

关于学位论文原创性和使用授权的声明

本人所提交的学位论文，是在导师指导下，独立进行科学研究所取得的成果。对在论文研究期间给予指导、帮助和做出重要贡献的个人或集体，均在文中明确说明。本声明的法律责任由本人承担。

本人完全了解山东农业大学有关保留和使用学位论文的规定，同意学校保留和按要求向国家有关部门或机构送交论文纸质本和电子版，允许论文被查阅和借阅。本人授权山东农业大学可以将本学位论文的全部或部分内 容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

保密论文在解密后应遵守此规定。

论文作者签名：张群华

导师签名：李进京

日期：2006.6.18

摘 要

在现代生产加工中,伴随着计算机技术与建模技术的发展,数控仿真技术已成为能够预测加工过程表现、产品质量及加工效率的一种有效方法,不但可以检验数控程序的正确性和合理性,避免由于机床走刀错误造成的数控设备损坏的危险,而且大大降低了以往采用工艺试切方法所带来的高消耗和高成本,大大缩短了新产品的开发周期。

在介绍了数控加工仿真技术的历史、发展现状以及存在问题的基础上,比较当前国内外数控仿真系统的理论与方法,结合个人技术水平与现有实验条件,对本院购置的SuperMan-200L数控机床进行了研究。根据数控仿真系统的要求构建了仿真系统的总体结构框架并确定了系统程序流程,将要实现的功能划分为几个具体的功能模块,为软件以后的扩充和维护提供了良好的开发环境。

按照系统开发的总体指导原则,根据仿真系统的需求选择了开发平台,利用Visual C++编程语言,采用模块化结构设计开发了直观、友好的人机交互界面。在分析了数控车削加工编程、仿真特点以及数控加工代码的基础上着重讨论了NC代码的处理,实现了译码模块中的预处理模块以及NC代码程序的处理模块,包括NC代码的读入与编辑,NC代码的词法检查、语法检查,解释、处理、翻译、编译等。通过对插补原理的算法分析实现了插补模块。最后讨论了刀具的动画显示和系统的帮助模块。

该系统具有良好的基于 Windows 全中文仿真加工环境,选择 Visual C++开发环境,利用它强大的界面设计功能设计出了相对简单形象的系统加工环境及交互界面,从而可以直观地显示仿真结果,使用户无需在实际加工的情况下观察加工效果。用户输入数控程序之后,仿真系统根据程序本身来仿真加工过程,体现了加工过程与数控程序的统一性。

关键词: 数控加工 仿真 Visual C++

Study on Simulation System of Superman-200L NC Machine Tool

Abstract

With the development of computer technology and modeling technology, the numerical control simulation technology has become a kind of effective method of foreseeing machining process performance, product quality and processing efficiency in the present production machining. It not only could inspect the rationality and correctness of NC program as a result it avoids the breakdown of NC equipment because of knife moving by error but also reduced the consumption and cost that test cutting brought before and shortened the development period of new product greatly.

Introduced the history, current condition and existing problem of the processing technology of NC simulation, compared NC simulation system theory and methods in existence domestic and international, combined the limit in technical level in person and experiment restrictions, the SuperMan-200L NC machine tool purchased by our college is studied in this paper. The overall structural frame and systematic program process are established according to the requirement of NC simulation system, the function that will be realized is divided in several specific function moduls, which offers good development condition for the following expansion and maintenance of the software.

According to the overall developing guideline and requirement of the simulation system, the develop platform is established on which the visual friendly human-computer interaction interface is developed in Visual C++ with modular structure. Analysed the characteristic of the NC turning programming and simulation, as well as the NC code, the handling of NC code is discussed emphatically in which the pretreatment module and handling of NC code module of decode module are realized. Which includes the readin and

edit, morphology inspection, grammar inspection, explanation, handling, interpolation and compile etc.of NC code. The interpolation module is completed through analyzing the algorithm of interpolation principle. Finally, the dynamic display of cutting tool and help module of system are mentioned.

The system has whole Chinese simulation circumstance based on Windows. Visual C++ is taken as background, whose powerful interface design function is used to design relative simple image and interactive interface of the system fully. So the user can observe the machining effect directly instead of actual machining. The system can simulate the machining process with the program after the user inputs the NC program. It caters for the trend of highly integration in current manufacturing.

Keywords: NC machining , simulate , Visual C++

1. 绪论

1.1 数控机床及数控自动编程技术的发展史

数控加工是 CAD/CAM 技术中最能明显发挥经济效益的生产环节之一。它可以保证产品达到极高的加工精度和稳定的加工质量；操作过程容易实现自动化；生产准备周期短，适宜产品快速更新换代的需要。生产对象的几何形状越复杂，加工精度要求越高，生产批量越小，数控加工的优越性就越能得到发挥^[1,2]。

控制数控机床加工的数控软件和程序编制系统也随之发展起来。大致经过了手工编程、数控语言自动编程和交互式图形编程三个阶段。起初根据图纸要求进行数据的计算，并按规定的指令代码手工编制零件加工程序及手工制作控制介质，编程效率低下、精度差。随着计算机技术的发展和算法语言的出现，为提高编程效率，降低编程人员的劳动强度，人们开始采用电子计算机来代替手工编程的工作。

1.2 数控仿真技术研究现状

在当代生产加工中，加工过程的建模在整个生产规划中显得尤为重要。把计算机仿真技术引入到零件的数控加工中，可以形象、直观地模拟数控加工的全过程，即数控加工仿真^[4]。随着计算机技术与建模技术的发展，数控仿真技术已成为能够预测加工过程表现、产品质量及加工效率的一种有效方法。就其本质而言，计算机仿真是基于数学模型的研究方法。根据仿真的结果，研究者可以得到在给定条件下的性能和系统运行过程中的各种参数^[3]。

1.2.1 数控仿真系统技术研究

NC(numerical control)图形验证分为几何验证和物理验证两个方面。几何验证不考虑切削参数、切削力和其它物理因素(如：加工过程的热效应等)的影响，以刀具和工件几何体为主要检验对象，目的是为了保证刀位数据的正确性。而 NC 物理验证则通过切削过程的动力学模型可实现刀具破损的检测监控、切削振动预报以及切削参数调整与控制，从而达到优化切削过程的目的^[5]。

1) 数控仿真加工环境、工件、刀具及夹具的几何建模技术

数控仿真系统的几何模型负责制造环境的建立和设备模型库的管理,通过各种数据结构建立图形数据库,并响应仿真系统的各种图形调用功能。

曲面模型可以精确定义零件的几何形状,有助于数控加工指令的自动生成和加工结果的验证。曲面模型可以很容易地将曲面棱缘像实体一样加以显示。对此,Jerard^[6]提出了一种基于曲面建模技术的数控仿真方法,将曲面离散成一系列网格点,并根据计算机图形消隐原理来显示最终加工零件。在此基础上,Chappel提出了一种点矢(point-vector)方法,借助以上理论,Oliver, Goodman, Fussell和Erosy^[7]开发了一个较完善的空间验证仿真系统,通过工件点矢与刀具扫描体相交,来获得零件实体模型,该算法还可以用来估计刀具的切入、切出角度及计算切屑的径向厚度,为物理过程仿真打下了基础。但在曲面模型中,形体究竟存在于表面的哪一侧,没有给出明确的定义,故在形体的表达上仍然缺乏完整性。在几何造型中,最高级的模型是实体模型^[8,9]。实体造型能为机械零件构造出完整的模型。Voelcker和Hunk^[10]最早将实体建模技术引入数控仿真之中。采用CSG(实体几何构造法)建模方法构造了一些简单的实体模型,并用来检验数控铣削的正确性。W.P.Wang和K.K.Wang^[11]开发了一种利用整体布尔运算方法的数控仿真系统。将实体扫描体分解为实体边界形成的扫描体的布尔并,与刀具扫描体相减,只需一次布尔减运算,节省了大量运算时间。由于CSG方法在对复杂形体表示上的局限性,很大程度上限制了进一步的开发和应用。因此,又有些学者提出了八叉树建模方法。这种方法可以表示任意复杂形状的形体,易于进行布尔运算,仿真速度快。但存在的问题是,对复杂零件需要一个庞大的八叉树结构,需要提高计算机自身性能。

2) 数控加工中物理模型(动力学模型)建立技术

在传统的加工过程中,动力学模型的建立通常有以下几种方法:解析、实验、机械和数值方法。解析方法与实验方法基于经验公式或某些假设,可给出较正确的物理模型。机械方法及数值方法是在前两种方法基础上伴随着计算机技术发展而刚刚开始的一种建模技术,由于它们与计算机仿真技术的紧密结合,正逐步成为数控加工中不可替代的物理建模方法。

机械建模方法在切削过程中综合考虑了切屑的承载力、切刀的几何形状、加工过程几何变形、加工环境、工件几何形状、工件刀具由切屑力引起的位移及位移反馈、加工过程中干扰(空刀及刀具疲劳)等对切屑力和表面精度的影响。Dever 和 Kapoor^[12]针对数控平面铣削,创建了一个复杂力学系统,在这个系统中他们充分考虑了切刀几何形状及干扰因素的影响。Altintas 和 Spense^[13]利用刀具路径生成算法提出了一种铣削力仿真技术,他们也注意到了刀具空刀、弯曲及系统弯曲变形的影响。通过这些系统的建立,他们可以对数控加工时的铣削力进行预测。

有限元方法是数值方法最为典型的应用。Strenowski 和 Car-roll^[14]最早将有限元建模方法引入数控加工过程仿真,他们提出了一种单元分离技术来模拟切削的成型过程,从中可以得到加工的一些特性。ZhangandBagchi^[15]也利用了有限元方法对加工过程进行了模拟仿真,并将结果与实验作了比较,结果吻合得较好。但由于有限元自身的特点,不适合对间断切削加工过程的建模,对加工过程的开始、结束过程模拟有很大困难。

1.2.2 数控仿真系统图象可视化技术

从某种意义上讲,数控加工过程的动态仿真就是利用动画及计算可视化技术描述实际的加工过程。通过仿真结果的可视化显示,不仅可以发现错误,修改工艺文件,还可以总结数据信息难以表达的规律,优化加工方案,最终达到经济地指导生产的目的。

数控仿真系统图象的动画显示多用于数控几何仿真系统中加工干涉及产品验证方面。在一个数控仿真系统中,当加工环境、刀具及夹具等几何模型建立之后,便可获得动画运动所需的数据结构。为制成完整的动画帧,仅需对运动进行一定的控制。即对运动数据进行分析,生成运动轨迹,这项技术已广泛地应用于刀位轨迹仿真中。实现动画及可视化结果的另一关键技术是消隐着色处理,它多用于加工产品及仿真数据验证上,通过相应的处理算法(如像空间算法),我们便可以通过仿真系统看到计算结果和最终的产品形象。

1.3 国内外研究状况

1.3.1 国外研究状况^[16]

数控加工仿真一直是国外 CAD/CAM 技术的一个研究热点。Kawabe 等人最早利用刀具轨迹,采用边界表示法来获取刀具加工零件的框架图,因为该方法简单,容易实现,所以被早期大多数的 CAD/CAM 系统所采用。但是,因为工件形状越来越复杂,刀具轨迹包含的信息量也越来越多,导致图形形状混乱不清,于是仿真研究开始向三维实体仿真算法方向发展,如:Anderson 研究了三维碰撞检验算法,Chappel 给出了利用向量来分析切削过程的算法,Sungurkin 等提出了特定造型设备上的数控仿真系统,K.K.Wang 研究了基于布尔运算的 NC 检验系统,J.H.Oliver 提出了直接尺寸检验方法等^[16, 17, 69]。

由于加工仿真具有诱人的应用前景,促使发达国家对其进行深入研究,并已出现许多成功的应用范例。在美国已形成了由政府、产业界、大学组成的多层次、多方位的综合研究开发力量^[70, 71]。主要研究机构有:(1)美国国家标准及技术局(NTST)制造工程实验室;(2)美国 Maryland 大学系统研究学院 CIM 实验室;(3)美国伊利诺斯大学芝加哥分校机械系工业虚拟现实研究所。

加拿大 WATERLOO 大学的机械工程系和集成制造系统管理研究中心也有部分与数控仿真相关的研究项目。

英国 Bath 大学机械工程系用 peninventor2.0 软件工具开发出了基于自己的 Svlis 几何造型工具的仿真系统。意大利 Bologna 大学用 B 样条曲面建立端铣刀与工作台模型,采用真实感图形显示铣床精加工过程^[72]。

日本和韩国也有许多高等院校和企业进行了数控仿真的研究。日本大坂大学机械工程系制造工程及系统研究室正在从事虚拟制造系统,三维建模,并行处理及虚拟现实等方面的研究。日本 SONY 公司研制的 FEEDAM 系统可对球头铣刀加工自由曲面进行三维仿真,并进行干涉、碰撞检查。韩国 Turbo-TEK 公司开发出面向培训的虚拟数控车削及铣削加工环境,能够实现数控加工的几何仿真,并配有声音信息。

目前,国外的数控代码检验系统已经发展到了实用化、商品化的阶段,有单独的数控加工仿真软件包:

(1)PTC 公司的 Pro/E 软件:它在数控加工的功能上提供多重走刀方法,在选择加工路径、干涉检验、加工时刀具的进退方式、多曲面加工、

刀具选择、毛坯材料、测量、通用后置处理等方面功能都很强。

(2) EDS Unigraphics 公司的 UG 软件：它将所有的数控加工仿真模块连接起来，从回转曲面车削到复杂曲面的铣削，为用户提供了一个方便实用的仿真环境。在刀具移动过程中，用户可以基于图形编辑刀具路径，它还允许输入来自各种基于 Parasolid 的 CAD 系统的实体模型，并且允许用户自定义菜单和对话框。

(3) GGTech 公司的 Vericut 软件包：该软件可在大部分工作站和微机上运行，能从大多数的 CAD/CAM 系统中读取刀位轨迹，用户使用它可以显示工件尺寸，初始工件、设备、夹具等，并可交互式地输入刀具数据，仿真刀具切削过程。据介绍其最新的模块还可以优化数控代码，减少数控机床的实际加工时间。

(4) Cimplex 公司的 NCV 软件包：它作为单独的软件包可运行于 IBM、Silicon、Sun 工作站，也可与该公司的其它模块连接，运行于各种高性能的图形工作站，它也具有与大多数 CAD/CAM 软件相接的性能。

(5) NCCS 公司的 IPV 软件：该软件能生成零件加工轨迹的仿真图形，修正代码的错误和缺陷。程序员可用它检验全部或部分刀具轨迹。对于零件、设备和夹具与刀具之间的碰撞，系统可以自动给出提示信息。

(6) SILMA 公司的 Cimstation 软件：它具有进行三维图形仿真加工全过程，刀具与机床、工件、夹具和附件等碰撞能很容易的识别出来。Cimstation 提供与现有的 NC 加工环境的接口，可用实际的数控代码驱动仿真图形。该软件可运行于 Unix 工作站，可读取 IGES 或直接的 CAD 接口文件，将 CAD 数据转换为 Cimstation 的数据。

(7) 还有一个应用广泛而且功能强大的数控加工仿真模块是 NC Verify。它支持多种数据输入方式，如 APT-CL 文件、标准 G 代码数控程序、CAM 文件或 CAD 图形文件；具有检查与测量功能，能够测量最终工件或加工过程中的工件；具有良好的可视功能，如动态旋转与截面功能，记录和重放功能；具有动态实时显示功能，及在仿真过程中不但能够显示材料的去除过程，而且实时显示正在加工的程序段；遇到加工错误，可以及时蜂鸣报警，且自动显示实际加工时间和加工进度。

1.3.2 国内 NC 验证的研究概况

我国开展数控加工图形验证技术的研究始于 80 年代末。但即使是由华中理工大学和清华大学合作研制的加工过程仿真器 MPS 以及由哈尔滨工业大学在国防科工委“八五”预研项目柔性制造系统 FMS 关键技术研究计划支持下开发的数控加工过程三维动态图形仿真器 NCMPS, 也存在着不具备精度检验的能力, 且皆以工作站为硬件基础。这种功能上的局限性和运行软件所需要的高昂硬件投资限制了软件系统的推广应用。

为了解决上述问题, 哈尔滨工业大学的刘华明教授带领课题组开发了一套基于微机的复杂曲面数控加工仿真验证软件包。该系统的三维模型没有采用 CSG 和 b-rep 表达, 而是在自己独有的复杂曲面离散技术的基础上构建的^[18]。这种曲面离散技术的优点表现在: 极大地降低了仿真验证系统对硬件的要求, 即使在 NC 程序验证过程中, 用户也可以选择任意角度观察加工过程, 这对以实体造型为基础的仿真验证系统而言是不可想象的, 使加工精度验证能够以一种更加简单的方法得以实现^[19]。另外该系统的开发是以 COM 技术为基础的, 系统的各项功能皆以组件 (COMPONENT) 的形式存在, 使系统更便于维护和升级换代。

1.4 目前数控仿真系统的问题

虽然仿真技术与数控技术都取得了迅猛的发展, 但目前在我国数控加工行业仍存在着以下问题:

- 1) 由于操作、编程复杂, 会操作、编程的人员少;
- 2) 在单件生产时, 感到编程麻烦, 宁可安排使用普通机床;
- 3) 编程和调试程序过程中稍有失误, 即造成设备和质量事故, 虽采用空车运转和软材料试切, 仍不能避免上述问题, 特别是复杂或重要工件的加工, 往往不敢使用数控机床;
- 4) 工件的编程量较大, 受控制系统容量限制而不能使用。遇有大型复杂工件, 即使计算机能自动生成数控代码, 操作者仍感程序调整困难;
- 5) 一些企业受资金不足的困扰, 无力购买大型软件;
- 6) 一些大型软件无法接受操作人员手工编制的数控代码, 或不能对这样的数控代码进行仿真。

