

基于面向对象的虚拟数控系统的研究

李仲阳 田娟秀

(湖南师范大学工学院, 长沙 410081)

摘 要 提出了虚拟数控机床的基本系统结构, 引入面向对象程序设计的基本概念于虚拟数控机床的设计。根据数控机床各零部件在加工过程中的特点, 把它们分成静态类、运动类和动态类 3 类。在此基础上, 利用 3 维实体 CAD (computer aided design) 软件系统进行虚拟数控机床加工过程中非变形物体 (静态类与运动类) 的外形设计并转换成 STL (stereolithography) 文件, 对动态变形物体采用特征造型方法, 与设计的成员函数即该零部件的运动函数有机地集成起来, 构成类, 最终实现数控机床各机构的动态仿真。引入了四元数的基本概念, 并将其应用于系统设计与实现。上述方法为虚拟数控系统的研究提供了基本框架。实验结果证明, 上述方法是可行的, 具有很好的实用价值和参考价值。

关键词 虚拟现实 数控机床 面向对象 图形句柄

中图法分类号: TP391.9 文献标识码: A 文章编号: 1006-8961(2006)02-0275-06

Study on System of Virtual NC Machine Tool Based on OOD

LI Zhong-yang, TIAN Juan-xiu

(Polytechnic College of Hunan Normal University, Changsha 410081)

Abstract In this paper, the basic system structure of the virtual NC (numerical control) machine tool is put forward and the basic conception of OOD (object oriented design) is introduced to the virtual system design of NC machine tool. Based on their respective characters, all parts of NC machine tool are classified into three classes: CStaticClass, CMoveClass and CDynamicClass. The STL files which containing the appearance feature of the undeformed objects are imported from solid modeling software (Pro/E) and feature modeling approach are adopted to describe the deformable objects, then combine with the respective animate functions to compose the classes, and finally accomplish the 3D dynamic simulation. The basic theory of quaternion is introduced and applied to the design of the system and the rotation of virtual NC machine tool. The methods provide a basic frame for studying on virtual NC system. The experimentations prove that the methods are feasible and have good practical value.

Keywords virtual reality, CNC (computerized numerical control) machine tool, object oriented, graphics handle

1 引 言

由于数控机床高效率、高质量和高柔性的特点, 使得它在制造业中迅速普及, 如何保证数控程序的正确性成为数控加工中的突出的问题。因此, 基于虚拟现实的数控系统的研究引起人们的广泛兴趣, 本文研究了虚拟数控机床的一般结构体系, 通过 3

维 CAD 软件进行数控机床实体设计, 在此基础上引入类的概念, 对虚拟数控机床的各零部件进行分类, 提出了基于面向对象的虚拟数控机床的设计思想, 将零部件的静态属性和动态属性有机集成, 通过对数控加工程序的译码, 同时引入图形句柄的概念, 方便地实现了图形的操作, 最终设计了一台可交互操作的虚拟数控机床。

基金项目: 湖南省自然科学基金项目 (02jjy3045)

收稿日期: 2004-07-23; 改回日期: 2005-04-26

第一作者简介: 李仲阳 (1963 -), 男, 教授。2000 年于华南理工大学获机械制造与自动化专业博士学位。主要研究方向为 CAD 与快速成型。已发表论文 20 余篇。E-mail: zhongyangli@163.net

2 虚拟数控机床的基本结构体系

虚拟数控系统是 VR(虚拟现实)在数控加工过程中的具体应用,其核心由计算机、配套硬件及软件系统组成。软件系统主要用来建立虚拟加工环境和加工过程仿真。

(1)虚拟加工环境由机床、工件、刀具和夹具等构成。根据对象化设计思想,可以把加工环境中的相关零部件看作是一个个对象。根据物体的特性不同对物体进行分类,构造类的形体属性、材质、类型、运动属性及实现运动的方法,最终实现虚拟加工环境。虚拟加工环境中还包括人机交互接口。系统通过人机交互实现不同的底层功能调用。

