文章编号:1671-4598(2020)08-0246-04 DOI:10.16526/j.cnki.11—4762/tp.2020.08.050 中图分类号:TP273 文献标识码:A

WPF/STK 集成的卫星轨道引导文件自动生成方法

薛乃阳[1]，丁 丹[2]，刘步花[1]
（1. 航天工程大学 研究生院，北京 101416；
2. 航天工程大学 电子与光学工程系，北京 101416）

摘要：针对获得轨道引导文件的传统方法比较繁琐和复杂的问题，详细介绍了采用 WPF（Windows 呈现基础）与 STK（卫星工具箱）软件集成的方法；利用 VisualStudio（可视化工作室）搭建仿真环境，以自动产生某卫星轨道引导文件的程序为例，具体介绍了集成开发的步骤流程及遇到的问题和解决方法；结果表明，利用 WPF 和 STK 进行系统集成，不仅能够获取符合任务要求的轨道引导文件，而且大大减小了 WPF 的编程工作量，同时通过个性化主界面设计，以可视化的形式对主程序集成，只需输入参数，便可快速获取相应的引导文件，避免了获取引导文件时需反复对 STK 参数进行修改；在测控任务开始前获取轨道引导文件的操作中，WPF/STK 联合仿真相较于仅使用 STK 软件进行处理而言，要更加简洁高效。

关键词：WPF 应用；STK；集成仿真；STK 对象模型；轨道引导文件

WPF/STK Integrated Satellite Orbit Guidance File Automatic Generation Method

Xue Naiyang[1]，Ding Dan[2]，Liu Buhua[1]
（1. Graduate School，Space Engineering University，Beijing 101416，China；
2. Department of Electronic and Optical Engineering，Space Engineering University，Beijing 101416，China）

Abstract：Aiming at the tedious and complicated traditional method of obtaining the orbit guidance file，this paper introduces the method of integrating WPF（windows presentation foundation）and STK（satellite tool kit）. Using Visual Studio to build simulation environment，the procedure of automatically generating a satellite orbit guidance fileas as an example，specifically introduces the steps of integrated development and the problems as well as solutions encountered. The results show that using WPF and STK for system integration can not only obtain track guide files that meet the task requirements，but also greatly reduce the programming workload of WPF. At the same time，through the personalized main interface design，the main program is integrated in a visual form. In this way，the corresponding boot file can be quickly obtained，avoiding to repeatedly modify the STK parameters. In the operation of obtaining the track boot file before the telemetry telecontrol & communication（TT&C）task，the WPF / STK joint simulation is more concise and efficient than STK software.

Keywords：WPF applications；STK；integrated simulation；STK Object Model；orbit guidance file

0 引言

WPF（windows presentation foundation）是微软新一代的图形系统，是专门用来编写程序表示层的技术和工具[1]，它为用户界面、2D/3D 图形、文档和媒体提供统一的呈现和操作方式[2]。WPF 引入一种全新的描述性标记语言 XAML（extensible application markup language）来定义应用程序的静态结构，后台逻辑则可使用 C#、C++等语言来完成。这样做既可以更好地协同界面设计与程序设计，又将界面显示与应用程序背后的逻辑解藕开来，降低了开发与维护的成本[3]。

获取轨道引导文件最常用的仿真软件是 STK（卫星工具箱），它是美国 AGI 公司开发的军事领域领先的系统分析工具。该软件可以应用于复杂海陆空天任务的分析，其强大的数据分析计算能力和良好的二三维展示功能帮助用户更好的理解场景，加上模型精确可靠、开发接口丰富，在国内外运用广泛[4]。由于 STK 中内置了很多的坐标系类型和轨道摄动外推算法，可以通过这些复杂的算法确定任意坐标系下任意时刻的卫星时基信息[5]。航天测控系统是航天系统的重要组成部分之一。在每次测控任务开始之前，测控站需要计算出目标航天器在相对于测站的位置坐标和速度信息（即卫星轨道引导文件），以保证后续任务能够顺利展开。目前获得引导文件的方法主要是单一操作 STK 软

收稿日期:2020-02-11； **修回日期**:2020-03-16。
基金项目:国家高技术研究发展计划("863"计划)基金资助项目(2015AA7026085)。
作者简介:薛乃阳(1997-),男,安徽滁州人,硕士研究生,主要从事航天测控方向的研究。
丁 丹(1980-),男,江苏南京人,副研究员,硕士研究生导师,主要从事航天测控方向的研究。
通讯作者:刘步花(1994-),女,重庆梁平人,硕士研究生,主要从事航天测控方向的研究。