机加工仿真的研究目前也存在以下问题:

1) 仿真的加工形式少, 研究范围窄

在众多的切削加工种类与形式中, 目前的仿真主要集中于铣、磨两种。即使在这两种加工方法上, 仿真也局限在很窄的范围内。如铣削中多是仿真棒铣刀和端铣刀, 而这种仿真系统对其他种类的铣刀(如加工成形表面用铣刀)却无能为力。其原因是机械加工种类繁多, 存在着车、铣、刨、磨、镗等多种加工形式; 另一方面加工理论复杂, 不同的加工方法, 刀具形状的加工模型有较大差别。同时, 目前的仿真系统大多进行几何仿真, 即对刀位轨迹、工件与刀具的干涉校验等, 有称之为 NC 校验 (NC Verification)。但在机加过程中, 几何校验只是前提条件, 更为重要的是切削力、刀具振动及刀具磨损等在切削过程中起决定因素的各物理量。

2) 物理仿真过程都是考虑理想切削状态, 与实际切削过程有较大差距

在目前的仿真系统中预先设定了大量假设因素, 如设定工艺系统刚性满足要求, 无振动; 加工材料结构统一, 无硬点等缺陷; 刀具无磨损; 切削要素不发生变化等。这种假定的理想状态不能将切削过程中的随即干扰如工件硬点造成的材质变化、振动造成的切深变化等因素考虑进去, 使仿真系统不能真实地反映实际切削过程。

3) 仿真手段限制仿真系统的发展

计算机技术的发展与仿真技术紧密相连, 过去由于计算机软硬件的限制, 仿真时间很长, 编码工作量大, 程序可读性、维护性差, 这些都为仿真工作带来困难。

以上问题已经引起研究人员的重视, 今后的机加工仿真系统将向着快速运行、面向多种加工形式、更加符合实际状况的方向发展^[20]。

从软件的应用方面看, 多数大牌软件还存在着一些问题。例如: 在技术支持方面, 国外的软件公司在国内发展不够成熟, 往往不能提供全面周到的技术服务, 没有全中文的操作界面和联机帮助, 产品的设计过程不符合中国人的设计习惯等等; 另外, 国外软件的价格昂贵, 性能价格比过高, 而且对运行环境的要求较高, 不符合中国国情。而我们国内大多数软件的图形显示功能不够强, 适应范围窄, 多是基于某一特定设备建立的, 难以扩充到其它不同类型的加工设备^[21]。

此外, 迄今为止, 国内外绝大多数已开发研制出的数控仿真系统都是

几何仿真系统。在现代加工生产中，为提高生产效率，获得较高的加工精度，关键也可以说是决定性的一步是，在加工之前能够给出加工参数(如合理选择刀具、主轴转速、进给率等)的合理评判及对产品质量的合理预测。如果我们能够建立起一个基于产品质量预测与分析的数控仿真系统，一方面它既可以对工件及刀具作出精确的几何描述，对数控程序进行验证；另一方面又可以对加工过程中任意时刻的几何信息进行提取(如切屑厚度，切屑几何形状，刀刃与工件啮合部分)，根据数控加工过程的动力学模型，对影响加工质量的刀具、夹具及工件的弯曲、疲劳、振动及温升进行科学的预测，来获得优化的加工过程参数(如合适的刀具、进给率等)。通过建立一个这样的仿真系统，在实际加工之前不仅可以获得优化的切削加工参数，避免了传统的加工参数依照手册或经验的保守选择，充分发挥了机床的潜能，大大提高了生产效率；而且可以对加工产品的精度进行预测，给出满足加工要求的误差补偿方法，设计出合理的切削工艺方案。只有这样，数控仿真系统才会发挥更大的作用，才能成为完善的、真正意义上的仿真系统。

1.5 课题意义

随着数控加工技术的发展，越来越多的数控程序采用自动编程软件编制，程序也越来越长，传统的检验方法已不能满足生产的要求。目前能够操纵这种机床的人才奇缺，这就要求高校教学中，特别是机械系、机制、模具等专业中，增开《数控编程》这门课，已越来越成为当务之急。但是，由于目前高校经费紧张，不可能拿出更多的资金来上一台数控加工中心，而只买几台数控机床又满足不了教学的需要；此外，工厂许多手工编程完成后，还要试切加工，这势必要增加加工成本。为此，编制数控机床仿真软件，以试图满足学校和工厂这两方面的需要就具有很强的经济意义^[23]。所以为了对数控程序的正确性进行快速、有效的检验，以及高效地为工厂培训人才使其尽快掌握数控机床的使用方法，以适应未来工作岗位，开发一套实用的数控加工仿真系统就非常有必要^[24]。

2. 仿真系统总体设计

2.1 软件选择

2.1.1 面向对象设计思想及其在程序设计中的应用

传统的软件开发是面向过程的，采用结构化的程序设计方法，按功能划分模块，模块与模块之间通过数据进行联系^[26]。但是若需要修改调整时，涉及面很宽，很多程序代码需要重新实现，开发效率低，软件的可维护性、可扩充性差，代码可重用性差。相反，面向对象的程序设计其着眼点是数据而不是功能，其设计思路是从确定表征客观实体的对象开始，建立对象之间的层次结构，确定反映对象状态的属性及施加于这些属性之上的操作，对象之间通过消息互相作用和联系，从而形成软件结构，这种方式符合人们对客观世界的认识过程^[27]。按此方法开发软件可克服传统软件的一些缺陷。

面向对象采用了数据抽象技术，程序对一个对象的数据访问，只能通过其公共界面，而不允许直接操作其数据，这就意味着改变数据的内部表达而不影响使用该数据的程序^[28]，这主要是通过对数据的封装实现的。封装性强调的是数据和代码的一致性，将表达对象状态的数据结构定义与对该数据施加的操作封装在一起，对外仅提供必要的表现和该对象特征的有关属性，而将该对象的具体实现细节及内部联系隐藏起来，对对象的访问只能通过公共界面中提供的操作函数来实现^[29, 30]，只要保证接口方式不变，内部实现可随意改动而不会影响使用数据的程序，这便保证了程序的安全可靠性。当系统不正常时，只能是对象的操作函数出了问题，使程序容易维护。由于模块对外接口关系简单明确，减少了模块间的耦合度，更易实现程序的模块化，提高了模块的可组合性和可重用性。本课题在对各个子功能模块的开发设计中，在进行每个类的特有属性与数据成员提取并封装隐藏起来，取得较好的编程效果。

继承性是面向对象技术不同于结构化设计的一个重要特点，是基类和派生类之间共享数据和机制^[31]。利用继承性，在定义和实现一个类的时候，可以在一个已经存在基类的基础之上来进行，把这个已存在的类所定义的内容作为自己的内容，并加入反映自身特征的新内容^[32]。实际上，

继承性体现的是对问题的差别进行编程的思想^[33]。在分类结构的高层定义公共属性和操作，而在低层只需针对特殊情况增添基类所没有的属性和操作，或者对基类的属性和操作进行修改和重置。

随着类层次结构的扩展，类库逐渐丰富，类的共享范围不断扩大，使得系统的开发不是从零开始，而是尽量利用已研制成的类，继承其有用的部分，倾力开发专用的新子类，这就可以显著地减少创建新类和新对象时的工作量，大大简化了系统的开发工作，并能有效地节省存储空间^[34, 35]。另外，继承性提高了软件的可重用性，使得开发的软件系统具有良好的开放性，新模块创建可以通过继承已有模块并加以补充修改来实现；同时也提高了系统的可靠性，因为通过继承改变模块属性的同时并未影响原模块在系统中的作用^[36]。因此，利用面向对象技术可高效便利地开发出一个开放型系统，这是结构化方法难以实现的。

从传统的各种软件统计结果表明，有关用户界面的程序量达 30%-70%^[37]。而良好的用户界面可以大大缩短人们与计算机之间的距离，使得软件的使用易学、易理解，方便用户，提高工作效率，减少使用计算机的出错率。早期的交互技术与用户界面是和应用程序相互渗透、嵌套、溶为一体的，因而严重地依赖于应用程序。八十年代初开始，把交互技术与用户界面从应用程序中独立出来并逐渐形成相应的学科。现在，随着各种功能强大的开发环境和平台的出现、可视化编程的普及，用户界面的设计变得越来越方便，同时各种应用软件的用户界面也设计得更加友好。

根据数控仿真软件要实现的功能要求，其对界面设计的主要要求有：

- 1) 要能对材料去除和刀具运动情况进行动态真实感图形显示；
- 2) 在动态显示材料去除和刀具运动的同时，程序要进行大量计算；
- 3) 材料的去除情况和刀具的运动轨迹是依据对输入的 NC 程序进行的大量计算，而不是用户操作命令；
- 4) 一个零件的仿真往往要运行相当长的时间。用户往往希望能够控制仿真过程，比如暂停或单步执行以仔细观察加工过程的某些局部情景。

综上所述，对于复杂的软件开发设计，用传统的面向过程的结构化设计方法来开发是困难的，而且系统存在很多的缺陷，诸如系统结构复杂，

代码重用性差, 可靠性差, 调试维护困难, 开发效率低等。面向对象技术更符合求解问题的过程, 而且较传统设计方法更易于把一个复杂问题分解为许多子问题, 以良性结构表达和实现设计问题, 充分体现了软件重用技术^[38], 为开发一个开放型的系统提供了优良的环境, 避免了许多不必要的重复工作, 从而大大提高了开发效率及系统的稳定性和可靠性。

2.1.2 VC++及 MFC 介绍^[39, 40, 41]

Visual C++6.0 是 Visual C++编译器最新且最好的版本, 它已经远非只是一个编译器了。它包括了综合的微软基本类库 (MFC Library), 这使得开发 Windows 应用程序变得简单且高效; 它提供有复杂的资源编辑器, 可以编辑对话框、菜单、工具栏、图象和其他许多 Windows 应用程序的组成元素; 它还有一个非常好的集成开发环境——Developer Studio, 用它可以在编写 C++程序时对程序的结构进行可视化的管理。

Visual C++(以下简称 VC)是一个功能强大的集成开发环境, 熟练地使用它有助于更好更快地编制应用程序。VC 编程的主要几方面有:

(1) MFC 类库

MFC 是 Microsoft Foundation Class Library, 它是 VC 的重要组成部分, 编程过程中可直接使用 MFC 中的类定义对象或在其基础上派生新类。

(2) AppWizard

AppWizard 是 VC 集成开发环境下的一个工具, 实质上是一个高级的代码生成器, 按照用户在其对话框中的设置, 创建 Windows 的基本应用框架, 该框架包括一些 C++语言的头文件、源文件和资源文件, 编译生成的可执行文件具有窗口系统的基本功能。对于不同层次应用的需求, 只需在该框架的基础上进行修改。

(3) ClassWizard

ClassWizard 是 VC 中对类进行管理的工具, 它创建的应用程序框架实际上由应用程序类、窗体类、文档类、视类(从 MFC 派生)以及这些类定义的对象组成。利用 ClassWizard 可以对这些类进行管理, 比如增加或删除成员变量和成员函数, 定义对键盘、鼠标和菜单响应的消息处理函数, 重载 MFC 虚函数等。

(4) 文档和视结构

典型的 VC 应用是文档和视结构，每个窗口都可以看作由窗体、文档和视组成，它们分别对应窗体类、文档类和视类。在 AppWizard 创建的基本应用框架中，窗体类对象管理窗口的边框、标题和菜单，文档类对象负责窗口中内容的存取，视类对象负责窗口中内容的显示。

(5) 消息响应机制

消息是人机交互的纽带，所有的键盘、鼠标、菜单事件都通过消息传递到应用程序中，然后再进行处理。窗体、文档和视对象之间的联系也是通过消息的发送和响应来完成的。此外，窗口的创建和销毁过程中都有大量的消息生成。VC 中，对消息的响应是由消息处理函数来完成的，通过 ClassWizard 可以方便地添加对某一消息的消息处理函数。此外，一个完全集成的 Debug 工具可以让你从各个角度来检查程序运行中的微小细节^[39]。

Visual C++6.0 改善了传统的编程手段，使得程序员可以直接在用户界面良好的可视化开发环境中进行工作。而且具有强大的图形功能，因此适于本系统的开发。

2.2 系统设计^[22]

仿真系统的使用对象是本院购置的江苏东恒集团杰必克超人 (SuperMan-200L) 型多功能数控机床，同时经过少量的修改，也可以用于其他数控系统。本系统采用基于数控代码的仿真检验方法进行模拟仿真，以 Windows2000 作为操作系统平台，运用面向对象的 VC++ 语言并采用 Windows2000 的多线程模式与模块化的设计方法。本系统的具体流程如图 2-1 示。

2.2.1 仿真系统功能模块

理论上讲，系统应该由毛坯选择模块，刀具选择模块，刀架、夹具选择模块，但为简化，均使用某一固定尺寸的部件为代表^[25]。

2.2.1.1 编写数控程序模块

该模块提供了一个供使用者输入和编辑数控程序的接口。操作者能够实现数控源代码的输入、编辑、存储等功能，录入的数控程序将用于仿真加工中刀具轨迹的计算。系统为用户提供了两种程序输入方式：直接输入和记事本输入。记事本输入：系统会打开一个记事本，用户在此记事本中

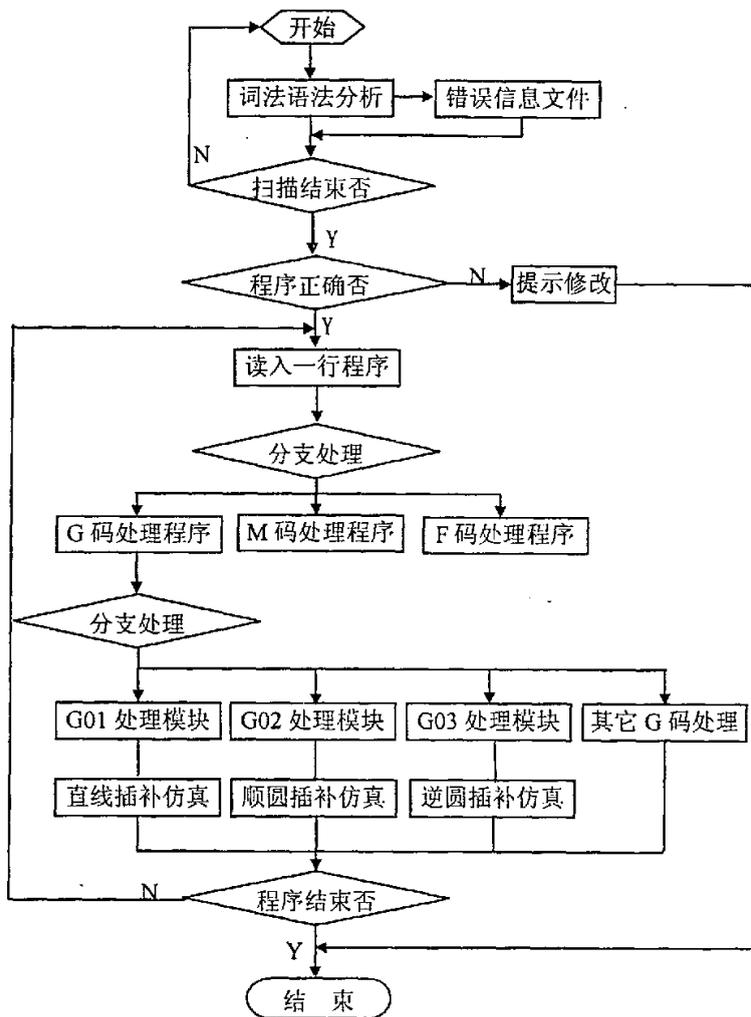


图 2-1 程序流程

fig2-1 program flow chart

输入、编辑、修改零件加工程序，完成后以指定的文件名存放在指定的目录下即可。直接输入：用户直接在零件编辑区域输入零件加工程序，并进行编辑、修改，直到输入的零件加工程序准确无误后，按下“完成”按钮，

被编辑的数控程序就会以指定的文件名存放在指定目录下。

2.2.1.2 数控程序分析模块

软件通过对数控程序的分析 and 译码获取加工信息,经刀具半径、长度补偿,再经过直线、圆弧插补、加减速快速移动等运算得到刀具在运动中某一时刻的位置数据,进而控制刀具图形完成毛坯形状在加工过程中的改变。

用户输入的加工程序必须经过编译、刀具补偿计算、辅助信息处理和进给速度计算等数据处理后,才能被插补程序所使用。译码的主要功能是以程序段为单位处理用户程序,将其中的轮廓信息(如起点、终点、直线和圆弧)、加工速度和辅助功能信息,翻译成便于计算机处理的格式,存放在指定的内存专用区域。译码系统读进地址字符,根据不同的字符做不同的处理。遇到功能代码(如:G、M等),将其之后的数据(G、M后为二位数)转换为整型数值。并存于相应的规定单元。数字串以空格或字母(下一个地址码)结束,处理完一个地址字后继续往后读,放弃地址之间的空格,读下一个地址字符,处理其后的数据,直到读到结束字符为止,即完成了一个程序段的译码^[42]。

编译系统一般经过两次或两次以上的扫描并进行相应的数据处理,就能完成一个程序段的编译,本次开发的编译系统采用5次扫描完成整个译码过程和数据处理。第一遍扫描将.nc数控文件读到相应的数据缓冲区,并进行相应的语法检查,如:是否程序段缺分号(即段结束符),或整个数控程序缺结束标志M02等;第二遍扫描对相应数据缓冲区的数据进行词法检查,如:是否有非法字符,数值出界等。如有错则显示出错提示;第三遍扫描对相应数据缓冲区的数据进行译码,并把译码后的数据存入到一个双向链表的数据结构中;第四遍扫描进行续效指令的处理;第五遍扫描进行刀补处理。