(2)虚拟加工过程包括数控程序译码、3 维动画仿真、碰撞干涉检查等功能模块。数控程序译码是用来识别用户编写的 NC(numerical control)文件,提取加工参数,把它转化成系统可识别的格式;3 维动画模拟数控加工过程,根据加工参数产生刀具位置数据,并以此数据控制机床运动部件和刀架运动。因此,虚拟数控机床基本结构可以用图 1 表示,其中,3 维建模模块用于生成虚拟环境中的对象;连接/动画编辑器用于将这些对象与其在虚拟环境中的运动联系起来;仿真管理器用于形成一个交互的运动系统;而用户应用系统面向具体问题,描述仿真的具体内容。

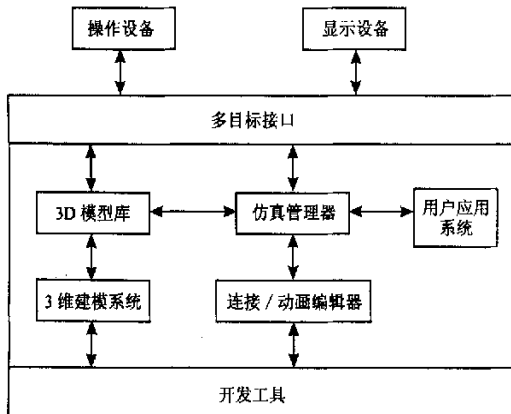


图 1 虚拟数控机床基本结构

Fig. 1 The basic structure of virtual NC machine tool

系统采用 3 维图形软件标准接口 OpenGL 进行 3 维环境的虚拟。OpenGL 的理论基础是矩阵运算,在 3 维中的动画是通过图形与其相应的运动矩阵(平移、缩放、旋转等)矩阵相乘后,在后台缓冲区形成一帧,然后

与前台显示缓冲区互换,在屏幕上重画实现的。为了开发一个具有交互性好、用户接口友好的虚拟加工系统,还必须借助可以处理事件和提供窗口操作的基本编程语言如 Visual C++, C++ Builder^[1]。

3 四元数与形体的 3 维几何变换

把虚拟车床加工环境中的各个组成部分视为图形对象,并以句柄作为图形对象的识别符。为此,必须找到一个可行有效的方法使得在面向对象的图形对象中可运用图形引擎进行运动控制。

对于绕任一轴旋转的 3 维变换,通常是采用平移与坐标轴旋转进行复合的方法实现,具体来说将其分解为平移、绕 x 轴旋转、绕 y 轴旋转与绕 z 轴旋转 4 种基本变换,即首先必须得到由选定旋转轴到另一个坐标轴的变换序列,然后根据指定旋转角求出绕该轴的旋转矩阵,最后得到将旋转轴变回到原来位置的逆矩阵序列。绕任意轴的旋转可以表示为

$$R(\theta) = T^{-1} \cdot R_x^{-1}(\alpha) \cdot R_y^{-1}(\beta) \cdot R_z(\theta) \cdot R_y(\beta) \cdot R_x(\alpha) \cdot T \quad (1)$$

其中, T 为平移矩阵, $R_x(\alpha)$ 为绕 x 轴旋转 α 的旋转矩阵, $R_y(\beta)$ 为绕 y 轴旋转 β 的旋转矩阵, $R_z(\theta)$ 为绕 z 轴旋转 θ 的旋转矩阵^[2]。

获得对给定轴旋转的更有效的方法是运用旋转变换的四元数表示。从物理内涵上看,四元数可理解为“绕某个矢量转动一定角度的物理量”,用四元数描述转动问题要比欧拉角法更为直观,在文献[3]~文献[5]中论证了四元数与旋转之间的关系,四元数方程是线性的,避免了欧拉角法固有的三角运算。通常四元数法的解算精度和速度比欧拉角法要好,并且避免了欧拉角法在大姿态角下可能出现的奇异现象。另外,四元数所需存储空间少于 4×4 矩阵^[2,4]。

