

基于 CATIA/CAA 的四轴自动钻铆定位系统 干涉检测编程方法研究及应用

Research and Application on the Collision Detection Programming Method of Four-Axis Automatic Riveting Allocation System Based on CATIA/CAA

中航工业昌河飞机工业(集团)公司 陈卫平
南京航空航天大学机电学院 徐岩
陆军航空兵住景德镇军事代表室 陈先有

[摘要] 干涉检测是自动钻铆加工前的重要准备工作,全文分别通过几种不同的干涉检测方法实现了运动仿真中的干涉检测,通过对各自方法取得的实际效果进行对比分析,得出通过基于 CAA C++ 的方式调用 CATIA 自有的干涉检测功能不仅运算速度快、干涉检查信息全,并且程序运行稳定,效果最好,同时应用于开发的四轴数控托架仿真系统,辅助自动钻铆机生产加工,具有较广泛的应用前景。

关键词: CATIA 组件应用架构 二次开发 干涉检测 仿真

[ABSTRACT] Collision detection takes an important part in the preparatory work for automatic riveting. The collision detection function during the movement simulating by using of several different methods, through the analysis on the practical effect for each method, the conclusions that this collision detection function runs fast and stably and gives the complete information of the collision for the operators that using the collision detection function of by use of the CATIA method based on CAA C++ are got. The method of using CATIA collision detection function is applied to the development of the movement simulation system for four-axis NC bracket simulation system, and assisting automatic riveting machine, having great practical value and wide applications in the future.

Keywords: CATIA Component application architecture(CAA) Secondary development Collision detection Simulation

在飞机制造中,铆接装配占有十分重要的地位,随着对飞机质量和性能要求的不断提高,企业愈来愈重视铆接质量,自动钻铆技术在国外航空制造企业得到了广泛应用^[1]。自动钻铆是在一台设备上一次性地连续完

成定位、夹紧、钻孔、镗窝、注胶、放铆和铣平等工序。与传统手工铆接相比,自动钻铆铆接质量高,工作稳定,而且可以同时节省大量的人力和时间^[2]。而干涉检测是自动钻铆加工前重要的准备工作,精确的干涉检测能够辅助操作人员提前发现编制的钻铆加工路径存在的问题,及时优化,提高了加工路径的可靠性及钻铆加工效率^[3]。因此,国内外研究人员对干涉检测技术进行了积极的探索研究,并取得了一定的成果。

目前国外在计算机仿真中实现的干涉检测技术已经十分成熟,应用比较广泛的仿真软件(如 UG 等)都具有功能强大的干涉检测功能。国内的一些研究院所和高校利用计算机仿真技术实现加工过程中的干涉检测,也取得了比较好的效果。纪玉坤^[4]等基于 UG 二次开发平台,对五轴数控机床进行加工仿真并检测运动部件间的干涉碰撞;余斌^[5]等采用 OpenGL 技术,开发了数控车、铣、钻的三维仿真软件,用以检查刀轨的正确性和检测运动部件是否存在干涉;冯潼能^[6]等采用 CATIA 二次开发技术应用 VB 语言对 DMU(电子样机)模块进行二次开发,建立了钻铆机的运动仿真系统,初步解决了系统加工过程中出现的碰撞与干涉问题。

本文针对航空企业引进的钻铆设备,在自行研制的四轴数控托架自动定位系统中,基于 CATIA 提供的开发平台 CAA(Component Application Architecture),开发带有干涉检测功能的四轴数控托架仿真系统,同时由于干涉检测理论的多样性,导致实现干涉检测方法的多样化,不同的编程方法,其进行干涉计算的速度、对运动仿真动态显示的影响、干涉信息的获取、程序的稳定性都是各不相同的。因此,为了找出实现干涉检测功能的最优编程方法,本文对实现干涉检测的编程方法进行了较为深入的研究。

1 CATIA 二次开发方法

1.1 基于宏的 CATIA 二次开发

大多数 CAD/CAM 软件都支持宏操作,可以对生成的宏文件添加或修改其判断、循环、选择等条件,再重新运行,这便是一个开发过程。这种方法直接、容易,可以用来实现一些简单功能,属于手动操作的自动化、条件化和重新整合。但是功能有限,运行不稳定,对话框界面风格不一致^[7]。