2.2.1.3 仿真及现实加工模块

根据数控程序分析模块得到的位置数据进行插补控制,完成数控加工动画仿真过程,同时根据这些位置数据控制刀具移动。

2.2.1.4 系统帮助模块

用于向使用者提供系统的使用说明及相关的技术说明,主要包括:系

统说明、系统能识别的指令及其功能和使用格式说明。

2.3 系统主要类及其主要属性和方法

系统用 VC++ 在 Windows2000 平台上开发，系统的主要框架继承了 MFC 类结构体系，采用 VC++ 的 AppWizard 自动生成系统的基本框架，VC++ 自动生成的类主要包括应用程序类、框架窗口类、文档类和视图类。在开发过程中主要是对自动生成的文档类和视图类进行修改和扩充，以及对工程中添加自定义的类，如文件转换类、中间转换文件结构定义类、仿真计算类、真实感图形生成类以及真实感图形显示类等等。这里主要介绍文档类和视图类的功能。文档类主要是为应用程序运行期间提供必要的数
据，如视图类完成打开 NC 文件并启动文件转换线程，将输入数据转换成 NC 中间转换文件格式，在仿真计算过程中需要 NC 中间转换文件的数据时，是通过文档来获得的。总之文档类似一种空间，存储着数据的基本集合，构成应用程序所使用的数据单元。

下面介绍系统中类的属性和方法，因为包含的属性和方法太多，不便一一详细列出说明，所以概括说明主要的方法，目的是着重介绍清楚主要轮廓。

文档类名为 CFangZhenDoc。其主要的属性和方法包括：

1) 一个链表结构 FangzhenList，FangzhenList 从 VC++ 的 CobjectList 类派生。

2) NC 中间文件所包含的数据，如刀具起点和终点坐标，起点和终点连线与 XOY 平面的夹角，刀轨长度等。

3) FangzhenList 数据的存储函数和从文件中读入 FangzhenList 数据的方法。

视图类是提供用户观察文档内容的视图和与用户、文档交互的方式：

每个活动的文档都有一个或多个视图与之联系。系统是单文档多视图的应用程序。一个文档类对应了四个视图类。主线程在视图中创建三个子线程，分别对另外三个视图类。视图类名为 CFangZhenView，其主要属性和方法有：

1) 创建和管理子线程所需要的一些数据和函数，如事件对象，子线程对象指针等。

2) 创建和管理子进程所需要的事件和函数,如共享内存的开辟方法,用来保持进程之间协调同步的事件对象等。

3) 用户控制仿真程序执行过程所需要的数据和方法。

4) 文件处理方法,包括 NC 文件的打开和数据读入,NC 中间文件的打开、数据读入和存储。

5) NC 文件向 NC 中间文件的转换方法

6) 动态真实感图形显示方法。包括图形环境初始化,动态图形显示,消隐等。

7) NC 原始文件和 NC 中间文件内容的动态显示方法。

3. 仿真加工环境的形成

在数控仿真系统中,人机交互界面主要用于进行人机交互处理。即在用户的配合下,利用各种交互设备完成CNC系统的操作、各种工作模式的选择、调试、运行管理和维护。用户按需求选择CNC系统的工作模式,进行相应的数据和命令的输入;CNC系统则适时针对用户的操作进行信息的反馈,向用户提供系统运行状态,提供错误警告信息,在设计过程中应充分考虑人的因素。

人机交互界面的开发是利用 Visual C++编程语言,采用模块化结构开发而成,整个界面具有直观、友好等特点,便于使用。本系统数控车床加工环境由四个部件库构成:夹具库、刀架库、刀具库和工件库。在 VC++系统中形成车削加工环境,就是要在模拟动态加工前先将上述各部件库中的部件用 VC++语言进行描述,使得各部件能够以其二维图形的概貌(车削主视图)显现在屏幕上。由于加工环境是多变的,所以各部件库中的部

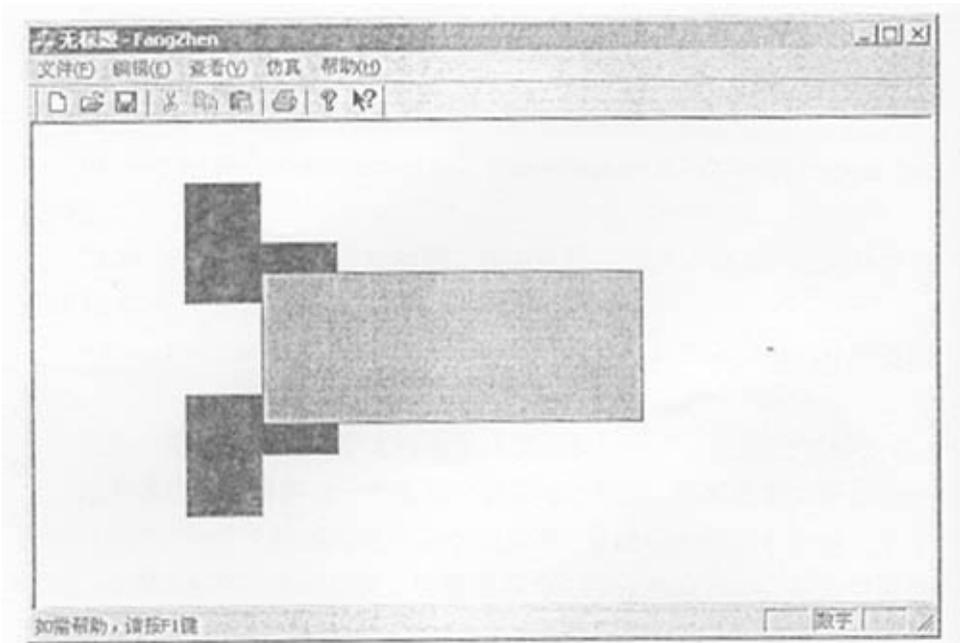


图 3-1 部件实例

fig3-1 parts example

件理论上都是可以无限扩充的。但为了界面的简洁并考虑能够说明问题，本系统对于不同的部件库仅以一个部件作为实例进行描述。考虑个人习惯，系统采用菜单和控件两种方式供使用者对加工环境进行选择。图 3-1 是用 VC++ 语言编写的界面和在其环境下调用工件后形成的加工环境实例。

3.1 用户界面综述

人机交互界面是一个系统功能的外在形象代表，它反映了一个系统的功能强大与否。从表面上看，界面向用户显示当前的系统运行的状态，同时又为用户提供了同系统交流信息发送指令的一个窗口平台，从深层来看，它还负责各个模块的调度应用，任务的调度与协调等功能。它在系统中扮演着关键的角色。因此，本文把人机交互界面作为一个单独的管理模块来进行开发。系统通过 CMACHINE_CONTRI 类提供标准的应用程序界面 API (Application Program Interface)。API 是系统功能模块单元进入系统平台的唯一途径，它向用户提供了方便、友好的人机交互环境，最终向用户提供了直观、高效、易学、易用的操作界面，向软件开发人员提供了功能丰富、使用方便的程序设计环境。

界面窗口的设计是一个面向对象的设计过程。在设计初期，首先要了解对象的内容，也就是操作者希望看到的信息，这些信息主要包括进行人机交互，控制刀具时需要用到的功能菜单，以及发送指令的窗口。因此，需要对界面需要包含的功能结构进行分析，对其功能进行规划。

Windows 下的数控仿真系统由于其功能繁多、结构复杂，因此从具体功能上来讲，它所提供操作者的人机交互界面也必须包含多方面的内容，如位置显示、程序编辑、程序运行状态显示、参数管理、报警显示等。通过对数控仿真加工设备控制需求的分析，以及参考 FUNIC、SIMENS、华中、航天等数控系统人机界面^[43, 44]，确定本仿真加工系统要实现的界面功能目标有如下几个方面：

1) 运行程序编辑功能

新建：新建一个加工程序；

打开：打开一个加工程序进行编辑；

保存：保存编辑中的程序；

插入：在加工程序中插入程序段；

删除：删除程序中的程序段；

此外还有粘贴、复制、查找等编辑功能。

2) 程序运行状态显示功能

当前执行程序信息：程序名，程序运行的行数；

当前运行代码的显示：在窗口中显示当前运行的加工段；

模拟仿真显示：对刀具的轨迹加工路线进行实时的模拟仿真，方便操作者对当前的运行路程和加工位置有大致地了解。

3) 运行坐标显示功能

绝对坐标值：刀具相对于机床原点的坐标值；

相对坐标值：刀具在编程相对坐标系中的坐标值；

加工剩余量：在每一个程序段中，各运动轴当前的坐标值与到达本段程序运行终点的坐标差值，在程序的设计中，根据加工过程的需要，我们对该值进行了放大显示；

机床原点坐标：显示机床原点的坐标值。

4) 加工模式

自动模式：

运行：运行加工程序；

停止：停止加工；

单步运行：单步运行程序；

试运行：试运行程序；

复位：系统复位，恢复到系统期待初始状态；

手动模式：

JOG操作：各运动轴的正向/负向移动。

5) 数据参数管理

刀具参数：刀具号、刀具类型、长度尺寸、半径尺寸等；

机床参数：正负软限位，加工区域，JOG速度等；

补偿参数：螺距补偿、反向间隙补偿、扭矩补偿；

加工视图：数据位置、图形参数设置；

坐标系偏置：工件坐标系零件偏移、工件坐标系零点重设；

坐标系预设：坐标系旋转、比例。

6) 故障诊断

在线故障诊断：根据各种输入信号进行故障诊断；

故障维修建议：根据诊断结果给出相应的处理方案。

此外还有一些其它的功能，以及为用户二次开发或者升级预留的程序接口，方便用户在功能上的增减。

系统的界面模块主要由人机交互界面和相应的回调函数组成。回调函数完成界面上控件所属事件的驱动操作。通过回调函数，进入下一级功能模块，如自动编程、文件管理、参数设置等功能模块^[45]。各模块的功能由相应的类来实现。系统界面各项功能单元模块的结构层次关系如图3-2所示：

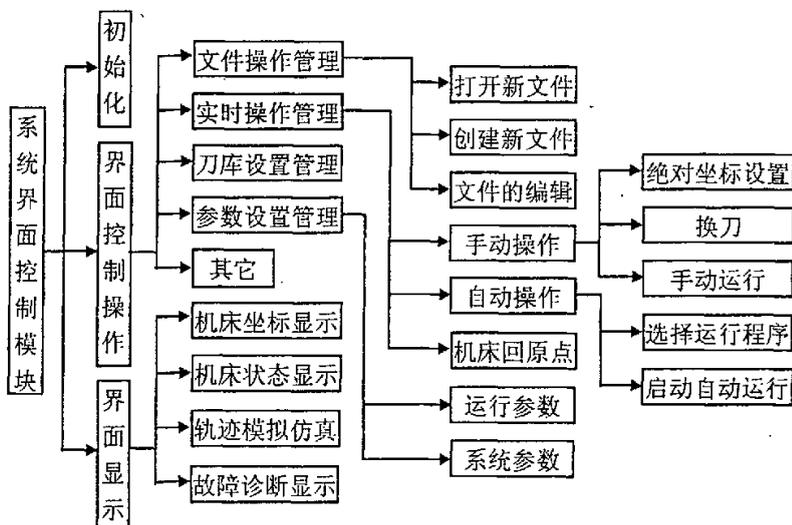


图3-2图形界面模块结构层次

fig3-2 figure interface module configuration arrangement

界面窗口是系统运行过程中的主窗口，也是人机交互的主界面，进入其它的子窗口，通常都要经过主界面进入，同时也是加工过程中显示的窗口。

界面类是基于 CFormView 类的，CFormView 类具有对话框类很多属性，便于界面的开发。CFormView 类是一个非常有用的类，具有许多无模式对话框的特点。象 Cdialog 派生类一样，CFormView 的派生类也与相应的对话框资源相联系。对话框资源中定义了格式化的框架特性，并且列举了它所包含的所有的控件。

类的设计如下：

```
class RunFace : public CFormView
{
    protected:
        RunFace();    //类的构造函数
        DECLARE_DYNCREATE(RunFace)

// Form Data

public:
   //{{AFX_DATA(RunSimuate)
    enum { IDD = IDD_MACHINECONTROL };
    CEdit m_editdata; //创建代码输出框
    CEdit m_runtime; //创建运行时间输出框
    CEdit m_tooltype; //创建刀具类型显示框
    CEdit m_spindlespeed; //创建主轴转速显示框
    CEdit m_feedrate; //创建进给速度显示框
    .....
    double m_xzuobiao; //定义X轴坐标变量
    double m_yzuobiao; //定义Y轴坐标变量
    double m_zzuobiao; //定义Z轴坐标变量
    .....
private:
    .....
};
```

以上程序大体上可分为三部分：创建控件、定义数据及定义接口函数。接口函数是主界面窗口和其它人机交互模块的连接部分，接口函数都采用标准的规范。

3.2 具体加工环境的描述

加工环境主要有机床、工件、刀具和卡盘这几部分，系统的操作人员可以使用这些定义加工环境并保存。

(1) 卡盘的定义

图2-4是卡盘形体的示意图。其中1到4的4个元素代表控制外形和尺寸的4个数据。由数据元素的坐标表示；直径表示卡盘的张开大小，由加工工件的直径决定。卡盘的原点在它的最右端。同时可以利用卡盘的数据元素定义不同类型的卡盘使用，当工件尺寸改变时，系统会自动根据工件的大小改变直径大小。

(2) 工件的定义

工件形体见图2-5，图中D，L分别表示工件的直径和长度，它们控制工件的形状大小。工件的原点在它的最左端。在加工时与卡盘的原点相对应。操作人员可根据需要确定工件原点与卡盘的偏距值，并以此来确定工

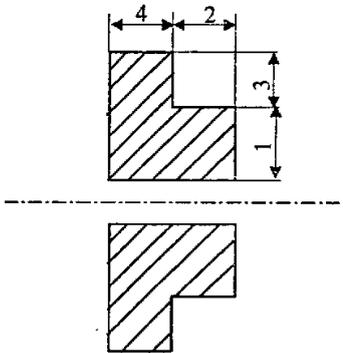


图3-3 卡盘示意

fig3-3chuck symbol

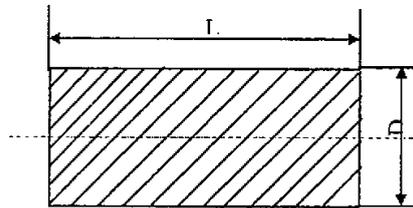


图3-4 工件示意

fig3-4workpiece symbol

件的坐标位置。

(3) 刀具的定义

刀具定义见图所示，刀具比较复杂，尺寸较多。各尺寸共同决定刀具的大小，型号等。原点在左上端。为简化起见，用图3-6所示的简图表示刀具。

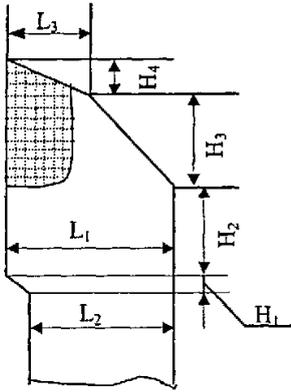


图 3-5 刀具示意

fig3-5 tool symbol

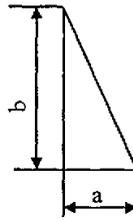


图 3-6 简化的刀具示意

fig3-6 simple tool symbol

3.3 人机交互界面介绍

3.3.1 系统启动界面

单击执行就先进入系统的启动界面，如图 3-7 所示。系统在弹出对话框窗口提示要求输入用户名与密码，这些是通过类 CDLLLoginCtrl 来完成的。用户名与密码在进入系统后可以设置与修改，保证操作者的合法性。在输入用户名和密码得到提交和确认后，就进入了系统的操作控制的主界面，如图 3-8 所示。

总体窗口采用当前流行的 Windows 界面的设计风格，结合数控仿真系统界面的特点设计。通过上面步骤，进入主控制操作界面后，可看到两个比较大的显示窗口、两个小的状态显示对话框，窗口的下端有一排功能按钮。

左侧较大的一个窗口是模拟仿真显示窗口，它的功能是对加工过程进



图 3-7 登陆对话框窗口

fig3-7 entry dialog box window

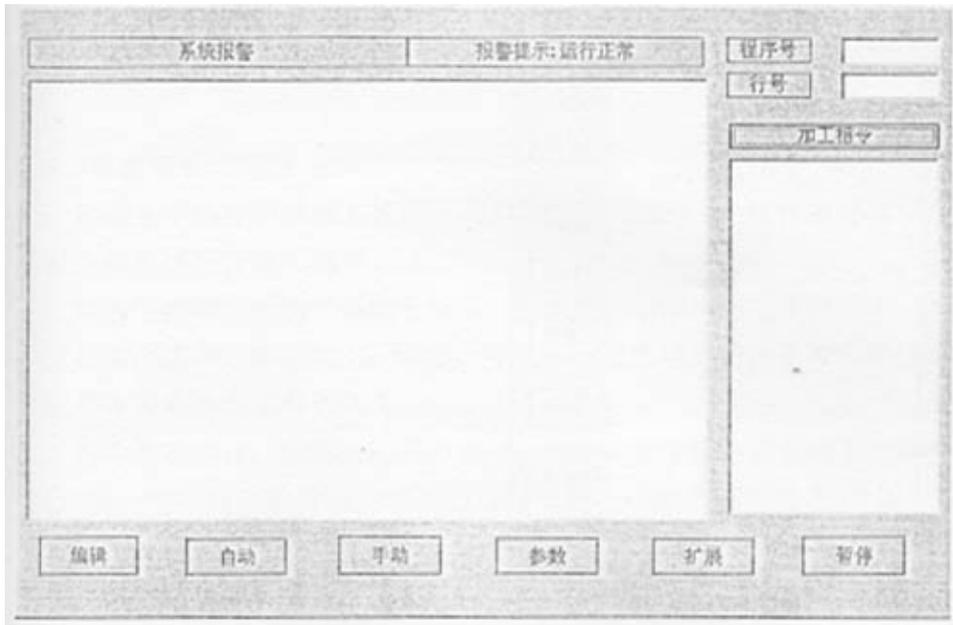


图 3-8 系统控制界面

fig3-8 the controlling surface

行实时仿真,右侧窗口是加工程序显示窗口,对当前正在执行的部分显示,并且对当前运行的程序段进行高亮显示。顶层的状态条是系统报警提示条,对当前系统运行的状态是否出错,或者在操作不当、编程错误等情况下,进行报警提示。最下面一排是系统进入功能模块窗口按钮,其中包括编辑、自动、手动、参数等功能选项。在下文中将对各个功能模块的图形界面和功能进行介绍。

