

基于 CATIA 二次开发的飞机质量特性批量提取方法

周建斌

(上海飞机设计研究院结构设计研究部,上海 200232)

Method for Batches of Weight Status Extraction
Based on Re-developing Technology of CATIA

Zhou Jianbin

(Structure Department of SADRI, Shanghai 200235, China)

摘要:质量特性数据在飞机研制过程中经常需要更新。通常情况下,质量特性的统计工作往往需要通过大量的人力来完成。分析了探求飞机质量特性批量提取方法的必要性及对此方法的要求,对实现批量提取的两种策略进行了比较,给出了实现批量提取的若干关键代码。通过对 CATIA 的二次开发,实现了飞机零件质量特性的批量提取及数据表的自动填写,减少重复性的体力劳动,提高统计工作的准确性。

关键词: CATIA;二次开发;Automation;质量特性;批量;递归

[Abstract] Weight status needs regular updating in the development of aircraft. Generally speaking, the updating work wastes a lot of manpower. Necessity of developing a method for batches of weight status extraction is analyzed. Comparison of tow means is completed. And pivotal code is provided. Based on re-developing technology of CATIA, batches of weight status extraction and automatic filling of datasheet are developed to reduce repeated human work and improve the veracity.

[Key words] CATIA; re-developing; automation; weight status; batches; recursion

0 引言

在飞机设计过程中需要不断地统计设备级和飞机级的质量特性(重量、重心、转动惯量),以进行飞机性能的分析。尽管 CATIA 已带有批量输出装配件中零件质量特性的功能,但却有如下缺点:①其输出结果为 TXT 格式,不便于编辑;②对于未赋材料属性的零件(若无主制造商的明确要求,供应商通常不会主动为数模赋材料属性;即便是自制件,若建模规范未要求对零件赋予材料属性或要求了却未严格执行,也必将出现大量无材料属性的数模),CATIA 在测量它们的质量特性时,默认使用水的密度,得到的结果与实际情况存在巨大偏差,而若由设计人员手工将材料属性全部补上,工作量又过于庞大。基于以上原因,对 CATIA 进行二次开发,实现能够灵活处理有材料属性和无材料属性的零件的飞机质量特性批量提取方法,在工程中是非常有用的。

结合工程实践,本文对质量特性批量提取方法提出几点基本要求:①应是基于全机坐标系进行质量特性的提取;②质量特性包括重量、重心、转动惯量($I_x, I_y, I_z, I_{xy}, I_{yz}, I_{zx}$);③应既能处理有材料属性的零件,又能处理无材料属性的零件。工程实践中关于质量特性还有众多其他要求,但与 CATIA 关系甚微,因此本文不做讨论。

1 CATIA 二次开发方式

CATIA 的二次开发主要有三种方式,第一种是

编写“宏(Macro)”。CATIA 提供了 VBScript 与 CATIAScript 两种语言来进行宏的编写,但宏能使用的人机接口命令只有 MsgBox(通过弹出对话框来传达信息)与 InputBox(通过弹出对话框来接收用户的输入),功能有限,而且 CATIA 提供的宏编辑工具没有自动提示功能,对于大部分人来说,不是一个理想的开发工具。

第二种是使用 CAA。CAA 是达索公司提供的 CATIA 专业二次开发工具,需要 VC 环境,制作的功能模块可以跟 CATIA 自带的命令相媲美。但 CAA 是需付费的。

第三种是使用 Automation 技术。通过 VB、VC 等高级编程语言,利用 Automation 技术来调用 CATIA,执行 CATIA 提供的众多命令。这种二次开发方法有如下的优点:①完全免费;②高级编程语言的 IDE 有较强的自动提示功能,降低开发难度;③利用高级编程语言可以制作出功能丰富的人机界面,弥补“宏”的不足。对于本文欲解决的质量特性批量提取问题来说,使用 Automation 技术最为合适。

因此本文将使用第三种二次开发方法。开发工具为 Visual Studio 2008 Express。

2 程序设计描述

2.1 思路

利用 CATIA 提供的 Inertia 类,可以方便地得到零件与装配件的质量特性。但尚需权衡两种策略:

第一种,由程序搜索出三维数模文件夹及其子文件夹下的所有零件,逐个提取质量特性并输出至数据表;第二种,由用户打开某专业最顶层的装配件数模,用“质量特性提取函数” Extract() 处理该装配件,逐一判断装配件下一级对象是否是装配件,若否,则提取其质量特性并输出至数据表;若是,则递归调用 Extract() 处理该对象。

