

基于 PMAC 通信的机器人镀膜路径算法实现

The design of path arithmetic in digital control based on PMAC communications platform

邓中亮 常耀斌 (北京邮电大学电子工程学院 100876)

摘要:针对 PMAC(Programmable Multi-Axis Controller)在数控系统中能够实现实时数据采集的应用优势,本文开发了基于 PMAC 通信的机器人镀膜系统应用程序。它是一套基于 PMAC 多轴运动控制器开发的高精度、多功能、自动化的系统。在讨论关于 PMAC 与 Windows 应用程序之间进行通信后,重点说明了如何设计可靠和高效的路径算法,通过路径规划算法,满足低存储量需求的链表数据结构分析,排序算法实现,实现了稳定性好,可读性好,执行效率高的,具有可移植性的数控嵌入式系统的算法。

关键词:PMAC PComm32 缓存区 路径算法 数据结构

1 引言

PMAC 是 American Delta Tau 开发的开放式多轴运动控制器。由于采用了 DSP56 系列的 CPU,实现了高效实时的数据采集和灵活控制多轴运动。软件方面 PMAC 的开放结构,编程的灵活性以及后台运行的 PLC(Programmable Logic Control)程序都为系统的开发提供了便利条件。机器人镀膜系统以工业控制机为平台、PMAC 为控制器构成主从式双微处理器结构。通过 PComm32 驱动程序,下载到机器人运动轨迹文件缓存区,通过多个定时器控制机器人在真空中实时精确镀膜。由于镀膜工件是不规则空间曲面,需要设计一种算法,让机器人自动识别不同形状的工件后,并能实时调整速度,角度和溅射路线,达到精确的自控镀膜。由于不同工件的工艺文件不同,需求算法能根据不同工件的数学模型,反解并拟合出相应的溅射曲线,PMAC 采集到溅运动轨迹文件的数据后,完成匀速的空间镀膜。基于上述需求的考虑,本文的设计的算法设计具有多种轨迹插补模式,经过多次实验,不仅符合高效率与低存储量需求,健壮性和可移植性很好。

2 PMAC 与应用程序的通信方法

关于 PMAC 与 Windows 应用程序之间进行通信,提供了 PComm32 通信驱动作为通信的渠道,其中

PComm32 是最有效的通信开发工具,不仅包含 PMAC 的通信方法,而且与 VC 具有很好的兼容性。PComm32 通信驱动程序有三部分组成:PMAC.dll, PMAC.sys, PMAC.vxd。

首先,本文通过显示链接动链接 PMAC.dll 到应用程序。下面是实现方法:

```
hPmacLib = LoadLibrary("Pmac.dll");
if(! hPmacLib)
{
    AfxMessageBox("不能与 pmac 卡通信!");
    return;
}
m_bDevice = (bool) OpenPmacDevicec(m_
dwDevice);
if(! m_bDevice)
{
    AfxMessageBox("Device can't open!");
    return;
}
FreeLibrary(hPmacLib);
```

其次,通信方法及函数实现简单介绍,如下采用下面五步来完成。

(1) 读取 PMAC 函数的地址,实现打开与 Pmac 卡

的通信 `OpenPmacDevicec(DWORD m_dwDevice)`。

(2) 实现 PMAC 进行命令传递函数的方法为 `GetPmacResponse(DWORD m_dwDevice, PCHAR m_pcResponse, DWORD m_dwMaxchar, PCHAR m_Colse)`。

(3) 读取 PMAC 函数的地址, 实现函数 `PmacDownloadFilec(DWORD m_dwDevice, char * m_pcResponse)` 用于文件的下载。

(4) 读取 PMAC 函数的地址, 实现函数 `PmacGetVariablec(DWORD m_dwDevice, char m_pcResponse, UINT num, short int m_number)` 用于得到 I, P, M 变量的值。