1.2 基于 CAA 的 CATIA 二次开发

作为 CATIA V5 的组成部分,组件应用架构 CAA 是 Dassault Systemes 产品扩展和客户进行二次开发的强有力的工具。它建构在 Microsoft Visual C++ 下,通过 API 函数调用 CATIA 的核心程序,并进行程序开发,从而实现了客户程序和原系统的紧密集成,完全可以实现对达索应用系统的深层次开发^[8]。

基于 CAA 的开发方式是一种功能更强大、层次更深入的开发方式,开发出的功能对话框与 CATIA 界面风格完全一致,程序运行十分稳定,本文所采用的就是基于 CAA 的开发方式^[9]。

2 空间物体干涉检测理论与方法

干涉检测一般包括静态干涉检测和动态干涉检测^[10]。在自动钻铆定位系统的运动仿真过程中,我们主要关心的是运动包络体是否存在零部件之间的运动干涉,所以本文所研究的是动态干涉检测。目前主要有 2 类技术被用于动态干涉检测:单步检测和扫描实体^[11]。单步检测是在物体移动过程中将轨迹分为很多时间步,在每一个时间步都进行静态干涉检测,判定运动的物体之间是否发生碰撞;扫描实体是指物体运动产生的扫描实体代表了物体在给定轨迹上移动所占有的体积空间,如果它们在各自的轨迹上运动时发生碰撞,则它们各自的扫描体将会静态干涉。本文基于功能实现的难易程度,选择了单步检测技术来实现动态干涉检测。

基于单步检测技术的空间物体动态干涉检测问题中,比较重要的是在每一个时间步 2 个几何模型间的干涉检测算法,其大致可分为 3 类:空间分解 (space decomposition) 方法、层次包围盒 (hierarchical bounding volumes) 方法^[12]和布尔求交 (boolean intersection) 方法。空间分解方法通常适用于稀疏的环境中分布比较均匀的几何对象间的干涉检测;层次包围盒方法应用得比较广泛,适用复杂环境中的干涉检测,但其干涉检测的精度较低;布尔求交方法在适用范围广泛的同时,相对于以上 2 种方法而言,干涉检测的精度更高,本文在这种方法的基础上实现了一种干涉检测方法。

3 基于 CATIA/CAA 的干涉检测编程方法

自动钻铆定位系统中干涉检测的任务主要是检测

钻铆运动仿真过程中是否存在因为加工运动路径不合理导致的零部件间的干涉与碰撞,从而帮助编程人员优化数控指令,更好地为实际加工服务。鉴于干涉检测功能在仿真中的重要作用,本文将详细阐述基于 CATIA/CAA 实现干涉检测的方法。

3.1 基于 CAA C++ 运用布尔求交方法实现干涉检测

本文是在单步检测技术的基础上实现了动态干涉检测,即把物体运动的轨迹分割成多个时间步,在每个时间步里分别对需要检测的零部件间进行一次布尔求交运算,布尔求交计算过程也是利用 CAA 提供的 C++ API 求交计算接口实现的,所以整个干涉检查功能的实现就变得简单了一些。我们在对任意 2 个零部件做布尔运算的时候,由于 CAA 提供的接口函数不能直接进行 2 个 part 的布尔运算,也不能直接对不同 part 中的个别 body 进行布尔运算,而只能对单一 part 中的 2 个 body 进行布尔运算,所以必须把 2 个 part 的布尔运算转化成单一 part 中的 body 的布尔运算,因此,就必须生成一个新的 part,将所要进行布尔运算的 part 中的 body 拷贝到新的 part 中,拷贝过程中必须保证数据、空间位置的正确性,然后通过 CAA 提供的求交函数实现布尔运算,整个方法的实现流程如图 1 所示。

