

数控铣削仿真系统关键技术研究

文 | 德州大陆架石油工程技术有限公司 马善坤

摘要: 提出了利用 B-rep 三维空间模型表示方法和 Z-MAP 表示法相结合的工步差异坯料模型表示方法,并在自动编程算法的基础上对坯料模型进行了离散。研究了切削刀具沿走刀路线移动的仿真方法和走刀过程中刀具与材料相交导致材料去除的模拟算法,提出了更为优秀的坯料离散点检索算法,仿真速度能够大大提高,最后开发了仿真系统,验证了该仿真算法的可行性和先进性。

关键词: Z-MAP; 走刀路径; 仿真; 材料去除

一、引言

当今世界,计算机科技与制造技术不断进步,加工过程中的仿真技术应运而生,并且在机械制造行业中得到了深入的推广。加工仿真的特点是在计算机屏幕上模拟加工,验证数控程序的准确性,对加工中的切削过程和刀具状态进行立体实时的仿真,同时对欠切、过切以及刀具和夹具、刀具和工件之间的碰撞干涉情况实现可视、定量的验证,能够形象、直观且快速地模拟数控加工的全过程,且不需要额外费用,同时能够达到试切等传统走刀路径的验证效果,极大地提高了数控机床的有效工时和使用寿命^[4-5]。对缩短产品的试制周期、提高数控加工效率、降低成本具有十分重要的意义^[1-3]。

国内自 20 世纪 90 年代初开始对数控加工仿真技术的研究,但至今先进的加工仿真技术仍然为国外制造业软件巨头所掌握,国内所开发的仿真系统尚未达到大规模应用的程度,尤其是在复杂零件数控加工方面,在仿真速度、真实性以及刀具路径规划等性能上国外一些主流系统差距较大。所以,攻克加工仿真的关键技术,深化刀具路径规划研究,提高仿真的真实性与仿真速度,是中国掌握具有自主知识产权的、先进实用的仿真加工系统的必经之路。本文研究了数控加工仿真中的几项关键技术,提出了基于 Z-MAP 建模方法、以走刀轨迹为驱动的工步差异坯料构建方法以及

在 OpenGL 环境下模仿工件材料去除的算法,能够有效地减少计算机的运算次数,提高仿真速度及质量。

二、刀具模型和坯料模型的构建

1、刀具模型

数控铣刀的种类主要有立铣刀、面铣刀、键槽铣刀、鼓形铣刀和成形铣刀等。由于刀具的多种分类和规格,因此刀具的几何形状也多种多样,若完全真实描述铣刀三维模型,工作量和难度都非常大;而且仿真过程中复杂的刀具模型在三维环境中的重绘时间也会更长,会影响加工仿真的实时性。因此在满足仿真要求的前提下,构建更为简单的刀具模型能够提高仿真效率和真实性。刀具在进行铣削加工时,本身会高速旋转,通常此时刀具的切削刃所形成的包络面为圆柱体,使用圆柱体来代替工作状态中的铣刀,既能节省刀具重绘时间,也不会影响仿真效果,因此这种方法在铣削仿真的研究中应用最为广泛。刀具在仿真过程中被认为是刚体,即不用对其做几何处理。

本文基于工艺信息中的刀具参数采用 OpenGL 提供的绘制圆柱体的库函数 `gluCylinder` (quad, base, top, height, slices, stacks) 来完成绘制铣刀切削刃部分模型。函数的形参中包括了刀具的直径、长度等基本信息。

2、坯料模型

模型边界表示方法 (B-rep 法) 是目前主流的三维空间

模型表示方法之一。在 B-rep 模型中,空间点是最基本的模型要素,拓扑出边、环和面直至完整的三维模型。一个完整的 B-rep 模型记录了全部的点、边、环和面的几何信息和拓扑信息,因此三维模型的每个面、面的每一条边界边及每一个顶点的信息都可以被直接读取,这样就有利于以面、边为基础来进行各种几何操作,如划分有限元网格、构建三维线框模型、计算数控切削过程的走刀路径、验证是否有碰撞干涉,以及显示具有超强真实感的图形等,因此 B-rep 模型表示方法与其他模型表示方法相比,具有很大的优势^[6-7]。B-rep 法将其模型按照实体、面、边、环和顶点描述,在计算机内部按网状的数据结构进行存储,如图 1 所示。

