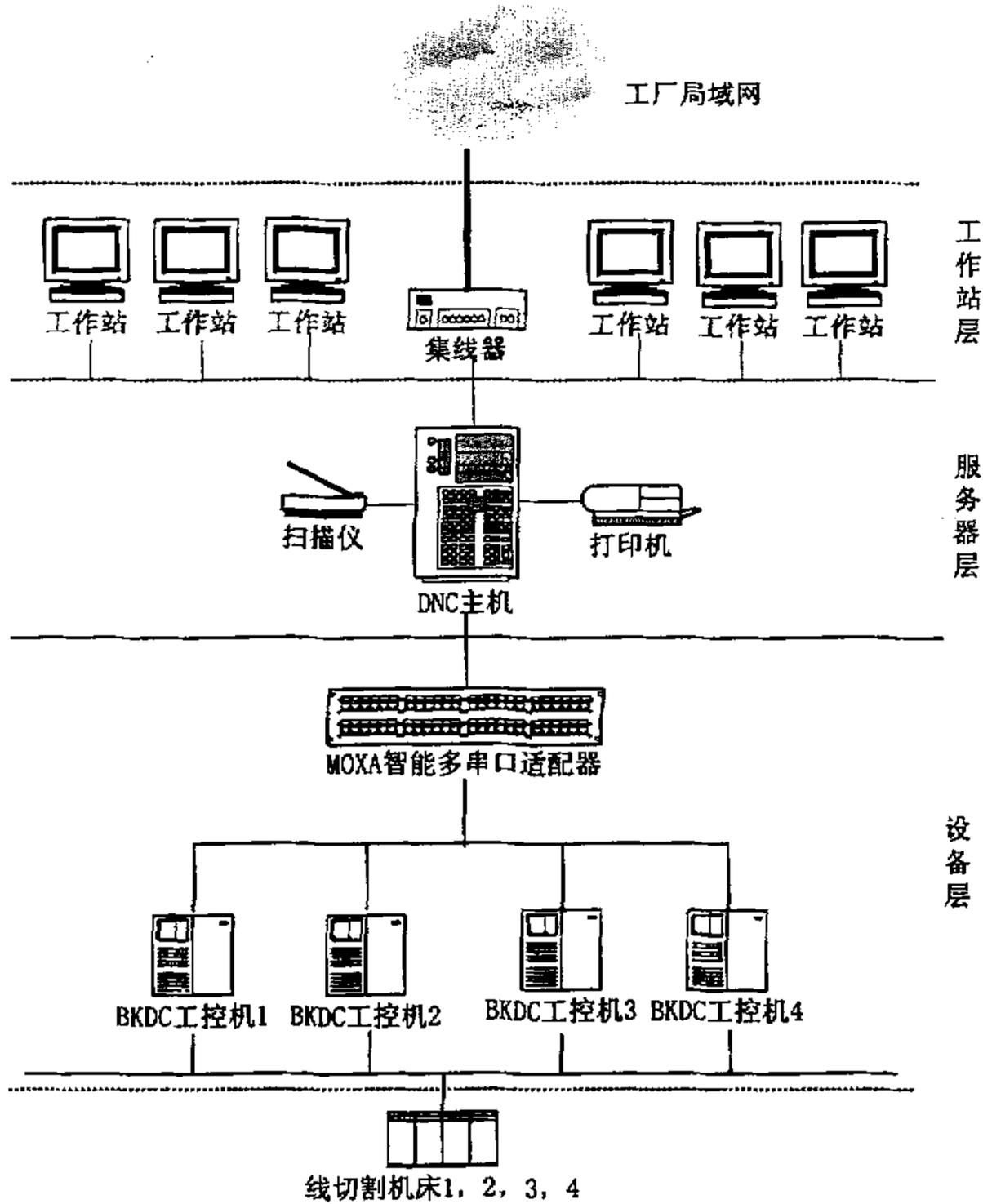


图 6-1 浙江大学电火花线切割二维创新设计 DNC 平台架构

### 6.1.3 系统实施后金工实习基地作业方式

我们按照前文提出的电火花线切割 DNC 加工系统，仍旧采用了工作站层、服务器层和设备层三层结构。所不同的是：

1. 考虑到金工基地是面向学生的，所以工作站层增加了工作站数量；
2. 考虑到车间只有一台高档的微机，并且金工实习基地对实时性要求不严格，所以服务器层把数据库服务器和 CAD/CAM 服务器集成到 DNC 主机内；