一个四元数是由一个实部和 3 个虚部组成的高维复数,写为

$$q = s + xi + yj + zk \quad (2)$$

其中, s 为标量部, x 、 y 和 z 是实数,为虚数项系数,参数 i 、 j 、 k 有以下特性:

$$i^2 = j^2 = k^2 = -1, ij = -ji = k$$

刻画一个四元数的另一方法是把它看作一有序对,由一个标量部和一个向量部组成:

$$q = (s, v) \quad (3)$$

其中, $v = (x, y, z)$

定义四元数的加法运算和乘法运算如下:

$$q_1 + q_2 = (s_1 + s_2, v_1 + v_2) \quad (4)$$

$$\begin{aligned} q_1 q_2 &= (s_1 s_2 - v_1 \cdot v_2, s_1 v_2 + s_2 v_1 + v_1 \times v_2) \\ &= (s_1 s_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) + \\ &\quad (s_1 x_2 + x_1 s_2 + y_1 z_2 - z_1 y_2) i + \\ &\quad (s_1 y_2 - x_1 z_2 + y_1 s_2 + z_1 x_2) j + \\ &\quad (s_1 z_2 + x_1 y_2 + y_1 x_2 - z_1 s_2) k \end{aligned} \quad (5)$$

四元数的模定义如下:

$$\|q\| = \text{Norm}(q) = \sqrt{s^2 + x^2 + y^2 + z^2} \quad (6)$$

若一个四元数的模为1,则称该四元数为单元四元数,只有单元四元数才可以表示旋转,因此必须把四元数化为单元四元数^[6]。

$$q = \frac{q}{\|q\|} = \frac{q}{\sqrt{s^2 + x^2 + y^2 + z^2}} \quad (7)$$

绕过原点的任一轴旋转首先由建立有下列标量和向量部的四元数来完成:

$$s = \cos\left(\frac{\theta}{2}\right) \quad v = u \sin\left(\frac{\theta}{2}\right) \quad (8)$$

其中, u 是沿所选旋转轴的单位向量, θ 是绕此轴的指定旋转角。任一点 P 绕此四元数旋转可以用四元数符号表示为

$$P = (0, p) \quad (9)$$

这里,点的坐标为向量部 $p = (x, y, z)$ 。然后点的旋转由四元数运算来实现:

$$p' = qpq^{-1} \quad (10)$$

其中, $q^{-1} = \frac{1}{\|q\|^2}(s, -v)$ 是式(8)中所给的标量和向量部组成的单位四元数 q 的逆。这一变换产生了标量部为0的新四元数:

$$P' = (0, p') \quad (11)$$

且向量部用点乘和叉乘表示为

$$p' = s^2 p + v(p \cdot v) + 2s(v \times p) + v \times (v \times p) \quad (12)$$

在OpenGL环境中,是通过使用矩阵来完成各种几何变换的,为了运用四元数来控制图形,需把四元数转换为矩阵,计算式(1)中的项 $R_s^{-1}(\alpha) \cdot R_y^{-1}(\beta) \cdot R_z(\theta) \cdot R_y(\beta) \cdot R_z(\alpha)$ 来获得复合旋转矩阵,可得一 3×3 的矩阵,但在OpenGL中均采用 4×4 的矩阵,将其扩展可得:

$$M_R(\theta) = \begin{bmatrix} 1-2y^2-2z^2 & 2xy-2sz & 2xz+2sy & 0.0 \\ 2xy+2sz & 1-2x^2-2z^2 & 2yz-2sx & 0.0 \\ 2xz-2sy & 2yz+2sx & 1-2x^2-2y^2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (13)$$

绕任意轴的旋转可以表示为

$$R(\theta) = T^{-1} \cdot M_R(\theta) \cdot T \quad (14)$$

由此可见,四元数将连续的旋转变换问题转化为两个四元数的相乘,避免了前后两矩阵相乘而带来的开销,避免了欧拉角法固有的三角运算,具有存储空间小、运算速度快和精度好的优点。基于以上知识,为了借助四元数的概念来清晰地描述物体旋转运动,可以把四元数的特性及其运算封装在一个类里,描述如下:

```
class CQuaternion
{
public:
    CQuaternion();
    virtual ~CQuaternion();
    //通过旋转轴向量及旋转角度建立四元数
    void CreateFromAxisAngle(GLfloat x, GLfloat y, GLfloat z,
        GLfloat angle);
    //将四元数转化为4×4矩阵,以便将其应用于OpenGL
    环境中
    void CreateMatrix(GLfloat *pMatrix);
    CQuaternion operator *(CQuaternion q);
private:
    //存放四元数的向量部系数和标量
    GLfloat x, y, z, s;
}
```