这两种策略都只能得到零件级的质量特性,最后都需要用户手工或编程计算出装配件级的质量特性。原因在 2.8 节中阐述。

第一种策略有十分严重的缺点。如果用户没有使用全机坐标系来创建零件,那么测量到的重心、转动惯量就是局部坐标系下的,就无法满足质量特性数据的相关要求。而且除非主制造商有明确的要求,否则供应商提供的成品件数模很有可能是在非全机坐标系下创建的。因此,第一种策略的通用性很差。

第二种策略的算法略微复杂,但其通用性要远远强于前者。由于专业级的顶层数模肯定是位于全机坐标系下的(否则专业间根本无法进行协调),因此使用 2.1 节中的第二种策略可以确保提取出的质量特性是位于全机坐标系下的,以满足引言中所提的要求。同时通过使用递归算法,即使顶层装配件还有第 2、3、4……级零部件,也同样可以做到全部提取。所以本文将使用该策略来完成批量提取飞机零件质量特性的工作。

2.2 获取零件质量特性的典型代码

CATIA 提供的 Inertia 类是获取质量特性的关键,Inertia 类的 Mass 属性中保存着被测量对象的重重(单位 kg), Density 属性中保存着密度(单位 kg/m^3), Volume 属性中则保存着体积(单位 m^3)。使用 Inertia 类的 GetCOGPosition 方法可以得到重心(单位 m),使用 GetInertiaMatrix 方法可以得到转动惯量(单位 $\text{kg} \cdot \text{m}^2$)。

下面是提取 CATIA 中已打开的且处于激活状态的装配件的下一级第 i 个对象(暂且假设它是零件,若是装配件,则需采用递归算法进行处理,这将在 2.6 节中阐述)的质量特性的代码。

```
Dim CATIA As Object
Dim oProductDoc As ProductDocument
Dim oRootProduct As Product
Dim oRootChildren As Products
‘获取 CATIA 对象
Try
```

```
CATIA = GetObject(, “CATIA. Application”)
Catch ex As Exception
MsgBox(“请手动打开 CATIA”)
Exit Sub
End Try
Try
oProductDoc = CATIA.ActiveDocument
Catch ex As Exception
Err.Clear()
MsgBox(“请打开一个 Product”)
Exit Sub
End Try
oRootProduct = oProductDoc.Product
oRootChildren = oRootProduct.Products
‘使用 Inertia 类进行测量
Dim oInertia As Inertia
Dim Mass As String
Dim oCoord(2), oMatrix(8)
Dim Ix, Iy, Iz, Ixy, Iyz, Izx As String
‘测量第 i 个零件的质量特性
oInertia = oRootChildren.Item(i). _
GetTechnologicalObject(“Inertia”)
‘得到重量
Mass = CStr(oInertia.Mass)
‘得到重心
oInertia.GetCOGPosition(oCoord)
Dim xcg As String = CStr(oCoord(0) * 1000)
Dim ycg As String = CStr(oCoord(1) * 1000)
Dim zcg As String = CStr(oCoord(2) * 1000)
‘得到转动惯量
oInertia.GetInertiaMatrix(oMatrix)
Ix = CStr(oMatrix(0)) \ Ix
Iy = CStr(oMatrix(4)) \ Iy
Iz = CStr(oMatrix(8)) \ Iz
Ixy = CStr(oMatrix(1)) \ Ixy
Iyz = CStr(oMatrix(5)) \ Iyz
Izx = CStr(oMatrix(2)) \ Izx
```

使用 Try...Catch ex As Exception...End Try 语句可以大大提高程序的容错性,作为用户来说,当然会乐于使用健壮的程序。

2.3 对无材料属性的零件的处理

若使用 Inertia 类对不含材料属性的零件进行测量,会发现 CATIA 使用的是水的密度 $1\ 000\text{kg}/\text{m}^3$,因此需要一些特殊处理。

首先有必要说明 CATIA 的 Inertia 类的工作机制。Inertia 类的各项属性中,若更新 Density 属性则会引起其他各项质量特性数据随之更新。因此对于无材料属性的零件,可以有两种处理方式:(1)由用户提供零件密度,程序对 Inertia 类的 Density 属性进行更新,然后便可以得到更新后的各项质量特性数据,此法适用于密度明确的零件;(2)由用户提供重量,程序计算出密度值,并更新 Density 属性,然后同样可以得到更新后的各项质量特性数据,此法适用于密度不明的零件,通常是标准件、成品件。