我们在原有车间资源的基础上，只在硬件方面增加了一块 MOXA 智能多串口适配器和一块集线器，系统没增加多少成本；软件方面，我们在 AutoCAD2000 环境下开发了电火花线切割自动编程系统和通讯软件以及简单的数据库管理软件。

但是系统实施后，金工基地的生产、教学模式发生了巨大的变化，其优越性体现在：

(1) 充分利用了原先闲置的设备资源，提高了设备利用率，让设备在实习过程中最大限度的被利用；

(2) 在生产实习过程中，可以让每个学生都可以同时进行自己的作业任务，充分利用了车间人员，由图 6-2 可知，可以让各台工作站上的学生同时进行绘图、编码、修改图形、文件入库，可以让 4 个学生同时进行代码传输并且加工自己创新设计的各种二维图案，一个同学进行打印，另一个同学进行扫描；

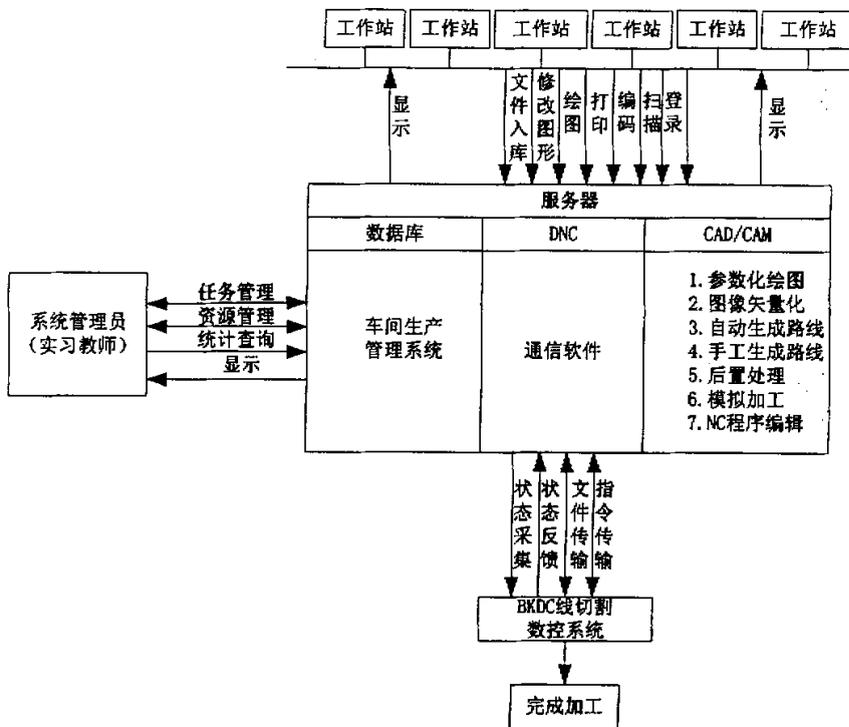


图 6-2 浙江大学电火花线切割二维创新设计 DNC 平台工作进程图

## 6.2 电火花线切割二维创新设计 DNC 平台的操作实例

### 6.2.1 电火花线切割二维创新设计 DNC 平台的主菜单

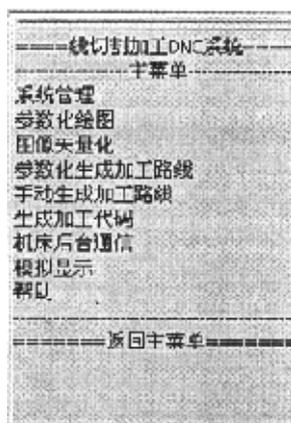


图 6-3 系统主菜单

### 6.2.2 齿轮的参数化设计实例

- (1) 点击右边的屏幕菜单项：参数化绘图菜单项下的子菜单项齿轮，如图 6-3
- (2) 如图 6-4 设置参数对话框。
- (3) 要求选择一点，作为齿轮的圆心。



