

编号

阶段标记

密级

公开

文件类型

版本

V2.3 Alpha

页 数

文件名称 航天任务设计工具箱 ATK

二次开发使用手册

编制单位

国防科技大学空天科学学院

编 写

ATK 项目组

校 对

张梦樱

审 核

杨 震

批 准

国防科技大学空天科学学院

修订记录

序号	章节名称	修订内容简述	修订日期	修订前版本号	修订后版本号	修订人	复核人
1		文档初稿	2022.12.30	无	V1.0		
2		删减章节	2023.04.10	V1.0	V1.1		
3		卫星集群对象、机动规划 LambertTarget 段、RPO 段	2024.03.29	V1.1	V2.1		
4		报告 可视化参数	2024.04.25	V2.1	V2.1.0. 1		
5		二三维、工具	2024.06.28	V2.3	V2.3.0. 0		
6							
7							
8							
9							
10							

目 录

1. ATK.Connect 功能及使用介绍	4
1.1. C++客户端使用介绍.....	4
1.1.1. atkOpen	5
1.1.2. atkConnect	5
1.1.3. atkClose	5
1.1.4. 案例操作流程.....	5
1.1.5. 案例展示.....	16
1.2. Matlab 客户端使用介绍	25
1.2.1. atkOpen	26
1.2.2. atkConnect	26
1.2.3. atkClose	26
1.2.4. 案例操作流程.....	26
1.2.5. 案例展示.....	35
1.3. Java 客户端使用介绍	44
1.3.1. atkOpen	45
1.3.2. atkConnect	45
1.3.3. atkClose	45
1.3.4. 案例操作流程.....	45
1.3.5. 案例展示.....	49
2. ATK.Connect 模式 MBSE 接口	58
2.1. 与插件相关的 SysML 语言要素	58
2.1.1. 模块定义图.....	59
2.1.2. 状态机图.....	62
2.1.3. 活动图.....	64
2.2. 相关软件部署过程.....	69
2.2.1. MagicDraw 安装.....	69
2.2.2. ATK 安装.....	69
2.2.3. MagicDraw 与 ATK 联合仿真插件安装.....	71
2.3. MagicDraw 与 ATK 联合仿真使用过程.....	72
2.3.1. 载人航天任务建模过程.....	72
2.3.2. 脚本编辑过程.....	74
2.3.3. 载人航天任务仿真过程.....	83
2.4. MagicDraw 与 ATK 联合仿真插件升级.....	86

3. ATK.Component 模式教程	88
3.1. C++语言操作流程.....	88
3.1.1. 新建项目.....	88
3.1.2. 文件配置.....	89
3.1.3. 项目新建项.....	90
3.1.4. 设置添加项的名称与位置.....	91
3.1.5. Debug 属性配置.....	92
3.1.6. Release 属性配置.....	95
3.1.7. 添加包含文件，编写代码.....	98
3.1.8. 查看生成文件.....	98
3.1.9. 仿真轨迹.....	99
3.1.10. 代码示例.....	99
3.2. Java 语言操作流程.....	105
3.2.1. swigwin-4.1.1 安装.....	105
3.2.2. Java 安装与环境配置.....	105
3.2.3. SWIG 将 C++转成 Java 封装文件.....	109
3.2.4. 依赖文件.....	126
3.2.5. 调用动态库时 Java 与 C++的语法不同.....	128
3.2.6. Java 测试代码示例（快速转移）.....	129
3.2.7. CMD 编译执行测试代码.....	134
3.2.8. 查看生成文件判断是否正确.....	135
3.2.9. 打开生成的想定文件查看效果.....	136
3.3. Python 语言操作流程.....	136
3.3.1. swigwin-4.1.1 安装.....	136
3.3.2. Python 安装与环境配置.....	137
3.3.3. 编写 SWIG 接口文件.....	140
3.3.4. 编译 SWIG 接口文件.....	158
3.3.5. 编写 SWIG 编译文件.....	159
3.3.6. 编译 SWIG 编译文件.....	170
3.3.7. Python 调用测试.....	170
4. ATK 发布说明	180
4.1. 重要声明.....	180
4.1.1. 权利声明.....	180
4.1.2. 技术声明.....	180
4.1.3. 使用条款.....	180

附录一 Connect 模式功能介绍..... 184

附录二 Component 模式功能介绍 184

ATK 二次开发模块是 ATK 为方便用户可以使用不同语言，不同软件对 ATK 进行操作的定制化模块。此模块提供了三种方式对 ATK 进行操作，包括 Connect 模式、MBSE 模式、Component 模式。

Connect 模式使用网络方式与 ATK 软件进行连接，用户可以通过编写脚本发送命令的方式对 ATK 软件进行操控。Connect 模式提供的库文件是此模块为用户提供的功能及其他信息集合，提供的接口函数作为客户端与 ATK 数据传输的识别方式，接口函数中输入的命令脚本为 ATK 解析操作的依据，以此达到通过脚本输入对 ATK 软件进行想定及属性设置的效果。此模式支持 C++、Matlab、Java 三种客户端方式调用，其中 Matlab 客户端支持 MatlabR2015b 版本，Java 客户端使用 1.8 版本。

MBSE 模式使用插件机制实现与 ATK 软件的数据传输，插件用于 MagicDraw 客户端与 ATK 之间的连接与数据传输与解析。用户通过在 MagicDraw 客户端构建模型，在模型中输入脚本命令，运行模型的方式对 ATK 软件进行操控。脚本命令格式与 Connect 模式脚本命令相同。其中 MagicDraw 使用 MagicDraw2019 版本。

Component 模式使用加载动态库的形式调用 ATK 相关功能，库文件包含此模块向用户提供的可操作接口，通过接口可以实现对 ATK 参数的修改与设置。用户可以在软件中加载相关库文件，编写代码的方式对 ATK 进行操控。此模式支持 C++、Java、Python 三种语言方式，其中 C++使用 Visual Studio2015 版本，Java 使用 1.8 版本，Python 使用 3.11.0 版本。

Connect 模式、MBSE 模式可使用的脚本命令请参考 Connect 模式功能介绍；Component 模式可使用的动态库接口请参考 Component 模式功能介绍。

1. ATK.Connect 功能及使用介绍

ATK.Connect 模式向用户提供多种语言客户端，提供不同的操作属性命令，以方便用户对 ATK 进行操作。具体使用方式及命令介绍如下。

1.1. C++客户端使用介绍

C++客户端是 ATK 软件向用户提供的一种可以输入脚本命令的窗口，C++客户端内嵌在 ATK 软件中，在“集成”菜单栏中，选择“客户端”按钮，即可

弹出客户端窗口。目前界面属性窗口不具备实时更新功能，故而设置属性后需重新打开对象属性界面属性数据才会更新。以下为 C++客户端提供的接口函数介绍。

1.1.1. atkOpen

用法：(1) conID = atkOpen();

(2) conid=atkOpen('192.168.0.12',6655);

说明：conID-连接句柄

默认连接地址为本机；也可输入需要连接的地址，ATK 端口号为 6655。

1.1.2. atkConnect

用法：atkConnect(conID, 'command', 'objPath', 'cmdParamString')

说明：conID-来自 atkOpen 的句柄

Command-具体请查看 6.1.1.2 Connect 命令库

objPath-接受命令的对象路径

cmdParamString-命令属性字符串

注意：拥有返回值的命令，会直接输出返回值字符串，目前 c++接口仅支持机动规划返回值。

举例：atkConnect(conID, 'Graphics', '*/Satellite/Satellite1 SetColor 12');

1.1.3. atkClose

用法：atkClose(conID)

说明：conID-来自 atkOpen 的连接句柄

1.1.4. 案例操作流程

以轨道倾角改变场景为例，此案例实现是从初始近地轨道到指定半径的地球同步轨道的转移轨道机动规划。以下为通过 C++客户端在 ATK 软件中新建想定并设置参数的使用介绍。

使用 C++方式与 ATK 进行连接，首先需要打开 ATK。不需要打开场景；若需新建场景，需将已打开的场景关闭。

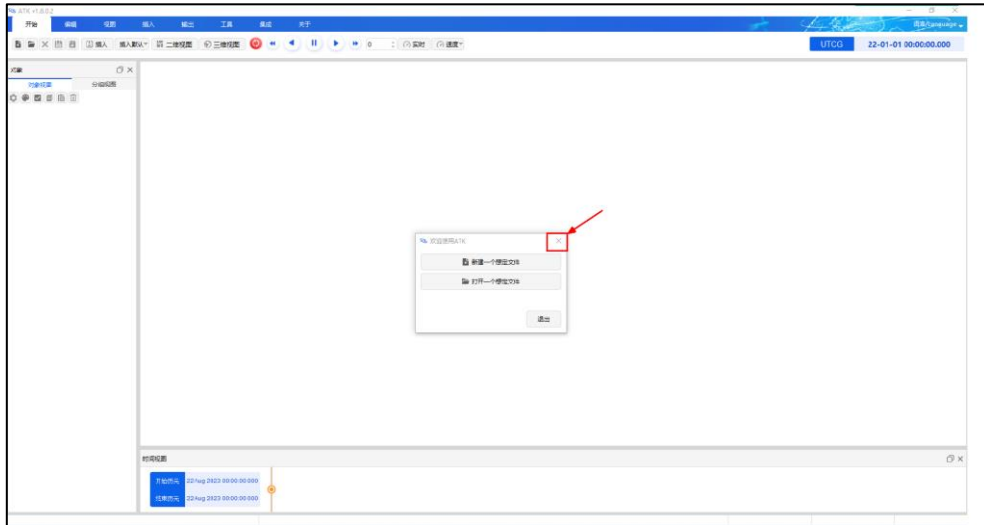


图 1-1 ATK 打开界面

1.1.4.1. 从命令窗口新建场景

注意：若有已打开场景，请先关闭场景。

1.1.4.1.1. 第一步：在菜单栏选择集成菜单，再选择客户端窗口。



图 1-2 集成按钮位置

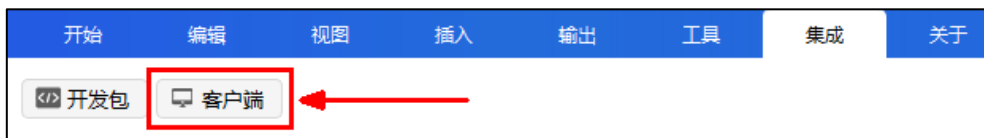


图 1-3 客户端按钮位置

点开客户端后，界面如下。

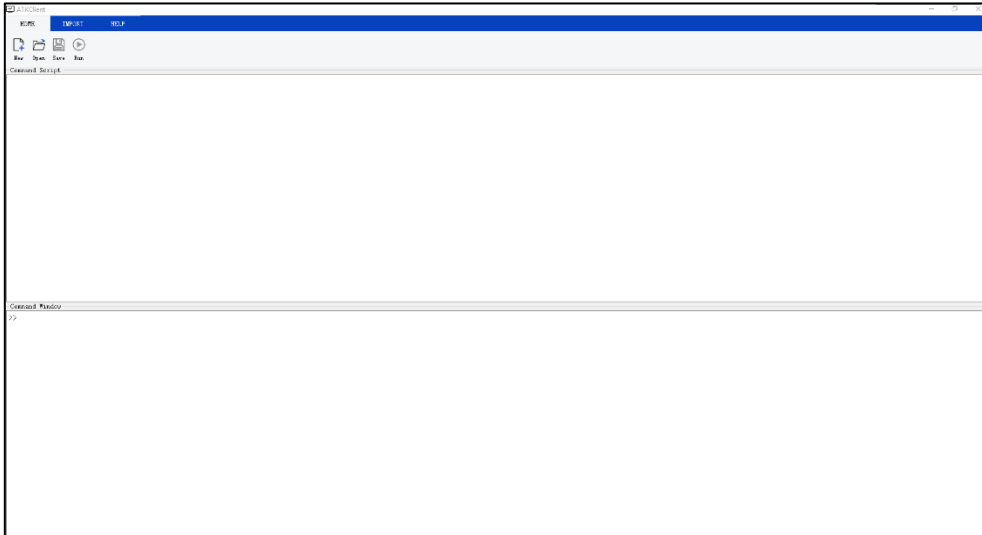


图 1-4 客户端界面

1.1.4.1.2. 第二步：在命令窗口输入命令，与 ATK 进行连接。

与 ATK 进行连接使用 [atkOpen](#) 命令。

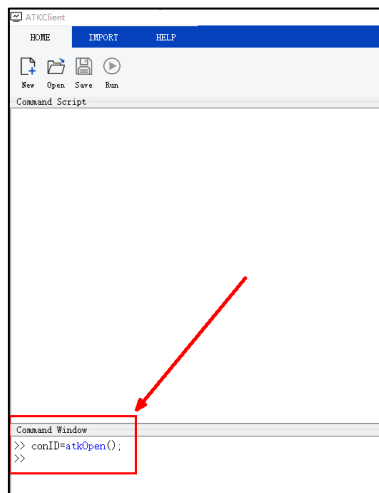


图 1-5 输入命令位置

1.1.4.1.3. 第三步：输入新建命令，新建想定并设置场景分析时间。

以新建轨道倾角改变场景为例，使用 [New](#) 命令。

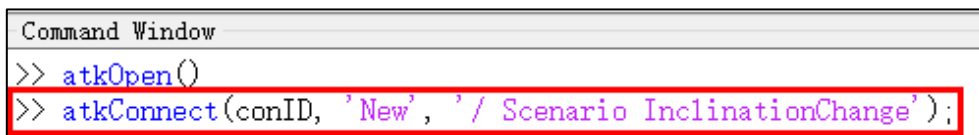


图 1-6 新建场景命令

执行命令后，相应界面如下。



图 1-7 新建的场景

使用 [SetAnalysisTimePeriod](#) 命令设置场景开始结束时间；使用 [Animate](#) 命令设置场景仿真状态。

```
Command Window
>> conID = atkOpen();
>> atkConnect(conID, 'New', '/ Scenario InclinationChange');
>> atkConnect(conID, 'SetAnalysisTimePeriod', '* "5 Nov 2022 00:00:00.000" "8 Nov 2022 00:00:00.000"');
>> atkConnect(conID, 'Animate', '* Reset');
>> |
```