件生成轨道/弹道星历数据[6]，其具体流程如图 1 所示。

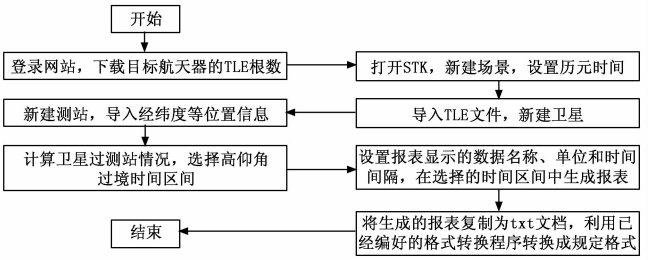

图 1　STK 软件获取引导文件的流程

虽然 STK 有效解决了轨道引导文件的获取问题，但由图 1 可知，如果在测控任务中更换目标卫星，就要停止仿真，重新按照流程逐一设定相关参数；此外，从 STK 上直接获得的报表格式与规定格式有差异，还需手动更改数据格式，才得到正确的文件。本文结合 WPF 和 STK 两个软件各自的优点，通过设置 WPF 的主程序界面及其后台逻辑，用 C＃指令控制 STK 的运行并进行文件格式处理，使仿真系统按照 WPF 预先设定的程序运行，从而简化了操作流程，提高了获取轨道预报文件的速度和正确率。本文以实际应用为背景，结合卫星轨道预报文件的自动获取问题，对其分析过程以及相应的编程给出了详细的阐述和说明。

1　WPF 和 STK 集成的途径

STK 支持多种开发任务的基础是 STK 以一系列控件、注册 COM 组件和类库的方式对外提供服务。其二次开发的方法主要有 3 种：Connect 模块、STK 对象模型（STKObjectModel）和应用程序对象模型[8]。在方式一中，用户只需调用相应的函数即可实现与 STK 之间的通讯；方式二灵活性强，集成度高，利用 STK 提供的开发包可以独立的开发应用程序。在现有研究中，方式一最多，但涉及 STKObjectMode 的研究极少。本文主要使用 STKObjectMode 进行 WPF 和 STK 的集成开发。在 VisualStudio 2017 中实现了 STK11.2 与 WPF 的集成，并实现了轨道预报文件的自动化获取。

1.1　STK 对象模型

STK 对象模型是一组 COM 类库，包括 STK Objects，STK X 等组件。这些组件分别用于控制 STK 实体，管理实体生命周期，获取数据，接近分析和覆盖计算，事件响应等。此外，STK 对象模型是基于 COM 技术构建的，可用支持 COM 的 C＃、C＋＋、Java 等语言开发，本文采用 C＃编程。

1.2　配置 WPF/STK 集成开发环境

在 WPF 中添加 STK 的程序集引用，配置 WPF/STK 集成开发环境，是二者集成开发的基础。首先，打开 Visual Studio 2017 软件，新建 WPF 工程项目，选择 .NET FrameWork 4 作为工程的框架。其次，添加程序集引用，在解决方案管理器中，右击“引用”下“添加引用”，打开引用管理器；在对话框中点击“程序集”，勾选“System. Windows. Forms”“WindowsFormsInt－egration”；点击“程序集”下面的“com”库，勾选“AGI STK Objects 11”、“AGI STK X 11”。最后，点击“com”库下面的“浏览”，依次选择路径“STK 安装目录”－＞“bin”－＞“Primary Interop Assemblies”，勾选程序集“AxAGI. STKX. Interop. dll”。注意，如果 STK 二次开发项目涉及较多程序集的话，应该添加其他相应的程序集，比如如果涉及到 STK 桌面软件相关操作，则应添加“AGI UI Application 11”等程序集。

1.3　STK 二次开发命令集

C＃的二次开发命令集（C＃ Code Snippets）涵盖了 STK 可以实现的大部分功能，对使用 STK 软件开发包进行自有应用程序开发有较大参考价值。存放位置在 Help 文档中的 integration. chm－＞UsingCoreLibraries－＞STKObjectModel－＞C＃CodeSnippets。用户在进行二次开发时，编写的主程序与 STK 之间实现信息交互需要使用这些命令并严格遵循其命令格式[9]。