6-4 齿轮参数对话框

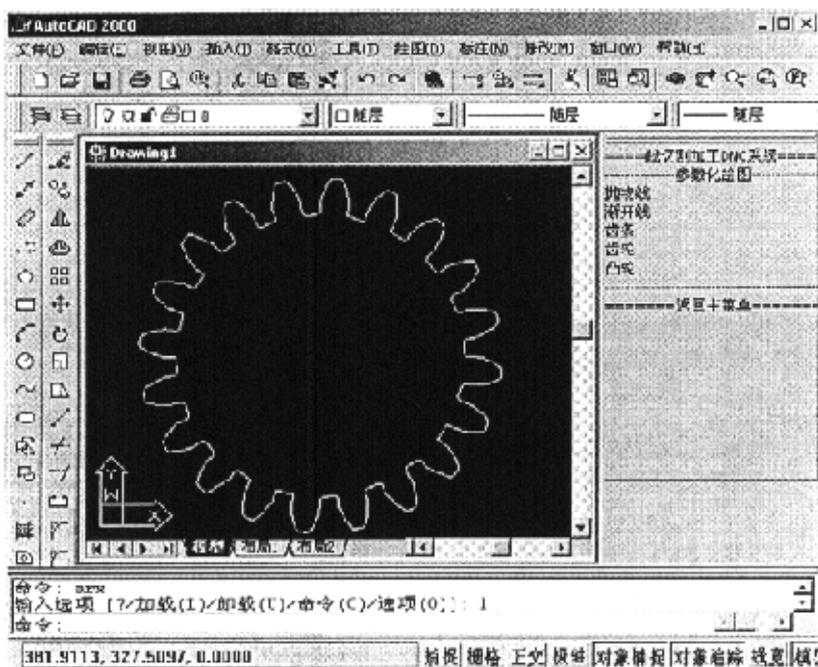


图 6-5 所绘制齿轮零件图形

### 6.2.3 手绘图“求是”标志的二维创新设计作品

- (1) 扫描手绘图纸，得到图像文件。

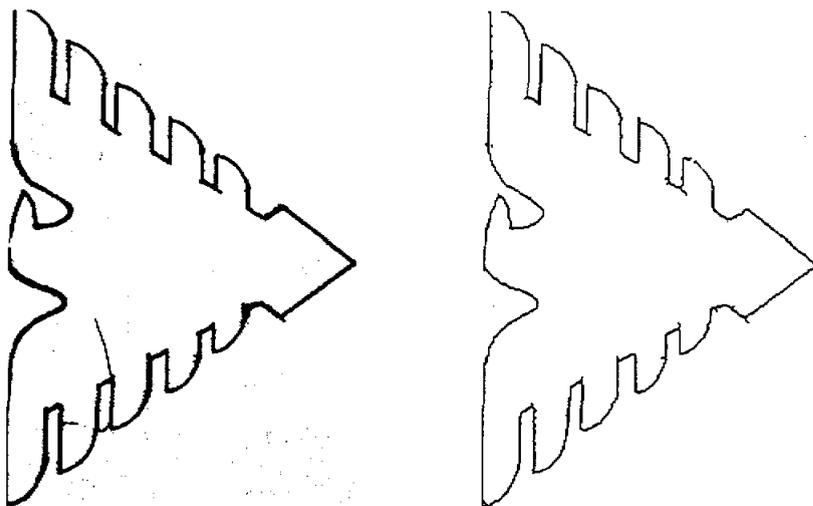


图 6-6 左图为未处理前的图，右图为细化后的图

- (2) 选择“图像矢量化”下的“细化”子菜单项，进行细化，如图 6-6 所示为细化前后的对比；
- (3) 选择“矢量化”菜单项，系统自动调入同名的 dxf 文件；