3.3.2 编辑界面

编辑界面包含两部分的功能,一方面是NC代码的编辑功能,包括编写、复制、粘贴等与Word类似的功能;另一方面是对NC代码检验功能,包括语法检测、词法检查、模拟仿真等功能。这些功能可以通过VC++提供的MFC基本类库完成^[46],如图3-9所示,中间竖排的控制按钮主要是编辑功能按钮,位于下部的横排按钮为检验功能按钮。

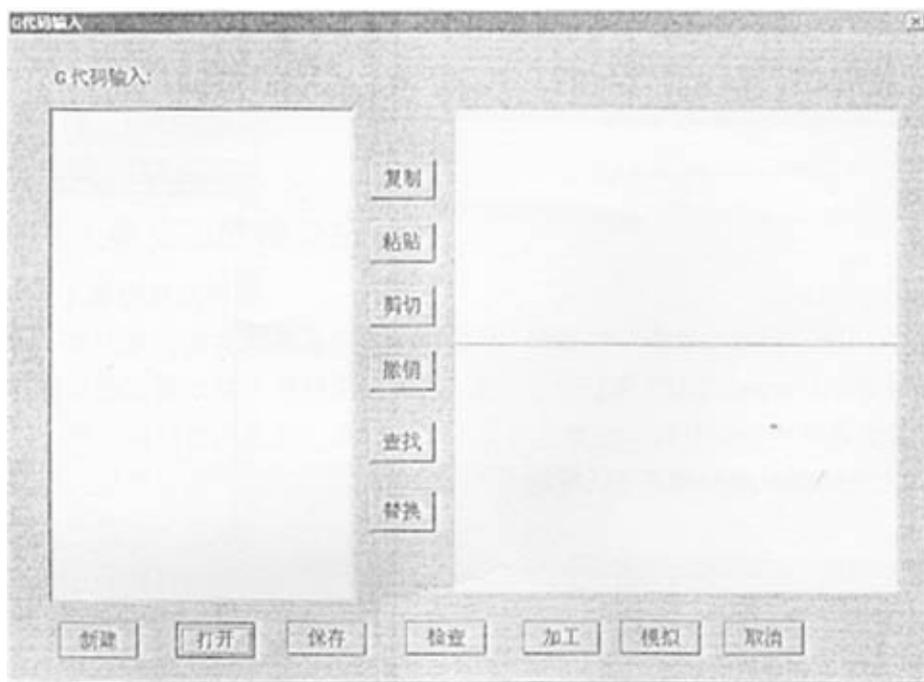


图 3-9 编辑界面

fig3-9edit surface

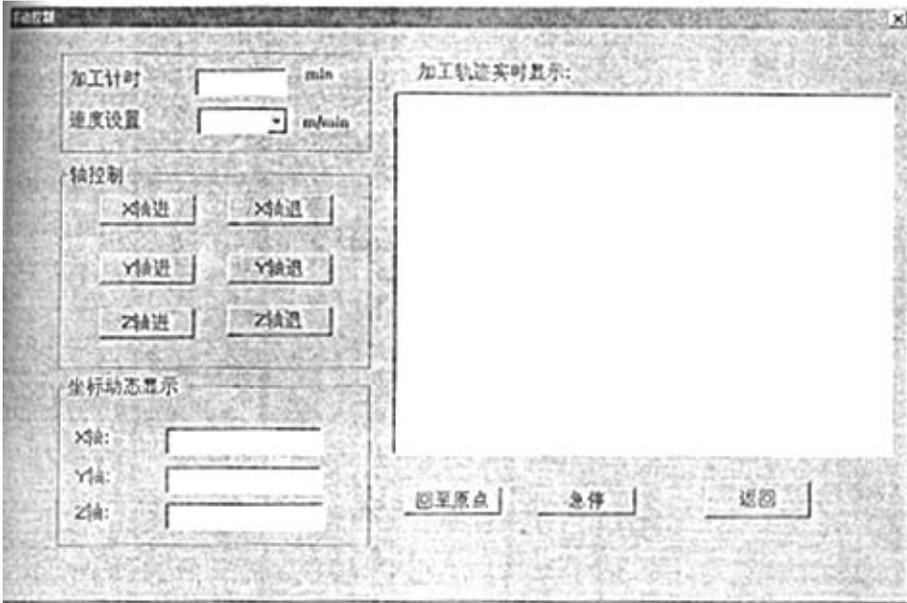


图 3-10 手动控制界面

fig3-10 handiwork controlling surface

3.3.3 自动与手动控制

自动与手动控制功能主要用于刀具的控制与操作上。操作者可以根据功能的需要进行合理的选择。

单击主控制界面的“自动”按钮，系统会自动弹出一个下拉菜单，其中包括程序选择、单段加工、运行、停止等功能选项。操作者可以通过选择这些选项实现对工件的加工。

作为手动操作，系统提供两种方式，一种是通过硬件设置的手轮进行操作；另一种是通过软件的控制界面进行操作。控制界面如图 3-10 所示。此外，系统保留了功能增补接口，方便以后在功能上的升级扩展。

4. 系统具体功能实现

4.1 数控加工编程及仿真的特点

数控车床的品种规格较多,应用也很广泛,但其功能基本相似,它们一般能自动完成内外圆柱面、圆锥面、球面、圆柱及圆锥螺纹等工序的加工,还能加工一些复杂的回转面。

在数控代码的转换这部分,尽管有很多种输入数据的格式,但是,文件的打开关闭,中间转换文件格式是单一的。所以,我们定义了一个一般类,主要完成文件的打开和关闭,定义中间转换文件的一些数据和对中间转换文件的写入数据的方法。一般类中也定义文件数据读入方法,但是,这方法被定义成纯虚函数。对不同的输入文件格式,再从一般类继承一个特殊类。特殊类可以继承一般类中的数据 and 文件处理,数据写入格式的方法等等。利用 VC++ 的动态联编功能,在系统运行时程序可以自动找到应该执行的函数。

在图形显示方面,我们注意到,多视图显示可以同步显示刀具运动、NC 程序内容,以及仿真进度。在同一时间输出信息量大大增加,便于用户查找错误在 NC 代码中的位置,有一定的实用价值。所以我们将系统的显示部分定义成整体与部分结构。

4.2 具体功能的设计与实现

在数控机床使用中,数控编程是一项十分重要的工作,编程质量如何直接决定着产品的加工质量。目前,数控程序的编制有两种方法:一种通过 CAD/CAM 软件直接生成所需的数控程序;另一种由人手工编制数控程序。第一种方法得到的程序可以直接用于加工,第二种方法得到的程序则必须通过仿真或试切才可以进行实际的加工,但仿真只能检查刀具轨迹方面的错误,对于编程语法错误则并不能识别,需要进行语法检查。

4.2.1 数控加工代码分析

数控机床一般采用 NC 代码(又称数控程序)编程。NC 代码编程语言属过程控制语言,其语句是控制数控机床运作的指令集。用 NC 代码编程的特点是:可以不考虑控制过程的具体细节,而是使用加工行业中的流行术

语，写上必要的指令，所以编程过程易学、易懂。工业现场的实际编程工作，一般均由机床操作工人根据工艺加工图纸直接完成或由CAD/CAM软件生成。

数控加工程序都是由若干个程序段按照一定顺序排列在一起组成的。所谓程序段，就是指为了完成某一项动作要求所需要的功能“字”的组合。

“字”是表示某一功能的一组代码符号。一个程序段中各字的排列顺序及其表达形式称为程序段格式，目前以字地址格式应用最广。在这种格式中每个字前有地址，各字的排列的先后顺序并不严格，数据位数可多可少（但不能超出规定范围），不需要的字以及与上一程序段相同的续效字可以不写。这种程序的优点是程序简短、直观、不易出错^[47, 48]。

地址符由英文字母构成，它确定了后续数字的含义。表示地址的英文字母可分为尺寸地址和非尺寸地址两种。其中表示尺寸的字地址英文字母有X, Y, Z, U, V, W, I, J, K, P, Q, R, A, B, C, D, E, H共18个字母；表示非尺寸的字地址有N, G, F, S, T, M, L, O共8个字母。其中数控机床中常用的字地址符有以下几种^[49]：

(1) 程序号字

英文字母O后跟4位以内的数字来表示。程序号必须加在每个程序之首，以区别不同的程序，其后可在括号内加程序名或注释。

(2) 程序段号

用字母N后跟4位以内数字表示，放在程序段前，以区别别的程序段，供检索时用。

(3) 准备功能字

它是指令动作方式的准备性工艺指令，用字母G加2位以内的数字表示。

(4) 辅助功能字

这类指令能够完成机床操作等辅助性的功能，用字母M后跟2位数字表示。

(5) 坐标轴移动指令字

由坐标轴代码和其后的该轴位移量数字码构成。对最小设定单位以下的位数，用四舍五入进行处理。

(6) 进给功能字

由地址符F与其后的几位数字组成，表示进给速度的大小。

(7) 主轴转速功能字

用地址符S与其后的几位数字构成，用来指定机床主轴的转速，单位为r/min。

(8) 刀具功能字

用地址符T与其后几位数字构成，表示现在应用的刀具。此外，针对不同的数控语言，有自己一些独特功能的字。

数控指令有以下几个特征：

(1) 数控指令具有连续性。数控加工中刀具的轨迹是连续的，不间断的，所以数控指令中的几何描述也是连续的，上一个程序段的结束点恰好是下一个程序段的起始点。

(2) 上一个顺序号N与下一个顺序号之间为一个节。数控指令中有些功能是多节有效，有些功能是单节有效，仿真系统必须体现这种要求。

(3) 同一节中几何运动只能在某一固定的平面(xy, yz或xz)中完成。

(4) 同一节中同一字出现多个，后者优先。

4.2.2 译码模块的具体实现

4.2.2.1 预处理模块

预处理模块主要任务是读入数控程序并对读入的数控程序进行词法检查和语法检查。

1) 文件的读入与编辑

对加工程序进行译码首先要将加工程序文件调入内存中。由于现在PC机上普遍装有256兆以上的内存，并且Windows操作系统又具有虚拟内存管理的功能，因此将文件全部下载到内存中已经不存在困难。数控程序源代码是以文件的形式存在的，在本系统中为了以后读取文件的方便，将数控源代码都存储为.nc的文件，C++为用户提供了强大的文件操作函数。根据数控源代码格式的特点：每一行都以“;”结束；各个字符之间有空格。C++提供的I/O流操作函数ofstream, ifstream, fstream恰恰满足了这个要求。类ofstream用于执行文件输出，类ifstream用于执行文件的输入。将加工文件在编辑的窗口中打开，并且实现相应的编辑功能。文件调入内存

后会在对话框中显示出来，然后系统开始对文件进行扫描，进行语法和词法方面的检查。其中文件的调入具体由以下方式完成：

```
void NcDlg::OnOpenFile() //打开程序文件函数
{CFileDialog fdlg(true); //调用打开文件对话框
  if(fdlg.DoModal()==IDOK) //显示打开文件对话框
  .....
  while(CReadString(strLine)) //读取文件
  {
    strFile+=strLine;
    strFile+="\r\n";
    m_edFile.SetWindowText(strFile); //显示代码
    .....
  }
```

在使用MFC提供的函数下载文件之后，我们定义了一个数据结构 Nc_struct 来保存每一个程序。数据结构 Nc_struct 为：

```
typedef char *line;
typedef struct Nc_struct{
  int fileno; //程序文件号
  char *filename; //文件名
  short beginline; //程序在文件中的起始行号
  int lines; //程序行数
  int numctrls; //程序中控制符的数目
  line *plines; //指向程序的指针
  label info *plabels; //指向程序标号的指针
  control_info ctrls[MAX, CTRL_COUNT]; //控制信息表
}Nc_struct, *Nc_pointer;
```

其中 plines 和 plabels 是动态分配的，这样能够节省内存空间，根据文件的大小合理的分配空间。

2) 数控程序词法检查的实现

词法分析是解释数控代码流程中最前面的一个模块。它的主要任务是从数控代码中指出数控代码不能识别的指令。如机床指令集中无 D 指令，

但是在程序中却出现了，就属于这种错误。

一个完整的数控代码是由多个指令组成。代码中一个指令单位称作一个程序段(record)，程序段与程序段之间用结束符来区别。一个程序段中的最大字符数不受限制。数控代码检查是以一个程序段为解释单位，一次解释一个程序段^[50, 51]。

因为词法检查主要是检查程序中是否有机床不能识别的指令。因此首先建立机床字符集，这里构造一个 char 类型的字符集，然后逐个读入 NC 文件中的字符，判断它是否属于字符集，如果不属于则说明程序使用了机床不能识别的指令，给出错误信息，指出错误所在行，并返回到程序编辑状态；如果该字符属于字符集则继续读入下一个字符。

词法检查和语法检查的关系有两种。一是在进行语法检查时，每检查一行时调用一次词法分析；另一种是在进行语法检查前，进行词法检查。本系统选择第二种，这样做的好处是可以节约系统资源、减少语法检查模块的任务。

3) 数控程序语法检查的实现

语法检查是数控代码检查模块中最重要的部分。它的主要任务是根据语法规则对数控代码进行语句分析。和词法检查相比，它所包含的内容更丰富。归纳起来，数控代码中主要有以下几方面的语法错误^[51, 52, 83, 84]：

(1) 指令搭配错误：如每句中不应包含同一组中的两条 G 或 M 代码。

(2) 指令顺序错误：如圆弧插补中 I, K 应在 X、Z 之后，循环指令中其它参数应在 X、Z 之后。

(3) 书写格式错误，包括两种情况：

a 知识不足造成的书写错误：

如程序开头应有程序号，程序号的长度应小于 5 位；行的开头有行号，行号也应小于 5 位；指令行中有指令格式要求的未满足要求；“十”、“一”和“，”没有被放在地址符的后面，R 指令被赋予负值。

b 疏忽造成的书写错误：

程序开头应该为“%”，行的开头应该为“N****”。程序号或行号中不应包含非数字字符；字段值中不应有无效的字母；行号增量应为正值；刀具号不能为无效刀；在数据中“+”和“-”只能出现在数值的前端；

地址符后无数值，突然出现下一个地址符或结束标志。

(4) 极限错误：如子程序号有一定的范围并且不能与主程序号冲突，主程序号有一定范围，子程序嵌套层数最多为 10 层，工件尺寸有一定范围限制，主轴转速和进给转速都有范围的限制。

(5) 程序开始和程序结束错误：如程序段开始无地址，立即输入了数字或程序段符号；还有无程序结束指令的错误，例如子程序应以 M17 结尾，主程序应以 M02 结束。

(6) 逻辑错误，包括三种类型：

a 加工顺序错误：

如刀具指令应放在 G 和 M 代码的前面，换刀前应先移动到换刀点，换刀后的首指令应为 G00。

b 加工参数错误：

如倒角或圆角应该在两条直线之间，加工时主轴和冷却液的状态应该正确，圆弧插补的终点和起点应在同一圆弧上，倒角长度不应小于较短的一条线段。

c 刀具半径补偿的错误：

如不能从 G41 直接到 G42 或反之，使用 G41/G42 的指令条数有一定限制，刀补指令应与 G00 或 G01 一起使用，换刀前应取消刀具补偿。

以上对可能出现的语法错误做了详细介绍，下面就其中一些主要错误检查的实现方法加以说明。

(1) 同一地址重复输入

在数控系统中一般规定数控地址不得重复输入(G, M 除外)，即在同一数控程序段中同一地址符出现的次数不能多于一次。在进行语法检查时，对数控代码中地址符的读入次数做一个统计标志，初态为 0，如遇到该地址符时标志增加 1，一程序段检查完后，检查各个地址符的统计标志，如大于 1 则发出错误信息，并返回编辑状态。

(2) 同组 G 代码重复输入

在同一程序段中，不允许重复出现同一组的 G 代码指令，但是不同的系统对这种情况有不同的处理方式。有的系统允许重复出现，认为最后的一个 G 代码为有效的，并且不产生错误报警。对这种错误检查方式为，

在程序初设置代码组的标志，且初态为 0，在检查代码时，如果有同一组代码出现，代码组的标志进行变化，同时将最近的这个 G 代码数值作为有效值填入该代码组的值中。

(3) 子程序的处理

程序中有固定的顺序和重复模式时，可将其作为子程序存放在存储器中，使程序简单化。为了编程处理的方便，规定用户必须将主程序和各个子程序放在同一文件中，并且子程序的位置应放在主程序的后面。对于子程序，处理是在初始时记录子程序的程序号，如果出现了相同程序号的子