2　问题提出与总体方案设计

配置过集成开发环境后，便可进行下一步的 WPF/STK 集成仿真应用。下面以自动获取轨道预报文件为例，详细介绍使用 WPF/STK 集成仿真方式获取卫星轨道数据等参数并进行处理的过程。

2.1　问题陈述与限定条件

1）问题陈述：设定测站的位置为北纬 40.3°，经度 116.23°，计算并选择卫星高仰角过境时间段；在选择的时间段内以 1 s 为间隔，生成卫星相对于测站坐标系的位置和速度坐标，创建轨道预报文件。

2）限定条件：卫星的轨道信息由插入的 TLE 轨道根数确定，格式要求轨道引导文件时间要转化为为北京时间（UTCG 时间加 8 小时），并且年、月、日和时、分、秒分别用八位和十位的数字来表示，表示位置的 x、y、z 坐标保留三位小数，表示速度的 x、y、z 分量保留六位小数，并在每一行都显示出卫星的名称。

2.2　总体方案流程设计

1）WPF 与 STK 集成并调试；

2）设计 WPF 主程序界面。根据任务需求，用 XAML 语言添加并定义相应的控件；

3）在后台逻辑中进行具体场景的设定与编程。用 C＃语言设置场景时间、新建卫星与测站、获取过境报表、获取轨道参数、更改报表格式及生成并保存轨道预报文件等。在主程序界面中通过导入卫星 TLE 文件，输入任务大致起始时间，先获取卫星过境报表并显示在主程序界面上；在界面上选择卫星高仰角过境时间段，并将选择的小范围起始时间段输入界面；点击相应的按钮，便可自动获取卫星轨道预报文件，任务场景的可视化仿真也可以在主程序界面中显示。总体的流程图如图 2 所示。

3　仿真程序设计

程序主要由两大部分组成，根据 WPF 设计的特点，可

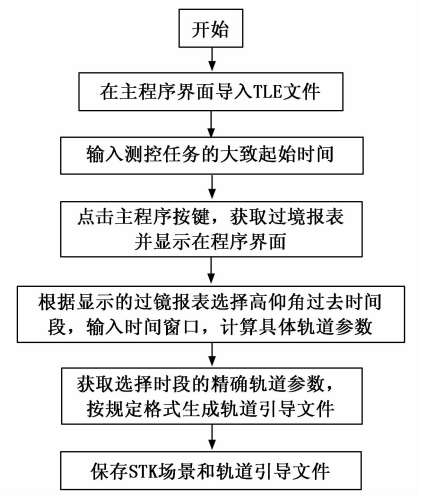

图 2 集成程序流程

分为用 XAML 语言定义的主界面程序和用 C＃编写的后台逻辑程序。

主界面程序的主要功能是获取和显示输入参数，用户点击对应的按钮控件并执行关联的后台逻辑程序，经运算程序运算后，主界面上会显示仿真 2D/3D 图像，并将最终的轨道引导文件保存到设定位置。

3.1 主界面程序设计及说明

主界面程序需要实现的功能是设置并定义控件信息、读取输入参数并传给后台逻辑程序。为了显示 STK 的可视化界面，需要利用 STKX 中的 AxAgUiAxVOCntrl 与 AxAgUiAx2DCntrl 类库，它们分别提供二三维可视化界面显示的功能。

配置好 WPF 与 STK 集成开发环境后，进行主程序界面设计的关键 XAML 语句如下：

1)在其命名空间引用中添加 STK 程序集，并命名为“stkLian”，程序语句为 xmlns:stkLian＝"clr－namespace:AxAGI.STKX;assembly＝AxAGI.STKX.Interop"。

2)在主程序窗口中添加 STK3D 视图窗口的控件，设置其显示在 WPF 中的 WindowsFormsHost 模板控件上。程序语句为＜WindowsFormsHost＞＜stkLian:AxAgUiAxVOCntrl＞＜/stkLian:AxAgUiAxVOCntrl＞