- (4) 选择自动生成加工路线模式，进行自动化编程；
- (5) 选择“保存代码”菜单项保存文件，并且选择机床进行传输、加工。

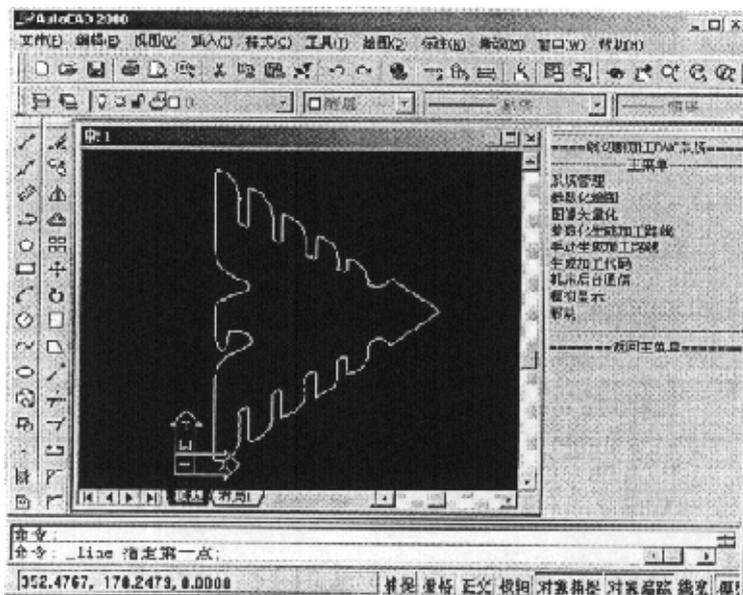


图 6-7 矢量化后的求是标志图

### 6.3 电火花线切割二维创新设计—学生作品

以下陈列的这些作品是由学生在实习过程中使用浙江大学电火花线切割二维创新设计 DNC 平台进行独力设计和完成的。





#### 6.4 本章小结

本章介绍了电火花线切割 DNC 加工系统实施的例子及系统的运行界面和操作步骤。浙江大学电火花线切割二维创新设计 DNC 平台的实现，进一步证明了电火花线切割 DNC 加工系统研究的使用价值。

## 第七章 总结与展望

### 7.1 论文总结

本论文通过对国内电火花线切割加工的现状的分析,结合目前的计算机网络技术、通信技术、数控技术和管理技术,提出了一种面向线切割加工车间的 DNC 加工系统。该方法适应我国目前电火花线切割加工车间的自动化加工需求,大大提高了线切割加工车间的资源的利用率,对于多品种、小批量生产零件的加工,显著提高了生产率,因此推广的前景比较广阔。随着我国数控机床改造的进程逐步加快,数控化日益提高,只要将这个系统稍微改造一下,在中小企业的数控加工车间也是比较适用的。

本文主要作了如下的研究工作:

1. 采用计算机网络技术、通信技术、数控和管理技术,提出了高集成度的电火花线切割 DNC 加工系统。并对整个系统进行了功能模块划分和关系模型分析,实现了部分功能模块。

2. 以 AutoCAD 2000 为应用软件系统的操作平台,采用 ObjectARX 和 Visual C++ 面向对象编程技术,将线切割 CAD/CAM 集成在 AutoCAD 中,包括参数化绘图、数字图像矢量化、加工模拟和 NC 文件编辑等功能,真正实现了设计和制造的有机集成。

3. 充分利用 AutoCAD 平台强大的 2D 功能进行绘图,提高了零件 CAD 效率。在实现手工生成加工路线的基础上,增加了参数化自动生成加工路线的方法,提出了一种矢量方法有效地解决了加工方向的自动判别和加工起点的最优选择,实现了刀具轨迹的自动补偿。