图 1-8 设置场景属性命令

1.1.4.1.4. 第四步：新建卫星对象

可以通过 New 命令，新建对象，以新建卫星为例。

```
Command Window
>> atkOpen()
>> atkConnect(conID, 'New', '/ Scenario InclinationChange');
>> atkConnect(conID, 'SetAnalysisTimePeriod', '* "5 Nov 2022 00:00:00.000" "8 Nov 2022 00:00:00.000"');
>> atkConnect(conID, 'New', '/ Satellite SatInclinationChange');
```

图 1-9 新建卫星命令

执行命令后，相应界面如下。

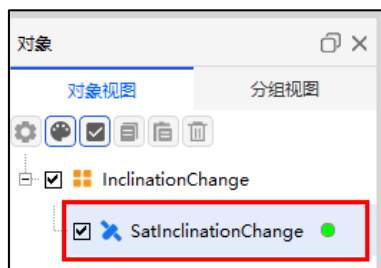


图 1-10 新建的卫星

1.1.4.1.5. 第五步：设置对象属性。

以设置卫星属性为例，具体命令格式请参考 Connect 模式功能介绍。

```

Command Window
>> atkConnect(conID, 'New', '/ Satellite SatInclinationChange');
>> atkConnect(conID, 'Animate', '* Reset');
>> atkConnect(conID, 'Graphics', '* /Satellite/SatInclinationChange Basic LineWidth 3');
>> atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange SetProp');
>> atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Propagate');
>> atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Target_Sequence');
>> atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.Target_Sequence.SegmentList.- Maneuver');
>> atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Propagate');
>> atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Target_Sequence');

```

图 1-11 设置卫星的属性命令

执行命令后，可在界面右键点击对象，选择属性按钮，查看设置结果。

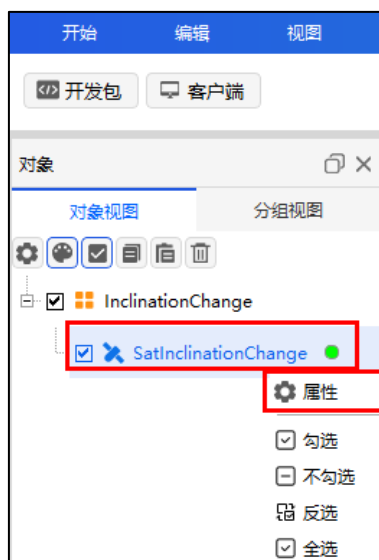


图 1-12 右键点击卫星，属性按钮位置

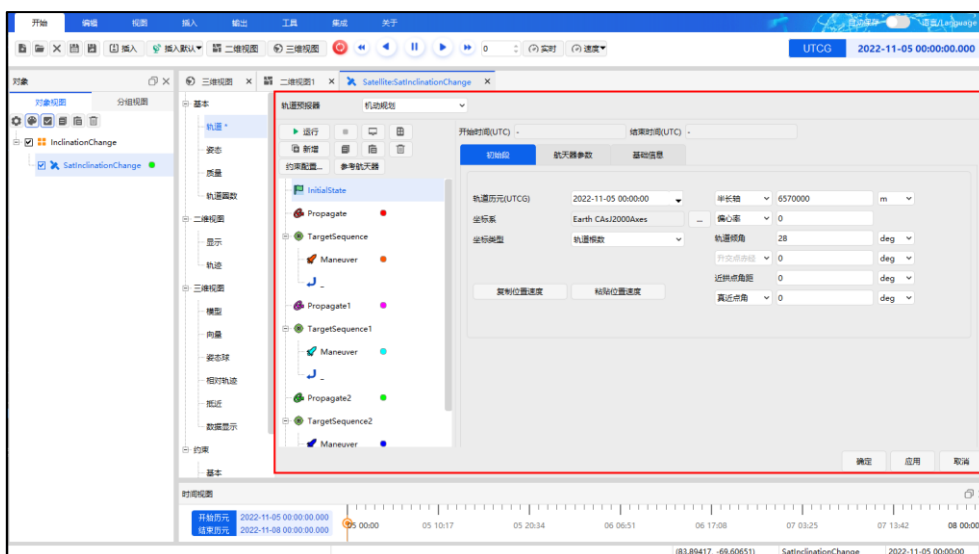


图 1-13 卫星属性界面

1.1.4.1.6. 第六步：开始仿真，属性设置完成后，请关闭连接。

机动规划使用 [Astrogator RunMCS](#) 命令运行机动规划。

```
Command Window
MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector_Manuever_Desired 0');
Weight 1');
>> atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Manuever "Eccentricity"
Desired 0');
>> atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetValue
MainSequence.SegmentList.Propagate3.SegmentColor 4278255615');
>> atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetValue
MainSequence.SegmentList.Propagate3.StoppingConditions.Duration');
>> atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetValue
MainSequence.SegmentList.Propagate3.StoppingConditions.Duration.TripValue 129600 sec');
>> atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetValue
MainSequence.SegmentList.Propagate3.StoppingConditions.Duration.Tolerance 0.00001 sec');
>> atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange RunMCS');
>>
```

图 1-14 运行机动规划命令

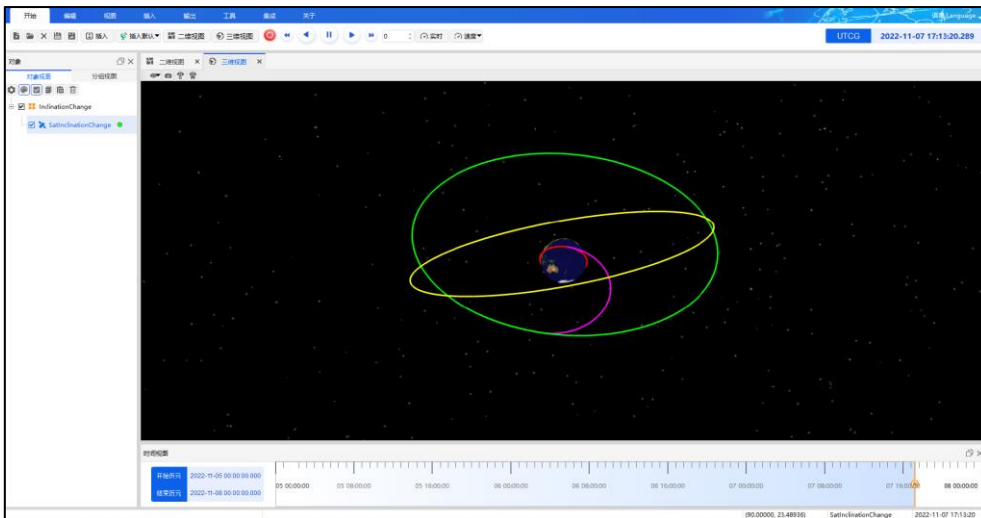


图 1-15 场景仿真界面

使用 [Save](#) 命令保存场景，若保存脚本文件，请在客户端界面点击保存按钮。

```
>> atkConnect(conID, 'Save', '/ *');
>>
```