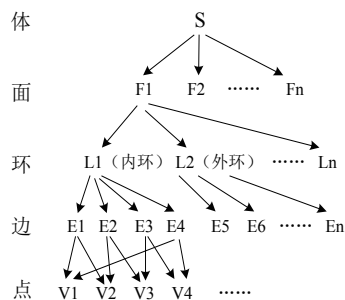


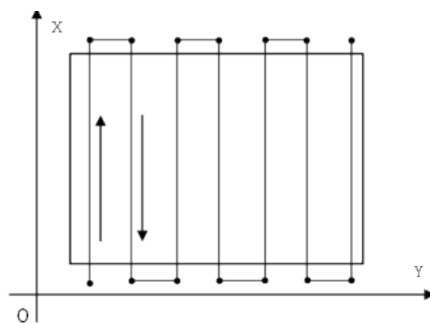
图 1 B-rep 模型拓扑结构

Z-MAP 表示法是 Yang 和 Hsu^[8]提出的, Z-MAP 是一种使用离散模型来表示三维空间实体的方法。该方法的基本思路是将初始的三维模型在 XOY 平面进行投影,在 XOY 平面得到一个二维平面,将该平面按照一定精度进行离散,形成均匀分布的许多正方形面片,因此可以认为整个三维实体是由多个以正方形为底面的小长方体的集合所构成。若使用 Z-MAP 方法构建三维实体来表示工件,则在铣削过程中,铣刀与毛坯相交进行材料去除的过程可以近似表示为离散小长方体的高度连续降低的过程,因此该方法被形象地称为“割草法”。

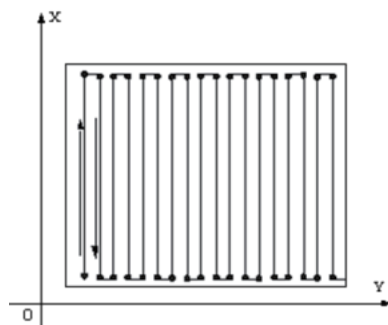
但传统的 Z-MAP 表示法是将整个三维实体模型进行离散,将整个模型表示为由许多以正方形为底面的小长方体的集合。将整个工件进行离散意味着离散算法的复杂以及计算机的运算量和显示工作量的增加,仿真速度和效果将会大大下降。

本文提出了基于 Z-MAP 方法和工序间几何差异构建工步差异坯料的方法。该建模方法的思路是:零件本体仍然使用 B-rep 表示方法表示,本道工序将要去除的区域内的材料以离散矢量建模方法表示。

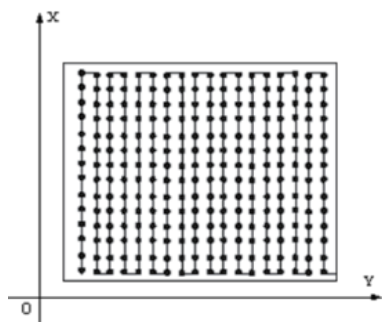
仿真系统通过读取数控工艺,生成刀具坐标和走刀路线,以简单的平面铣削为例,得到的刀具坐标和走刀路径如图 2 (a) 所示,刀具坐标存储在 CutterPointList 链表中。基于数控工艺中的起刀点和铣削方向,系统自动在铣削区域内生成致密的虚拟刀具坐标,如图 2 (b) 所示,虚拟刀具坐标在链表中的排列顺序与实际刀具坐标完全一致,可以看成是切出量为零,刀具横向移动极小的刀具坐标链,虚拟刀具坐标是工件坯料离散的基础。然后在虚拟刀具坐标之间插入均匀的离散点,插入离散点后效果如图 2 (c) 所示。坯料上的离散点统一存储在 RoughPointList 链表中。最后基于离散点的坐标构建底面为正方形的长方体,长方体的高度即为被去除材料的总高度。