4. 介绍了 ARX 与 MFC 编程技术,实现了对 AutoCAD 增加新实体的功能,如抛物线、双曲线等,并且推出了其插补算法,使系统能完成除直线、圆弧以外的其它非圆二次曲线轮廓的加工。扩充了 AutoCAD 的产品设计能力,以提高一些常见模具零件复杂轮廓(如齿轮、凸轮)的绘制及编程效率。

5. 通过对数字图象的预处理和矢量化算法实现对手绘图片或者实物零件进行线切割仿形加工,使得在线切割加工中实现二维创新设计成为可能。

6. 通过对国内线切割加工车间机床的接口问题,针对性的提出了通信方法,解决了 DNC 主机与异构线切割机床问题。

通过上述研究,得出了下列结论:

1. 在 AutoCAD 平台下,开发电火花线切割加工 DNC 系统中的自动编程模块软件是非常实用的,并且不失为一种解决当前电火花加工中所存在问题的切实有效的方法。

2. 完全可以用目前成熟的通讯技术结合制造商,彻底解决异构线切割机床的联网通讯问题,真正实现 DNC 主机和线切割机床的加工信息的双向传递。

3. 本文提出的电火花线切割 DNC 加工系统广义上讲不仅仅是一种系统,而是一种解决方案。浙江大学金工实习基地二维创新设计 DNC 平台就是本系统的一个很成功的实施例

子。

## 7.2 展望

由于作者有很多的工作集中于系统框架的构建，功能模块的划分以及建模。由于时间、精力和物力诸多原因，研究工作还存在以下几个方面的不足，需要进一步完善，也是我将来需要深入研究的课题：

1. 电火花线切割 DNC 加工系统中生产管理模块有待于进一步开发、实现和完善，以真正实现成产订单、设备、工具、材料、工艺文件、NC 程序、数据备份、质量等资料的集中管理。实现与上一级 CAD/CAPP/CAM/MIS 系统的集成。
2. 开发和完善数值脉冲电源，并在试验基础上建立基于专家系统的工艺数据库，开发加工参数优化选取系统，以帮助操作者根据不同的加工条件和要求合理选用加工参数，充分发挥机床潜力，增加本系统的智能化。
3. 结合制造厂家开发和完善开放式的数控系统。使得电火花线切割 DNC 加工系统的通信软件能充分利用机床数控系统提供的开放性软件接口，真正实现线切割机床的有效管理和控制功能。