图 1-16 场景保存命令

场景及对象设置完成后，使用 [atkClose](#) 命令与 ATK 断开连接。

```
>> atkClose(conID);
```

图 1-17 与 ATK 断开连接

1.1.4.2. 从输入脚本窗口新建场景

除去逐条运行命令新建场景外，也可以多条命令批量处理。

注意：若有已打开场景，请先关闭场景。

1.1.4.2.1. 第一步：在菜单栏选择集成菜单，再选择客户端窗口。



图 1-18 集成按钮位置

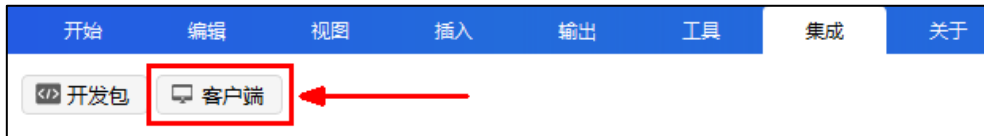


图 1-19 客户端按钮位置

点开客户端后，界面如下。

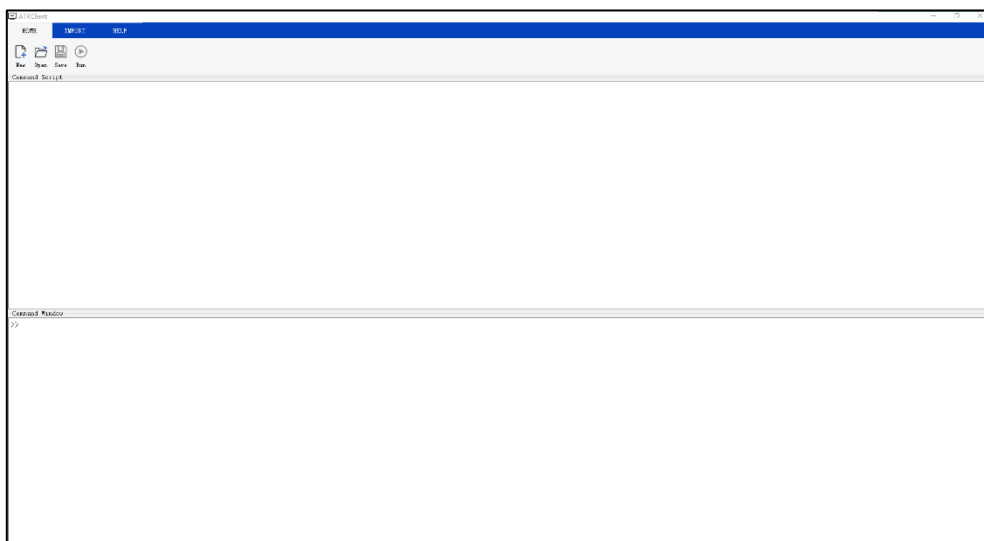


图 1-20 客户端界面

1.1.4.2.2. 第二步：点击“New”按钮新建脚本文件。

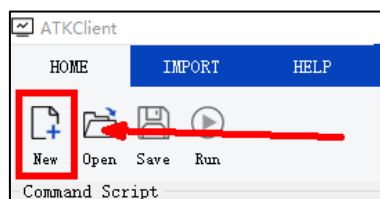


图 1-21 New 按钮位置

1.1.4.2.3. 第三步：输入脚本命令，新建想定以及对象，并对其属性进行设置。

- (1) 与 ATK 进行连接使用 [atkOpen](#) 命令。
- (2) 以新建轨道倾角改变场景为例，使用 [New](#) 命令新建场景，新建卫星

对象。

(3) 使用 [SetAnalysisTimePeriod](#) 命令设置场景开始结束时间；使用 [Animate](#) 命令设置场景仿真状态。

(4) 以设置卫星轨道属性为例，具体命令格式请参考 Connect 模式功能介绍。

(5) 使用 [Save](#) 命令保存场景，若保存脚本文件，请在客户端界面点击保存按钮。

(6) 使用 [atkClose](#) 命令与 ATK 断开连接。

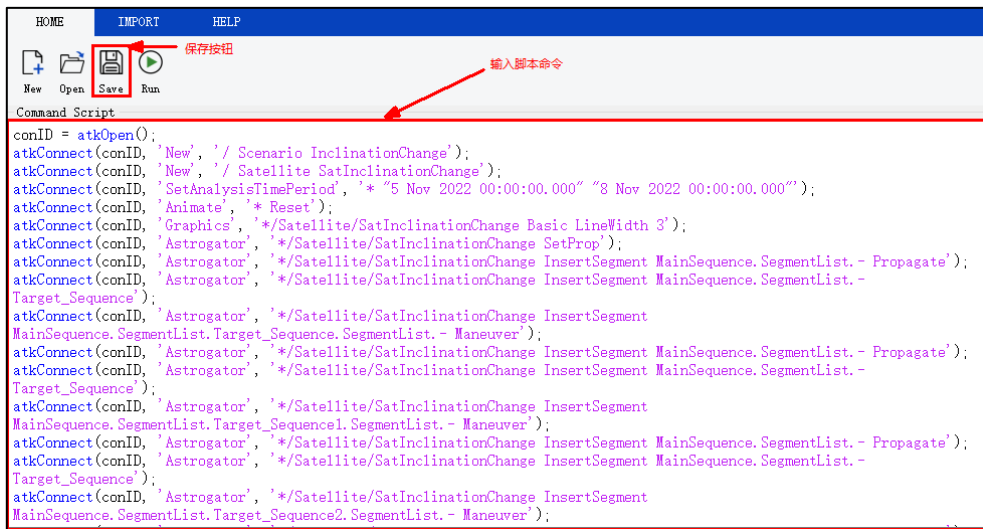


图 1-22 输入的命令及保存脚本文件按钮位置

1.1.4.2.4. 第四步：点击 Run 按钮，即可开始仿真。

场景及对象属性设置完成后，即可运行脚本。

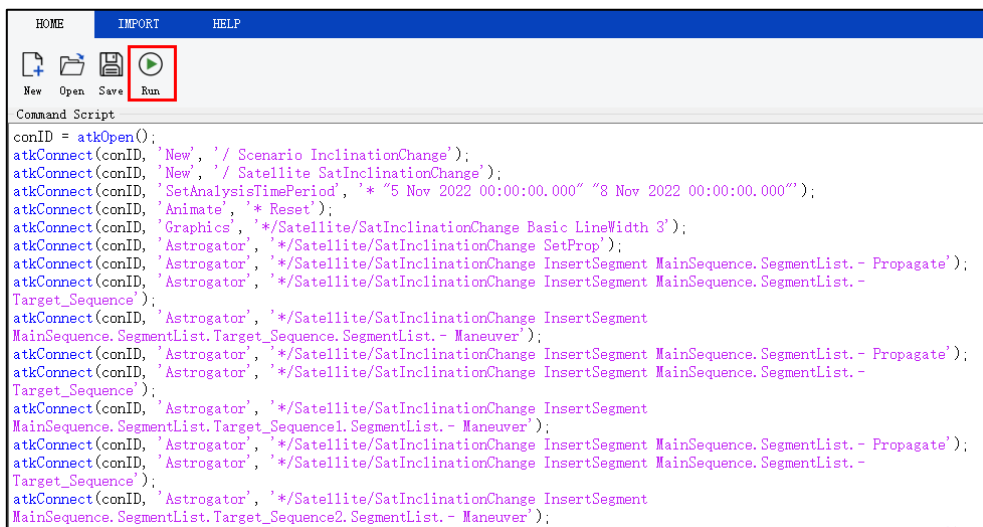


图 1-23 Run 按钮位置

脚本运行结束后，相应界面如下。

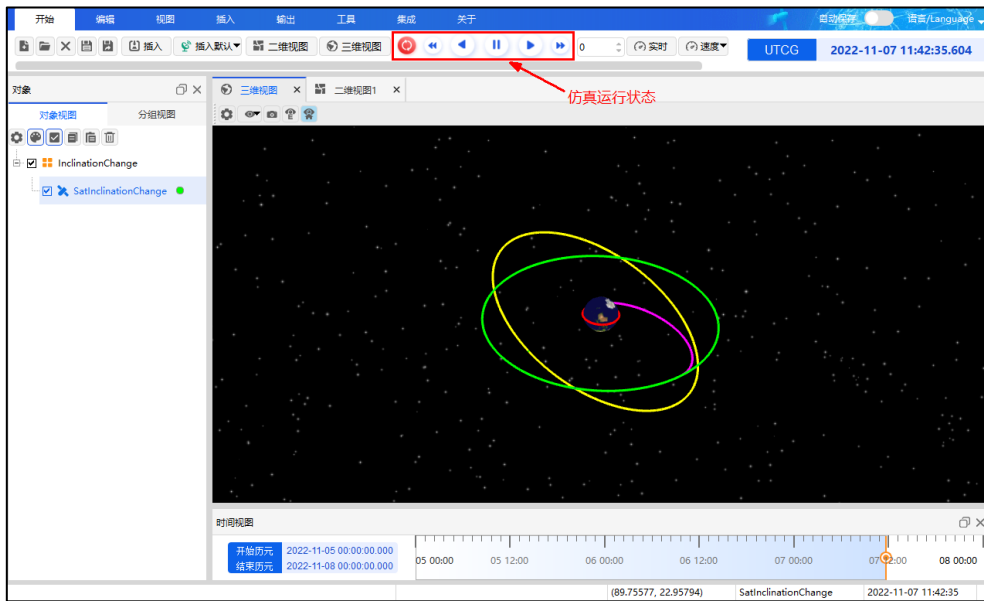


图 1-24 场景仿真界面

1.1.4.3. 打开已保存场景

注意：若有已打开场景，请先关闭场景。

1.1.4.3.1. 第一步：在菜单栏选择集成菜单，再选择客户端窗口。



图 1-25 集成按钮位置

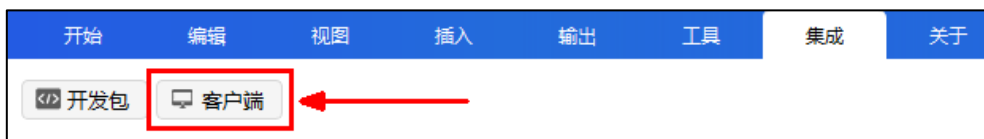


图 1-26 客户端按钮位置

点开客户端后，界面如下。

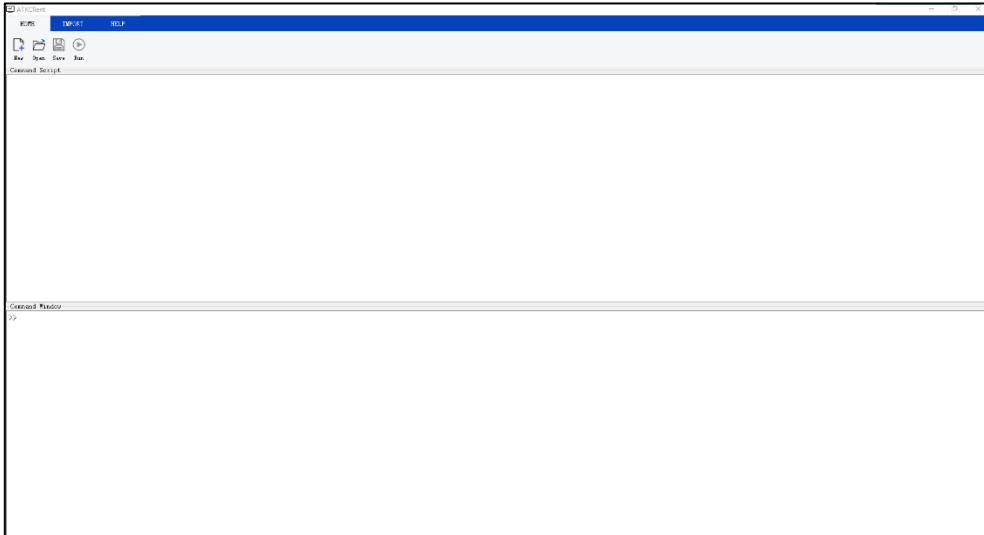


图 1-27 客户端界面

1.1.4.3.2. 第二步：在客户端主菜单点击 Open 按钮。

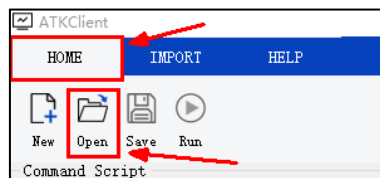


图 1-28 Open 按钮位置

1.1.4.3.3. 第三步：选择需要打开的.atks 脚本文件，点击打开按钮。

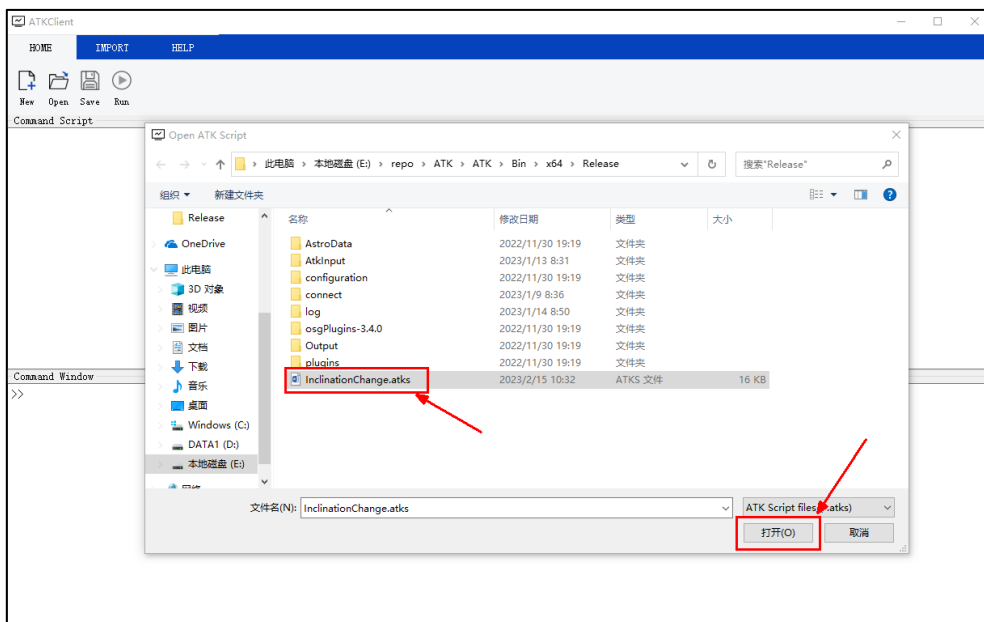


图 1-29 选择文件并打开

文件打开后，相应界面如下。

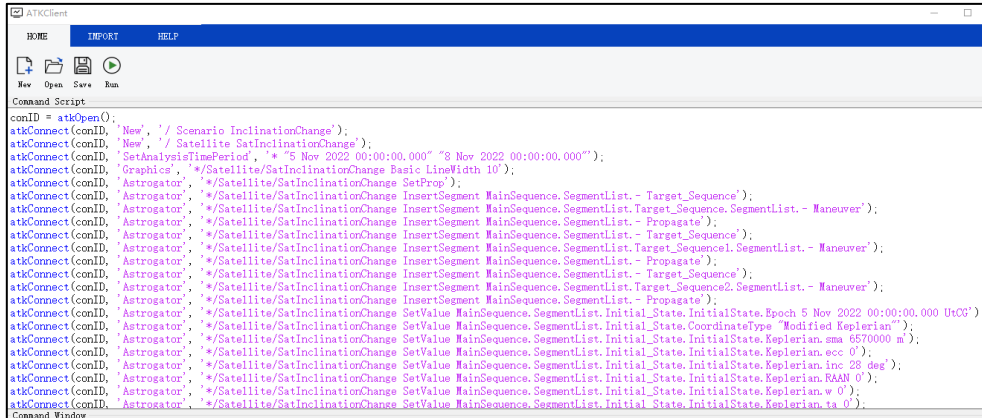


图 1-30 打开文件后的客户端界面

1.1.4.3.4. 第四步：打开后点击主菜单中的 **Run** 按钮，即可运行。

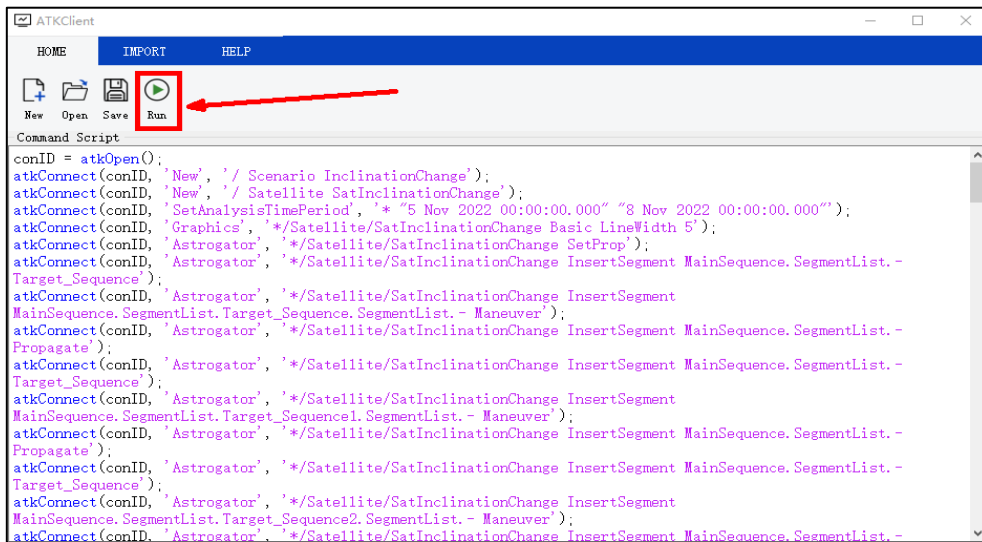


图 1-31 Run 按钮位置

脚本文件执行后，相应界面如下。

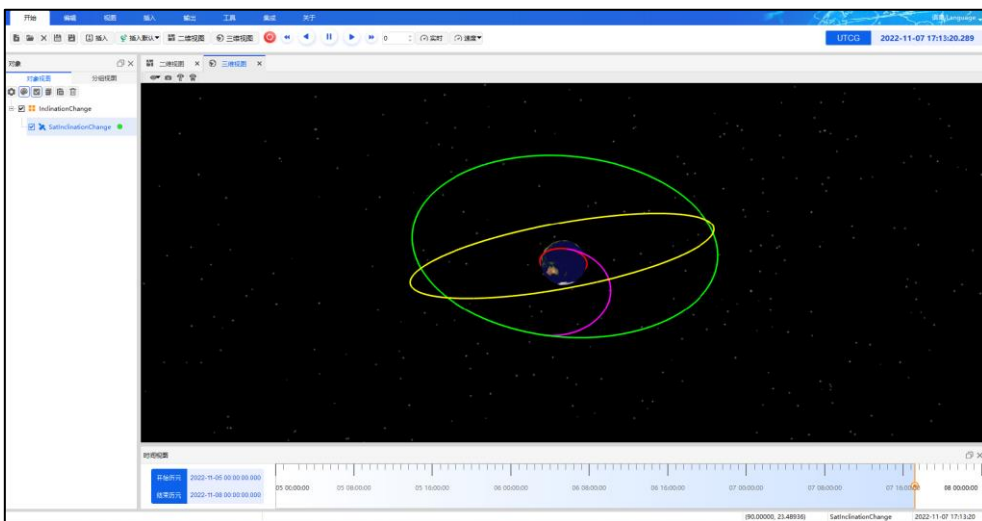


图 1-32 场景仿真界面

1.1.5. 案例展示

1.1.5.1. InclinationChange 场景案例

通过二次开发 Connect 模式，使用 C++接口创建 InclinationChange 场景，首次设置为机动规划，会默认添加一个初始段，必须保持 ATK 为打开状态，具体命令分类包括：

