

文章编号:1005-0523(2006)04-0115-03

数控加工中心虚拟系统的设计

熊世文, 郭烈恩, 严建华

(南昌大学 机电工程学院, 江西 南昌 330029)

摘要:介绍了数控加工中心虚拟系统特点, 功能要求, 构架了系统的框架, 对系统的各个模块的实现进行了详细的阐述. 采用了三维造型软件实现了机床的几何模型, 使用实体造型构建了工件模型, 开发了基于3D平台的刀具轨迹模型. 对系统的编译系统做了简略的介绍.

关键词:虚拟制造; 计算机仿真; 数控技术

中图分类号:TH12

文献标识码:A

0 引言

数控加工中心虚拟系统的目的是构建一个具有真实感的机床加工仿真系统, 属于虚拟制造范畴. 虚拟制造是通过计算机来模拟制造并预估产品功能、性能及可加工性等各方面可能存在的问题的一种验证方法. 它提高人们的预测和决策水平, 使得制造技术走出主要依赖于经验的狭小天地, 发展到了全方位预报的新阶段^[1].

1 系统的特点及功能要求

虚拟加工环境的目标是建立一个真实的数控加工环境. 在这个环境中, 需要建立机床模型和加工过程模型^[2]. 仿真系统应具备的功能:

- 1) 编辑数控文件.
- 2) 对数控程序编译, 插补, 生成机床驱动文件.
- 3) 具有真实感的机床几何模型.
- 4) 能模拟机床运动及加工过程.
- 5) 对机床状态进行反馈, 实时监控机床的运动状态.

6) 具有真实感的人机操作界面, 能对机床参数及状态进行控制.

2 系统整体构架

仿真过程为: 在控制面板上调入 NC 程序或就地编辑 NC 程序, 然后对准备好的加工程序进行检查, 轨迹仿真, 确认无误, 准备加工. 加工前对整个系统进行必要的设置, 刀具信息, 工件坐标系等. 进行加工, 显示机床运动动画及工件切削动画, 对机床状态进行监测, 显示监测的信息, 如果有非法操作, 越程等信息, 发出相应的警报, 自身作出必要的措施.

从上面的过程^[3]来看, 整个系统本质上是由命令字符串驱动一个虚拟机床的动画运动的过程. 系统显然有两大块: 一个是 NC 处理, 一个是模拟机床运动.

NC 系统主要工作有数控程序编辑, 刀补, 插补, 编译生成虚拟机床驱动文件等.

模拟机床运动则需要对真实机床零部件进行造型, 同时提供必要的运动接口, 刀具运动接口, 机床运动接口等用于 NC 系统驱动加工. 除此两大主

收稿日期: 2005-10-08

作者简介: 熊世文(1982-), 男, 江西南昌人, 硕士研究生, 研究方向为电火花特种加工.

功能之外,还需要实现人机界面,机床参数设置及机床状态信息反馈及监测等功能。

3 各个模块的设计:

1) 几何模型

(1) 机床本体模型, 刀具模型, 换刀架, 切削液喷管等复杂几何模型:

这些模型比较复杂, 直接采用汇图编程的方法很难实现这么复杂的图形, 即使实现了也需要花费极大的时间和精力, 绘制出来的效果也很难以达到预期效果. 因而采用一些成熟绘图软件如 3dsMax, UG, ProE 等来实现这些几何模型. 遗憾的是并不能直接调用这些软件生成几何模型, 只能得到这些几何模型的描述性文件. 从而, 不得不对这些文件进行研究, 找出需要的几何信息, 然后转换成程序中能够使用的几何实体. 幸运的是, 可以找到一种标准的文件格式, 3DS 文件格式, 几乎所有的 3D 绘图软件都支持这种文件格式, 能转化成这种文件格式. 那么该模块的工作便是编写一个文件接口, 将 3DS 二进制文件读入转换成 OpenGL 几何实体^[4].

(2) 刀具轨迹及零件几何模型:

这个模块用于刀具轨迹仿真, 验证 NC 程序是否正确及显示加工后零件几何模型, 如点, 线, 圆弧, 平面, 直纹面等几何元素的绘制. 点, 向量, 矩阵的各种运算. 在 Windows 的 GDI 编程中可以得到设备环境 CDC, CDC 可以绘制直线, 圆弧, 矩形等. 在 OpenGL 环境中, 可以相似地构造出一个设备环境类, 让它绘制出一些基本的几何元素, 直线, 圆弧等. 这个层里面设计的类如下^[5]:

```
class Point;//点模型
class CVector3D;//向量模型
class glCDC;//设备环境类, 即用于直线, 圆弧等元素的绘制
```

(3) 工件模型:

这个模型用于工件切削运动. 模拟加工过程其实就是模拟工件切除过程, 因而需要进行实体的布尔运算, 工件就不能是一个整体的实块而是又无数过细小的离散体组成, 那么加工过程就是这些小快体是显示还是不显示的一个布尔运算过程. 因此, 只将工件在 X, Y 平面上进行分割, Z 方向用 top 值来表示. 构建的模型的如下:

```
class PexSel//离散的小方块实体模型
整个工件可表示为:PexSel Box[x][y]; //x,y 为
```