(a) 实际走刀路线和刀具坐标

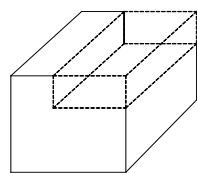


(b) 虚拟刀具坐标

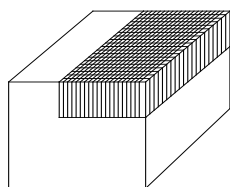


(c) 插入离散点

图 2 坯料离散点阵的生成



(a) 待加工特征



(b) 构建局部毛坯

图 3 工步差异坯料构建效果

图 3 (a) 中所示，在某个工序内如果要铣削一个台阶面，则仅对台阶特征的材料区域进行坯料的构建，工步差异坯料构建效果如图 3 (b) 所示。

三、走刀轨迹的仿真

走刀轨迹的仿真即铣刀沿其切削路径的移动。本文中，走刀路径由 **CutterPointList** 链表（刀具坐标链表）中的所有铣刀坐标按顺序连接形成，走刀路径仿真的本质就是铣刀在 OpenGL 三维环境中沿走刀路径不断重绘的过程。在进行仿真时，铣刀不但要依次到达刀具坐标链表中的刀具坐标，还需要演绎刀具在两个刀具坐标之间行进的过程，这就需要在两个刀具坐标之间插入新的更加致密的仿真用的虚拟刀具坐标，以便获得更好的仿真效果。其方法是首先计算出下一个虚拟刀具坐标，然后拭除原有刀具坐标上的铣刀模型并在新的刀具坐标上构建新的铣刀模型，因此走刀轨迹仿真的关键是计算下一个虚拟刀具坐标的坐标和刀具在 OpenGL 环境中的重绘。

1、新刀具坐标的计算方法

假设第一个轨迹点的坐标是 (x_1, y_1, z_1) ，第二个轨迹点的坐标是 (x_2, y_2, z_2) ，则这两点之间的长度为：

$$L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (1)$$

设数控机床每一次在两轨迹点连线方向上的微小的移动长度为 **step**，则 **step** 在 x, y, z 三个方向上的分量分别为：

$$\begin{aligned} \text{step}_x &= \text{step} \cdot \frac{x}{L} \\ \text{step}_y &= \text{step} \cdot \frac{y}{L} \\ \text{step}_z &= \text{step} \cdot \frac{z}{L} \end{aligned} \quad (2)$$

因此下一个刀具坐标的坐标为当前点 **CutterPoint** 的坐标加上 **step** 在 x, y, z 三个方向上的分量，然后将一个刀具坐标的坐标赋值给当前刀具坐标 **CutterPoint**，即：

$$\begin{aligned} \text{CutterPoint } x &= \text{CutterPoint } x + \text{step}_x \\ \text{CutterPoint } y &= \text{CutterPoint } y + \text{step}_y \\ \text{CutterPoint } z &= \text{CutterPoint } z + \text{step}_z \end{aligned} \quad (3)$$

最后当下一个虚拟刀具坐标到达或已经超过 **CutterPointList** 链表中第二个刀具坐标时，则将第二个刀具坐标赋值给当前刀具坐标 **CutterPointList**。然后以第二个刀具坐标作为下一轮的第一个刀具坐标，继续进行，直到 **CutterPointList** 链表中的最后一个点。

为了真实演绎仿真效果，上述对刀位坐标的计算必须在很微小的时间间隔内进行，本文通过 **SetTimer()** 函数来进行时间间隔的设定，然后触发 **OnTimer()** 函数中进行刀位坐标的计算。