- (1) 与 ATK 进行连接。
- (2) 新建场景和多个卫星。
- (3) 设置场景属性。
- (4) 设置卫星轨道属性。
- (5) 保存场景后与 ATK 断开连接。

1.1.5.2. 脚本展示

1.1.5.2.1. 与 ATK 进行连接。

```
//与 ATK 进行连接  
conID = atkOpen();
```

1.1.5.2.2. 新建场景和多个卫星。

```
//新建一个名为 InclinationChange 的场景  
atkConnect(conID, 'New', '/ Scenario InclinationChange');  
//新建一个名为 SatInclinationChange 的卫星  
atkConnect(conID, 'New', '/ Satellite SatInclinationChange');
```

1.1.5.2.3. 设置场景属性。

```
//使用 UTCG 格式设置场景分析时间段，开始时间为 2022-11-05 零点，结束时间为 2022-11-08 零点  
atkConnect(conID, 'SetAnalysisTimePeriod', '* "5 Nov 2022 00:00:00.000" "8 Nov 2022 00:00:00.000" ');  
//仿真重置  
atkConnect(conID, 'Animate', '* Reset');
```

1.1.5.2.4. 设置卫星轨道属性。

```
//设置卫星轨迹线宽为 2
```

```

atkConnect(conID, 'Graphics', '*/Satellite/SatInclinationChange Basic LineWidth 2');
//设置卫星轨道预报器类型为机动规划；第一级目录默认存在一个初始段，一个
预报段
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetProp');
//在第一级目录下添加一个预报段，默认段名为 Propagate
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Propagate');
//在第一级目录下添加一个瞄准序列段，默认段名为 Target_Sequence
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Target_Sequence');
//在瞄准序列段内添加一个机动段，默认段名为 Maneuver
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.Target_Sequence.SegmentList.- Maneuver');
//在第一级目录下添加一个预报段；由于已经存在一个预报段，所以默认此段段
名为 Propagate1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Propagate');
//在第一级目录下添加一个瞄准序列段；由于已经存在一个瞄准序列段，所以默
认此段段名为 Target_Sequence1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Target_Sequence');
//在 Target_Sequence1 下添加一个机动段，默认段名为 Maneuver
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.Target_Sequence1.SegmentList.- Maneuver');
//在第一级目录下添加一个预报段；由于已经存在两个预报段，所以默认此段段
名为 Propagate2
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Propagate');
//在第一级目录下添加一个瞄准序列段；由于已经存在两个瞄准序列段，所以默
认此段段名为 Target_Sequence2
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Target_Sequence');
//在 Target_Sequence2 下添加一个机动段，默认段名为 Maneuver
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.Target_Sequence2.SegmentList.- Maneuver');
//在第一级目录下添加一个预报段；由于已经存在三个预报段，所以此段段名为
Propagate3
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Propagate');
//设置初始段轨道历元
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Epoch 5 Nov 2022 00:00:00.000
UTCG');
//设置初始段坐标类型为轨道根数

```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.CoordinateType "Modified Keplerian" ');
//设置初始段半长轴为 6570000 米
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.sma 6570000 m');
//设置初始段偏心率为 0
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.ecc 0');
//设置初始段轨道倾角为 28 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.inc 28 deg');
//设置初始段升交点赤经为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.RAAN 0');
//设置初始段近拱点角距为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.w 0');
//设置初始段真近点角为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.ta 0');
//设置第一个预报段颜色为红色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate.SegmentColor 4278190335');
//设置第一个预报段停止条件为 Duration
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate.StoppingConditions Duration');
//设置第一个预报段停止条件 Duration 的触发值为 7200 秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate.StoppingConditions.Duration.TripValue 7200 sec');
//设置第一个预报段停止条件 Duration 的误差值为 0.0001 秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate.StoppingConditions.Duration.Tolerance 0.0001 sec');
//设置第一个瞄准序列段内的机动段颜色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence.SegmentList.Maneuver.SegmentColor
4278212095');
//设置第一个瞄准序列段内的机动段在脉冲类型下推力坐标轴为 VNC
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence.SegmentList.Maneuver.Impul-
siveMnvr.ThrustAxes "Satellite VNC(Earth)" ');
//设置第一个瞄准序列段控制变量为机动段的直角坐标 X
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange AddMCS-
SegmentControl MainSequence.SegmentList.Target_Sequence.SegmentList.Maneu-
ver ImpulsiveMnvr.Cartesian.X');

```

```

//第一个瞄准序列段添加属性页
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector');
//设置第一个瞄准序列段的控制变量-机动段直角坐标 X 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Active true');
//设置第一个瞄准序列段的控制变量-机动段直角坐标 X 的最大步长为 100 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');
//设置第一个瞄准序列段的控制变量-机动段直角坐标 X 的累计校正量为 2456.42862556706 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Correction 2456.42862556706 m/sec');
//设置第一个瞄准序列段的控制变量-机动段直角坐标 X 的摄动量为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
//设置第一个瞄准序列段的控制变量-机动段直角坐标 X 的归一化参数为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
//设置第一个瞄准序列段内的机动段的约束为 Radius Of Apoapsis
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.Results " Radius Of Apoapsis " ');
//设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver "Radius Of Apoapsis" Active true');
//设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 的收敛误差为 0.1 米
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver "Radius Of Apoapsis" Tolerance 0.1 m');
//设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 的归一化参数为 1 米
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver "Radius Of Apoapsis" Scale 1 m');
//设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 的权重系数为 1

```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Cor-
rector Maneuver "Radius Of Apoapsis" Weight 1');
//设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 的期望值为 42160000
米
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Cor-
rector Maneuver "Radius Of Apoapsis" Desired 42160000 m');
//设置第二个预报段的颜色为亮紫色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate1.SegmentColor 4294902015');
//设置第二个预报段的停止条件为 Apoapsis
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate1.StoppingConditions Apoapsis');
//设置第二个预报段停止条件 Apoapsis 的重复次数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate1.StoppingConditions.Apoapsis.RepeatCount 1');
//设置第二个预报段停止条件 Apoapsis 的误差值为 0.0001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate1.StoppingConditions.Apoapsis.Tolerance 0.0001');
//设置第二个瞄准序列段的机动段颜色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence1.SegmentList.Maneuver.SegmentColor -256');
//设置第二个瞄准序列段内的机动段没冲类型下推力坐标轴为 VNC
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence1.SegmentList.Maneuver.Impul-
siveMnvr.ThrustAxes "Satellite VNC(Earth)" ');
//设置第二个瞄准序列段的控制变量为机动段直角坐标 X
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange AddMCS-
SegmentControl MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneu-
ver ImpulsiveMnvr.Cartesian.X');
//第二个瞄准序列段添加属性页
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence1.Profiles Differential_Corrector');
//设置第二个瞄准序列段的控制变量-机动段直角坐标 X 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Active true');
//设置第二个瞄准序列段的控制变量-机动段直角坐标 X 的最大步长为 100 米每
秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');

```

```

//设置第二个瞄准序列段的控制变量-机动段直角坐标 X 的累计校正量为
1480.06329844802 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Correction 1480.06329844802 m/sec');
//设置第二个瞄准序列段的控制变量-机动段直角坐标 X 的摄动量为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
//设置第二个瞄准序列段的控制变量-机动段直角坐标 X 的归一化参数为 1 米每
秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
//设置第二个瞄准序列段内的机动段约束为 Eccentricity
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence1.SegmentList.Maneuver.Results "Eccentricity"
');
//设置第二个瞄准序列段内的机动段约束 Eccentricity 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Active true');
//设置第二个瞄准序列段内的机动段约束收敛误差为 0.001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Tolerance 0.001');
//设置第二个瞄准序列段内的机动段约束归一化参数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Scale 1');
//设置第二个瞄准序列段内的机动段约束权重系数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Weight 1');
//设置第二个瞄准序列段内的机动段约束期望值为 0
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Desired 0');
//设置第三个预报段的颜色为绿色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate2.SegmentColor 4278255360');
//设置第三个预报段的停止条件为 AscendingNode
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate2.StoppingConditions AscendingNode');

```

```

//设置第三个预报段的停止条件 AscendingNode 的重复次数为 2
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate2.StoppingConditions.AscendingNode.RepeatCount
2');
//设置第三个预报段的停止条件 AscendingNode 的误差值为 0.000001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate2.StoppingConditions.AscendingNode.Tolerance
0.000001');
//设置第三个瞄准序列段的机动段颜色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence2.SegmentList.Maneuver.SegmentColor -
65536');
//设置第三个瞄准序列段内的机动段脉冲属性下推力坐标轴为 VNC
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence2.SegmentList.Maneuver.Impul-
siveMnvr.ThrustAxes "Satellite VNC(Earth)" ');
//设置第三个瞄准序列段的控制变量为机动段的直角坐标 X
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange AddMCS-
SegmentControl MainSequence.SegmentList.Target_Sequence2.SegmentList.Maneu-
ver ImpulsiveMnvr.Cartesian.X');
//第三个瞄准序列段添加属性页
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence2.Profiles Differential_Corrector');
//设置第三个瞄准序列段控制变量-机动段直角坐标 X 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Active true');
//设置第三个瞄准序列段控制变量-机动段直角坐标 X 的最大步长为 100 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');
//设置第三个瞄准序列段控制变量-机动段直角坐标 X 的累计校正量为-
359.983751789857 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Correction -359.983751789857 m/sec');
//设置第三个瞄准序列段控制变量-机动段直角坐标 X 摄动量为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
//设置第三个瞄准序列段控制变量-机动段直角坐标 X 归一化参数为 1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');

```



```

//设置第三个瞄准序列段的控制变量为机动段直角坐标 Y
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange AddMCS-
SegmentControl MainSequence.SegmentList.Target_Sequence2.SegmentList.Maneu-
ver ImpulsiveMnvr.Cartesian.Y');
//设置第三个瞄准序列段控制变量-机动段直角坐标 Y 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y Active true');
//设置第三个瞄准序列段控制变量-机动段直角坐标 Y 的最大步长为 100 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y MaxStep 100 m/sec');
//设置第三个瞄准序列段控制变量-机动段直角坐标 Y 的累计校正量为-
1444.72488992759 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y Correction -1444.72488992759 m/sec');
//设置第三个瞄准序列段控制变量-机动段直角坐标 Y 的摄动量为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y Perturbation 0.1 m/sec');
//设置第三个瞄准序列段控制变量-机动段直角坐标 Y 的归一化参数为 1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y Scale 1 m/sec');
//设置第三个瞄准序列段内的机动段约束为 Inclination 和 Eccentricity
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence2.SegmentList.Maneuver.Results "Inclination"
"Eccentricity" ');
//设置第三个瞄准序列段内机动段约束 Inclination 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Inclination" Active true');
//设置第三个瞄准序列段内机动段约束 Inclination 的收敛误差为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Inclination" Tolerance 0 deg');
//设置第三个瞄准序列段内机动段约束 Inclination 的归一化参数为 1 弧度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Inclination" Scale 1 rad');
//设置第三个瞄准序列段内机动段约束 Inclination 的权重系数为 1