4 虚拟加工系统中物体的分类及类描述

面向对象的程序设计方法是虚拟加工系统中处理各图形对象的有力工具,它根据系统的对象及其相互关系来构造仿真模型,模型的对象表示实际系统中相应的实体,具有直观性且便于理解^[7]。直接引入面向对象程序设计中的类、属性与方法的概念于虚拟加工系统中。在虚拟加工的制造系统中,刻画系统中各部分的属性主要有几何形状、颜色和材质等。而类的方法是类的动态特征,是这一类事物在收到某种消息后所作出的反映。根据面向对象的这一思想,将虚拟加工中制造系统的各物体分成如下3类:

(1)静态类 在虚拟加工过程中位置和形状都不会改变的物体,如机床导轨、主轴箱和其他辅件如灯罩、操作面板等,它们之间除机械连接外,没有相互的运动联系;

(2)运动类 在虚拟加工过程中位置可以改变但形状不会改变的物体,如刀具、卡盘尾架和刀架等,可分为程控型运动和随机型运动。刀具、刀架、尾架等在加工过程中做平移运动,而卡盘、夹头、丝杆等在加工过程做旋转运动;

(3)动态类 在虚拟加工过程中,位置和形状都可以改变的物体,如工件。

根据上述分类并应用类的概念,将物体抽象为节点,节点主要由如下项目构成:物体名称;物体的材质与颜色;刻画物体几何形状的一组 3 维数据;物体的位置;描述物体 3 维旋转运动的四元数;物体的运动成员函数。

5 虚拟加工过程的实现

5.1 静态对象与运动对象的设计

静态对象与运动对象的一个重要特性是在加工的过程中形状不会发生变化,因此对这两类对象可采用 3 维造型软件如 Pro/E 进行特征造型并导出实体 STL 文件以建立实体文件库,使类的设计得到了简化。通过 OpenGL 重构实体 STL 文件可以获得对象的外观属性,在此基础上定义其材质、类型及运动函数即可构造类。作为对象成员的实体 STL 文件的实质是对几何形体的刻画,表现为一组浮点数据。它是一个用三角形网格面片对几何形体进行离散生成的数据文件,每一个三角形用 3 个顶点坐标和一个法向量来描述。在一个具体的非变形零件对象中,零件表面特征由 STL 文件描述,并且该零件的运动体现为 STL 三角形面片经过旋转变换或平移变换后逐帧放出的图像,图 2 是由 STL 文件重构的车床卡盘的一个实例。

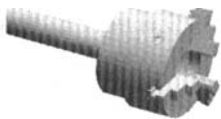


图 2 一个由实体 STL 文件重构的实例

Fig. 2 A instance of rebuilding an object from STL file

整个车床是由各个零件对象组成的,可通过设计车床相关零件成员及其运动来实现。运动函数中各个参数是对象的公有成员,仿真管理器通过改变其参数以达到使其运动的效果。

以下是一个运动类的描述:

```
class CMoveClass
{
public:
    CMoveClass( );
    virtual ~ CMoveClass( );
    //生成显示列表
    void    buildlist();
    void    Animating();
    char    *filename;
    //STL 文件名
    FILE    *fp;
    GLdouble position[3];
    GLdouble color[3];
    int list_name; //显示列表标识
    CQuaternion *rotate; //用于控制旋转的四元数
```

卡盘是一个运动对象,在加工过程中绕 x 轴线作旋转运动,可借助四元数来实现旋转,代码如下:

```
CMoveClass *MyChunk;
...
//绕  $x$  轴旋转 angle 角度,将其转化为四元数表示
MyChunk.rotate.CreateFromAxisAngle(1.0, 0.0, 0.0, angle);
MyChunk.rotate.CreateMatrix( pMatrix );
glMultiMatrix( pMatrix ); //将通过四元数建立的矩阵乘以当前矩阵
MyChunk.rotate.Animating( );
```