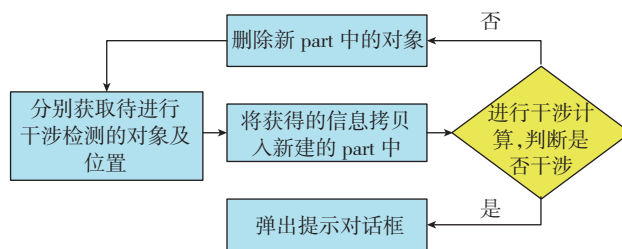


图1 运用布尔运算实现干涉的方法流程图

Fig.1 Method chart for realizing collision by using boolean operation

图 2 所示为基于布尔运算实现的干涉检测,自动钻铆定位系统在运动仿真过程中同步进行干涉检测,在每一时间步内,将需要检测的零部件复制入新建的 part

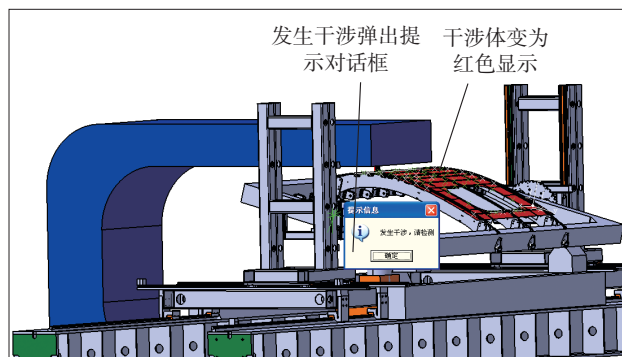


图2 基于布尔运算实现的干涉检测

Fig.2 Collision detection based on boolean operation

中,然后进行布尔求交运算,并检测到某些零部件间发生了干涉,弹出提示对话框。这种方法实现了我们预期的检测功能,但是由于需要在各个实体零部件间进行多次布尔运算,计算量比较大,虽然现在计算机的运算能力有了很大提升,但是对于如此大的计算量,还是不可避免地影响了仿真运动的动态显示,进而影响了仿真效果,同时对于干涉实体间干涉量的获取,还得通过另外的程序来实现。

3.2 基于自动化对象编程 AUTOMATION 的干涉检测

CATIA 自身自带的空间分析模块中包含了干涉检测功能,该功能检测速度快、精度高,同时还给用户提供了完备的干涉信息,便于用户分析处理。虽然该干涉检测的函数目前没有与之对应的对用户开放的 C++ API 接口,但是我们可以通过 AUTOMATION API 接口来调用该函数。这其实是一种基于宏的开发方式,会存在界面风格不统一的问题,我们可以通过写宏函数 WriteMacro() 与运行宏函数 RunMacro() 将宏嵌入基于 CAA 组件应用框架的开发程序中,既避免了手动运行宏,又解决了界面风格不统一的问题。图 3 所示为调用 CATIA 内部函数实现干涉检测的方法的整个流程。

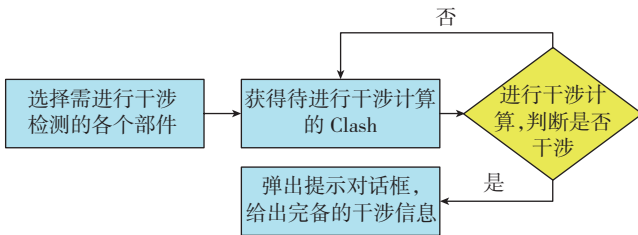


图3 调用CATIA内部的干涉检测函数实现干涉检测的方法流程图
Fig.3 Method chart for realizing collision detection by using CATIA internal collision detection function

通过调用 CATIA 内部函数实现的仿真系统如图 4 所示,在每个时间步内运行一次宏进行干涉检测,当发生干涉时,弹出提示对话框,同时还能为用户提供干涉信息,并且干涉检测的速度也很快,对仿真运动的动态

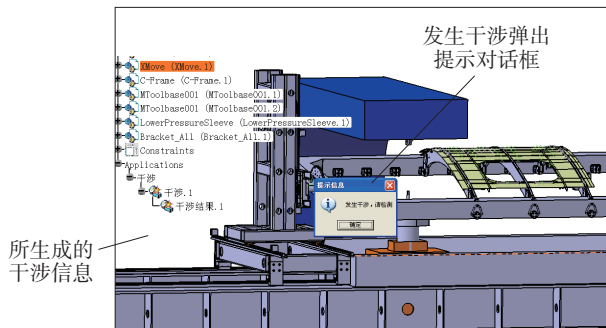


图4 调用CATIA内部函数实现的干涉检测
Fig.4 Collision detection by using CATIA internal function

显示基本没有影响。但是由于这是一种基于宏的开发方式,是一种操作层面上的开发,在仿真过程中连续运行宏,其实就是反复地打开、执行、关闭 CATIA 提供的干涉检测功能,这样会导致屏幕不断闪烁,影响了仿真的效果,另外,操作层面上开发的不稳定性,还会导致宏程序有时候无法响应。