```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Inclination" Weight 1');
//设置第三个瞄准序列段内机动段约束 Inclination 的期望值为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Inclination" Desired 0 deg');
//设置第三个瞄准序列段内机动段约束 Eccentricity 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Active true');
//设置第三个瞄准序列段内机动段约束 Eccentricity 的收敛误差为 0.001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Tolerance 0.001');
//设置第三个瞄准序列段内机动段约束 Eccentricity 的收敛误差为 0.001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Scale 1');
//设置第三个瞄准序列段内机动段约束 Eccentricity 的权重系数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Weight 1');
//设置第三个瞄准序列段内机动段约束 Eccentricity 期望值为 0
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Desired 0');
//设置第四个预报段的颜色为黄色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate3.SegmentColor 4278255615');
//设置第四个预报段的停止条件为 Duration
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate3.StoppingConditions Duration');
//设置第四个预报段停止条件 Duration 的触发值为 129600 秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate3.StoppingConditions.Duration.TripValue 129600
sec');
//设置第四个预报段停止条件 Duration 的误差值为 0.00001 秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate3.StoppingConditions.Duration.Tolerance 0.00001
sec');
//运行此算例
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange RunMCS');

```

1.1.5.2.5. 保存场景后与 ATK 断开连接。

```
//保存场景  
atkConnect(conID, 'Save', '/' *');  
//与 ATK 断开连接  
atkClose(conID);
```

1.2. Matlab 客户端使用介绍

Matlab 客户端是一种方便用户与 ATK 软件进行网络连接并操作的方式，目前界面属性窗口不具备实时更新功能，故而设置属性后需重新打开对象属性界面属性数据才会更新。Matlab 客户端需用户自行下载安装（ATK 测试使用 MatlabR2015b 版本），ATK 提供库文件，用户需将使用库与使用函数添加到使用目录，如下图：

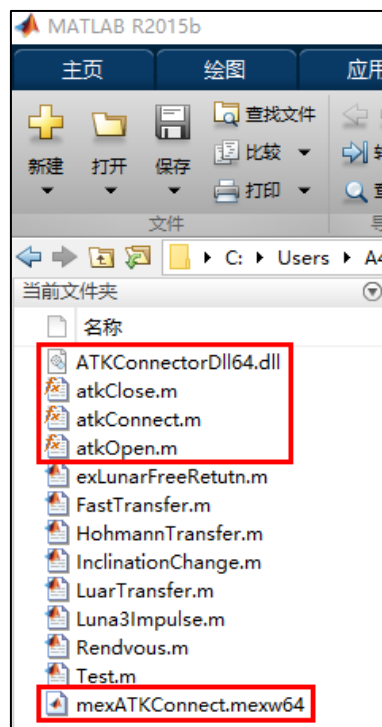


图 1-33 ATK 打开界面

ATKConnectorDll64.dll 为基于 Connect 模式提供的动态库，用于和 ATK 建立网络连接，传递命令数据和解析返回结果；

mexATKConnect.mexw64 是一个可执行的 Mex 文件，提供用于 Matlab 环境的 MEX 函数，方便 Matlab 和 ATKConnectorDll64.dll 之间传递数据；

atkOpen.m、atkConnect.m、atkClose.m 是 Matlab 函数式 M 文件，通过使用

MEX 函数完成 Matlab 和 ATK 之间的连接建立和数据传递。

1.2.1. atkOpen

用法: `conID = atkOpen('hostPortStr')`

说明: conID-连接句柄

hostPortStr-进行连接的网络地址端口号

1.2.2. atkConnect

用法: `rtnData = atkConnect(conID, 'command', 'objPath', 'cmdParamString')`

说明: conID-来自 atkOpen 的句柄

Command-具体请查看 Connect 命令库

objPath-接受命令的对象路径

cmdParamString-命令属性字符串

rtnData-从 atk 返回的响应的字符串

举例: `atkConnect(conID, 'Graphics', '*/Satellite/Satellite1 SetColor 12');`

1.2.3. atkClose

用法: `atkClose(conID)`

说明: conID-来自 atkOpen 的连接句柄

1.2.4. 案例操作流程

以轨道倾角改变场景为例，此案例实现是从初始近地轨道到指定半径的地球同步轨道的转移轨道机动规划。以下为通过 Matlab 客户端在 ATK 软件中新建想定并设置参数的使用介绍。

使用 Matlab 向 ATK 发送命令前，必须打开 ATK，不需要打开场景；若需新建场景，需将已打开的场景关闭。ATK 网络通信端口为 6655。

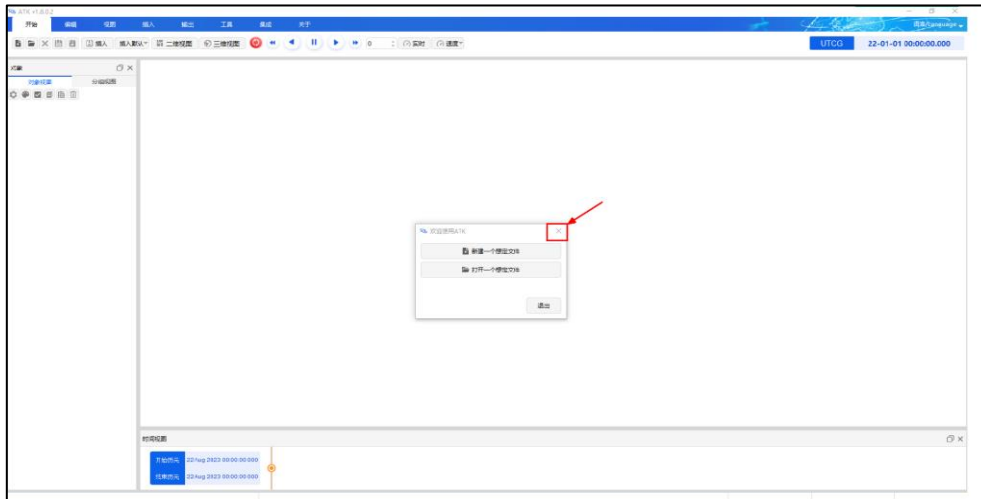


图 1-34 ATK 打开界面

1.2.4.1. 从命令窗口新建场景

注意：若有已打开场景，请先关闭场景。

1.2.4.1.1. 第一步：打开 Matlab，在命令行窗口输入命令与 ATK 进行连接。

与 ATK 进行连接使用 [atkOpen](#) 命令。

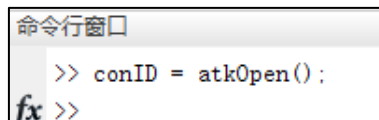


图 1-35 与 ATK 连接

1.2.4.1.2. 第二步：输入新建命令，新建想定并设置场景分析时间。

以新建轨道倾角改变场景为例，使用 [New](#) 命令。

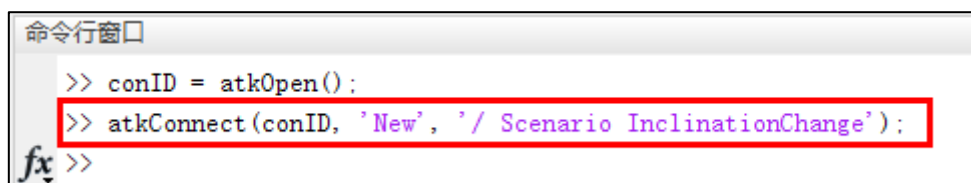


图 1-36 新建场景命令

执行命令后，相应界面如下。



图 1-37 新建的场景

使用 [SetAnalysisTimePeriod](#) 命令设置场景开始结束时间；使用 [Animate](#) 命令设置场景仿真状态。

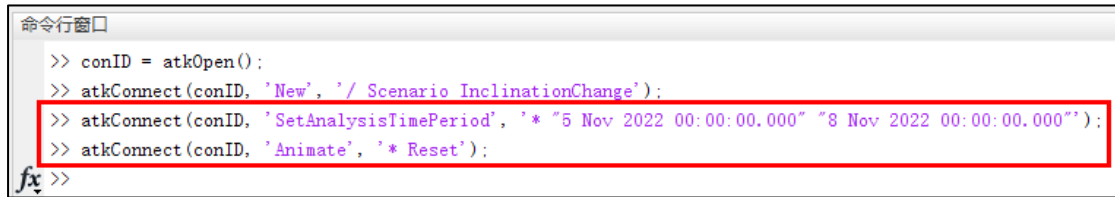


图 1-38 设置场景属性命令

1.2.4.1.3. 第三步：新建卫星对象并设置对象属性。

可以通过 New 命令，新建对象，以新建卫星为例。

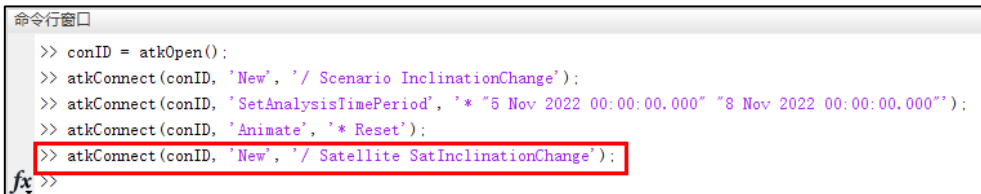


图 1-39 新建卫星命令

执行命令后，相应界面如下。

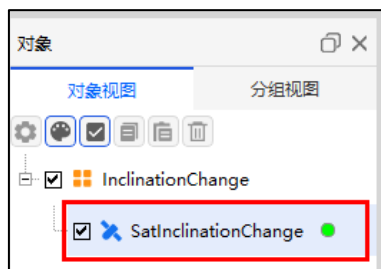


图 1-40 新建的卫星

以设置卫星属性为例，具体命令格式请参考 Connect 模式功能介绍。

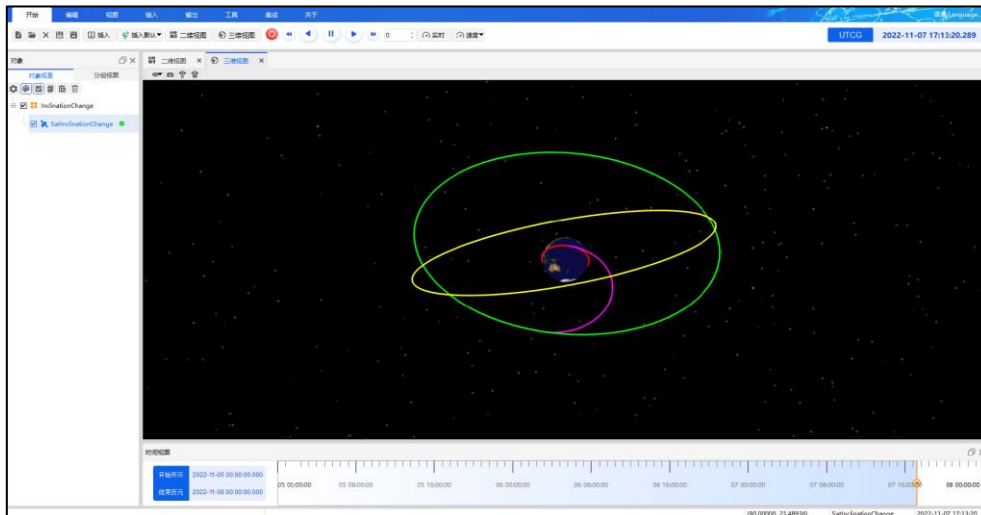


图 1-44 场景仿真界面

使用 [Save](#) 命令保存场景，若保存脚本文件，请在客户端界面点击保存按钮。

```
>> atkConnect (conID, 'Save', '/' *);
```

图 1-45 场景保存命令

场景及对象设置完成后，使用 [atkClose](#) 命令与 ATK 断开连接。

```
>> atkClose (conID);
```

图 1-46 与 ATK 断开连接

1.2.4.2. 从输入脚本窗口新建场景

除去逐条运行命令新建场景外，也可以多条命令批量处理。

注意：若有已打开场景，请先关闭场景。

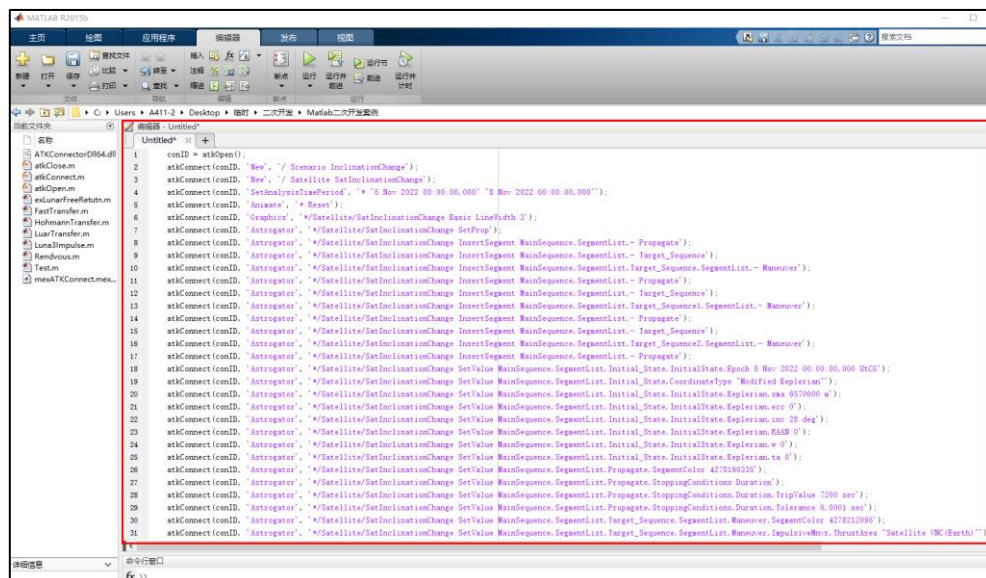
1.2.4.2.1. 第一步：打开 Matlab，在主页菜单栏下点击新建脚本按钮或新建按钮。



图 1-47 新建脚本和新建按钮位置

1.2.4.2.2. 第二步：输入多条命令，新建想定以及对象，并对其属性进行设置。

- (1) 与 ATK 进行连接使用 `atkOpen` 命令。
- (2) 以新建 `InclinationChange` 场景为例，使用 `New` 命令新建场景。
- (3) 使用 `SetAnalysisTimePeriod` 命令设置场景开始结束时间；使用 `Animate` 命令设置场景仿真状态。
- (4) 新建卫星对象，设置卫星为机动规划类型，并设置每个段的参数后运行算例。
- (5) 使用 `Save` 命令保存场景。
- (6) 使用 `atkClose` 命令与 ATK 断开连接。