5.2 工件的动态生成与加工过程仿真

在虚拟数控车削加工中,工件实体是一个动态的回转体,它可以采用沿轴向方向将其分成多个高度较小的圆台近似地表示出来^[8,9](如图 3 所示)。

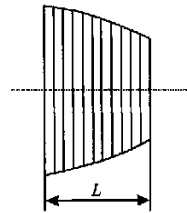


图 3 回转体的多个圆台近似表示

Fig. 3 Solid of rotation approximately represented by multiple circular truncated cone

为了实现任意形状的数控车削加工仿真,采用从 NC 程序中提取工件信息与加工轨迹信息。首先对 NC 程序进行译码,同时提取刀具加工轨

迹,然而进行加工预处理,包括通过刀具加工轨迹来生成刀位控制点,每一对控制点与一个圆柱面(圆台)的两端面半径相对应。如精加工图4所示的手柄,车床对毛坯进行圆弧插值。由于工件是一个回转体,可以在 xy 平面内来考查它。可从圆弧的中心点、方程及圆弧间的内外切特点来求得圆弧间的切点。切点将圆弧插值部分沿 x 轴分成了3段,每一段满足相应的圆方程。然后,按照图3所示的方法,将圆弧插值部分沿 x 轴细分为小圆台,根据 x 坐标值和相应段的圆方程求得相应的 y 值,也就是刀位控制点。因此,动态生成棒状工件问题可转化为动态生成各个圆柱面、圆台面和圆端面的过程。GLU(openGL utility library)是图形标准OpenGL的一个工具库,此工具库提供了变换坐标、网格化多边形、纹理操作和绘制常用形体等方面的操作。动态加工是通过时钟或键盘、鼠标等输入设备系统来引发事件,然后在其事件处理函数中实现动态显示功能。可设置两个时钟,一个用来管理纵向进给加工,一个管理横向进给加工,并根据加工的速度和长度设置时钟间隔的长短。在加工过程中,事件处理函数根据刀位控制点来改变相应参数,再重新绘制出来,从而驱动卡盘部件的转动、刀具和刀架沿加工轨迹的平移以及刀具平移时经过控制点相应的动态物体所发生的形变,即得到动态加工过程的仿真。

其具体步骤可描述如下(对于外圆加工):

(1) 初始化 根据加工速度 F 和加工长度 L ,设置横向进给时钟间隔 T_1 及每时钟间隔横向进给量 L_x ,根据纵向进给量 h 、总切削量 H 及横向进给时钟间隔 T_1 设置纵向进给时钟间隔 T_2 ,已纵向进给变量 $h_p=0$;

(2) 根据刀具加工轨迹生成自动控制点;

(3) 对刀,纵向进给 h ,加工开始;

(4) 驱动卡盘及工件转动,若发生横向进给时钟事件,刀具沿轴向平移 L_x ,并且驱动平移所掠过路径的一个或多个圆台作出相应的形变;

(5) 纵向进给时钟事件发生否? 是,执行第6步;否,跳到第4步;

(6) 退刀至刀具零点, $h_p=h_p+h$, $h_p=H$? 是,则执行第7步;否,则跳至第3步;