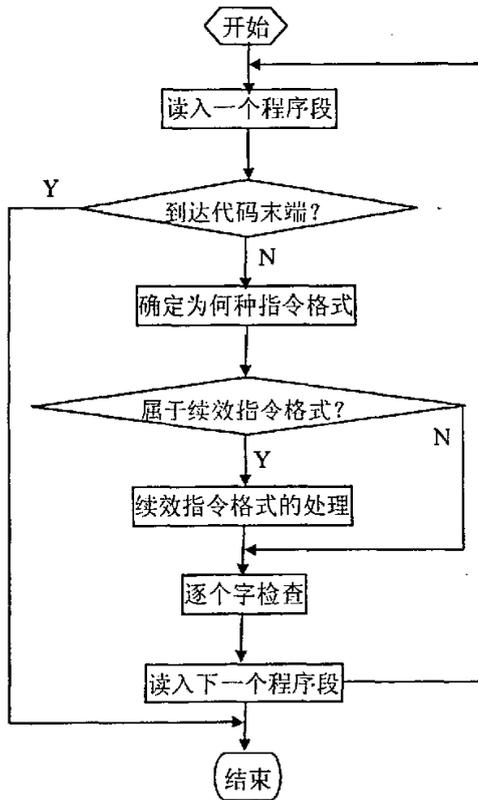


图 4-1 语法检查的算法流程

fig4-1 phrasing check arithmetic flow

程序则给出错误信息。

(4) 语法检查的算法

语法检查的总体思想是逐行逐个进行检查。对于程序段，首先确定属于哪种指令格式，然后进行指令格式的处理，再进行逐个字段的程序检查以及上下文之间的关系检查。语法检查的算法流程如图 4-1。

为了方便进行语法检查和代码翻译逐行逐字段进行，将代码文件存储到一个单向链表数据结构中是一个很好的选择，这样语法检查和代码翻译部分可以方便地对程序进行顺序的查看和检查。单向链表数据结构中，每个结点存储一行代码，本文采用 File_text 结构存储每行的各字段，链表的具体数据结构如下。

```
Struct File_text
{
    char text[100];
    int length;
    File_text *Next_Link;
}File_text;
```

其中字符串 text 用来存储读入文件的每一行程序段，根据 NC 代码文件格式可以知道，NC 代码文件的每一行长度都不超过 80 个字符，所以这样处理是合适的。其中字符串 text 用来存储读入数控程序的每一行程序段，整型数 length 用来记录文件每一行程序的实际长度，即文件中每一行的实际字符数，这一变量在解释每一行各关键指令字时是非常重要的。指针 Next_Link 指向链表的下一个结点，这样就形成了单向链表结构，当前结点是此链表的最后一个结点时，它的 Next_Link 指针为空——NULL；此链表头指针为 Head_Link，由此指针可以访问到链表的任一结点，当将数控加工程序全部读入到链表中时，就可以对这一链表进行分析和处理，即对各指令字和指令字后的特征值进行语法检查。

(5) 代码翻译

代码翻译是在程序检查的基础上，将程序源代码翻译成仿真运行需要的加工数据，存储在中间双向链表中，传递给仿真运行模块。

1) 中间双向链表的设计

考虑到 NC 文件是以代码行为基本单元的,同时各代码具有上下文之间的联系,为便于代码行的来回搜索,决定采用双向链表的数据结构作为程序检查和仿真运行模块的中间接口。

由于数控机床的代码指令是有限的,所以这里双向链表数据结构的设计也以这些代码为根据,双向链表的一个结点对应程序的一个指令行,存储该行中的各代码值。双向链表具体的数据结构如下:

```
enum correction{Cancel, Left, Right};
struct Process_Data
{
    short int Gcode;//G 代码值, 主要指各种加工指令
    bool Turn;//X 和 Z 坐标的顺序, 0 为 X 坐标在前, 1 为 Z 坐标在前
    float Xzuobiao;//X 坐标值
    float Zzuobiao;//Z 坐标值
    float Uinon1;//对 G01 为 P0, 对 G02 为 I, 对 G84, G85 为 P0
    float Union2;//对 G01 为 P1, 对 G02 为 K, 对 G84, G85 为 P2
    int Feed;//进给率、螺距 F
    int Speed;//主轴转速 S
    short int Tool[2];//刀号
    enum correction ToolCorrection;//刀具半径补偿
    short int Mcode;// M 代码 Mcode[2]为程序结束
};
```

2) 代码翻译算法

代码翻译是以前面建立的读入代码行双向链表为基础的,它从读入代码行双向链表中依次读取程序行并将加工数据存储到中间双向链表中。代码翻译的算法流程图如图 4-2 示。为了方便将加工数据存储到中间双向链表中,建立一个 Process_Data 结构体变量 Field_Data 首先将加工数据存储到 Field_Data 中,然后将其作为参数传递给中间双向链表的插入成员函数。由于一些加工数据有续效性,因此在存储该行数据之前,首先应该继承上一行的某些加工数据^[55, 56, 57, 58, 59]。

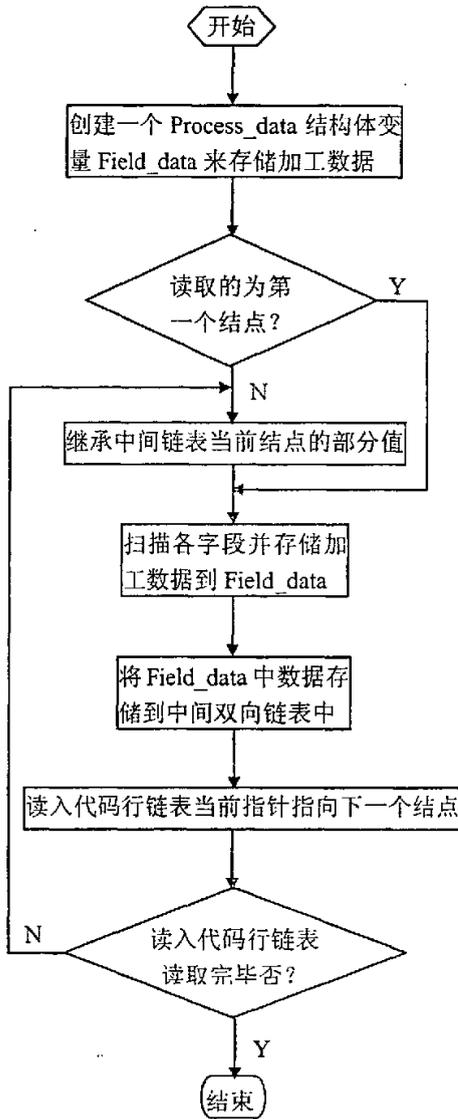


图 4-2 代码翻译算法流程

fig4-2 code translator arithmetic flow

4.2.2.2 NC 程序各程序段的分析模块

在程序读入模块中,我们已经将 NC 程序文件以 ASCII 码的形式读入到一单向链表 File_text 的字符串 text 中,因此本模块就根据每个字符串 text 中的内容来进行进一步分析得到各程序段内各指令字后的数值。具体由以下方式完成:

```
get_data
{
for(i=0;i<File_text->length;i++)
{
switch(File_text->text[i])
{
case 'N' : get_n_val();
break;
case 'G' : get_g_val();
break;
case 'X' : get_x_val();
break;
case 'Y' : get_y_val();
break;
.....
case 'M' : get_m_val();
break;
default;
break;
}
}
line_data=line_data->next;
}
```

通过以上方法可以将各指令字后的数值存储到以下的双向链表结构中:

```
typedef struct line_data
```

```

{
    float   tool_center_x;
    float   tool_center_y;
    int     line_No;
    int     G_code;
    int     G_code_inter;
    int     G_code_plane;
    int     G_code_add;
    int     M_code;
    struct  Arc_pos
    {
        struct  ARC2 {float xs, ys, xe, ye, xc, yc, r}ARC2;
        struct  ARC3 {float xs, ys, xe, ye, xc, yc, r}ARC3;
        struct  CIR {float xs, ys, xe, ye, xc, yc, r}CIR;
    }Arc_pos;
    struct  line_data *prior;
    struct  line_data *next;
}line_data;

```

其中, `tool_center_x`, `tool_center_y`, 分别存储每一程序行中刀具中心位置的坐标值。`int line_No` 存储每一程序行的行号, 即指令字“N”后的数值。`int G_code` 存储此行中的 G 指令值, 大小从 00~99; 为了处理 G 指令中的续效指令, 又定义了三个整型变量 `G_code_inter`, `G_code_plane`, `G_code_add`。结构体 `Arc_pos` 用来记录 NC 指令中顺圆、逆圆和整圆插补时的起点、终点、圆心坐标值及半径值。指针 `prior`、`next` 用来指向链表的前一个和后一个结点, 然后定义两个全局指针变量 `begin`、`end` 指向链表的第一和最后一个结点, 由此可访问到整个链表数据。

4.2.2.3 NC 程序的处理模块

我们已经获得了一个完全有效的结构 `line_data`, 但仅有这些结点中的值仍不能实现刀具运动的仿真, 因为没有进行插补运算, 刀具轨迹上除 NC 程序段中所含的刀具坐标值以外其他坐标值仍是未知数。本模块的作

用是根据 line_data 各结点中的插补指令值和起点和终点坐标值来完成插补运算，主要完成了直线插补和圆弧插补功能。

所有工作分为两部分：(1)完成对 line_data 每一插补指令的解释，从每个结点中的刀心坐标值、插补命令值等信息中获得插补运算所需的各种值(诸如象限判断、终点判别等信息)；(2)完成 NC 程序的插补运算，从而获得刀具运动轨迹上每一点的坐标值。这里我们采用逐点比较法进行插补运算，一是因为这种方法本身的运算简单，插补精度较高；二是因为这种方法的象限转换和终点判别也是非常简单的。

这里要用到一个重要数据结构——Cmd_data 单向链表。

```
typedef struct Cmd_data
{
    unsigned int goo_4,g17_19;
    union XYZ
    {
        struct { unsigned int x, y;}g17;
        struct { unsigned int z, x;}g18;
        struct { unsigned int y, z;}g19;
    }xyz;
    union IJK
    {
        struct { unsigned int i, j;}g17;
        struct { unsigned int k, i;}g18;
        struct { unsigned int j, k;}g19;
    }ijk;
    unsigned int quad;
    struct Cmdcode *next;
}Cmd_list;
```

此链表结构由一全局指针变量 first 指向第一个结点，并由此可以访问到各个结点。其中：

goo_4 存储 line_data 表中每个结点的插补指令值。

g17_19 存储 line_data 表中每个结点中坐标平面的 G 代码值。

quad 是由 line_data 的当前结点和前一结点的 X、Y 坐标判别当前结点所要求的插补运动所处的象限值。

两个联合 XYZ 和 IJK 是用来进行逐点比较法插补运算的终点判别和象限转换。

在实现过程中,针对直线插补和圆弧插补,其具体步骤主要在象限判别,这和终点判定是不同的。逐点比较法中,随着终点相对起点所处的象限的不同,机床向各轴进给的方向也是不同的,为了能够统一地应用第一象限的插补运算的判别公式,必须进行象限的判别和转换。对直线插补,此过程是相对比较简单的,可由下述函数的返回值直接赋予此结点 Cmdlist 中的变量 quad 即可:

```
xx=x_end - x_start;
yy=y_end - y_start;
int quadant(float xx, float yy)
{int q;
if(xx>0&&yy>=0)q=1;
elseif(xx<=0&&yy>0)q=2;
elseif(xx<=0&&yy<=0)q=3;
elseif(xx>=0&&yy<=0)q=4;
return q;
}
```

其中 x_start、y_start、x_end、y_end 分别为前后程序段中的刀心的插补起、终点坐标值。

4.2.2.4 解释处理模块

解释处理模块负责提取有关命令动作和状态信息,即通过对已经存入缓冲区的数据结构进行分析扫描,提取出与仿真有关的动作及状态信息。G 代码是数控代码中最重要的代码,几乎控制了数控机床所有的功能和重要设置,必须对其进行合理有效的处理。从众多的 G 代码中发现,只有如下的代码与仿真有关:G00 点位控制;G01 直线插补;G02/G03 圆弧插补;G41/G42 半径补偿;G90/G91 设置绝对或相对坐标等。为此,在本

文研究中忽略其他 G 代码而只对上述 G 代码进行分析, 并对 M, S, T, F 等代码进行状态分析。

1) 指令特征值的提取

在预处理模块中, 对词法、语法进行分析后, 进行第三遍扫描, 主要任务是: 根据链表 Head_Link 每个结点的字符串 text 中的内容来提取各程序段内各指令字后的特征值, 如把 X、Z、I、J 等坐标数值, 还有 M, S, T, F 等指令和 G 指令后面的代码数值存入到全局链表指针 TNUMps 中。TNUMps 指向的结构体是:

```
struct TNUMps
{
    int n; //行号
    float x; //X坐标
    float z; //Z坐标
    float I; //圆弧 I
    float K; //圆弧 K
    float F; //进给速度值
    int tool; //刀具号
    int M; //M指令
    int G; //G指令
    int S; //S指令
    TNUM *prior; //前一个指针
    TNUM *next; //后一个指针
}
```

2) 数控程序续效指令的处理

在数控程序中, G 代码和 M 代码有两种使用状况: 模态和非模态之分, 模态指令又叫续效指令, 它的特征是在本字段产生之后就一直有效, 直到遇到同组的模态指令为止, 原来的指令才会失效, 由新的指令值代替。在编制数控加工程序时, 当不需要改变某个续效指令值时可以省略不写。例如: X, Y, Z 等坐标值, 还有 M, S, T, F 等指令和 G 指令的同组指令等。G 指令分为几组, 同组中指令一经采用, 直到出现同组其他 G 指

令时才会失效，否则保留继续有效，而且可以省略不写，在同一程序段中出现非同组的几个绩效指令时，并不影响 G 指令的绩效。

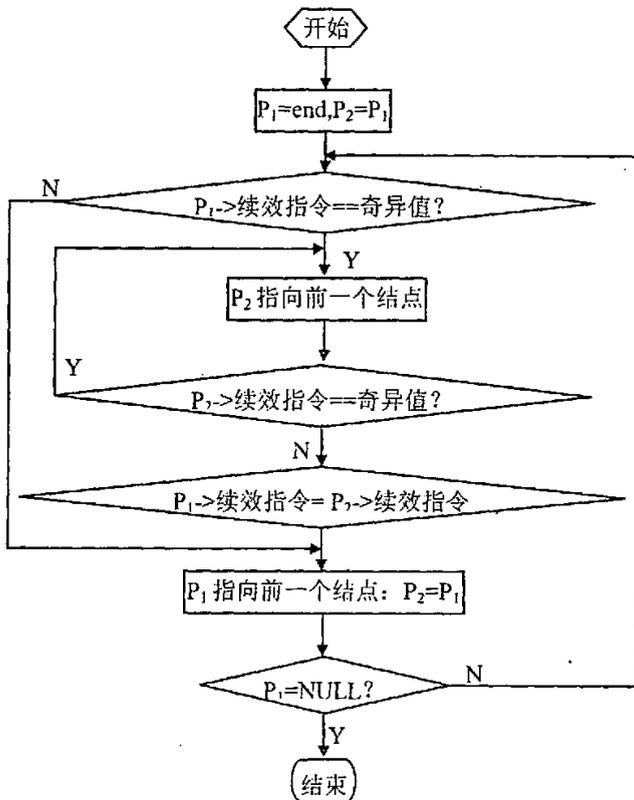


图 4-3 绩效指令处理过程

fig4-3 disposal course of mode instruction

由绩效指令的定义可以知道，链表 TNUMps 的许多结点的绩效值是空的，为了得到轨迹上的刀位点必须对绩效指令进行处理，把指令补充完整，因此在把链表 Head_Link 每个结点的字符串 text 中各指令字后的特征值存入到全局链表指针 TNUMps 之前，应先对 TNUMps 新开辟的结点进行初始化，如 X、Z、G 等绩效指令值赋奇异值，即赋予它们定义域以外的其他值，如 G 可以赋 1000 等，对各绩效指令组来说，凡在值不等于初

始值的，两结点之间的续效指令的值应等于前一个结点相应的指令值，在增加完 TNUMps 的结点之后，用以下算法处理续效指令，具体过程如图 4-3 所示。框图中 P₁、P₂ 初始为链表的尾指针，指向最后一个结点，在对续效指令处理时，对数据结构采取由后向前的搜索策略，经初值判断后，可以得到各程序段的模态指令状态。

4.2.2.5 语法检查功能模块的设计与功能实现

在代码编辑功能完成后，就要进行代码处理。代码处理的过程就是译码的过程。目前通常的译码工作有两种形式：一种是“解释”方式；另一种是“编译”方式。解释方式就是一边翻译一边执行，即在插补模块进行插补时，解释程序预解释下一条程序，等本条程序插补完后，再将下条程序的预解释结果进行插补，不产生被执行的目标程序而是执行源程序本身。这种方法实现简单，但各模块之间是顺序的，串行的，效率低，不易处理程序间的转接，容易造成停顿与过切。编译方式就是把源程序翻译成等价的目标程序存入缓冲区，开始加工时直接执行此目标程序。这种方法速度快，效率高，但是需要较大的内存，并且编译与加工之间存在时间间隔。尤其是当利用小段直线进行复杂零件的插补运算时，极易造成系统资源紧张，降低系统总体性能。由上可知，这两种方法都不能很好的完成译码工作^[42]。本文提出了另外一种“层叠法”来控制代码的翻译。首先在进行第一次处理代码时，主要进行词法、语法的错误检查，及简单的词意解释，但一般不进行数据处理。当代码的词法和语法都基本正确时，再进行第二次处理，即扫描代码整体进行译码和数据处理工作。这种方法节约系统资源。只需要一般配置即可，不需要额外开销。同时它适应于大程序的译码和复杂数据处理，不会降低系统的数据处理性能。其具体如下：

首先应该进行语法检查，检查出错误并给出修改提示。我们采用逐层检查法，逐层检查法就是按照程序结构检查、语法检查和词法检查的顺序进行检查。程序结构检查主要是检查程序结构的完整性，比如程序开始字符、程序号、程序结束代码等。语法检查主要是查看数控指令是否符合数控系统的语法规则。词法检查是对关键字后面的数字类型和数字范围的检查。本系统中定义了数十种语法和词法规则，同时也可以根据系统需要添加新的语法规则。这样，经过检查可以基本保证程序的语法正确性。一旦

检查出错误，系统将给出出错行号和诊断信息，以便于修改。

1) 语法分析方法

目前数控程序的程序段格式都是采用字地址符格式，常用功能字有准备功能字、辅助功能字、尺寸字、主轴转速功能字、刀具功能字和进给功能字。对于尺寸字、主轴转速、刀具及进给功能字则给出了参数取值范围。功能最多的是准备功能字和辅助功能字。ISO允许准备功能可以从G00-G99，辅助功能同样也是从M00-M99。但这些功能并没有全部使用，依据数控系统的功能的不同而有所不同。ISO只对其中少部分常用的功能作了标准化，留给制造商一定的余地开发自己的新功能。同时G功能有模态和非模态之分。另外，G功能通常分成不同的组。要求同组的G功能在同一程序段中只能出现一个。在开发数控语法检查系统时，参考了计算机高级语言编译原理中的思路和方法，将整个语法检查过程分成三部分：结构分析、词法分析和语法分析。

a 结构分析

一个完整的程序必须包括程序的开始部分，程序的内容部分和程序结束部分。程序开始部分常用%表示程序开始，随后写上程序号，如：

```
% 10
```

程序内容部分是整个程序的核心部分，由一个个程序段组成，表示数控机床要完成的全部动作。程序结束部分以程序结束指令构成一个最后的程序段。程序结束指令常用M02。

一个程序段由三部分组成：起始是序号字为N****，接着是若干个功能字。常用的功能字有：准备功能字G，坐标功能字X，Y，Z，辅助功能字M，进给功能字F，主轴转速功能字S和刀具功能字T等六种。

b 词法分析

结构分析之后是词法分析。以每一行为单位作为字符流，完成滤去空格、拼数字、拼复合词的任务，变成“字地址符+数字”形式的指令单词序列。数控系统常用单词符号：

字母：X，Y，Z，G，M，S，T，F，H，D，N，I，K，J

数字：0，1，2，3，4，5，6，7，8，9

符号：“+”、“-”、“;”、“.”