(5) 关闭与 Pmac 卡的通信 `ClosePmacDevicec(DWORD m_dwDevice)`。

通过上述通信方法和流程实现了 Pmac 卡的通信过程。

3 算法研究与实现

3.1 路径规划算法与数据结构分析

10410 任何物体均可被分解成许多不同形体元素的集合。较简单的表示形体的图形组合可以构成较复杂的图形, 较复杂的图形组合又可以构成更复杂的图形, 一层一层地构造下去, 就可以得到整个几何图形。几何图形一般都有层次结构, 利用层次结构可以很容易地实现几何图形模块化设计。

3.2 用链表数据结构保节点信息方法

工件作为一个曲面图形来说, 可以被分解为若干个三角形的集合, 每个三角形是整体工件曲面的一个模块。三角形可以由三个顶点在三维空间的坐标 (x_1, y_1, z_1) 、 (x_2, y_2, z_2) 、 (x_3, y_3, z_3) 和三角形所在平面的法向量 (a, b, c) 所唯一确定。考虑到工件的这些图形特点, 它的图形数据结构可以采用链表结构描述, 工件的每个构成三角形占用链表的一个结点。而溅射曲线则可以被分解为若干个直线段的集合, 每个直线段是整体溅射曲线的一个模块, 记录相邻直线段的交点和直线段所在直线的方向矢量, 就可以唯一地描述该曲线。同样可以采用链表结构描述, 链表的每个结点保存一个交点和方向矢量信息。

3.3 排序算法实现直线段部分拼合

确定溅射曲线, 首先需要按照溅射靶的直径对工件曲面进行分层, 用层面对工件曲面进行切割, 求得层面切割曲线, 即用层面和工件图形数据结构中的所有构成三角形求交点, 并同时记录求得交点所在三角形平面的法向量。这是因为靶每次必然完成正好等于直径宽度的溅射。然后, 对每层面求得全部交点进行排序, 求出交点合成曲线——工件曲面和层面的交线段逆时针顺序集合。一个三角形和层面至多有两个交点, 两个交点之间的连线为通过该三角形的直线段。由此计算出所有直线段后, 需要将直线段拼合起来。思考三角形的拓扑结构, 可以知道, 除了位于工件曲面边界三角形上的交点外, 其余的交点必然同时属于两个三角形。交点所在不同三角形平面的法向量的合成向量可近似被看作交点合成曲线在该交点的方向向量, 这样就可以将溅射曲线分属于两个相邻三角形的不同两个直线段部分拼合起来。

3.4 如何拟和切割曲线

求出各个层面的层面切割曲线后, 需要将各个层面切割曲线进行拟和, 用工件曲面的边界曲线在相邻层面之间的曲线段连结相邻层面的层面切割曲线。为了构成行走路径不发生重复溅射曲线, 每次都从与上一次相反的一侧进行拟和, 若拟和起点的位置角度小于 90° 度, 则直接连接原先的逆时针层面切割曲线, 反之, 需先倒, 转层面切割曲线为顺时针方向, 在连接到溅射曲线上。连接上全部的层面切割曲线即求得了完整的溅射曲线。

3.5 进行坐标变换

用溅射曲线对机器人进行行动指导之前, 还需要将溅射曲线进行坐标变换, 变换为机器人的各个电机的直接命令参数, 用三维坐标和两个旋转坐标描述。然后, 按溅射曲线方向, 依次向目标文件写入命令参数。

3.6 生成目标文件

至此对工件的分析结束, 实际溅射时, 打开目标文件, 按照命令参数控制机器人的各个电机, 指导它的行动, 即可完成该种工件的溅射。Patchfile() 函数的执行是由类中的函数调用的, 调用此函数主要是为了生成工件的镀膜文件, 调用此函数前首先要选择工艺号(也

就是所要操作的 stl 文件), 在函数中首先要调用 `Realfile()` 函数, 该函数中可以生成机器人的运动轨迹文件。在函数中得到了当前的靶号, 这是在离线界面调试中选择的。这样得到了取靶的文件。 `pView -> m_page1 -> PmacDownloadFile(0, "OpenBuffer.txt")` 函数是向 pmac 卡中下载打开循环缓冲区的文件。