(7) 退出。

数控外圆车削的实例如图4、图5所示(因印刷关系不能显示动态效果)。

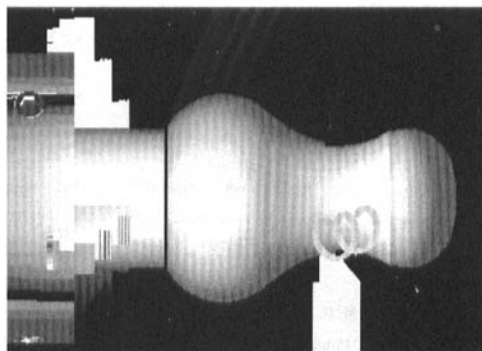


图4 精加工手柄局部图

Fig. 4 The part picture of finish machining the handle

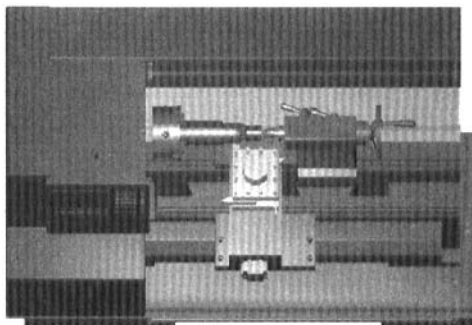


图5 车外圆的动态加工过程

Fig. 5 The dynamic machining process of cylindrical turning

6 结 论

通过引入了面向对象程序设计的思想,将虚拟加工环境中各个零部件根据其形状和位置是否发生改变进行分类。对非动态物体的外形采用了采用3维造型软件如Pro/E进行特征造型并导出实体STL文件以建立实体文件库,使类的设计得到了简化,而对动态物体即工件的外形采用实时生成的方法,并把复杂的外形以圆柱、圆台等简单的形状来近似地表示,把加工过程的形变的仿真转化为改变简单形体的参数来实现。实践表明,面向对象的虚拟数控机床的设计及其算法是可行的。上述方法为虚拟数控系统的研究提供了基本框架,具有很好的实用价值。

本文中研究的动态物体是针对回转体的,这种方法还不能对螺纹车削中的工件进行逼真的几何描述。采用何种模型来真实地表示螺纹工件、如何生

成具有真实感的切屑以及采用何种可靠的快速算法对数控加工过程中可能出现的碰撞进行检测,将是进一步研究的内容。

参考文献 (References)

- 1 Jiang Zao, Wang Hong-cheng. OpenGL VB/VC Graphics Programming[M]. Beijing: Science Press, 2001:3~8. [江早, 王洪成. OpenGL VC/VB 图形编程[M]. 北京: 科学出版社, 2001: 3~8.]
- 2 Hearn D, Baker M P. Computer Graphics[M]. Translated by Cai Shi-Jie, Beijing: Publishing House of Electronics Industry, 1998: 305~312. [赫恩, 贝克著. 计算机图形学[M]. 蔡士杰译, 北京: 电子工业出版社, 1998:305~312.]
- 3 Shoemake K. Animating rotation with quaternion calculus[J]. ACM SIGGRAPH Computer Graphics, 1985, 19(3): 245~254.
- 4 Eberly D. Quaternion Algebra and Calculus[EB/OL]. <http://www.magic-software.com/Documentation/Quaternions.pdf>, 2002-09-27.
- 5 Bobick N. Rotating Objects Using Quaternions[EB/OL]. http://www.gamasutra.com/features/19980703/quaternions_01.htm, 1998-07-03.
- 6 Void S. Quaternion Powers[EB/OL]. <http://gamedev.net/reference/article/article1092.asp>, 2000-07-06.
- 7 Zhu Heng, He Han-wu, Xiong You-lun, et al. Modeling and simulation of virtual manufacturing system[J]. China Mechanical Engineering, 1996, 7(3): 30~32. [朱恒, 何汉武, 熊有伦等. 虚拟制造系统建模与仿真[J]. 中国机械工程, 1996, 7(3): 30~32.]
- 8 Ge Yan-jun, Shi Zhi-hui, Lu Bi-hong, et al. Study on the virtual manufacturing environment technology of NC lathe[J]. China Mechanical Engineering, 2002, 13(8): 672~673. [葛研军, 施志辉, 卢碧红等. 数控车床虚拟制造环境技术研究[J]. 中国机械工程, 2002, 13(8): 672~673.]
- 9 Wang Xiao, Liu Hui-xia, Yang Run-dang, et al. Modeling about virtual equipment and simulation of machining processes[J]. China Mechanical Engineering, 2003, 14(17): 1489~1491. [王霄, 刘会霞, 杨润党等. 虚拟设备的建模和加工过程仿真[J]. 中国机械工程, 2003, 14(17): 1489~1491.]