## 参考文献

- 【1】 刘晋春, 赵家齐, 赵万生等. 特种加工. 北京: 机械工业出版社, 1986
- 【2】 孔德音. 模具制造学. 北京: 机械工业出版社, 1995
- 【3】 许鹤峰, 闫光荣. 数字化模具制造技术. 北京: 化学工业出版社, 2001
- 【4】 李明辉. 电火花线切割加工技术的研究现状及发展趋势. 见: 中国机械工程学会年会论文集●特种加工技术. 北京: 机械工业出版社, 2001. 138144
- 【5】 高立群, 黄因慧, 张祖尧. WDNC 线切割群控系统. 电加工, 1995 (2)
- 【6】 周继烈, 倪益华, 程耀东. 面向模具制造系统的电火花线切割 DNC 系统研究. 电加工与模具, 2000 (2)
- 【7】 Flavio Testi. DNC SYSTEMS: a story that never ends. Journal of society of Engineers, 1996, 77(3): 31~43
- 【8】 Toh K T K. The future role of DNC in metalworking SMES. International Journal of Production Research, 1996, 34(3): 863~867
- 【9】 Donald L F. Broad scope DNC. Manufact Eng. 1987, 99 (2): 49~51
- 【10】 Tho K T K. DNC network maximises return on investment. Sheet Metal Industries, 1995, 72 (2): 30~32
- 【11】 刘飞, 杨丹, 王时龙. 制造自动化. 北京: 机械工业出版社, 1997
- 【12】 王时龙, 刘飞. DNC 集成技术发展综述. 计算机集成制造系统 CIMS, 1998(4): 45~47
- 【13】 王时龙, 刘飞, 张旭梅等. 基于软插件技术的 DNC 通信系统研究. 机械工程学报, 1999, 35(3): 51~54
- 【14】 张旭梅, 刘飞. 集成 DNC 车间的敏捷运行模式. 机械设计与制造工程, 1999 (5)
- 【15】 张旭梅, 刘飞, 王时龙. 集成 DNC 系统及其运行模式探讨. 计算机集成制造系统 CIMS, 1998(1): 32~34
- 【16】 程春峰, 李长星等. ST—DNC 2.1 分布式计算机数控系统. 制造技术与机床, 1995 (7)
- 【17】 唐文虎, 童劲松等. 基于 CORBA 的 DNC 系统的实现. 组合机床与自动化加工技术, 1999(10): 24~28
- 【18】 龙伟等. 异构数控机床 DNC 接口与通信. 机械, 2000, 27 (2): 5~6
- 【19】 陈昌文, 席华荣等. 最新 DNC 软件功能分析. 机械工程师, 2000(1): 25~28
- 【20】 刘飞, 杨丹. CIMS 制造自动化. 机械工业出版社, 1997
- 【21】 熊斌, 谭建荣等. 敏捷 DNC 系统的理论探讨. 机械设计与制造, 1998 (6)
- 【22】 王时龙, 刘飞. DNC 集成技术的发展. 中国机械工程, 1998, 9(2)
- 【23】 聂秋根 张洪兴等. DNC 技术的研究现状与展望. 机械制造, 2002, 40(8): 24~26
- 【24】 魏孝斌, 王宁生等. 企业集成 DNC 技术的研究与应用. 机械与电子, 2002(1)
- 【25】 Beard, Tom. Look What's Happened To DNC. Modern machine shop, 1992, 65(6)

- 【26】方水良. 柔性制造系统的应用研究—两层局域网柔性 DNC 系统的研究和开发: [博士学位论文]. 南京: 南京航空学院, 1990
- 【27】张旭梅. 集成 DNC 及其运行模式探讨. 计算机集成制造系统, 1998, 4(1): 32~35
- 【28】庄熙明. 基于网络的柔性 DNC 系统的设计与实现. 机械设计与制造工程, 1999, 1(28): 29~30
- 【29】杨萍, 刘飞. CAN 总线型 DNC 系统通信技术研究. 制造技术与机床, 1998(12): 7~9
- 【30】许兴华, 马铭辰等. 集成 DNC 在设计制造集成系统中的应用. 制造业自动化, 2002, 24(8)
- 【31】李兴兰, 张友根等. DNC 分布式数控加工系统研究. 计算机与现代化, 2002 (4)
- 【32】王霄. 制造车间异构数控机床的 DNC 集成研究. 机床与液压, 2000(4)
- 【33】闰伟国, 王敏杰等. DNC 技术的内涵和发展. 组合机床与自动化加工技术, 2001(8)
- 【34】李应, 王治森, 郑家强. 企业集成DNC方案探讨. 合肥工业大学学报, 2000 (6)
- 【35】张宇, 初学丰. 模具CAD/CAM/DNC一体化技术. 昆明理工大学学报, 1999, 24(4): 67~72
- 【36】俞成勇, 周继烈等. 一个基于网络的电火花线切割加工 DNC 系统的实例. 见: 中国机械工程学会年会论文集●特种加工技术. 北京: 机械工业出版社, 2001. 167~170
- 【37】俞成勇, 周继烈. 基于网络的电火花线切割加工 DNC 系统研究. 机床与液压, 2003 (3)
- 【38】张浩, 王素敏. 面向车间的微机 CAD/CAM/DNC 系统集成方案及应用. 组合机床与自动化加工技术, 1997(5)
- 【40】周继烈, 黄长林, 倪益华. 数控线切割加工 CAD/CAM 集成化参数化研究. 浙江大学学报, 1997(1)
- 【41】倪益华, 黄长林, 周继烈等. 基于 AutoCAD 环境的线切割编程系统. 模具工业, 2001(5)
- 【42】曾周末, 薛欣伟. 基于 AutoCAD 的线切割加工 CAD/CAM 技术. 制造技术与机床, 2000(4): 47~48
- 【43】罗东梅, 梁桂庆. 基于 CAD/CAM 集成化的线切割自动编程技术. 电加工, 1995(2)
- 【44】李世国. AutoCAD 高级开发技术 ARX 编程及应用. 北京: 机械工业出版社, 1999
- 【45】李世国, 潘建忠等. AutoCAD 2000 ObjectARX 编程指南. 北京: 机械工业出版社, 2000
- 【46】江思敏, 曹默等. AutoCAD 2000 开发工具—ObjectARX 开发工具与应用实例. 北京: 人民邮电出版社, 1999
- 【47】薛欣伟. 数控加工 CAD/CAM 集成技术的研究: [硕士学位论文]. 天津: 天津大学, 1999
- 【48】黄明宇. 计算机模拟显示数控线切割加工轨迹. 金属成形工艺, 2001(2)
- 【49】郝泳涛, 周峰等. 数控线切割 3B 加工指令的图形化自动编程. 机械, 1999, 26(1): 34~36
- 【50】冉伟伟, 杨岳等. 基于 ARX 的线切割自动编程及加工仿真系统. 机械与电子, 1999(6)