在词法分析中可以首先排除下列语法错误：①非法字符(字母或符号)；②非法指令；③地址后的数字不满足规定的范围；④小数点出错；⑤程序无结束指令；⑥字地址后无数字。完成以上工作后就要用语法分析来识别所给出的单词符号序列是否符合给定语法规则。

c 语法分析

语法规则也是需要进行设定。表中列出了与相关指令相匹配的功能字。语法分析从读入第一个单词开始，与语法规则中的相关指令相比较，如果在规则表中列出了相关的要求，则调用相关的语法检查单元，按语法规则要求继续读入单词，并判断当前读入的单词是否是与之相匹配的功能字，从而逐段地判断。如果在程序段中出现了附录中没有列出的功能字，则出错^[60]。如下表列出了部分指令：

指令	功能匹配字			
	尺寸字	进给功能字	主轴功能字	其它功能字
G00	X、Z	NULL	NULL	NULL
G01	X/U、Z/W	F	NULL	NULL
G02	X/U、Z/W、I、K	F	NULL	NULL
G03	X/U、Z/W、I、K	F	NULL	NULL
M00	NULL	NULL	NULL	NULL
M02	NULL	NULL	NULL	NULL

表4-1 指令语法规则表

chart4-1 phraseological rule repertoire

2) 语法检查模块的程序设计

根据以上分析，语法检查类应该封装各个层次的语法检查功能，因此类的设计如下：

```
class CCodeCheck: public CObject
protected:
CStructcheckQ; //程序结构检查
Cwordcheck(); //程序词法检查
```

```

CGrammercheck() ;//程序语法检查
Public:
    CCodeCheckp; //构造函数
    .....
}
    
```

在类中定义了三个成员函数，封装了相应的结构检查、词法检查和语法检查，在需要相应的功能的时候只需要对相应函数进行调用。语法检查的程序流程图如图4-4所示。

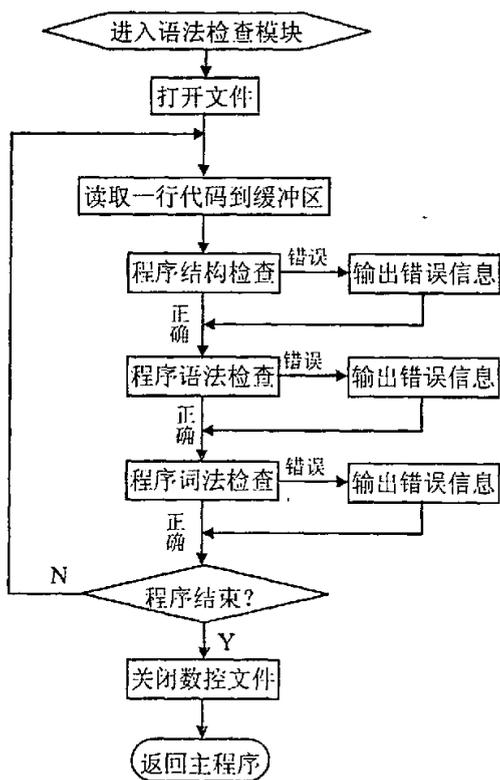


图 4-4 语法检查程序流程

Fig4-4 phrasing check program flow

4.2.3 插补模块的设计与实现

仿真系统的主要问题是怎样控制刀具或者工件的运动,一个连续切削控制的仿真系统必须控制刀具相对于工件的已给定的速度、沿着指定的路径运动。插补模块的任务就是要完成在轮廓起点到终点的中间点的坐标值计算,对于轮廓控制来说插补是最重要的计算任务,插补必须是实时的,即必须在有限的时间内完成计算任务,对各坐标轴分配速度或位置信息。本系统中,在对插补算法进行分析和改进的基础上,给出了相应的插补程序及程序验证。

4.2.3.1 数控插补概述

数控系统的根本作用是对机床运动的轨迹进行控制,复杂形状零件加工是数控加工技术的重要应用对象。一般情况下已知运动轨迹的起点坐标、终点坐标和轨迹的曲线方程,由数控系统实时地计算出各个中间点的坐标。这就是说需要通过数控系统的计算“插入、补上”运动轨迹各个中间点的坐标,通常把这个过程称为“插补”。插补的结果是输出运动轨迹的中间坐标值,机床伺服驱动系统根据这些坐标值控制各坐标轴协调运动,加工出预定的几何形状^[42]。插补可分为硬件插补和软件插补,软件插补就目前普遍应用的算法而言可分为两大类^[61]:脉冲增量插补和数字增量插补。

4.2.3.2 插补的算法分析

由于本系统只对一些基本的数控代码进行仿真,因此采用了最基本的逐点比较的算法,它能够实现直线、圆弧和非圆弧二次曲线的插补而且插补精度比较高。其基本思路是被控制对象在按要求的轨迹运动时,每走一步都要和要求的轨迹相比较,由比较结果决定下一步移动的方向,这种算法的特点是运算直观,插补误差小于一个脉冲当量,输出脉冲均匀,而且输出脉冲的速度变化小,调节方便,因此在两坐标数控机床中应用较为普遍。直线和圆弧是构成工件轮廓的基本线条,因此大多数CNC系统一般都具有直线和圆弧插补功能,只有在有些要求较高的系统中才具有抛物线、螺旋线插补功能。本系统根据实际加工需要,具备直线和圆弧的插补功能^[62, 63, 64, 65, 66]。

1) 直线插补的算法分析

在xy平面第一象限内，假设待加工零件轮廓的某一段为直线，若该直线加工起点坐标为坐标原点O，终点A的坐标为 (X_e, Y_e) ，设点 $P(X_i, Y_i)$ 为任一动点，正处在直线OA上，则下式成立：

$$\frac{Y_i}{X_i} = \frac{Y_e}{X_e} \text{ 即 } X_e Y_i - X_i Y_e = 0;$$

若点P正处在直线OA上方，则下式成立：

$$\frac{Y_i}{X_i} > \frac{Y_e}{X_e} \text{ 即 } X_e Y_i - X_i Y_e > 0;$$

若点P正处在直线OA下方，则下式成立：

$$\frac{Y_i}{X_i} < \frac{Y_e}{X_e} \text{ 即 } X_e Y_i - X_i Y_e < 0.$$

于是有如下结论：令 $F = X_e Y_i - X_i Y_e$ ：

- 当 $F=0$ 时，点 $P(X_i, Y_i)$ 落在直线上；
- 当 $F>0$ 时，点 $P(X_i, Y_i)$ 落在直线的上方；
- 当 $F<0$ 时，点 $P(X_i, Y_i)$ 落在直线的下方。

当偏差值 $F>0$ 时，应向x轴正方向发出一进给脉冲，刀具从现加工点向X轴正方向前进一步；当偏差值 $F<0$ 时，应向Y轴正方向发出一进给脉冲，刀具从现加工点向Y轴正方向前进一步。这样就实现了第一象限的直线插补，其它象限的插补原理类似。

直线插补的程序片断如下：

.....

```

m_result=0; //置偏差初值为零
OnlineChabu0; //直线插补函数入口
{
if(m_result>=0)
{
if (m_xe>=0&&mye>=0)//一象限x轴进给
{themotor.MotorX(); //驱动X轴电机正向进给

```

```

m_result=m_result-m_je;//偏差计算
}
.....其它象限X轴进给
}
else
{if(m_xe>=0&&m_je>=0)//一象限Y轴进给
{themotor.MotorY();
m_result=m_result+m_xe;
}
.....//其它象限Y轴进给
}
}

```

2) 圆弧插补的算法分析与改进

加工一个圆弧，很容易联想到把加工点到圆心的距离和该圆弧的名义半径相比较来反映加工偏差。设要加工第一象限顺时针走向的圆弧，半径为R，以原点为圆心，起点坐标为 (X_0, Y_0) ，圆弧上任一加工点P的坐标为 $P(X_i, Y_i)$ ，P点与圆心的距离R的平方为 $R^2 = X_i^2 + Y_i^2$ ，推出逐点比较法偏差计算公式。

若点 $P(X_i, Y_i)$ 正好落在圆弧上，则下式成立：

$$X_i^2 + Y_i^2 = X_0^2 + Y_0^2 = R^2$$

若点 $P(X_i, Y_i)$ 正好落在圆弧外侧，则下式成立：

$$X_i^2 + Y_i^2 > X_0^2 + Y_0^2$$

若点 $P(X_i, Y_i)$ 正好落在圆弧内侧，则下式成立：

$$X_i^2 + Y_i^2 < X_0^2 + Y_0^2$$

设逐点比较法圆弧的偏差判别式为：

$$F = (X_i^2 - X_0^2) + (Y_i^2 - Y_0^2)$$

当 $F > 0$ 时，向y轴发出一个负方向运动的进给脉冲，当 $F < 0$ 时，向X轴发

出一个正方向进给的运动脉冲，这样就实现了第一象限圆弧逆时针的插补。进给一步后新的加工偏差经递推为：

$$\text{当 } F > 0 \text{ 时，应走 } -\Delta Y, \text{ 新偏差 } F_{i+1,j} = F - 2Y_i + 1$$

$$\text{当 } F < 0 \text{ 时，应走 } +\Delta X, \text{ 新偏差 } F_{i,j+1} = F + 2X_i + 1$$

其它各象限圆弧的顺时针和逆时针插补的基本原理和第一象限圆弧的顺时针插补的原理是一致的。表4-1为顺圆四象限轨迹插补的偏差和坐标计算方法。

象限	$F \geq 0$		$F < 0$	
	偏差计算	坐标计算	偏差计算	坐标计算
I	$F_{i+1} = F - 2Y_i + 1$	$Y_{i+1} = Y_i - 1$	$F_{i+1} = F + 2X_i + 1$	$X_{i+1} = X_i + 1$
II	$F_{i+1} = F + 2X_i + 1$	$X_{i+1} = X_i + 1$	$F_{i+1} = F + 2Y_i + 1$	$Y_{i+1} = Y_i + 1$
III	$F_{i+1} = F + 2Y_i + 1$	$Y_{i+1} = Y_i + 1$	$F_{i+1} = F - 2X_i + 1$	$X_{i+1} = X_i - 1$
IV	$F_{i+1} = F - 2X_i + 1$	$X_{i+1} = X_i - 1$	$F_{i+1} = F - 2Y_i + 1$	$Y_{i+1} = Y_i - 1$

表4-2 顺圆四象限轨迹插补的偏差和坐标计算方法

char4-2 the deviation and coordinate arithmetic of track interpolation in four quadrant

圆弧轨迹插补不同于直线，一段圆弧可能跨越几个象限。从表2可以看出，每当一段圆弧跨过象限后，原来的偏差计算、坐标计算式必须进行修改。在插补过程中每走一步必须检查是否会跨越象限，一旦查到跨越象限，就要根据顺圆或逆圆改进进入该象限的一套计算方法。传统的过象限处理比如查表法等非常麻烦。我们对逐点比较法圆弧插补算法进行了改进，根据动点坐标本身位置的变化自动转换运算和判别，自动转入其他象限，改变偏差计算、坐标计算式。

改进后的圆弧插补原理和实现过程如下^[67, 68]：设判别数N，顺圆时N=1，逆圆时N=0：

设判别式 $P=F_f \oplus N_f \oplus X_f \oplus Y_f$ ，计算结果当 $P=0$ 时，进给X方向，当 $P=1$ 时，进给Y方向。这里 F_f 为偏差值 F 的符号位， $F \geq 0$ 时， $F_f=0$ ； $F < 0$ 时， $F_f=1$ 。这里 X_f 为 X 的符号位，当 $X \geq 0$ 时， $X_f=0$ ；当 $X < 0$ 时， $X_f=1$ 。同样， Y_f 为 Y 的符号位，当 $Y \geq 0$ 时， $Y_f=0$ ；当 $Y < 0$ 时， $Y_f=1$ 。

设判别式 $(\Delta X)_f = (N \oplus Y_f)$ ， $(\Delta X)_f=0$ 时， $\Delta X=1$ ； $(\Delta X)_f=1$ 时， $\Delta X=-1$ ， $(\Delta Y)_f = N \oplus X_f$ ， $(\Delta Y)_f=0$ 时， $\Delta Y=1$ ； $(\Delta Y)_f=1$ 时， $\Delta Y=-1$ 。

偏差计算变为 $F_{i+1} = F + 2\Delta XX_i + 1$ 和 $F_{i+1} = F + 2\Delta YY_i + 1$ ，坐标计算变为 $X_{i+1} = X_i + \Delta X$ ， $Y_{i+1} = Y_i + \Delta Y$ 。

顺圆四象限的轨迹插补用符号判别法的偏差、坐标计算方法，如表4-3所示。表4-3在偏差、坐标计算方法上比表4-2简单，且能达到顺圆圆弧轨迹插补的要求。

象限	$F \geq 0$		$F < 0$	
	偏差计算	坐标计算	偏差计算	坐标计算
I	$F_{i+1} = F + 2\Delta YY_i + 1$	$Y_{i+1} = Y_i + \Delta Y$	$F_{i+1} = F + 2\Delta XX_i + 1$	$X_{i+1} = X_i + \Delta X$
II	$F_{i+1} = F + 2\Delta XX_i + 1$	$X_{i+1} = X_i + \Delta X$	$F_{i+1} = F + 2\Delta YY_i + 1$	$Y_{i+1} = Y_i + \Delta Y$
III	$F_{i+1} = F + 2\Delta YY_i + 1$	$Y_{i+1} = Y_i + \Delta Y$	$F_{i+1} = F + 2\Delta XX_i + 1$	$X_{i+1} = X_i + \Delta X$
IV	$F_{i+1} = F + 2\Delta XX_i + 1$	$X_{i+1} = X_i + \Delta X$	$F_{i+1} = F + 2\Delta YY_i + 1$	$Y_{i+1} = Y_i + \Delta Y$

表4-3 偏差和坐标计算方法

chart4-3 deviation and coordinate arithmetic

综上，我们可以设计出圆弧插补的程序流程图，如图4-5所示。

根据以上流程图，圆弧插补部分程序设计如下：

.....

m_xi=m_x0;//动点X坐标初始值

m_yi=m_y0;//动点y坐标初始值

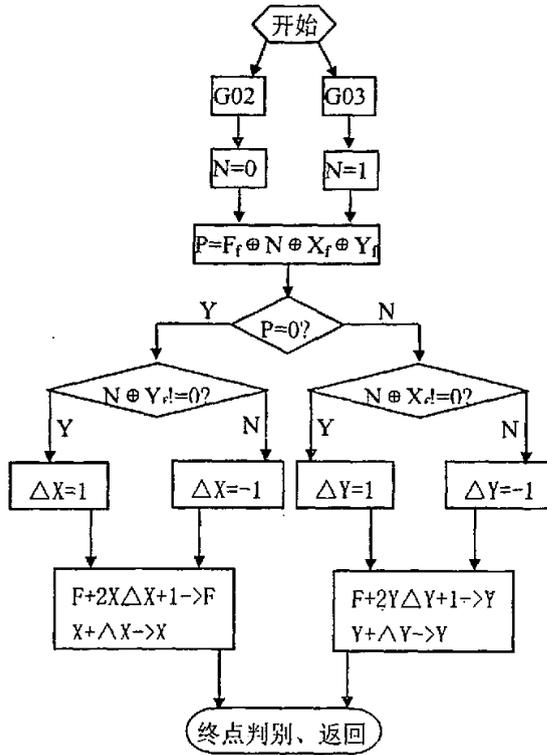


图4-5 圆弧插补程序流程图

fig4-5 circular arc interpolation program flow

$m_result = (m_xi^2 + m_yi^2) - (m_x0^2 + m_y0^2); //$ 偏差初始值计算

.....