```
1 cmdID = atkOpen();
2 atkConnect(cmdID, 'New', '/ Scenario InclinationChange');
3 atkConnect(cmdID, 'New', '/ Satellite SatInclinationChange');
4 atkConnect(cmdID, 'SetAnalysisTimePeriod', '* * 8 Nov 2022 00:00:00.000 * 8 Nov 2022 00:00:00.000');
5 atkConnect(cmdID, 'Animate', '* Rest');
6 atkConnect(cmdID, 'Orbitator', '* /Satellite/SatInclinationChange Basic Line[With 2]');
7 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetProp');
8 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Propagate');
9 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Target_Sequence');
10 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Target_Sequence.SegmentList - Maneuver');
11 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Propagate');
12 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Target_Sequence');
13 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Target_Sequence.SegmentList - Maneuver');
14 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Propagate');
15 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Target_Sequence');
16 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Target_Sequence.SegmentList - Maneuver');
17 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Propagate');
18 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Initial_State.InitialState.Epoch 8 Nov 2022 00:00:00.000 UTC0');
19 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Initial_State.CoordinateType 'Modified Keplerian');
20 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Initial_State.InitialState.Epulsion.ma 6070000 m');
21 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Initial_State.InitialState.Epulsion.ecc 0');
22 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Initial_State.InitialState.Epulsion.inc 28 deg');
23 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Initial_State.InitialState.Epulsion.RAAN 0');
24 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Initial_State.InitialState.Epulsion.w 0');
25 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Initial_State.InitialState.Epulsion.ua 0');
26 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate.SegmentColor 4278180235');
27 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate.StoppingConditions.Duration');
28 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate.StoppingConditions.Duration.TripValue 7200 sec');
29 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate.StoppingConditions.Duration.Tolerance 0.0001 sec');
30 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.SegmentColor 4278180235');
31 atkConnect(cmdID, 'Astrogator', '* /Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.ImpulsiveThrust.ThrustAxis 'Satellite YWC(Earth)');
```

图 1-48 输入脚本命令

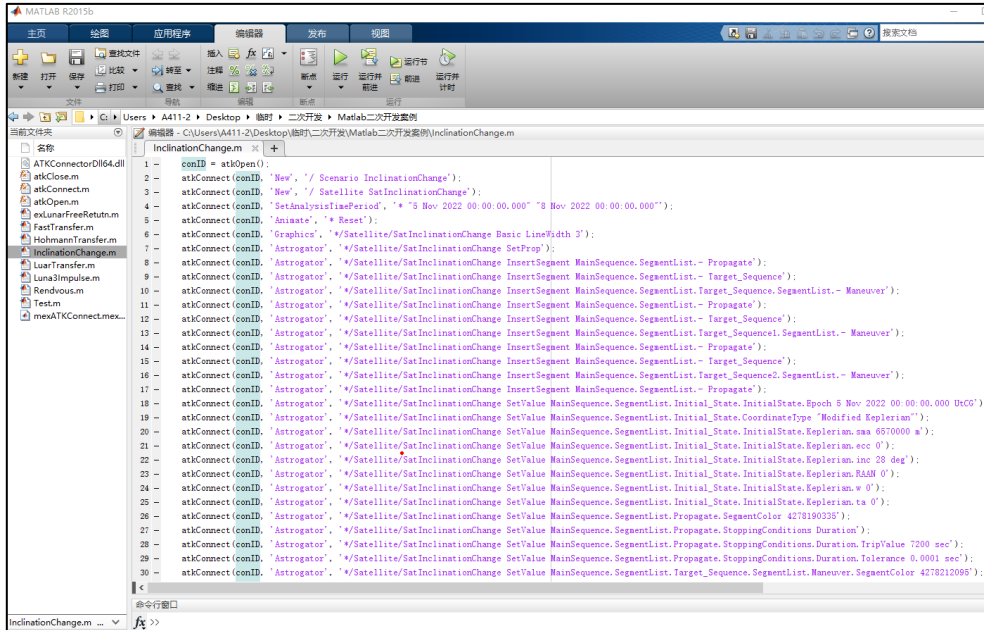


图 1-49 文件打开界面

1.2.4.2.3. 第三步：点击运行按钮，保存文件，即可开始仿真。

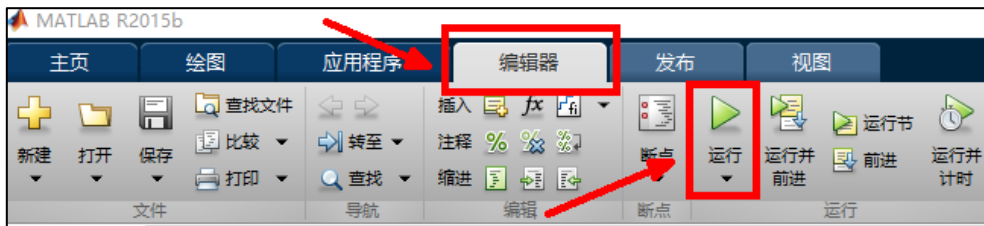


图 1-50 运行按钮位置

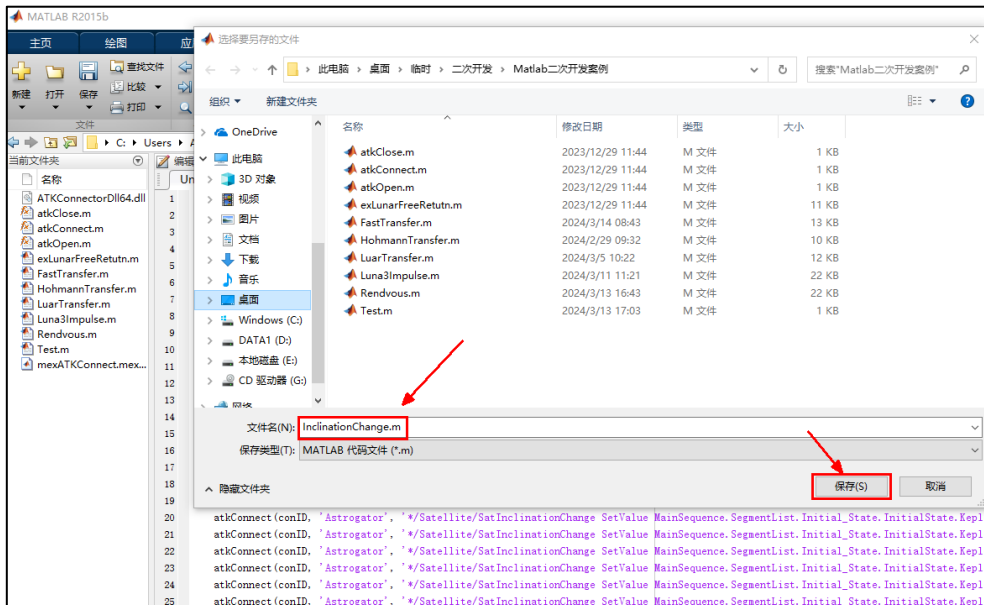


图 1-51 修改文件名字并保存

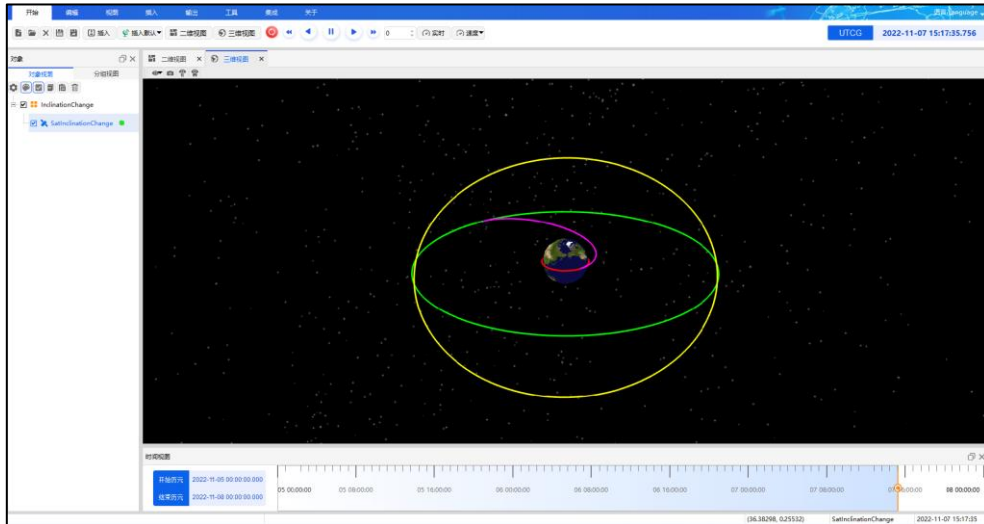


图 1-52 场景仿真轨迹

1.2.4.3. 打开已保存场景

注意：若有已打开场景，请先关闭场景。

1.2.4.3.1. 第一步：打开 Matlab，在主页菜单栏下点击打开按钮。



图 1-53 打开按钮位置

1.2.4.3.2. 第二步：选择需要打开的脚本文件，点击打开按钮。

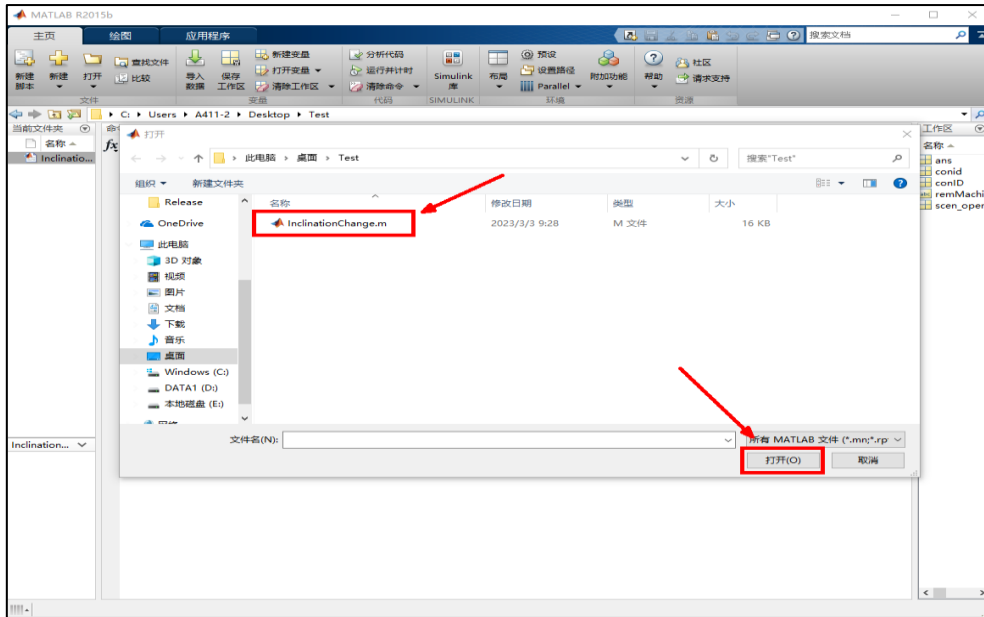


图 1-54 选择文件并打开

文件打开后，相应界面如下。

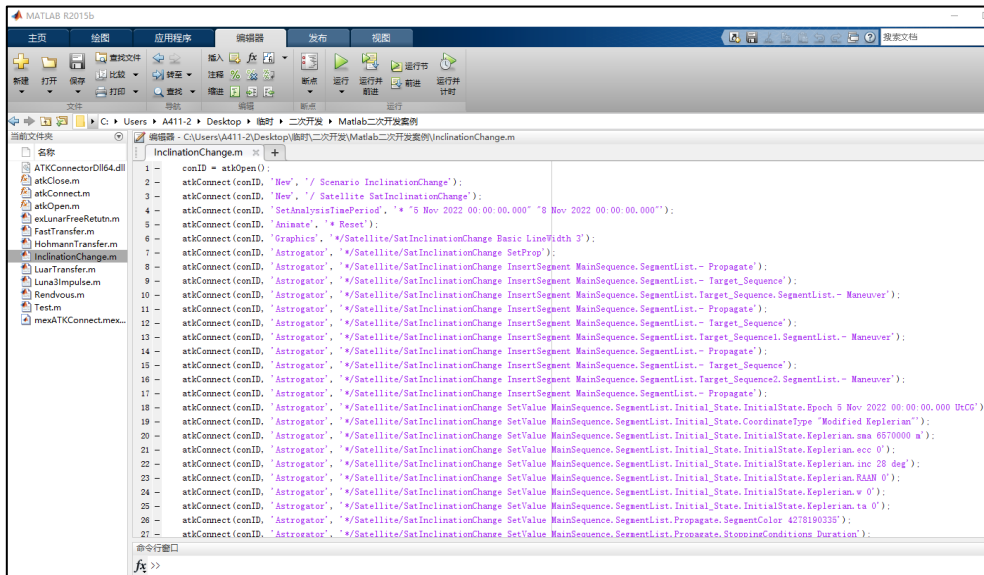


图 1-55 文件打开界面

1.2.4.3.3. 第三步：点击运行按钮，即可开始仿真。



图 1-56 运行按钮位置

脚本文件执行后，相应界面如下。

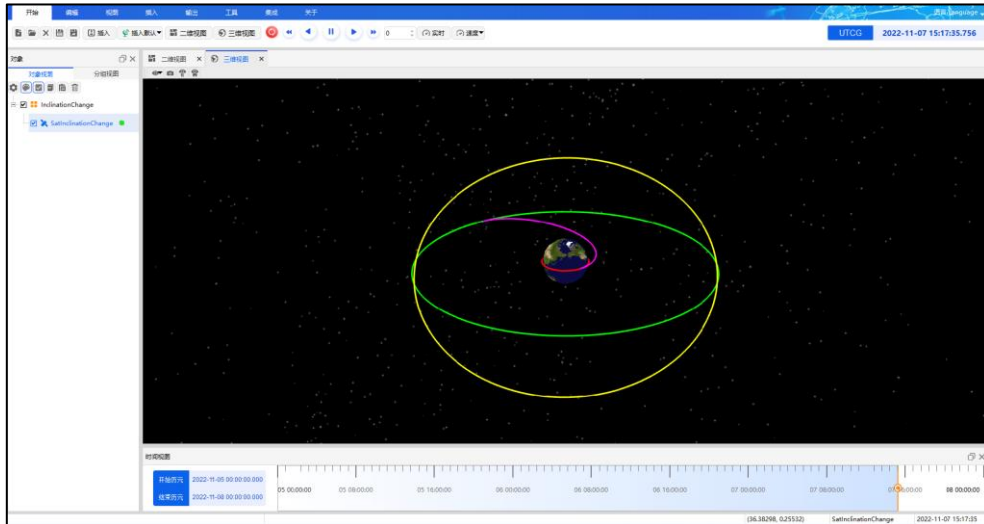


图 1-57 场景仿真轨迹

1.2.5. 案例展示

1.2.5.1. InclinationChange 场景案例

通过二次开发 Connect 模式，使用 Matlab 接口创建 InclinationChange 场景，首次设置为机动规划，会默认添加一个初始段，必须保持 ATK 为打开状态，具体命令分类包括：

- (1) 与 ATK 进行连接。
- (2) 新建场景和多个卫星。
- (3) 设置场景属性。
- (4) 设置卫星轨道属性。
- (5) 保存场景后与 ATK 断开连接。

1.2.5.2. 脚本展示

1.2.5.2.1. 与 ATK 进行连接。