＜/WindowsFormsHost＞

3)在主程序窗口中添加 STK2D 视图窗口的控件。程序语句为＜WindowsFormsHost＞

＜ stkLian: AxAgUiAx2DCntrl ＞ ＜/stkLian: AxAgUiAx2DCntrl＞＜/WindowsFormsHost＞

4)设置主程序界面上的按钮和文本框等控件，并为其添加事件处理程序，与后台逻辑代码相对应，设置控件代码在此不赘述。

3.2 后台逻辑程序设计及说明

后台逻辑程序就是用 C＃语言给添加的 STK 程序集发送指令，编写在主程序中添加的各个事件处理程序，使其分别完成场景创建、测站创建、卫星创建、过境计算等任务，之后进行报表参数设置、存储并输出相应数据并转换格式，最后生成轨道引导文件。根据实际问题需要，相应的 C＃控制语句可以在上文介绍的 C＃开发命令集查询，语句如下：

1）在窗体的类文件头部加入命名空间引用，代码为：using AGI.STKObjects。

2）对于每一个 STK 应用程序，有唯一的根节点 AgStkObjectRoot 对象[8]，可通过此对象控制 STK 场景。此外，这是唯一可以直接创建的对象，其余对象模型必须间接获得。代码语句为：

```
AGI.STKObjects.AgStkObjectRoot root; public MainWindow()
{InitializeComponent();
root = new AGI.STKObjects.AgStkObjectRoot();}
```

3）编写主程序窗口 xxxx 按键的时间处理器程序，程序语句示例为：

```
private void xxxx_Click(object sender, RoutedEventArgs e)
{//根据具体需求编写调用信息处理函数}
```

下面给出主要步骤在开发命令集中的名称及含义：

a. 创建场景，并设置场景时间，代码名称为 Create a new scenario；

b. 创建测站，设置其经纬度，语句名称为 Create a facility on Earth at lat/lon/alt；

c. 新建卫星，代码名称为 Create a satellite (on current scenario central body)；

d. 对卫星设置 SGP4 轨道预报器，代码名称为 Set the satellite to use the SGP4 propagator；

e. 用 TLE 轨道根数定义卫星运行轨道，代码名称为 Configure the SGP4 propagator with file source，示例程序中的“2215”代表卫星的 SSC 序列号(TLE 文件中航天器 SSC 目录编号)，tleFilePath 代表 TLE 文件在电脑中的位置；

f. 计算卫星过境情况并获取报表，代码名称为 Configure the SGP4 propagator with file source；

g. 保存场景，代码名称为 Save a scenario。

4 发现的问题归纳及解决

4.1 注释的分类及含义

在 C＃开发命令集中，表头是代码名称，注释在程序前，并用“//”表明，如图 3 所示。这些注释分为两种，第一种是说明在使用这段代码前需要声明的变量和需要在命名空间中引用的命令集，第二类注释主要用来说明本段代码的功能。通过注释内容和在文件中的位置可以判断注释类型，如图 3（a）所示，第一类注释在上侧，两类注释之间用两行隔开；若文件中无第一类注释，第二类注释会与表头空两行，如图 3（c）所示。

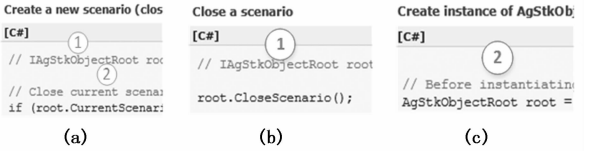

图 3 开发程序集的内容及注释分类

分清注释类型，读懂其含义是编写正确代码的必要条

件。比如，在创建测站等场景时的第一类注释为“//IAgStkObject－Root root：STK Object Model root”，表明这是通过根节点 AgStkObjectRoot 提供的方法和属性加载的对象，需要在前文声明好根节点。此外，利用 TLE 创建卫星时步骤如图 4 所示，仿真卫星轨道需使用“SGP4”预报器。“SGP4”（简化的通用摄动预报器）是美军空间司令部的标准预报器，需要与 TLE 双轨道根数一起使用[10]。在 STK 11.2 设置 SGP4 预报器的示例代码段中，第一种注释为“// IAgSatellite satellite：Satellite object”，表明写入这段代码之前先要创建卫星对象。

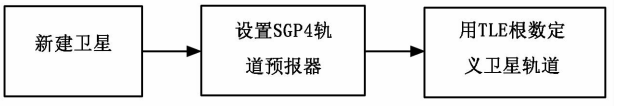

图 4　插入 TLE 轨道根数卫星的流程

4.2　引导文件内容设置程序要点

在 STK 中，计算和轨道预报的功能集中在“Report”中，可以点击其属性按钮为报告定义时间段、设置报告内容、设置单位和输出数据。在示例代码段中，定义时间段和设置单位的代码比较简单，直接套用即可，在此不赘述。下面着重介绍设置报告内容和输出数据的代码，表 1 给出了所需要数据在 STK 中的存储位置和层数。