3.7 路径规划算法流程图如下

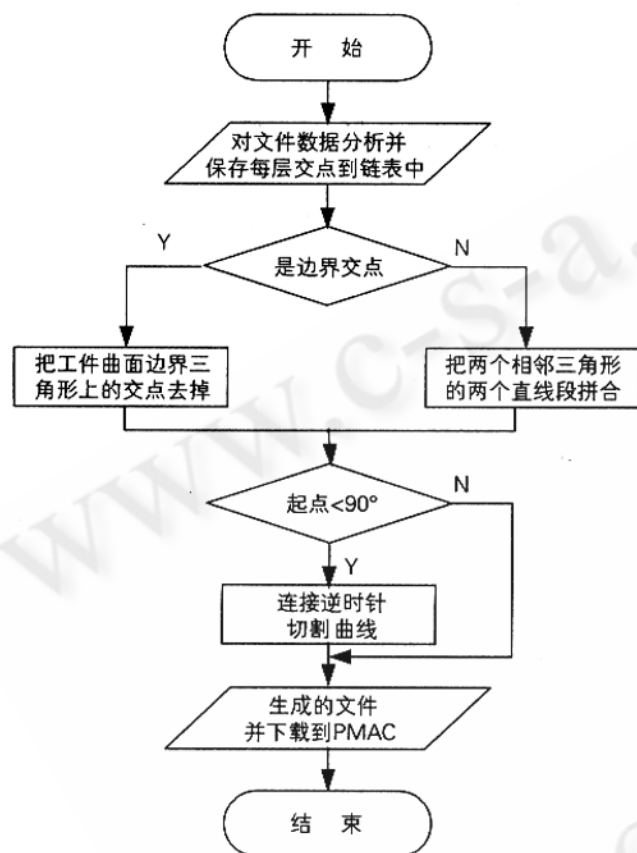


图 1 路径规划算法流程图

3.8 算法需要生成的文件结构定义如下

CUtil 是算法的类, 需要生成拟合数据的文件, 反解数据的文件, 工件数据的文件, 原始数据的文件。

```

CUtil: CUtil(char * F, char * M, char * S, char * A)
{
    Init(); //文件初始化
    //拟合数据的文件名
    m_FitDataFileName = F;
    //反解数据的文件名
  
```

```

    m_MoveDataFileName = M;
    //工件数据的文件名
    m_STLFileName = S;
    //原始数据的文件名
    m_OriginalName = A;
  }
  
```

3.9 算法的设计分析

该算法的控制结构和原操作结合很好, 对结点采用选择排序算法, 稳定性好, 由于用 c 开发, 所以可读性, 执行效率比其他高级语言更好。由于采用链表实现了低存储量的需求。

4 实验与结论

在 VC++6.0 平台下, 通过 OpenGL 来产生曲线和曲面达到对算法的仿真效果, OpenGL 可以生成任意角度的多项式曲线, 并可以将其他类型的多边形曲线和曲面转换成贝塞尔曲线和曲面。这些求值器能在任何度的曲线及曲面上计算指定数目的点。由于可让 OpenGL 计算在曲线上所需的任意数量的点, 因此可以达到应用所需的精度。VC 的 MFC 中提供了 `SetTime()` 设置定时器的基本信息。并在定时器引发的时候发送消息 `WM_TIMER`。路径算法仿真可以通过用 `SetTime()` 和 `OnTime()` 两个函数对 `Timer` 进行控制。用 OpenGL 绘制算法生成的文件。经过仿真和多次实验, 该算法已在数控嵌入式系统中得到成功应用。

参考文献

- [美] S 巴斯. 计算机算法: 设计和分析引论. 朱洪等译. 上海: 复旦大学出版社, 1985.
- 严蔚敏, 数据结构, 北京: 清华大学出版社, 1992.
- 周启平, VxWorks 下设备驱动程序 BSP 及开发指南 [M], 北京: 中国电力出版社 2004 年. 1-312.
- 张乃孝, 算法与数据结构, 高等教育出版社, 2002.
- 胡谋, 计算机容错技术, 北京: 铁道出版社, 1991.
- A Practical Introduction to Data Structures and Algorithm Analysis.