2、刀具在 OpenGL 环境中的重绘

本文通过 **ToolDraw()** 函数来完成刀具在三维环境中的重绘。**ToolDraw()** 函数的形参即为下一个刀具坐标。在此坐标上调用刀具的绘制函数 **gluCylinder(quad, base, top, height, slices, stacks)** 即可。

四、坯料材料去除算法的研究

坯料材料去除的过程即是刀具扫描体与坯料进行求交运算并对坯料更新的过程。刀具由一个刀具坐标移至另一个刀具坐标时，会扫略一片空间区域，在图 4 中，由 A、B、C、D、E、F 所组成的阴影面积即为一次走刀后，刀具的扫略区域，称为刀具扫描体。

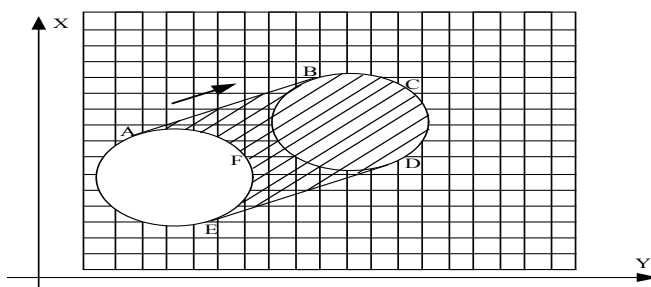


图 4 刀具扫描体的形成

进行工件的材料去除仿真算法时，刀具每走到一个刀具坐标，都要检验是否需要与坯料进行求交运算。若坯料上的离散小长方体需要和刀具扫描体进行求交计算，则应该满足

两个条件：坯料上的离散点在刀具扫描体内；离散的小长方体的高度超过铣刀当前坐标的 Z 坐标值。

若以上条件都能满足，则修改小长方体的高度，使其与当前刀具的 Z 坐标值一致，当小长方体的高度变为0时，说明刀具扫描体内材料已经被完全切除，此时可以将其对应的离散点从 RoughPointList 链表中删除；若以上条件不能同时满足，则无法进行切削过程的求交计算。在实际铣削加工仿真中，虚拟刀具坐标之间的距离越小，模拟连续走刀的效果越好，因此刀具扫描体所覆盖的区域也极小。

传统的方法在每次刀具扫描体与坯料进行求交计算时，是遍历坯料离散点链表中的空间点，判断其是否能同时满足上述两个求交运算的条件。由于坯料的离散点通常极多，几十万甚至上百万个，若每次刷新时都使用传统的方式进行搜索，则每次都要执行数十万次的判断，以现在的计算机处理速度，仿真效果和速度将不理想。本文基于工件坯料的构建原则，提出了一种优化的搜索算法，在完全保证搜索精度的前提下，仅需要搜索离散点链表中的一小部分，因此能够大大减少搜索量，提高计算机的仿真速度和效果。算法流程图如图5所示。

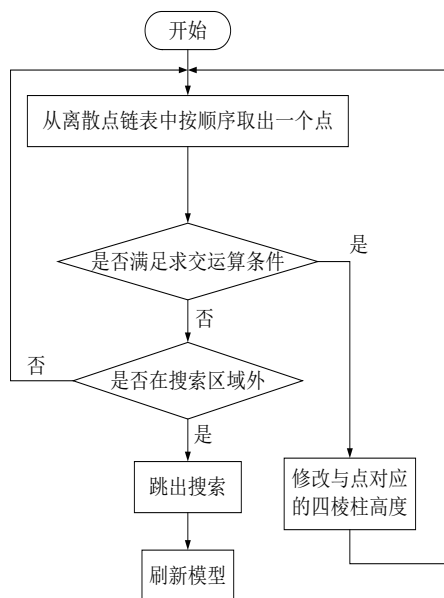


图5 坯料离散点搜索算法

搜索范围的定义：搜索范围是铣刀在下次变换走刀方向之前所将要扫略过的材料区域，如图6所示。由于坯料离散点在链表中的存储顺序和本工步中的刀具路径的走势一致，因此如果判断出一个点不在搜索范围之内，则表明之后