3.3 运用 C 语言访问 AUTOMATION API 接口实现干涉检测

AUTOMATION API 接口提供的开放函数可以通过基于宏的方式来调用,也可以通过基于 CAA C++ 的方式来调用,因为其底层语言依然是 C 语言。因此,我们可以在 CAA 安装目录下找到与 AUTOMATION API 接口提供的干涉检测功能函数相对应的库函数,就可以在 C 环境下调用库函数中实现干涉检测的一系列函数,从而通过基于 CAA C++ 的方式实现干涉检测功能。具体流程如图 5 所示。

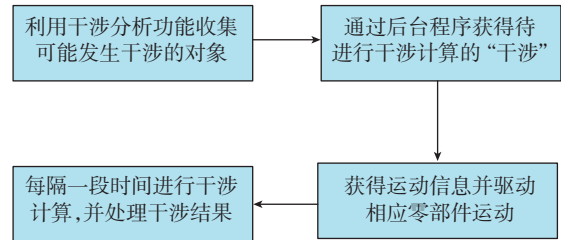


图5 运用C语言访问AUTOMATION API接口实现干涉检测流程图
Fig.5 Flowchart of access AUTOMATION API by C language to realize interference detection

流程中所涉及的主要功能函数如下:

// 获得待计算的“干涉”对象

```
CATIA Clash * FindClash(CATDocument * ipDocument, CATIPProduct * ipProductOnRoot)
```

// 根据驱动信息驱动相应的运动部件,并每隔一段时间,计算干涉结果

```
void SetTimer(UINT iElapse, CATDocument * ipDocument, double iAngleSpeed, double iXSpeed, double iYSpeed, double iZSpeed, CArray<double, double> & iListOnAngle, CArray<double, double> & iListOnX, CArray<double, double> & iListOnY, CArray<double, double> & iListOnZ, CATDocument * iTopDocument, CATIA Clash * ipClash, CArray<int, int> & iMovelist1, CArray<CATUnicodeString, CATUnicodeString> & Ncline)
```

该干涉检测方法同 3.2 介绍的方法仅在开发方式上有所区别,所采用的检测理论基础是相同的,即均借助组件应用架构(Component Application Architecture, CAA)提供的干涉检测接口 CATIA Clash 实现的,因此所

显示的干涉结果是相同的,如图4所示。不同的是,该干涉检测方法不再是操作层面上的开发,可有效避免屏幕闪烁、运行的不稳定性以及运行效率低等不足之处。同时,通过调用CATIA提供的干涉检测功能的函数,也不存在基于CAA C++运用布尔求交实现干涉检测的方法中干涉检测计算量大、速度慢、影响动态显示的问题。

3.4 3种方法的比较

3.4.1 3种编程方法对于动态显示流畅性的影响

3种干涉检测的编程方法基于不同的干涉理论,实现干涉检测功能的途径也各不相同,因此在自动钻铆加工仿真中进行干涉检测时3种方法的干涉计算量也各不相同,间接地影响到仿真过程的动态显示的流畅性,我们通过1组数据来具体对比各种方法对流畅性的影响(本文所获得的实验数据和仿真图片都是在主频为2.16GHz、内存为1.99GB的PC机这样的硬件条件下获取的)。例如:设置机床A轴的转动速度为 $2^\circ/\text{s}$,X轴的移动速度为100mm/s,Y轴的移动速度为100mm/s,Z轴的移动速度为50mm/s,数控托架从钻铆位置点A($15.3^\circ, 205.24\text{mm}, 151.17\text{mm}, 50.67\text{mm}$)运动到钻铆位置点B($20.5^\circ, 343.12\text{mm}, 287.35\text{mm}, 72.62\text{mm}$),在仿真运动过程中分别采用不同干涉检测方法,模拟完成整个运动过程的时间,如表1所示。