工件分辨率

如果是斜面型的工件, 操作如下:

对离散的 PexSel 的参数如 top 进行设置, 使其形状成为一个斜面

```
for(int j=0;j<x;j++)
{
for(int i=0;i<y;i++)
{
box[j][i].x=box[j][i].x+i*box[j][i].lenth;
box[j][i].y=box[j][i].y+j*box[j][i].lenth;
box[j][i].top=box[j][i].top+i*1;
}
}
```

绘制斜面工件

```
for(j=0;j<x;j++)
{
for(int i=0;i<y;i++)
{
box[j][i].Draw();
}
}
```

2) 运动模型

运动模型有机床本体运动, 刀具运动, 加工切削运动, 其他运动(如换刀等)等, 这其实属于动画制作过程. 可以让一张张相关的图片以较快的速度进行切换, 就能得到连续的运动效果. 相似地, 在一定时间里绘制 N 张相关的图片, 就能得到计算机动画效果. 先设置一个系统时间, 让它不停的刷新画面, 接下来的工作就是绘制这些相关的图片了.

图形的绘制, 把它封装成按参数化形式绘制, 只要将其参数进行修改就可以实现动画控制.

比如一个正方体绘制可写成:

```
glTranslatef(m__x,m__y,m__z);
DrawBox(length,width,height);
```

那么只要控制 m__x, m__y, m__z 三个变量进行控制, 然后让画图模块不停地按参数绘制即可实现动画.

机床本体运动:

机床的运动有最基本的 X, Y, Z 三方向运动, 刀具旋转运动同时要控制转速和进给速度, 用如下变量来控制:

```
double m__x; ////////////////
double m__y; //位置控制
```

```
double m __z; //变量
double m __angle; ///////////////
double S; //转速
double F; //进给速度
```

有了上面几个变量显然是不够的,还需要在这上面做进一步的工作.指令机床不是一步一步怎么走而是让他走直线,圆弧甚至是高级曲线.必须构建一些控制器,让它们完成指令机床走这些曲线的细节步骤.

构建如下位置控制计算器类,直线,圆弧等:

```
class FunCircle //实现圆弧运动计算器
class FunLine //实现直线运动计算器
...
```

3) 切削运动:

切削运动是一个布尔减运算过程,建立了工件几何模型,对工件模型参数进行设置即可实现切削动画效果.

```
class LineBool //刀具沿直线运动布尔运算
class AreBool //刀具沿圆弧运动布尔运算
...
```

4) 编译模块:

编译模块主要划分为四个部分:词法分析、语法分析、目标代码生成和出错处理.编译过程是输入数控加工程序(G代码),输出目标代码或错误信息.本系统采用逐行扫描方式,以词法分析程序和语法分析程序为核心,出错处理作为一个独立的过程,目标代码的生成则在错误为零的情况下生成.

设计一个编译类 Compile.

输入:CString m __NcCode; //一段 NC 代码

功能函数:

Wordcheck() //词法检查

SyntaxCheck() //语法检查

输出:CString errInfo //错误信息

操作数据对象:

ProgramNode NcSegmentStruct; //编译后生成的中间文件.

CTypedPtrList < COBList, CPart * > * m __curveList; //编译后生成的刀具轨迹链表

5) 机床信息反馈:

(1) 机床参数:

设计一个后台数据库 CDaoDatabase m __db; //机床数据库

后台数据库使用微软公司的 ACCESS 制作.

(2) 状态监测,设计一个类:

```
class RunErrCheck;
```

实现功能包括非法报警,工件与刀具干涉,非法操作,越程等.

(3) 人机交互:

几乎所用的操作,控制都在这里,那么几乎所有的模块都在这里汇集,可以是指针,实体,用来实现整个机床及加工过程的控制.

设计一个 NcPanel 类,这个类提供以下控制变量,用于机床参数设置,机床运动控制等等:

```
Machine * machineControl; //指向机床本体指针,用来控制机床运动
```

```
ProgramNode NcSegmentStruct; //编译后生成的中间文件,既用来驱动机床 NC 代码段.
```

```
CTypedPtrList < COBList, CPart * > * m __curveList; //刀具轨迹链表
```

```
CDaoDatabase m __db;
////////////////////
```

```
CDaoTableDef * m __table; //机床数据库(坐标系,刀长,刀径等数据)
```

```
CDaoRecordset * m __record;
////////////////////
```

```
CString m __posInformation; //刀具位置信息
```

还有其他一些控制变量如机床状态,运行模式等:

```
PANEL m __panelState; //机床状态,处于编辑模式等
```

```
WorkMode runMode; //运行模式,循环,单段执行等
```

```
...
```

4 小结

本系统实现了虚拟机床的基本功能,机床几何模型,刀具模型,刀具轨迹模型取得了非常好的效果.虚拟环境非常真实.然而不足的是,工件模型采用了实体造型模型,对于复杂的零件显的力不从心,分辨率取得太低,零件不逼真,如果分辨率取得太高,计算量太大,动画太慢,甚至会死机,需要进行进一步研究.

参考文献:

[1] 肖田元.虚拟制造研究进展与展望[J].系统仿真学报, 2004, 9.

[2] Y Yamaguchi&F. Kimura. A constrain Modeling System for