的所有点也都不会再搜索范围内，也就是说该点之后的其他坯料离散点都不会与铣刀的刀具扫描体相交，所以可以在该点处结束搜索从而减少计算机运行次数。

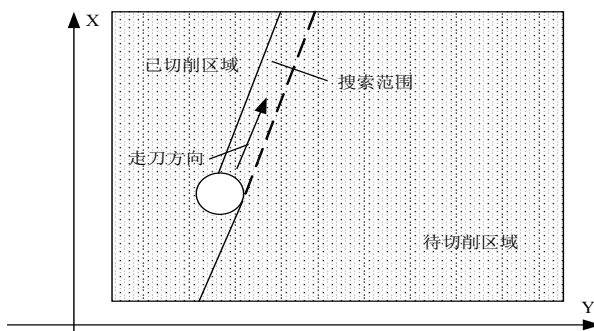


图6 刀具切削范围

(1) 从毛坯离散点链表的第一个点开始，按照点的存储顺序，将离散点依次读取。每读取一次即进行一次判断，确定该点是否满足两个求交运算条件。由于坯料点链表中坯料点的排列顺序与刀具路径走势一致，因此链表中的前一部分点肯定满足求交运算条件，因此本搜索算法可以快速中靶。

(2) 若一个点满足求交运算的条件，则修改与该点对应的小长方体的高度，使之与刀具坐标的纵坐标一致，若小长方体的高度不为零，则将该点移至 RoughPointList 链表的尾部；若小长方体的高度变为零，则将该点直接从 RoughPointList 链表中移除。

(3) 若某点不满足求交运算，则判断该点是否在搜索范围内。若该点在搜索范围内，则重复步骤1，否则结束搜索并刷新工件的三维空间模型。

工步差异坯料构建能够有效地省计算机运行时间，提高仿真的速度，由于零件本体不与刀具扫描体进行求交运算，因此无法模拟过切效果。而对过切现象的判断是加工仿真非常重要的内容，本文通过在走刀时判断刀具包络面是否切入了零件本体内部，从而判断是否产生过切，发生过切的特征面采用特定的颜色标识。