笔者使用图 1 中的对话框由用户输入密度或重量。

程序还可将零件件号与其密度值(若用户提供的是重量,则需计算出密度值)以结构体的形式保存在一个数组(图 2 所示的流程图中称之为“历史数据”)中,当再次遇到同一零件时,则从数组中调取,而无需用户重复输入。

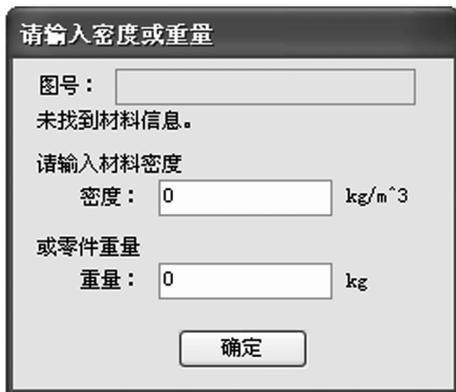


图 1 输入密度或重量

2.4 对参考零件的处理

一个装配件中难免会含有参考数模,比如一位系统专业的设计员肯定会在他的数模中使用结构专业的数模作为参考,但在获取质量特性时,他并不需要机体结构的质量特性,而只需要获取本专业负责的零部件的质量特性就足够了。在此情况下,用户可以将其隐藏,程序便会忽略它,不提取它的质量特性。通过 CatVisPropertyShow 变量与 Selection 对象的搭配使用可以判断装配件下一级的第 i 个对象(即下文代码中的 oRootChildren. Item(i))是否被隐藏。

‘Selection 对象是一个集合

‘在 CATIA 中选中的对象均被纳入其中

```
Dim sel As Selection
```

```
sel = CATIA.ActiveDocument.Selection
```

```
sel.Clear()
```

```
sel.Add(oRootChildren.Item(i))
```

Dim attr As CatVisPropertyShow ‘CatVisPropertyShow 是一个用于表明隐藏或

’显示状态的变量类型

```
sel.VisProperties.GetShow(attr)
```

```
If attr = CatVisPropertyShow. _
```

```
catVisPropertyNoShowAttr Then
```

‘此处插入对隐藏对象的处理代码

```
End If
```

2.5 EXCEL 表格的填写代码

获得零件的质量特性后,还需填入 EXCEL 数据表。此处以填写重量(保存于变量 Mass 中)的代码为例,假设当前要填写的单元格是“C2”。

```
Dim objExcelFile As Excel.Application
```

```
Dim objWorkBook As Excel.Workbook
```

```
Dim objImportSheet As Excel.Worksheet
```

‘创建 EXCEL 对象

```
objExcelFile = New Excel.Application
```

‘打开并显示数据表

```
objWorkBook = objExcelFile.Workbooks. _  
Open(strExcelFileToFillPath)
```

```
objImportSheet = objWorkBook.Sheets(1)
```

‘填写重量

‘Mass 中保存了重量值

```
objImportSheet.Range("C2").Value = Mass
```

其他质量特性数据的填写操作与之类似。笔者建议有材料属性的零件与无材料属性的零件的数据单元格应赋予不同的背景色,以便于区分。设置单元格的背景色可使用形如 Range(“A2”).Interior.ColorIndex=22 的代码来实现,其中“22”为颜色序列号,“22”为砖红色。

至此,再配合 For 循环语句,遍历装配件下属所有对象,已完全可以实现对不含组件的装配件下属零件质量特性的批量提取。若装配件中还含有组件,则需要用到递归算法。

2.6 递归算法

采用递归算法的目的是为实现“一次打开,全部提取”,只需要打开最顶层的装配件,就可以把其中包含的所有零件的质量特性全部提取出来,做到真正的批量提取。

将提取装配件中零件质量特性的函数命名为 Extract(),称之为“批量提取函数”。该函数会遍历装配件下一级的对象,若为零件则输出其质量特性,若为组件则调用自身去提取该组件下属零件质量特

性,以此来形成递归。

2.7 批量提取函数的程序流程图

在前述的基础之上,给出批量提取函数 Extract () 的流程图,见图 2,其中以“Product”表示将被提取零件质量特性的装配件,“Product(i)”表示“Product”下一级第 i 个对象。