OncircleChabuQ //圆弧插补入口函数

{

if((m_result < 0) ^ (m_g0 < 0) ^ (m_xi < 0) ^ (m_yi < 0) == 0) //P值计算判断

{if(m_g0 < 0) ^ (m_yi < 0) != 0) //进给方向判断

m_result = m_result + 2 * m_xi + 1; //偏差计算

m_xi = m_xi + 1; //坐标计算

themotor.Motor.X(); //X轴正向进给

```
    }  
    else  
    {m_result=m_result-2*m_xi+1;  
    m_xi=m_xi-1;  
    themotor.MotorXI();  
    }  
    }  
    else  
    {if(m_g0^(m_xi<0)!=0)  
    {m_result=m_result-2*m_yi+1;  
    m_yi=m_yi-1;  
    themotor.MotorYI();  
    }  
    else  
    {m_result=m_result+2*m_yi+1;  
    m_yi=m_yi+1 ;  
    themotor.MotorY();  
    }  
    }  
    .....  
}
```

4.2.4 动画显示和动画生成

4.2.4.1 常用的动画技术

在计算机图形技术中，目前常用的动画技术主要有以下三种：多页面切换技术、图段变换技术和帧动画技术。

(1) 多页面切换技术

这种技术称为“页面共振”技术。对于某些图形系统，显示模式允许有一个以上的显示页面，其中一页被定义成主显示页，其余页面作为图形显示页。在主显示页显示的同时，下一幅图形可放置在工作页上，然后再把工作页切换成主显示页，如此反复进行，每次用新图形代替旧图形，从

而形成动态变化过程。

(2) 图段变换技术

图段是图形的一部分或者是整体。在光栅系统中, 点阵图段(图符)可以直接从屏幕上获得, 获取的图段作为一个点阵文件或图块存储到磁盘上或缓冲区中, 然后再调用另一图段显示操作, 将它们放置在屏幕的指定位置, 通过变换不同的输出屏幕位置, 就产生了图段图形的连续移动, 也就产生了图段变换动画。由图段动画的定义可以看到, 图段动画和其他动画技术相比有其特定的优点, 图段可以预先定义和预先存储, 并将这些图段(过程或者点阵)作为公共子过程或图符文件存放起来, 在需要的时候调出直接进行绘制和标记, 它可以大大减小计算量, 但增加了存储量。

(3) 帧动画技术

帧动画技术是对显示帧缓冲区直接操作和填写的过程, 根据填写的内容不同而产生的快速屏幕切换, 形成了类似于放电影的一帧一帧的逐帧切换。帧动画的原理是: 屏幕上的图形是存储在显示帧缓冲区中颜色信息的直接映射, 只要将象素的颜色信息直接写入帧缓冲区的指定位置, 在屏幕的对应位置上就能立刻显示出相应的图形和图象。

4.2.4.2 仿真中的动画显示

(1) 双缓冲技术

目前 OpenGL 图形技术标准在计算机图形的研究与应用中处于领先地位。双缓冲技术是 OpenGL 所提供的先进的动画技术, 双缓冲技术是前述多页面切换技术的具体实现。OpenGL 在帧缓冲区存储、操作象素数据, 帧缓冲区由颜色、深度、累加、模板缓冲区等一系列逻辑缓冲区组成, 颜色缓冲区本身也是由一系列逻辑缓冲区组成, 包括前左、前右、后左、后右和一些辅助缓冲区。OpenGL 支持双缓冲图像, 可以使应用程序在脱离屏幕的缓冲区计算和绘图, 当图像可以显示时, 把该缓冲区的内容拷贝到屏幕缓冲区, 结果使图像变化平滑。双缓冲技术应用于数控加工动态仿真对获得刀具切削过程平滑的动画效果具有重要意义。

(2) 局部刷新技术

由于在数控加工仿真中计算和显示的数据量相当大, 在显示过程中如果对工件和刀具所有的几何对象和信息都参与运算和显示, 则在刀具的每

一步切削过程中必然会占用相当长的时间,结果导致图像显示不平滑,降低图像的整体真实感性能。因此,必须采用一种特殊的显示方法来解决这个问题。作者在借鉴目前先进显示技术和方法的基础上,采用局部刷新技术来减小在显示过程中计算和显示对象的范围,从而加快图形计算和显示速度,增强系统显示的动画效果。

局部刷新是图段变换动态技术的具体实现。在数控加工动态仿真的显示过程中,局部刷新技术就是确定刀具在某一运动过程或位置处的工件模型几何信息改变的空间范围,仅对该“切削”范围内的工件和刀具模型进行显示计算,在此范围之外的对象并不参与运算,在屏幕图形的显示中,只有该范围内的图像发生变化,而该范围之外的图像保持原样。随着刀具运动和切削过程的连续进行,刀具切削范围内的图像不断地发生变化,观察者形象地感觉到刀具切削位置处的工件材料不断地被“挖去”,从而从图像效果上达到模拟切削过程的目的。

4.2.4.3 图形驱动与接口设计

在数控仿真系统模块与图形仿真模块的接口中,译码结果将提供各项图形驱动参数及加工状态信息。加工信息多而复杂,为研究问题方便,将刀尖抽象为“切点”,则“切点”相对工艺中心的距离即为工件瞬时半径。在系统中,译码的输出数据结构定义如下:

```
typedef struct daowei
{
    int x, y, z;      //刀具行进位置
    int xingti;      //指令方式
    int number;      //NC 代码段行号
    int tool_dir;    //刀具转向
    int tool_number; //刀具号
}daowei
```

为提取数据方便,数据结构中有一定的冗余信息。代码与仿真的接口由一个数组来实现。仿真时一边读取数据,一边实现动画,每次读取一组数据,这一交换过程用伪码表示如下:

驱动数组(数组记录元素)

```
{  
    if(翻译代码结束)返回;  
    else  
        {读取数组元素, 并仿真;  
          数组记录元素值增加;  
        }  
}
```

4.2.4.4 刀具动画的显示

在仿真过程中还有刀具的动画显示。衡量数控车削仿真系统性能的一个重要指标是加工过程显示的动画生成。因此如何通过动画的形式形象直观地模拟数控加工的切削过程和结果就成为数控加工仿真的关键。必须将刀具的切削运动过程逼真地反映出来,使得用户可以最大限度地从图像显示中获得对整个切削过程的分析 and 检验。刀具显示的最大问题是如何实现在前后数个切削位置处刀具图像的交替显示,而且在图像的交替显示过程中的计算和显示的速度不能影响整个图形的动画效果。刀具运动的动画显示有两种实现方法:异或方法和图像编辑方法。

(1) 异或方法

异或显示是交互式图形技术中的一种重要的绘图模式和技术。在异或绘图模式下,在原有图像位置处重新显示一遍原图像,颜色在写入帧缓存时与缓存中原有颜色值进行异或逻辑运算,该异或操作消除了上一次绘图的结果,并在图形显示效果上恢复上一次绘制之前的结果。异或技术常用于交互式图形系统中的图形牵引技术和橡皮筋技术,为用户提供良好的交互手段。

(2) 图像编辑方法

在刀具的显示中,采用图像编辑的方法能生成具有较强真实感的刀具运动效果。计算机的图形显示信息被保存在帧缓存存储器中,系统将该存储器中的图形数据信息通过光栅扫描方式映射到屏幕上,从而形成图像。应用程序可以通过对帧缓存存储器中的图像信息进行拷贝、复制、移动和缩放等编辑操作,生成所需要的图像,图像编辑方法是图段变换技术和帧动画技术的综合应用。

二维图形仿真采用了异或技术作为刀具显示的实现方法,异或显示是交互式图形技术中的一种重要的绘图模式和技术。异或技术为用户提供良好的交互手段。因为异或技术是数控加工仿真中实现刀具动态显示的一个有效手段,在某一切削位置处,在异或模式下连续绘制两次刀具,后一次的显示擦除了前一次显示的效果,在整个仿真过程中刀具的显示是连续和平滑的,很好地模拟了真实空间中刀具的运动过程,实现也较为方便。因此我们实现了对各种形状的二维图形仿真,达到了几何仿真的要求,实现了对代码的检验。并且通过对最终的图形检测,操作人员可以得到所加工零件在具体某一位置的形状大小。二维仿真可以完成对圆弧、阶梯、倒角等形体的加工演示。

切削过程的动态仿真过程有下面几个方面:(1)取刀具:以刀具号为标识向刀具数据库取刀,提取刀具参数并绘在屏幕上。(2)实现刀具运动:在仿真中,刀具的移动靠计算机动态显示技术来实现,程序设计中,在刀具所经过的每一个轨迹点均要画出刀具图形,这样整个轨迹线上会布满刀具。为了形象地显示刀具移动的样子,既简单又常用的方法有两种:一是“画一擦一画”法,基本原理是刀具每移动一个位置就擦去原来的刀具图形,画出新的刀具图形,这种方法只适于简单的图形,其缺点是会将重叠于图形之上的不动部分图形也擦去。第二种方法是采用“异或”显示模式绘图,即本位置上的图形与上一位置的图形作异或运算后再显示,如此循环就生成动态的图形。这种方法不仅使代码紧凑,而且对静止不动的那部分图形无影响。本程序采用了这种方法,它使得刀具产生运动效果,而工件不运动。(3)实现工件上材料的切除:画出零件的毛坯形状,在屏幕上模仿出车刀一刀刀切下毛坯上材料的样子是切削加工仿真的重要组成部分。用软件实现的基本方法是将被切下的部分重新涂色,当涂成与背景一样的颜色时,给人的视觉就是被刀具切削下来了。具体实现方法是在刀具的刀尖部位定义一个小区域为填色区。这样,在刀具的切削运动中,伴随着切削指令,刀具阴影部分所经过的区域都变成了背景色,实现了切削过程动态仿真。

4.2.5 帮助模块

实用的帮助系统是应用程序开发者必须考虑的一个问题,良好的帮助

系统将给用户带来方便、快捷的操作。数控程序的编制无疑是一项非常复杂的工作，如果能够及时提供在线帮助，对操作人员来说，无疑是一件非常有利的东西。在系统的编制过程中充分考虑到了这一点。为了使用户更加方便的使用本系统，采用了基于 Windows 的帮助系统，它能够支持超链接，上下文敏感帮助，具有索引能力，浏览能力及相应的图形综合能力。系统设计的帮助模块对系统的操作和数控编程的知识做介绍。操作者在遇到困难时，可通过单击帮助按钮或按 F1 键得到及时的帮助。系统操作详细介绍了如何使用本系统，使操作者能够按照系统要求进行仿真；对于那些数控编程基础较弱的人员来说，可以通过帮助系统及时解决在编程方法中遇到的问题，提高他们的学习效率。

5. 车削仿真实例检验

以加工下图所示零件为例检验本仿真系统：

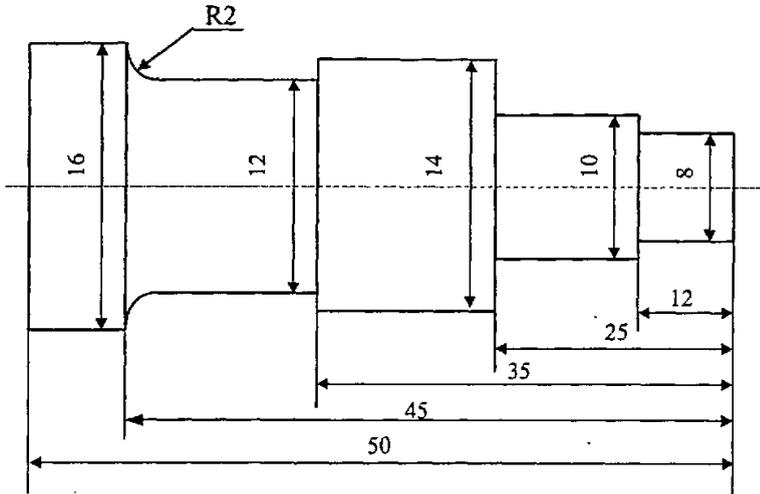


图 5-1 零件图

fig5-1 part drawing

NC 程序如下：

```

n0010 g92 x50 z100      设工件编程坐标原点
n0015 t10                换端面刀
n0020 g00 z0
n0030 g01 x30 f300
n0040 g01 x-2 f30        车端面
n0050 g01 z1 f200
n0060 g00 x20
n0070 g01 z-50 f50
n0080 g01 u1 w1 f100
n0090 g00 z2
n0100 g01 x18 f200

```

n0110 g01 z-50 f50
n0120 g01 u1 w1 f50
n0130 g00 z2
n0140 g01 x17 f200
n0150 g01 z-45 f50
n0160 g01 u1 w1 f50
n0170 g00 z2
n0180 g01 x14 f200
n0190 g01 z-15 f50
n0200 g01 z-15 f50
n0210 g00 z2
n0200 g01 x13 f200
n0230 g01 z-15 f50
n0240 g01 u1 w1
n0250 g00 z2
n0260 g01 x10 f200
n0280 g01 z-5 f50
n0290 g01 u1 w1
n0300 g00 z2
n0310 g01 x7 f200
n0320 g01 z-5 f50
n0330 g01 u1 w1
n0340 g00 z2
n0350 g00 x11.5
n0000 g22 12
n0000 g01 u-2.5 f100
n0360 g01 z-5 f50
n0370 g01 u4 f50
n0380 g01 z-15 f50
n0390 g01 u2

```
n0400 g01 u4 w-8 f50
n0410 g01 u-4 z-40 f50
n0430 g02 u4 w-2 r2 f50
n0440 g01 z-45
n0450 g01 u1
n0460 g01 z-50 f50
n0470 g01 u1 w1
n0480 g00 z2
n0490 g01 u-12 f200
n0500 g80
```

以上程序为粗车外轮廓

```
n0510 g00 x50
n0020 g00 z0
n0000 g01 x6.2 f100
n0360 g01 z-5 f50
n0370 g01 u4 f50
n0380 g01 z-15 f50
n0390 g01 u2
n0400 g01 u4 w-8 f50
n0410 g01 z-40 f50
n0430 g02 u4 w-2 r2 f50
n0440 g01 z-45
n0450 g01 u1
n0460 g01 z-50 f50
n0470 g01 u1 w1
n0480 g00 z2
n0490 g01 x6 f100
n0360 g01 z-5 f50
n0370 g01 u4 f50
n0380 g01 z-15 f50
```

n0390 g01 u2
n0400 g01 u4 w-8 f50
n0410 g01 u-4 z-40 f50
n0430 g02 u4 w-2 r2 f50
n0440 g01 z-45
n0450 g01 u1
n0460 g01 z-50 f50
n0470 g01 u1 w1
n0480 g00 z2
n0490 g00 x50 z100

以上程序为精车外轮廓

n0500 t20
n0510 g00 z-15
n0520 g00 x20
n0530 g01 x8 f20
n0540 g01 x20 f30
n0550 g00 x50
n0560 g00 z100
n0600 g00 z-50
n0610 g00 x30
n0620 g01 x20 f300
n0630 g01 x-1 f15
n0700 g01 u1 w1 f30
n0710 g00 x50
n0720 g00 z100
n0800 m02

换割刀切槽

用割刀切断棒料

说明:

一号刀: 端面刀

二号刀: 割刀

棒料直径: 20mm

在本程序中起刀点、换刀点，及加工完后的终刀点为同一点，这样的程序编制对应零件的批量加工比较合适。

在系统中调用该程序后仿真过程及仿真结果如图 5-2 和 5-3 所示：

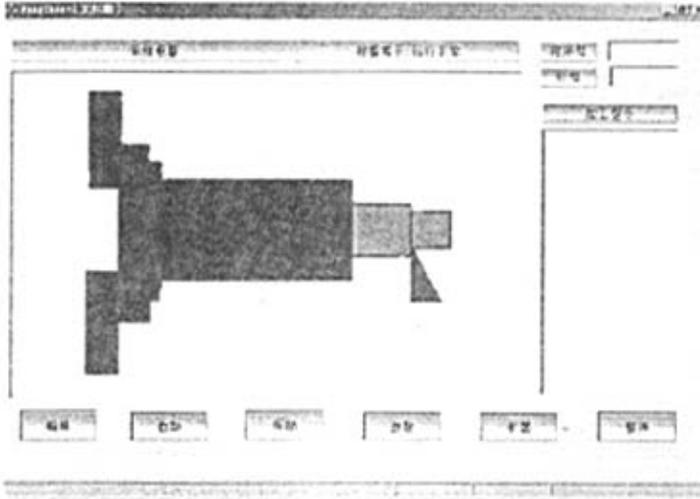


图 5.2 车削仿真过程图

fig5.2 Turning simulation process demo

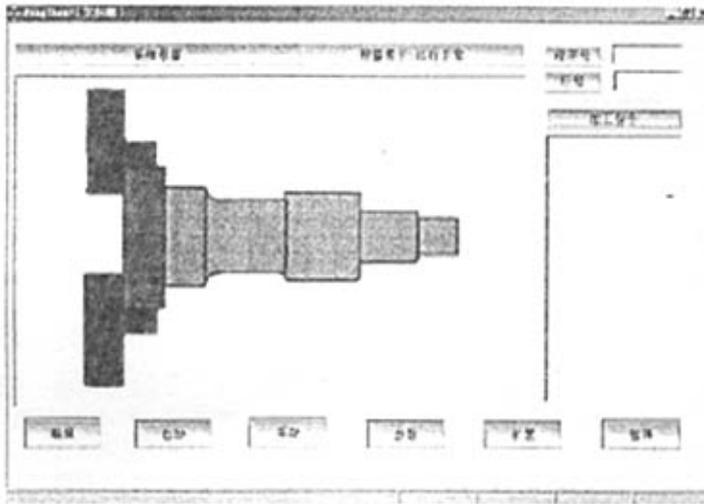


图 5-3 车削实例演示结果

fig5-3 Turning example result demo

上面的实例证明,本系统能够按照设计的车削仿真加工的流程顺利进行,仿真过程中程序运行良好,可以检验数控程序的正确性和合理性。

6. 总结与展望

课题根据面向对象的设计思想与软件工程理论, 开发Superman-200L数控机床仿真系统, 由于时间与精力关系完成了部分工作, 现将完成的内容总结如下:

1) 按照系统开发的总体指导原则, 根据仿真系统的要求选择了开发平台, 并构造了仿真系统的软件体系结构, 后续的程序开发均在此结构框架内进行。

2) 通过参考其他数控仿真系统的界面, 以以人为本的原则, 构建了本系统的人机交互界面, 形成了最基本的仿真加工环境。

3) 按照系统的软件体系结构, 完成了部分功能模块的实现。

该系统是在VC++环境下开发的, 具有以下特点:

1) 实用性

此数控车削加工模拟系统是在进行车削加工研究的同时, 针对Superman-200L数控车床研制的。此加工模拟系统的开发, 缩短了对编制的数控加工程序的试切时间, 节省了试切材料, 提高了机床的效率, 避免了数控车床可能因程序差错而引起的破坏。因而此数控车削加工模拟系统的实用性强, 本模拟系统选不同的后置处理模块即能对不同数控系统的数控车床加工进行加工模拟, 能适应不同的数控车床, 对科研、生产有实用价值。

2) 简便性

该系统的操作, 采用汉化菜单人机交互方式, 操作者只需按微机上汉化提示和询问就能方便地运行本系统, 这样对现场操作人员的要求不高, 便于推广。

3) 可扩充性

基于VC++的数控车削加工图形仿真系统功能的扩充, 由于采用了模块化设计方案, 使得仿真操作功能的实现都直接或间接调用了通用的函数和过程, 有的功能实现过程实际上就是一个独立的模块。因此只要在原仿真系统中添加一些简单的接口函数, 即可使主程序与新的模块相连接, 实现软件功能的扩充。

数控仿真系统的开发设计是一项复杂的工程, 要使系统真正的完善起

来, 还有相当多的工作要做。另外本系统作为一个可扩展、易维护的系统, 它还有许多需进一步开发的地方。通过本文的研究, 作者认为有必要在以下方面进一步开展研究工作:

1) 首先本系统的各部件库需要建立并完善, 这样才能作为真正意义上的仿真系统, 可以模拟数控机床上的任意刀具的切削。

2) 各交互界面虽然基本形成, 但还不尽如人意, 还没有真正考虑到各辅助功能, 例如冷却液的开启、主轴转速的控制等等。

3) 作为仿真系统还应进行刀具与工件的干涉碰撞检查。

4) 仿真系统应该考虑到NC文件出错后的处理, 可以给出出错报告或给出警告, 并能返回到出错处以便进一步编辑改正。

参 考 文 献

- [1]唐荣锡, CAD/CAM 技术, 北京: 北京航空航天大学出版社, 1994。
- [2]申丽国, 张昆, 黄征 国外数控技术的研究动向和发展趋势 机械工业
自动化, 1996, Vol.18, No.2: 19-23。
- [3]张翠华, 李进京, 数控仿真系统技术的研究现状, 山东农机, 2005. 7。
- [4]马玉林, 孙宏伟 基于质量保证的数控切削仿真技术研究展望 制造
业自动化 2000. 4 Vol. 22No. 4。
- [5]李吉平, 张文铭, 黄田 N C 图形验证与仿真技术的研究概况计算机仿
真 2001. 9Vol. 18No. 5。
- [6]OLIVERJH,etal.DirectdimensionalNCverification[J].CAD,1990 (20).
- [7]FUSSELLBK,etal.ComputergeneratedCNCmachiningfeedrates[J].Procee
dings92JapanUSASumposiumonFlexibleAutomation,ASME,1992.
- [8]任仲贵, CAD/CAM 原理[M], 清华大学出版社。
- [9]孙家广, 杨长贵, 计算机图形学[M], 清华大学出版社。
- [10]VOLECKERB,HUNTWA.The role of solid modeling in machine-process mo
deling and NC verification[J].Proceedings of SAE 1981 International Congre
ss and Exposition. Michigan. 1981, 2.
- [11]WANGW,WANGK.Geometric modeling for swept volume of moving solids
[J].compu. Grah. & Applic, 1986, (22).
- [12]DEVORRE, KAPOORSG. Mechanistic model for prediction of the forces syste
m in face milling operations[J]. ASME Journal of Engineering for Indus-try, 19
84, (106).
- [13]ALTINTASY, SPENCEA. End milling force algorithm for CAD system[J]. An
nalsof CIRP, 1991, (40).
- [14]STREMKOWSKIJS, CARROLLJT. An orthogonal metal cutting model base
d on an eulerian finite element method, manufacturing processes, machines and
systems[J]. Proceedings of 13 NSF Conference on Production Research and Tec
hnology, 1986.
- [15]ZHANGB, BAGCHIA. Finite elements simulation of chip formation and compa
rison with machining experiment[J]. ASME Journal of Engineering for Indust

- ry,1994,(116).
- [16]Donald R.Stovicek. NC Verification-Machining it right the first time. Tooling&Production. 1992,(3):31-36
- [17]W.P.Wang, K.K.Wang. Real time Verification of Multi-axis NC Programs with Rhaster Graphic.Proc. IEEE International Conference on Robotics&Automation 1986,Vol1:167-171
- [18]LiuHuaming,LiJipingandRenBingyin.DiscreteRepresentationofSculpturedSurfaceforDimensionalNCVerification.InternationalConferenceonAdvancedManufacturingTechnology.Xi'an,P.R.China,1999.NewYork,SciencePress:240-244.
- [19]李吉平,刘华明,张文铭,黄田,任秉银.复杂曲面加工精度检验中曲面法矢与刀具扫描体求交算法的研究,计算机辅助设计与图形学学报,2000,12(12):931-935。
- [20]黄雪梅,高国利,王启义.虚拟制造的机械加工过程仿真.组合机床与自动化加工技术。
- [21]许立,数控车削加工仿真系统的研究与实现。
- [22]张翠华,李进京,Superman-200L数控机床仿真系统的研究,农业装备技术,Vol.32, No.2, 2006.4。
- [23]宋晓梅,赵婉,数控机床仿真系统的设计与实现,青岛化工学院学报,Vol.18, 1997.3。
- [24]李峰 基于VC的数控仿真系统的开发 湖北工学院学报 第19卷第4期 2004.8。
- [25]曹文杰,徐燕申,利用VC++编程实现数控车削仿真,机械,Vol.29, 2002.1。
- [26]田军,基于面向对象的CAD系统开发,电工技术与自动化,2001(4)。
- [27]谭浩强主编,面向对象的设计思想,清华大学出版社,1998。
- [28]万建成,李杰主编,C++的面向对象设计原理与方法,山东工业大学计算机系,1999。
- [29]蔡希尧,陈平,面向对象技术,西安电子科技大学出版社,1995。
- [30]载同,CAD/CAPP/CAM基本教程,机械工业出版社,1999。

- [31]李志刚, 于炜, 深入 C++编程, 北京航空航天大学出版社, 1994。
- [32]陈亮, 面向对象技术在机械 CAD 中的应用出探, 机械研究与应用, 1999, 12 (1)。
- [33]Edward Yourdon & Argila, OBJECT ORIENTED ANALYSIS & DESIGN PRENTICE HALL.
- [34]avid J, Kruglinski, Scot Wingo, Programming Visual C++6.0 北京希望电子出版社出版, 1999, 12。
- [35]B, R, Lawn, M, V, Swain, Microfracture beneath point indentation in brittle solids, J, of Materials J, of Materials, 1975(10).
- [36] M, V, Swain, B, R, Lawn, Indentation fracture in brittle rocks and glasses, Int, J, Rock, Mech., Min, Sci, &Geomech., Abstr, 1976, 311.
- [37]孙家广, 杨长贵, 计算机图形学, 北京, 清华大学出版社, 1995, 5
- [38]R, Schulze, Trenchleif von Harttgestein, Doctor Dissertation, TU Hanover, 1980.
- [39]Jon Bates, Tim Tompkins, 何健辉, 董方鹏译, 实用 Visual C++6.0 教程, 清华大学出版社, 2000. 8。
- [40]马安鹏, Visual C++6 程序设计导学, 清华大学出版社, 2002. 4。
- [41]龙腾科技, Visual C++6.0 循序渐进教程, 红旗出版社, 2005. 2。
- [42]罗学科, 谢富春, 数控原理与数控机床, 化学工业出版社, 2004, 1。
- [43]HideoMatsuka. Japanese PC-based Open Control Systems for Manufacturing Equipment Int.J.PapanSoc.Prec.Eng, 1996(3):204-209。
- [44]M. W.Bailay, etal Sawing in the stone and civil engineering industries, IDR. 1979. 2, 56-60.
- [45]杨箭, 通用型数控系统设计, 机电一体化, 1997(2):25-26。
- [46]李志刚, 于炜. 深入 C++编程. 北京航空航天大学出版社, 1994:12-14。
- [47]毕承恩, 丁乃件, 现代数控机床. 机械工业出版社, 1991。
- [48]李诚人. 机床计算机数控. 西北工业大学出版社, 1988。
- [49]李峻勤, 费仁元. 数控机床及其使用与维修国防工业出版社, 2000。
- [50]郭培全, 王红岩, 数控机床编程与应用, 机械工业出版社, 2000。

- [51]方沂, 数控机床编程与操作, 国防工业出版社, 1999。
- [52]张国彬, 林亨, 车削中心数控加工仿真系统程序检查模块的开发, 机械设计与制造, 2001.6: 13-15。
- [53]郭培全, 王红岩, 数控机床编程与应用, 机械工业出版社, 2000。
- [54]BEIJING-FANUC series 0-TD 操作说明书卜北京法那科机电有限公司, 1997。
- [55]王先逵, 段广洪, 吴丹, 计算机辅助制造, 清华大学精仪系, 1997。
- [56]冯辛安, 葛巧秦主编, CAD/CAM 技术概论, 北京: 机械工业出版社, 1995。
- [57]范炳炎, 数控加工程序编制, 北京: 航空工业出版社, 1990。
- [58][美]Michael.J.Young 著, 邱仲潘等译, Visual C++6 从入门到精通。北京: 电子工业出版社, 1999。
- [59]殷人昆等, 数据结构(用面向对象方法与 C++描述), 北京: 清华大学出版社, 1999。
- [60]黄新燕, 蔡理, 张雯, 数控程序编辑及语法检查系统的开发, 机床与液压, 2002(6)。
- [61]孙丽梅, 开放式数控系统的软 CNC 的研究开发[D], 北京: 北京工业大学机械系, 2003。
- [62]邵俊鹏, 董玉红。机床数控技术, 哈尔滨工业大学出版社, 1996。
- [63]刘文信, 孙学礼。机床数控技术。机械工业出版社, 1994。
- [64]林奕鸿, 机床数控技术及应用, 机械工业出版社, 1994。
- [65]任玉田, 焦振学, 王宏普, 机床计算技术控技术, 北京理工大学出版社, 1996。
- [66]王永章, 杜君文, 程国全, 数控技术, 高等教育出版社, 2001.12。
- [67]吴晓苏, 逐点比较圆弧轨迹插补在过象限时用符号判别法的分析, 机电工程, 2002(5)。
- [68]染芝芸, 孙雷, 王亮德等, 一种新型的插补方法—自动过象限卡诺图最小偏差法, 测控技术, 2000(9)。
- [69] G.S.Catania. A computer-aided prototype system for NC rough milling of free-form objects. Computer Industry. 1993, 20(3):275-283.

- [70]高国利,王启义,高真感虚拟数控车削加工环境及实用系统。1999年度技术工作总结。
- [71]王洪成,NC加工三维图形动态仿真系统的研究与开发,东北大学硕士学位论文。
- [72]宋显健等,虚拟制造单元仿真技术与系统。计算机辅助设计与制造1998,第9期:41-46。
- [73]张旭强,基于面向对象的石材异型制品加工设备数控系统开发,山东大学硕士学位论文,2002。
- [74]肖成志,面向石材异型制品加工设备的开放式数控系统研究与开发,山东大学硕士学位论文,2004。
- [75]许文达,石材异型制品加工数控系统中译码器与人机界面的开发,山东大学硕士学位论文,2004。
- [76]原彬,数控车削加工仿真系统的研究与实现,大连铁道学院硕士学位论文,2001。
- [77]钟军,数控车削加工过程几何仿真,大连铁道学院硕士学位论文,2000。
- [78]罗学科,王东宇,徐兴建,数控代码的计算机图形仿真,机械设计与制造,2000.6, No3。
- [79]王润孝,机床数控原理与系统,西北工业大学出版社,1993。
- [80]彭炎午,计算机数控(CNC)系统,西北工业大学出版社,1988。
- [81]FridshalR, ChengKP, DuncanD, ZuckerW, numerical Control Part Program Verification System, Proc. Conf. CAD/CAM technology in mechanical engineering MIT, Cambridge, MA, USA, 1982。
- [82]赵继政 多轴数控加工的动态图形仿真系统的研究与开发 西北工业大学硕士学位论文,1995(本文1998年4月6日收到)(编辑张学)
- [83]胡德计,基于面向对象的CAD/CAPP/CAM/CNC集成系统建模及其实现[D],天津大学博士论文,1997,7。
- [84]葛研军,虚拟数控车削加工系统研究[D],东北大学博士论文,1998,12。
- [85]Daoh-Tiam zhang and Jen-GuoChen.An NC Lathe simulation for part programming and machine operation training,Computer in

industry,1993,21:139-148.

[86]Donald Hearn,M.Pauline Baber, 计算机图形学, PRENTICEHALL, 1998. 4.

[87]熊光悛, 计算机仿真及其在制造业中的应用[J], 计算机仿真, 1996, (1)。

[88]高华, 基于 VB 的数控车削几何仿真系统研究与开发, 山东农业大学硕士学位论文。

附 录

SuperMan 车床数控系统功能代码表

表 1 G 代码表

代 码	功 能	说 明	模 态
G00	快速点定位	F 设定范围: 2000~ 6000mm/min(Z)	模态
G01	直线插补	F 设定范围: 6~2000mm/min(Z)	
G02	顺时针方向圆 弧插补	自动过象限 F 设定范围: 6~600mm/min(Z)	模态
G03	逆时针方向圆 弧插补	自动过象限	模态
G04	程序延时	延时范围: 0.00~99.99 秒	非模态
G22	程序循环		非模态
G23	矩形循环		模态
G26	X、Z 轴同时回换 刀点		非模态
G27	X 轴返回换刀点		非模态
G29	Z 轴返回换刀点		非模态
G32	英制螺纹车削	导程范围 3~33.5 牙/英寸	模态
G33	公制螺纹切削	导程范围 0.25~12mm	模态
G36	X、Z 向回机械原 点		非模态
G37	X 向回机械原点		非模态
G39	Z 向回机械原点		非模态
G80	循环注销		非模态
G92	建立工件坐标 系		非模态

表 2 辅助功能 M 代码表

代 码	功 能	代 码	功 能
M00	程序暂停	M07	冷却开
M02	程序结束	M08	冷却关
M03	主轴正转	M97	程序跳转
M04	主轴反转	M98	子程序调用
M05	主轴停止	M99	子程序返回

主轴变速功能 (S 功能)

SuperMan 车床数控系统具有对配置有变频器的可变主轴电机实现 S 指令调速的功能。

代码形式: S××××, S 后面所带数值为 0~9999 的整数, 如果该数值超出系统参数设定里的最大值, 主轴实际转速将为系统设定中的最大值。

例如, N0010 S6000, 在系统中设定的主轴最大值为 3000 转/分, 执行该指令后主轴的事件转速为 3000 转/分。S 指令后的数值不可以为负值。

S 指令发出后所对应的转速将不会改变, 只有 S 指令重新发出或是改变了主轴传动比。

刀具功能 (T 功能)

- T1: 换第 I 把刀 (T01)
- T2: 换第 II 把刀 (T02)
- T3: 换第 III 把刀 (T03)
- T4: 换第 IV 把刀 (T04)
- T10: 换第 I 把刀, 并进行刀具补偿
- T20: 换第 II 把刀, 并进行刀具补偿
- T30: 换第 III 把刀, 并进行刀具补偿
- T40: 换第 IV 把刀, 并进行刀具补偿

致 谢

论文是在导师李进京教授的谆谆教诲指导下完成的。从论文的选题、课题的开展研究到论文撰写整个过程，无不倾注了李老师大量的心血。李老师开拓进取的工作作风、严谨求实的治学态度、循循善诱的教导方式和渊博的知识使我的课题得以顺利完成，同时开朗豁达的生活态度和孜孜不倦的探索精神也深深感染着我。在此成文之际，谨向辛勤培育我的导师致以崇高的敬意和衷心的感谢。

在论文选题和研究方案制定过程中，课题组的刘贤喜老师、赵立新老师都给了我很多的指导和帮助，并提出了宝贵建议；在数控机床学习方面得到了华磊老师、林成厚老师、张军老师的大力帮助。在此一并向他们表示最诚挚的感谢。

最后，谨向亲切关怀我的院领导、所有关心、帮助过我的师长、同学和朋友表示感谢！感谢他们一直以来对我的支持、鼓励和关心！

攻读学位期间发表论文

- [1] 数控仿真系统技术的研究现状. 山东农机, 2005 (7): 6~8。
- [2] SuperMan-200L 数控机床仿真系统的研究. 农业装备技术, 2006 (2): 29~31。