表1 采用不同干涉检测方法的模拟加工所需时间

	所需时间 /s	相对误差 /%
实际加工(理论上所需时间)	5.78	—
采用3.1所述方法的模拟加工	16.82	191.0
采用3.2所述方法的模拟加工	8.24	42.6
采用3.3所述方法的模拟加工	6.56	13.5

由表1可知,采用3.3所述的干涉检测方法,对于相同的干涉检测任务,较其他2种方法而言干涉计算量最小,计算速度最快,模拟加工所需时间与实际加工理论上所需时间最接近,时间误差也在许用范围之内,对仿真的动态显示影响最小,对其流畅性基本没什么影响。

3.4.2 基于不同编程方法的检测精度对仿真系统可靠性的影响

检测精度在干涉检测过程中起着至关重要的作用,直接影响整个钻铆加工仿真的可靠性,在不影响仿真效果的前提下,检测精度越高,仿真系统的可靠性就越好。基于不同干涉理论实现的干涉检测方法的检测精度各不相同,我们通过加工仿真过程中几组检测精度的数据,来具体对比各种方法对仿真系统可靠性的影响。

根据企业实际生产的情况,本次开发的仿真系统的检测精度为2mm就可以满足实际生产加工的需要。通

过表2,对不同的干涉检测方法分别在3次加工仿真过程中的检测精度进行比对,我们不难发现3种方法达到的检测精度都是符合实际生产要求的,但是3.2与3.3所述的干涉检测方法达到的精度明显高于3.1所述方法,因而采用后2种方法实现干涉检测的仿真系统可靠性更好。

表2 采用不同干涉检测方法的检测精度 mm

	过程一	过程二	过程三	平均精度
采用3.1所述方法的检测精度	1.574	1.603	1.552	1.576
采用3.2所述方法的检测精度	0.878	0.795	0.732	0.802
采用3.3所述方法的检测精度	0.878	0.795	0.732	0.802

4 应用

为了逐步缩小我国在自动钻铆技术与国外先进水平的差距,从技术经济综合效益考虑,某航空制造企业仅从国外引进了钻铆机。为配合钻铆机的使用,南京航空航天大学研制了数控托架,如图6所示。同时,为了提高钻铆的自动化水平,应用上述所探讨的干涉检测的方法,自主开发了四轴自动钻铆定位系统,系统操作界面如图7所示。



图6 自动钻铆系统

Fig.6 Automatic riveting system

为了满足企业对加工零件范围的要求,托架设计人员使用通用的运动仿真软件难以计算加工位于壁板边缘的铆钉所需要转动的角度以及Z向所需的行程。使用自动仿真软件可以满足以上要求,设计人员在设计过程中利用该系统对模型的不间断验证与编辑修改,有效地减少了设计失误率,并大大缩短设计周期,提高了设计质量。

编程人员可利用自动钻铆仿真系统生成数控代码,同时直观地验证加工程序的正确性,提前发现加工过程

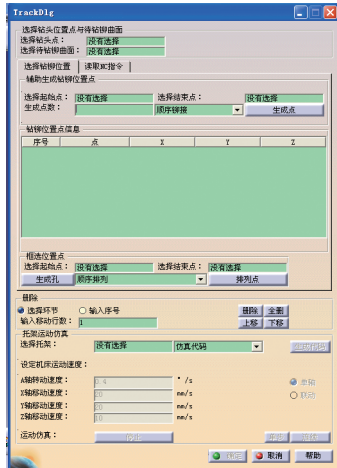


图7 四轴自动钻铆定位系统操作界面

Fig.7 Operating interface of four-axis automatic riveting allocation system

中部件之间的干涉,进而优化加工路径,提高设备的利用率。

自动钻铆仿真系统为工艺人员设计工装提供设计依据。以单曲度铆接壁板为例,由于壁板自身占用一定的行程,加工其边缘铆钉也需要Z向行程,然而受喉深(C形架深度)、托架Z向行程以及托架的旋转角度等因素的限制,如果工装设计不当,将会导致多次装夹,甚至发生不能够加工的状况。为达到一次装夹,即可加工整个壁板,工装的高度以及工件的安装位置需要满足一定的条件,工艺人员可以借助仿真系统确定加工壁板边缘铆钉需要的Z向行程以及旋转角度,进而合理地摆放工件,提高加工效率。