```
%与 ATK 进行连接
conID = atkOpen();
```

1.2.5.2.2. 新建场景和多个卫星。

```
%新建一个名为 InclinationChange 的场景
atkConnect(conID, 'New', '/ Scenario InclinationChange');
```

```
%新建一个名为 SatInclinationChange 的卫星
atkConnect(conID, 'New', '/ Satellite SatInclinationChange');
```

1.2.5.2.3. 设置场景属性。

```
%使用 UTCG 格式设置场景分析时间段，开始时间为 2022-11-05 零点，结束时间为 2022-11-08 零点
atkConnect(conID, 'SetAnalysisTimePeriod', '* "5 Nov 2022 00:00:00.000" "8 Nov 2022 00:00:00.000" ');
%仿真重置
atkConnect(conID, 'Animate', '* Reset');
```

1.2.5.2.4. 设置卫星轨道属性。

```
%设置卫星轨迹线宽为 2
atkConnect(conID, 'Graphics', '* / Satellite / SatInclinationChange Basic LineWidth 2');
%设置卫星轨道预报器类型为机动规划；第一级目录默认存在一个初始段，一个预报段
atkConnect(conID, 'Astrogator', '* / Satellite / SatInclinationChange SetProp');
%在第一级目录下添加一个预报段，默认段名为 Propagate
atkConnect(conID, 'Astrogator', '* / Satellite / SatInclinationChange InsertSegment MainSequence.SegmentList.- Propagate');
%在第一级目录下添加一个瞄准序列段，默认段名为 Target_Sequence
atkConnect(conID, 'Astrogator', '* / Satellite / SatInclinationChange InsertSegment MainSequence.SegmentList.- Target_Sequence');
%在瞄准序列段内添加一个机动段，默认段名为 Maneuver
atkConnect(conID, 'Astrogator', '* / Satellite / SatInclinationChange InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList.- Maneuver');
%在第一级目录下添加一个预报段；由于已经存在一个预报段，所以默认此段段名为 Propagate1
atkConnect(conID, 'Astrogator', '* / Satellite / SatInclinationChange InsertSegment MainSequence.SegmentList.- Propagate');
%在第一级目录下添加一个瞄准序列段；由于已经存在一个瞄准序列段，所以默认此段段名为 Target_Sequence1
atkConnect(conID, 'Astrogator', '* / Satellite / SatInclinationChange InsertSegment MainSequence.SegmentList.- Target_Sequence');
%在 Target_Sequence1 下添加一个机动段，默认段名为 Maneuver
atkConnect(conID, 'Astrogator', '* / Satellite / SatInclinationChange InsertSegment MainSequence.SegmentList.Target_Sequence1.SegmentList.- Maneuver');
%在第一级目录下添加一个预报段；由于已经存在两个预报段，所以默认此段段名为 Propagate2
atkConnect(conID, 'Astrogator', '* / Satellite / SatInclinationChange InsertSegment MainSequence.SegmentList.- Propagate');
```

```

%在第一级目录下添加一个瞄准序列段；由于已经存在两个瞄准序列段，所以
默认此段段名为 Target_Sequence2
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Target_Sequence');
%在 Target_Sequence2 下添加一个机动段，默认段名为 Maneuver
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.Target_Sequence2.SegmentList.- Maneuver');
%在第一级目录下添加一个预报段；由于已经存在三个预报段，所以此段段名
为 Propagate3
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Propagate');
%设置初始段轨道历元
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Epoch 5 Nov 2022 00:00:00.000
UTCG');
%设置初始段坐标类型为轨道根数
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.CoordinateType "Modified Keplerian" ');
%设置初始段半长轴为 6570000 米
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.sma 6570000 m');
%设置初始段偏心率为 0
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.ecc 0');
%设置初始段轨道倾角为 28 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.inc 28 deg');
%设置初始段升交点赤经为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.RAAN 0');
%设置初始段近拱点角距为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.w 0');
%设置初始段真近点角为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.ta 0');
%设置第一个预报段颜色为红色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate.SegmentColor 4278190335');
%设置第一个预报段停止条件为 Duration
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate.StoppingConditions Duration');
%设置第一个预报段停止条件 Duration 的触发值为 7200 秒

```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate.StoppingConditions.Duration.TripValue 7200 sec');
%设置第一个预报段停止条件 Duration 的误差值为 0.0001 秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate.StoppingConditions.Duration.Tolerance 0.0001 sec');
%设置第一个瞄准序列段内的机动段颜色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.SegmentColor 4278212095');
%设置第一个瞄准序列段内的机动段在脉冲类型下推力坐标轴为 VNC
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.ImpulsiveMnvr.ThrustAxes "Satellite VNC(Earth)" ');
%设置第一个瞄准序列段控制变量为机动段的直角坐标 X
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange AddMCS-SegmentControl MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver ImpulsiveMnvr.Cartesian.X');
%第一个瞄准序列段添加属性页
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence.Profiles Differential_Corrector');
%设置第一个瞄准序列段的控制变量-机动段直角坐标 X 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Active true');
%设置第一个瞄准序列段的控制变量-机动段直角坐标 X 的最大步长为 100 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');
%设置第一个瞄准序列段的控制变量-机动段直角坐标 X 的累计校正量为 2456.42862556706 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Correction 2456.42862556706 m/sec');
%设置第一个瞄准序列段的控制变量-机动段直角坐标 X 的摄动量为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
%设置第一个瞄准序列段的控制变量-机动段直角坐标 X 的归一化参数为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
%设置第一个瞄准序列段内的机动段的约束为 Radius Of Apoapsis

```



```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.Results " Radius Of Apoapsis " ');
%设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver "Radius Of Apoapsis" Active true');
%设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 的收敛误差为 0.1 米
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver "Radius Of Apoapsis" Tolerance 0.1 m');
%设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 的归一化参数为 1 米
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver "Radius Of Apoapsis" Scale 1 m');
%设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 的权重系数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver "Radius Of Apoapsis" Weight 1');
%设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 的期望值为 42160000 米
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver "Radius Of Apoapsis" Desired 42160000 m');
%设置第二个预报段的颜色为亮紫色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate1.SegmentColor 4294902015');
%设置第二个预报段的停止条件为 Apoapsis
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate1.StoppingConditions Apoapsis');
%设置第二个预报段停止条件 Apoapsis 的重复次数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate1.StoppingConditions.Apoapsis.RepeatCount 1');
%设置第二个预报段停止条件 Apoapsis 的误差值为 0.0001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate1.StoppingConditions.Apoapsis.Tolerance 0.0001');
%设置第二个瞄准序列段的机动段颜色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.SegmentColor -256');
%设置第二个瞄准序列段内的机动段没冲类型下推力坐标轴为 VNC
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.ImpulsiveMnvr.ThrustAxes "Satellite VNC(Earth)" ');
%设置第二个瞄准序列段的控制变量为机动段直角坐标 X

```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange AddMCS-
SegmentControl MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneu-
ver ImpulsiveMnvr.Cartesian.X');
%第二个瞄准序列段添加属性页
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence1.Profiles Differential_Corrector');
%设置第二个瞄准序列段的控制变量-机动段直角坐标 X 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Active true');
%设置第二个瞄准序列段的控制变量-机动段直角坐标 X 的最大步长为 100 米每
秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');
%设置第二个瞄准序列段的控制变量-机动段直角坐标 X 的累计校正量为
1480.06329844802 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Correction 1480.06329844802 m/sec');
%设置第二个瞄准序列段的控制变量-机动段直角坐标 X 的摄动量为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
%设置第二个瞄准序列段的控制变量-机动段直角坐标 X 的归一化参数为 1 米每
秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
%设置第二个瞄准序列段内的机动段约束为 Eccentricity
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence1.SegmentList.Maneuver.Results "Eccentricity"
');
%设置第二个瞄准序列段内的机动段约束 Eccentricity 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Active true');
%设置第二个瞄准序列段内的机动段约束收敛误差为 0.001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Tolerance 0.001');
%设置第二个瞄准序列段内的机动段约束归一化参数为 1

```

```

atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Scale 1');
%设置第二个瞄准序列段内的机动段约束权重系数为 1
atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Weight 1');
%设置第二个瞄准序列段内的机动段约束期望值为 0
atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Desired 0');
%设置第三个预报段的颜色为绿色
atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate2.SegmentColor 4278255360');
%设置第三个预报段的停止条件为 AscendingNode
atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate2.StoppingConditions AscendingNode');
%设置第三个预报段的停止条件 AscendingNode 的重复次数为 2
atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate2.StoppingConditions.AscendingNode.RepeatCount
2');
%设置第三个预报段的停止条件 AscendingNode 的误差值为 0.000001
atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate2.StoppingConditions.AscendingNode.Tolerance
0.000001');
%设置第三个瞄准序列段的机动段颜色
atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence2.SegmentList.Maneuver.SegmentColor -
65536');
%设置第三个瞄准序列段内的机动段脉冲属性下推力坐标轴为 VNC
atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence2.SegmentList.Maneuver.Impul-
siveMnvr.ThrustAxes "Satellite VNC(Earth)" ');
%设置第三个瞄准序列段的控制变量为机动段的直角坐标 X
atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange AddMCS-
SegmentControl MainSequence.SegmentList.Target_Sequence2.SegmentList.Maneu-
ver ImpulsiveMnvr.Cartesian.X');
%第三个瞄准序列段添加属性页
atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence2.Profiles Differential_Corrector');
%设置第三个瞄准序列段控制变量-机动段直角坐标 X 为使用状态
atkConnect(conID, 'Astrogator', '*/*Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Active true');

```