结合读者所在企业的企业规则,此流程图或许会需要调整,而且通过编写代码还可以对已填好的质量特性数据表进行更多的操作,如计算装配件级的质量特性、对零件进行排序等。

2.8 后续处理

如果装配件中含有无材料属性的零件,那么直

接测量装配件得到的质量特性肯定是错误的。但又由于装配件中通常含有多种密度不同的零件,因此通过更新装配件的密度来提取其质量特性的做法也会带来较大的误差。由此可见,要提取装配件质量特性不能简单地套用提取零件质量特性的方法。为了得到准确的装配件质量特性,可以使用以下两种方法:(1)完善下属所有零件的材料属性,删除参考参数模,再测量装配件;(2)采用本文提供的方法编制软件得到零件级的质量特性数据表,再计算出装配件的质量特性。显然后者所花的精力要少得多,并且由零件质量特性计算出装配件质量特性的工作亦可通过编程实现自动化。

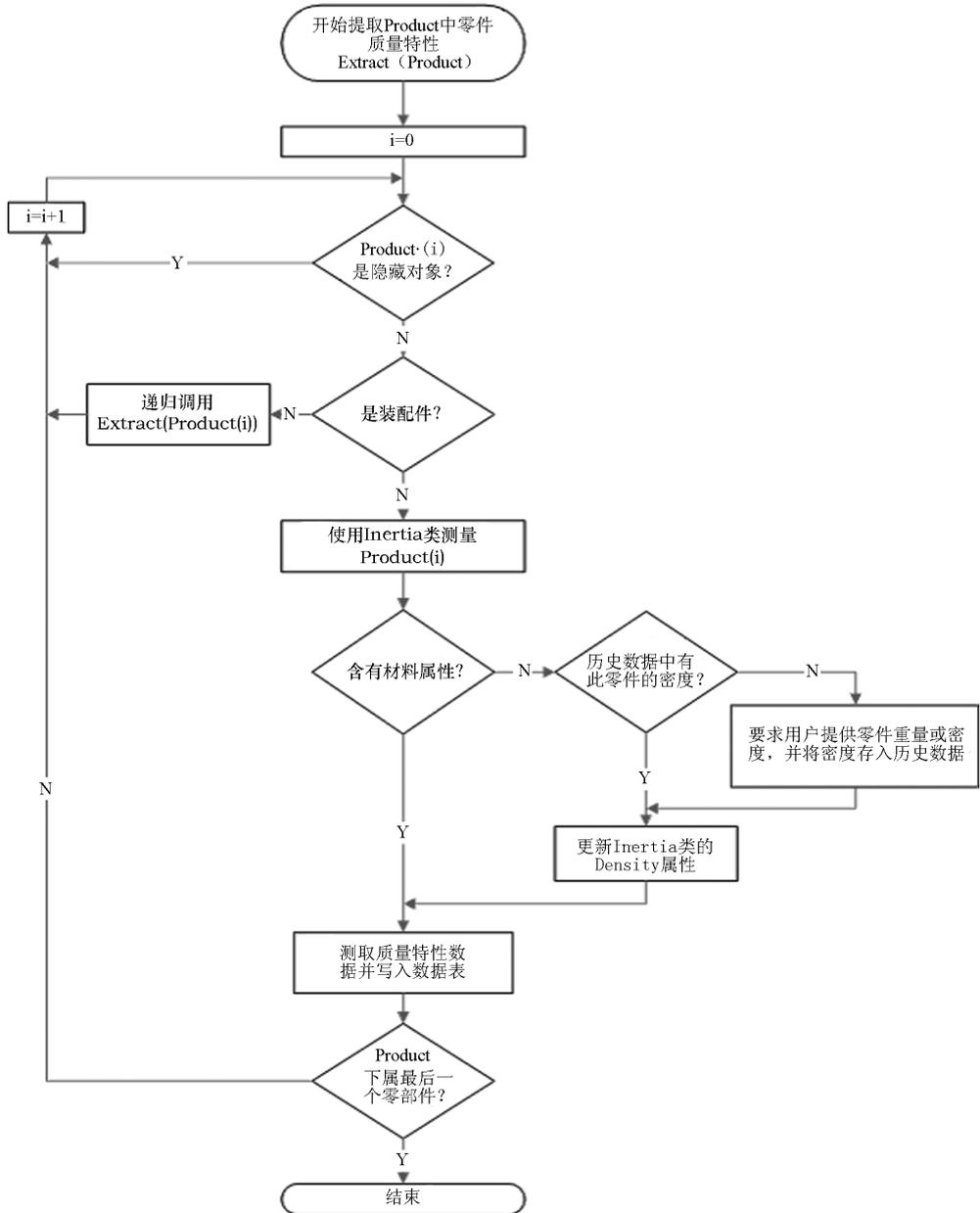


图 2 批量提取函数 Extract () 流程图