表 1　数据名称位置表

数据名称	数据类别在 DataProviders 位置	层数
卫星过境时间和高度角	Constraint Data－＞Time（或 FromElevationAngle）	2
卫星相对测站的时空坐标	Vectors(Fixed)－＞Position－＞Time，x，y，z	3
卫星相对于测站的速度分量	Vectors(Fixed)－＞Velocity－＞ x，y，z	3

由表 1 可以看出，设置“Report”报告内容是在“DataProviders”选择的，用户需要通过子对象集合层层向下获取所需要的对象。但是根据所需对象的位置所处的层数，可以分为两层和三层结构。以获取卫星过测站的时间和高度角为例，获取两层结构的数据代码如下：

```
Array dataPrvElements = new object[]{ "Time", "FromElevationAngle"};
IAgDataPrvTimeVardp = access.DataProviders[ "Constraint Data"] as IAgDataPrvTimeVar;//选择 ConstraintData
```

以获取某时段卫星相对于测站的时间和位置坐标为例，三层结构的程序如下：

```
Array elems = new object []
{ " Time", " x", " y", " z"};
IAgDataProviderGroupdpVectorChoose = sat1.DataProviders[" Vectors (Fixed)"] as IAgDataProviderGroup; //选择 Vectors (Fixed)
IAgDataProvidergroupPosition = dpVectorChoose.Group [ " Position"] as IAgDataProvider; //选择 Position
```

4.3　生成初始报表

C＃开发命令集中的示例程序没有把获取的数据写入 txt 文档的功能，可以通过添加用 System.IO 中 Stream 类的语句来修改原代码，实现数据的读写操作。首先，在后台逻辑程序的命名空间中添加引用“usingSystem.IO”；其次，使用 IO 大类中的的 StreamWriter 类和 StreamReader 类来实现对文本文档的读写功能，具体代码不赘述；最后，把示例代码中的“WriteLine”全部改为“sw.WriteLine”，就可以实现存入 txt 文档的目的。

4.4　更改引导文件格式

对照上文对引导文件的限定条件，更改初始报表格式。首先，在资源管理器中添加一个储存轨道预报表数据的 Code 类，右击项目名称－＞“添加”－＞“新建项”，打开添加新项，点击“VisualC＃”下面的“Code”－＞“类”，并把新建的这个类命名为“SatelliteData.cs”，在这个类下面按显示项目的类型分为八类，分别从初始的轨道预报文件中存入到相应的类中。月份由英文简写转为两位数字可以通过编写“Dictionary”类来解决，将月份和对应的简写存入字典类，在后面的转化中通过编写查阅此类代码来完成转换。

5　结果与分析

5.1　仿真过程与图像

确定测控任务大致时间段为 2020 年 3 月 12 日至 15 日，卫星名称为“Satellite1”，导入相应的 TLE 文件。主程序界面输入的仿真结果如图 5 所示。

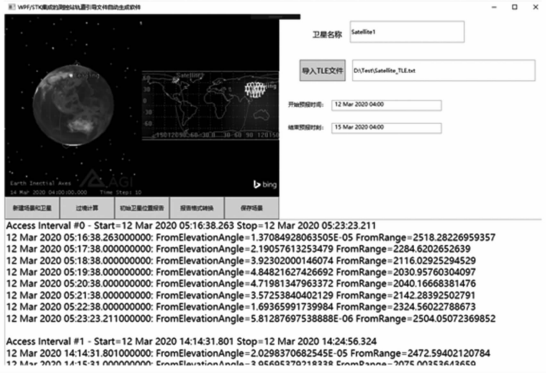

图 5　主程序界面输入参数

5.2　仿真结果分析

在主程序界面下方的卫星过境报表中选择高仰角过境时段。在界面显示的过境报表中，表示卫星过境仰角的数据名称为“FromElevationAngle”。通过与其他时段的比较可得，在卫星第 14 次过境的时间窗口中可出现最大高度角，为 68.17°；因此选择这一点附近的时间段：3 月 15 日 03 时 44 分 13 秒至 03 时 50 分 13 秒，再次输入时间窗口，依次点击其余的按钮控件，得到最终的轨道引导文件表头及部分数据信息如图 6 所示。

（下转第 264 页）