- 【51】罗学科,姚刘聪等. 基于 Windows 平台的交互式数控线切割自动编程系统. 新技术新工艺, 1999(5)
- 【52】孔振羽,刘晋春等. 基于 Windows 的新型线切割自动编程系统. 计算机辅助设计与制造, 1998(2)
- 【53】张祖媛等. 数控线切割自动编程系统的开发. 四川工业学院学报, 1999, 18(2)
- 【54】赵万生,李论,郭艳玲. 电火花线切割 CAD/CAM 集成系统的研究.电加工与模具, 2001 (6)
- 【55】吕凤军. 数字图像处理编程入门. 北京: 清华大学出版社, 1999
- 【56】郭明,周继烈,倪益华. 线切割二维仿形加工的图像处理方法. 电加工与模具, 2002(1)
- 【57】何立民. 单片机应用系统设计. 北京: 北京航空航天大学出版社, 1990
- 【58】李华等. MCS-51 系列单片机实用接口技术. 北京: 北京航空航天大学出版社, 1993
- 【59】肖本贤. 用 VB 实现 PC 机与多个 MCS-51 单片机主从式通讯的实例. 制造业自动化 2001, 23(2).
- 【60】何维,杨涛等. WIN95 环境下微机与 8031 单片机的串行异步通信. 半导体技术, 2001, 26(5)
- 【61】杨涛,何维等. 微机与 8031 组成数据采集系统中的多线程处理. 半导体技术, 2001, 26(1)
- 【62】温伟强,梁浩源. 一种 C 语言实现 PC 机与多台单片机通信的新方法. 电子技术应用, 1996(11)
- 【63】袁立新,余汉城. 在打印口上实现向线切割控制器传送数据. 电加工与模具, 2002(3)
- 【64】蒋涛. 多串口通讯编程解决方案分析. 四川大学学报(自然科学), 2001, 24(5)
- 【65】谢小星. CAXA 数控加工造型. 编程. 通信. 北京: 北京航空航天大学出版社, 2002
- 【66】何立民. 单片机应用系统设计. 北京: 北京航空航天大学出版社, 1990
- 【67】范逸之. Visual Basic 与 RS232 串行通讯控制. 北京: 中国青年出版社, 2000
- 【68】李建华,郭明. RS232 和调制解调器高级通信编程. 北京: 人民邮电出版社, 2001
- 【69】苏州沙迪克三光机电有限公司. BKDC 电火花线切割控制机使用说明书. 2000
- 【70】苏州沙迪克三光机电有限公司. WAP2000 线切割编程系统用户指南. 2000
- 【71】王时龙,刘飞等. 一种新型的集成化 DNC 系统. 中国机械工程, 1998, 9(5): 25~28
- 【72】余世浩,李熙亚. 冲裁模 CAD/CAM 系统. 北京: 机械工业出版社, 1996
- 【73】金以文. C/C++与数据结构. 杭州: 浙江大学出版社, 1997
- 【74】张乐益. 线切割仿形自动编程系统研究. 电加工, 1996(3)
- 【75】陈怀国. 基于 AutoCAD 输入图形的线切割自动编程. 常州工业技术学院学报, 1996, 9(4)
- 【76】郭伟. 车间级CAD/CAM/DNC网络系统. 计算机辅助设计与制造, 1996(9)
- 【77】赵新. 面向车间单元的 CAD/CAM/DNC 系统. 机械制造, 1999, 37(4): 44~45