5 结论

随着虚拟现实技术的不断发展,加工仿真被越来越广泛地运用到航空航天制造等领域,一些从事自动钻铆技术研究及应用的企业和科研院所也在积极探索自动钻铆中干涉仿真技术的实现方法。但由于该技术处于起步阶段,国内可借鉴的资料和经验较少,同时国外专业从事钻铆仿真技术研究的企业也不多,并且对技术封锁非常严密,可利用的技术资源也十分有限。因此,本文对基于CATIA平台上自动钻铆中干涉检测技术进行研究,并成功应用于自主开发的自动钻铆仿真系统,得出以下研究成果:

(1)本课题克服种种技术难题,摸索出一套基于VC语言的CATIA仿真模块二次开发新方法,有效解决了目前国内广泛使用的基于VB语言开发方法所不可避免的兼容性差、运行不稳定、功能模块开发受限等问题。

(2)本课题基于多种干涉理论与开发方法,在CATIA仿真模块中实现了多种干涉检测方法,改变了

CATIA中干涉检测实现方法单一的现状。

(3)本课题是与某航空制造企业合作的横向课题,在实际加工前的仿真模拟中实现了干涉检测,提前发现可能存在的问题,及时解决,为企业提高了生产效率,节约了劳动成本。

(4)本课题将虚拟仿真技术与自动钻铆加工紧密融合,并成功应用于大型壁板件的钻铆加工,为后续更为复杂的钻铆仿真技术垫底了一定的基础。

参考文献

- [1] 李少波,陈翔鹏.自动钻铆技术的应用和无头铆钉安装.航空制造技术,2007(9):50-52.
- [2] 卜泳,许国康,肖庆东.飞机结构件的自动化精密制孔技术.航空制造技术,2007(24):61-64.
- [3] 王仲奇,殷俊清,康永刚.飞机自动钻铆仿真技术.航空制造技术,2008(20):44-47.
- [4] 纪玉坤,曹利新.基于UG的五轴数控机床加工仿真.计算机仿真,2006,23(5):215-218.
- [5] 余斌,刘荣忠.基于OpenGL的数控加工仿真系统研究.四川大学学报:工程科学版,2001,33(4):16-19.
- [6] 冯潼能,王黎明,初爱国.基于CATIA V5的自动钻铆机工艺过程仿真.航空制造技术,2009(21):45-47.
- [7] 胡挺,吴立军.CATIA二次开发技术基础.北京:电子工业出版社,2006.
- [8] 董伊鑫,席平.基于CATIA二次开发的五坐标数控机床自动仿真系统.机械工程师,2005(3):41-43.
- [9] 何朝良,杜廷娜,张超.基于CAA的CATIA二次开发初探.自动化技术与应用,2006,25(9):37-40.
- [10] 黄娟,顾寄南.装配仿真中碰撞干涉检查研究的综述.江苏大学学报,2002,23(2):17-21.
- [11] Moore M, Wilhelms J. Collision detection and response for computer animation. Computer Graphics, 1988, 22(4):289-298.
- [12] 曾柳杨,刘毅.基于CATIA/CAA的虚拟装配约束关系分析与干涉检查研究[D].南京:南京航空航天大学,2005.

(责编 夏宛)

(上接第83页)

尼龙搭扣一边缝纫在隔音层上,另一边用胶粘剂粘贴在蜂窝板背面,这种连接形式既有利于操作,也便于飞机隔音层的维护和更换。

4 结束语

经过详细的工艺审查使内装饰的设计具有良好的工艺可行性,在零件生产前进行了多次的工艺试验和分析研究,明确了内装饰制造的关键点和难点,并从材料、工装、成型工艺及参数、装配、装饰层的包敷等多个工艺环节采取了许多工艺措施,保证了零件制造的质量。某机型内装饰是首次加工制造的隔音软性内装饰,它的成功研制提高了我公司内装饰零件的加工制造水平,为以后新型飞机内装饰的研制积累了丰富的经验。

(责编 深蓝)