五、实例验证

(1) 基于本文所述理论，开发了数控加工仿真系统，系统通过解析数控工艺，基于当前工步的工艺参数生成刀具坐标和数控代码（图7），并且在坯料上显示走刀路径（图8）。

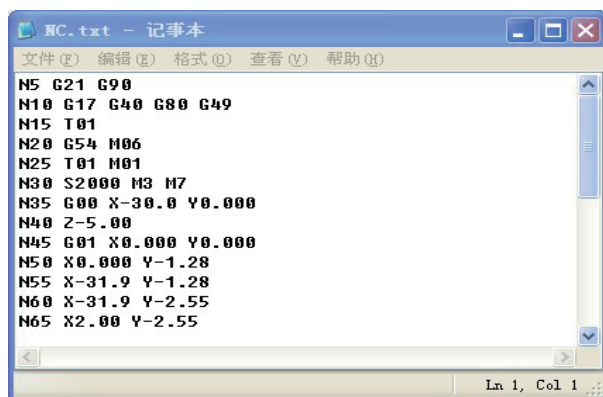


图 7 NC 代码

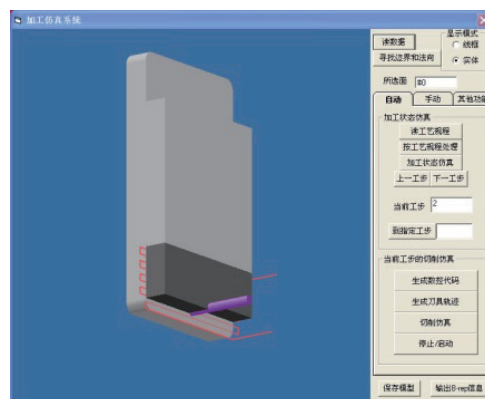


图 10 动态切削仿真

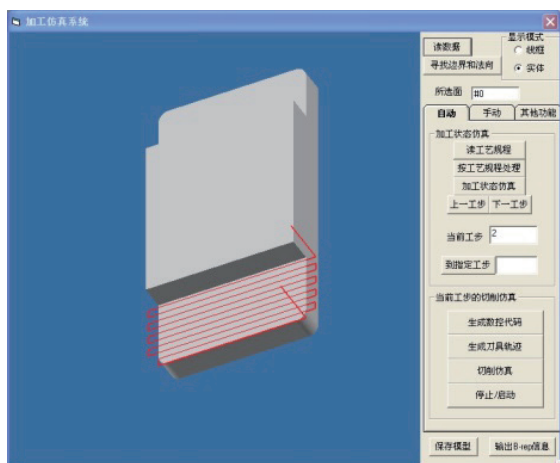


图 8 走刀路径

(2) 系统根据数控工艺自动生成工步差异坯料模型和刀具切削刃部分模型,如图9所示,为了显示工步差异坯料构建效果,本文对坯料模型采用了与零件本体不同的颜色。材料去除的动态仿真效果如图10所示。

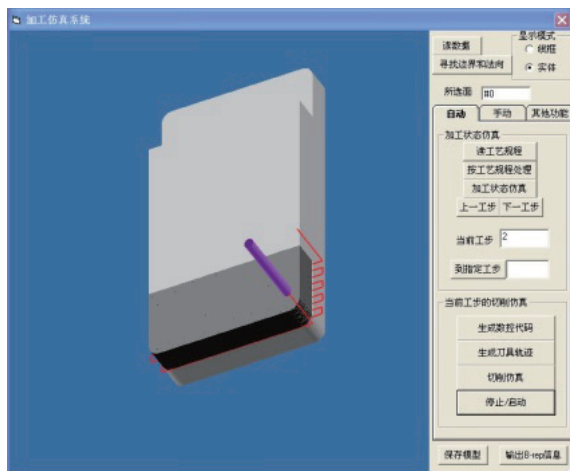


图 9 坯料模型和刀具切削刃部分模型

六、结语

本文以数控工艺为驱动基础;利用 Z-MAP 方法实现了对铣削过程中的坯料进行局部重建,研究了走刀路径的仿真方法和切削过程材料去除的仿真算法,提出了优化的坯料离散点搜索算法,大大降低了计算机的运算次数,提高了仿真的速度和质量,并通过仿真实例验证了材料去除的动态仿真算法的可行性。

参考文献:

- [1] 陈文义, 刘志新, 常跃峰. 虚拟制造的铣削加工过程仿真技术研究[J]. 煤矿机械, 2005, (5): 105-106
- [2] 余斌, 刘荣忠. 基于 OpenGL 的数控加工仿真系统研究[J]. 四川大学学报, 2001, 33(5): 16-19
- [3] 江早, 王洪成. OpenGL VC/VB 图形编程[M]. 北京: 科学出版社, 2001, 329-334
- [4] 李鹏, 王太勇, 李宏伟等. 虚拟环境下的数控加工实时 3 维仿真系统的研究与开发[J]. 制造业自动化, 2004, 26(8): 37-42
- [5] Liu T T, HOU S L, TAO Y D, et al. Processing information extracting and machining simulation of the revolving model parts[C]. 2009 International Conference on Measuring Technology and Mechatronics Automation, China · Hunan · Changsha, 2009: 292-295
- [6] Zheng Bo Chuan, Peng Wei, Zhang Yin et al. A survey on 3D model retrieval techniques. Journal of Computer Aided Design & Computer Graphics, 2004, 16 (7) : 873-881
- [7] Gao S, Shah J J. Automatic recognition of interacting machining features based on minimal condition subgraph. Computer Aided Design, 1998, 30(9):727-739
- [8] San-Kyu Lee, Sung-Lim Ko. Development of simulation system for machining process using enhanced Z map model[J]. Journal of Materials Processing Technology, 2002(130-131): 608-617