- 【78】施群, 秦观生等. 基于 WWW 的 DNC 系统设计. 制造技术与机床, 2001(2)
- 【79】M.T. Yan, Y.S. Liao. Adaptive Control of WEDM Process Using the Fuzzy Control Strategy. ISEM, 1995(6)
- 【80】Steven, D. Jechson, Ravi O.Mittal. Automatic Generation of 2-axis Laser-cutter NC Machine Program and Path Planning from CAD. Computers in Industry, 1993,21: 223~231
- 【81】W.J.M.Kickert, Van Nauta Lemke. Application of a Fuzzy Constroller in a Warm Plant. Automation, 1976, 12: 301~308
- 【82】J.G.Griffiths. Toolpath based on Hilbert's Curve. Gomputer-Aided Design, 1994,26(11): 839~844

## 致 谢

值此论文定稿之际,衷心的感谢我的导师周继烈老师。在我近三年的攻读硕士学位期间,周老师在学习和科研上给予了我悉心的指导,生活上给予了无私的关怀,论文的字里行间无不凝聚着老师的心血。周老师严谨求实的治学作风、诲人不倦的师者风范时刻激励着我克服困难,认真完成本论文的工作。他求是创新的科研精神和言传身教的师者风范,将使我在以后的研究工作中受益匪浅。他热情随和、宽大为怀的个性修养更将使我受益终身!

感谢教研室的狄瑞坤、倪益华、王家平和应济老师,在平时学习和工作中给予的帮助和支持。特别向倪益华、黄长林老师对我论文的指导和关心表示衷心的感谢。

感谢两年多来和我一起共事过的师兄弟们,他们是:许建广、郭民、徐红启、王建萍、王晓亚、项国峰、傅龙珠、金昊炫、张为民、肖伟杰、欧阳珍、秦丰。感谢他们给予我的鼓励和帮助和启发。

感谢机制 2000 硕的所有同学,特别是:曾伟良、赵明岩、叶良鹏、俞新东、刘家豪等。感谢他们两年多来给予我的关心和帮助。

感谢挚友童晓明先生、宋朋金女士和胡旭东小姐,感谢他们在生活上对我的关心和支持,永远难忘读硕期间和他们共同度过的那段美好时光。并祝愿他们在以后的人生路上一帆风顺!

最后感谢我的父母对我的多年培养,他们殷切的期望和无尽的关怀是鼓励我不断进取的动力。浓浓亲情,无以言报!感谢哥哥和嫂嫂多年来给予我精神上和物质上的支持,是我漫漫求学路上的力量源泉。正是他们无私的奉献和深深的爱,才使我得以顺利完成学业,我愿与他们共享收获的喜悦!

谨以此文献给所有关心、帮助和支持过我的人们!

俞成勇

二零零三年一月于浙大求是园