```

%设置第三个瞄准序列段控制变量-机动段直角坐标 X 的最大步长为 100 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');
%设置第三个瞄准序列段控制变量-机动段直角坐标 X 的累计校正量为-
359.983751789857 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Correction -359.983751789857 m/sec');
%设置第三个瞄准序列段控制变量-机动段直角坐标 X 摄动量为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
%设置第三个瞄准序列段控制变量-机动段直角坐标 X 归一化参数为 1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
%设置第三个瞄准序列段的控制变量为机动段直角坐标 Y
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange AddMCS-
SegmentControl MainSequence.SegmentList.Target_Sequence2.SegmentList.Manue-
ver ImpulsiveMnvr.Cartesian.Y');
%设置第三个瞄准序列段控制变量-机动段直角坐标 Y 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y Active true');
%设置第三个瞄准序列段控制变量-机动段直角坐标 Y 的最大步长为 100 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y MaxStep 100 m/sec');
%设置第三个瞄准序列段控制变量-机动段直角坐标 Y 的累计校正量为-
1444.72488992759 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y Correction -1444.72488992759 m/sec');
%设置第三个瞄准序列段控制变量-机动段直角坐标 Y 的摄动量为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y Perturbation 0.1 m/sec');
%设置第三个瞄准序列段控制变量-机动段直角坐标 Y 的归一化参数为 1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y Scale 1 m/sec');
%设置第三个瞄准序列段内的机动段约束为 Inclination 和 Eccentricity

```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence2.SegmentList.Maneuver.Results "Inclination" "Eccentricity" ');
%设置第三个瞄准序列段内机动段约束 Inclination 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver "Inclination" Active true');
%设置第三个瞄准序列段内机动段约束 Inclination 的收敛误差为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver "Inclination" Tolerance 0 deg');
%设置第三个瞄准序列段内机动段约束 Inclination 的归一化参数为 1 弧度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver "Inclination" Scale 1 rad');
%设置第三个瞄准序列段内机动段约束 Inclination 的权重系数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver "Inclination" Weight 1');
%设置第三个瞄准序列段内机动段约束 Inclination 的期望值为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver "Inclination" Desired 0 deg');
%设置第三个瞄准序列段内机动段约束 Eccentricity 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver "Eccentricity" Active true');
%设置第三个瞄准序列段内机动段约束 Eccentricity 的收敛误差为 0.001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver "Eccentricity" Tolerance 0.001');
%设置第三个瞄准序列段内机动段约束 Eccentricity 的收敛误差为 0.001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver "Eccentricity" Scale 1');
%设置第三个瞄准序列段内机动段约束 Eccentricity 的权重系数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver "Eccentricity" Weight 1');
%设置第三个瞄准序列段内机动段约束 Eccentricity 期望值为 0
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver "Eccentricity" Desired 0');
%设置第四个预报段的颜色为黄色

```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate3.SegmentColor 4278255615');
%设置第四个预报段的停止条件为 Duration
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate3.StoppingConditions Duration');
%设置第四个预报段停止条件 Duration 的触发值为 129600 秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate3.StoppingConditions.Duration.TripValue 129600 sec');
%设置第四个预报段停止条件 Duration 的误差值为 0.00001 秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate3.StoppingConditions.Duration.Tolerance 0.00001 sec');
%运行此算例
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange RunMCS');

```

1.2.5.2.5. 保存场景后与 ATK 断开连接。

```

//保存场景
atkConnect(conID, 'Save', '/ *');
//与 ATK 断开连接
atkClose(conID);

```

1.3. Java 客户端使用介绍

Java 客户端是 ATK 软件向用户提供的一种可以输入脚本命令的窗口，Java 客户端由 ATK 提供中，在 ATK 软件安装包中，点击“IntegratingWithATK”文件夹-“Connect”文件夹-“win”文件夹-“Java”文件夹-

“startATKConnector.bat”，即可弹出客户端窗口。目前界面属性窗口不具备实时更新功能，故而设置属性后需重新打开对象属性界面属性数据才会更新。以下为 Java 客户端提供的接口函数介绍。目前 Java 客户端与 ATK 使用 atkOpen 命令进行连接，使用 atkConnect 命令进行属性设置，使用 atkClose 命令与 ATK 断开连接。

客户端使用 Java1.8.0 版本，打开 Java 客户端与 ATK 进行连接，如下图：

名称	修改日期	类型	大小
jre1.8.0_311	2024/1/6 16:03	文件夹	
ATKConnectorTools.jar	2023/12/27 10:41	Executable Jar File	69 KB
startATKConnector.bat	2023/12/27 10:23	Windows 批处理...	1 KB

图 1-58 ATK 打开界面

客户端提供 Java 可执行程序；提供 ATKConnectorTools.jar，用来完成 ATK 与 Java 客户端的数据传输与解析，其中包含 `atkOpen`、`atkConnect`、`atkClose` 函数用以完成 ATK 与 Java 客户端的链接与参数设置，使用 `atkOpen` 进行连接，使用 `atkConnect` 进行属性设置。使用 `atkClose` 与 ATK 断开连接；提供 `startATKConnector.bat` 文件用来实现客户端命令的输入输出。

1.3.1. `atkOpen`

用法：`conID = atkOpen('hostStr', PortStr)`

说明：`conID`-连接句柄

`hostStr`-进行连接的网络地址

`PortStr`-进行连接的端口号

1.3.2. `atkConnect`

用法：`rtnData = atkConnect(conID, 'command', 'objPath', 'cmdParamString')`

说明：`conID`-来自 `atkOpen` 的句柄

`Command`-具体请查看 `Connect` 命令库

`objPath`-接受命令的对象路径

`cmdParamString`-命令属性字符串

`rtnData`-从 `atk` 返回的响应的字符串

举例：`atkConnect(conID, 'Graphics', '*/Satellite/Satellite1 SetColor 12');`

1.3.3. `atkClose`

用法：`atkClose(conID)`

说明：`conID`-来自 `atkOpen` 的连接句柄

1.3.4. 案例操作流程

以轨道倾角改变场景为例，此案例实现是从初始近地轨道到指定半径的地球同步轨道的转移轨道机动规划。以下为通过 Java 客户端在 ATK 软件中新建想定并设置参数的使用介绍。

使用 Java 客户端向 ATK 发送命令前，必须打开 ATK，不需要打开场景；若需新建场景，需将已打开的场景关闭。ATK 网络通信端口为 6655。

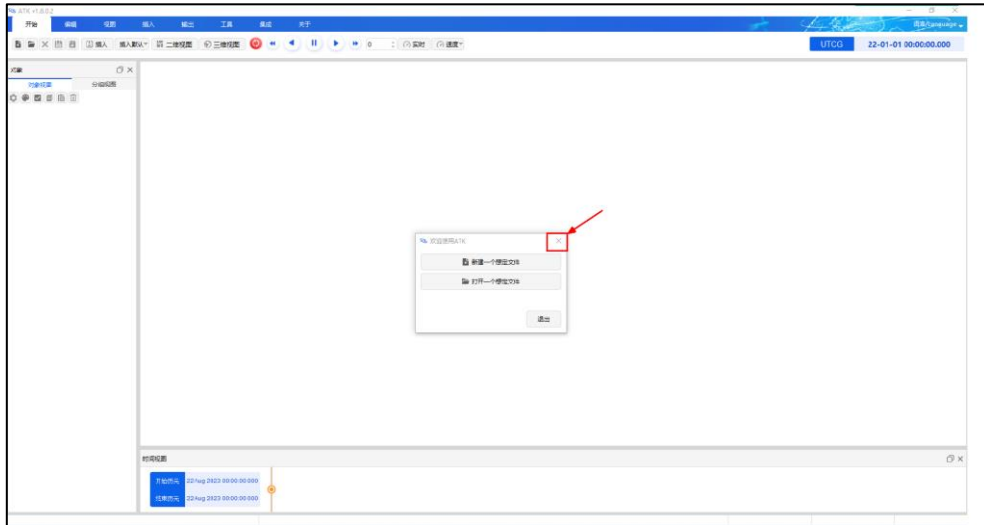


图 1-59 ATK 打开界面

1.3.4.1. 从命令窗口新建场景

注意：若有已打开场景，请先关闭场景。

1.3.4.1.1. 第一步：打开 JAVA 客户端，在命令行窗口输入命令与 ATK 进行连接。

与 ATK 进行连接使用 `atkOpen` 命令。

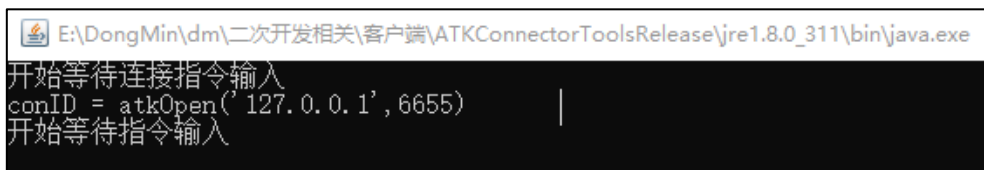


图 1-60 与 ATK 连接

1.3.4.1.2. 第二步：输入新建命令，新建想定并设置场景分析时间。

以新建倾角改变轨道场景为例，使用 `New` 命令。

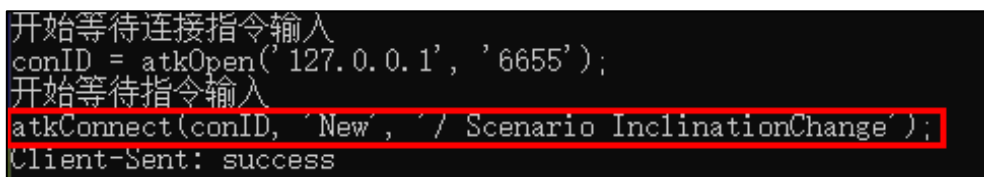


图 1-61 新建场景命令

执行命令后，相应界面如下。



图 1-62 新建的场景

使用 [SetAnalysisTimePeriod](#) 命令设置场景开始结束时间；使用 [Animate](#) 命令设置场景仿真状态。

```
开始等待连接指令输入
conID = atkOpen(127.0.0.1, '6655');
开始等待指令输入
atkConnect(conID, 'New', '/ Scenario InclinationChange');
Client-Sent: success
atkConnect(conID, 'SetAnalysisTimePeriod', '* "5 Nov 2022 00:00:00.000" "8 Nov 2022 00:00:00.000" ');
Client-Sent: success
atkConnect(conID, 'Animate', '* Reset');
Client-Sent: success
```

图 1-63 设置场景属性命令

1.3.4.1.3. 第三步：新建卫星对象并设置对象属性。

可以通过 New 命令，新建对象，以新建卫星为例。

```
开始等待连接指令输入
conID = atkOpen(127.0.0.1, '6655');
开始等待指令输入
atkConnect(conID, 'New', '/ Scenario InclinationChange');
Client-Sent: success
atkConnect(conID, 'SetAnalysisTimePeriod', '* "5 Nov 2022 00:00:00.000" "8 Nov 2022 00:00:00.000" ');
Client-Sent: success
atkConnect(conID, 'Animate', '* Reset');
Client-Sent: success
atkConnect(conID, 'New', '/ Satellite SatInclinationChange');
Client-Sent: success
```

图 1-64 新建卫星命令

执行命令后，相应界面如下。

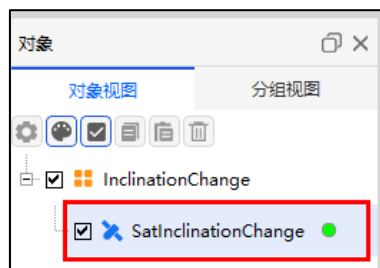


图 1-65 新建的卫星

以设置卫星属性为例，具体命令格式请参考 Connect 模式功能介绍。

```

开始等待连接指令输入
conID = atkOpen('127.0.0.1', '6655');
开始等待指令输入
atkConnect(conID, 'New', '/ Scenario InclinationChange');
Client-Sent: success
atkConnect(conID, 'SetAnalysisTimePeriod', '* "5 Nov 2022 00:00:00.000" "8 Nov 2022 00:00:00.000" ');
Client-Sent: success
atkConnect(conID, 'Animate', '* Reset');
Client-Sent: success
atkConnect(conID, 'New', '/ Satellite SatInclinationChange');
Client-Sent: success
atkConnect(conID, 'Graphics', '* /Satellite/SatInclinationChange Basic LineWidth 2');
Client-Sent: success
atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange SetProp');
Client-Sent: success
atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Propagate');
Client-Sent: success
atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Target_Sequence');
Client-Sent: success
atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList - Maneuver');
Client-Sent: success
atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Propagate');
Client-Sent: success
atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList - Target_Sequence');
Client-Sent: success
atkConnect(conID, 'Astrogator', '* /Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList.Target_Sequence1.SegmentList - Maneuver');

```

图 1-66 设置卫星的属性命令

执行命令后，可在界面右键点击对象，选择属性按钮，查看设置结果。

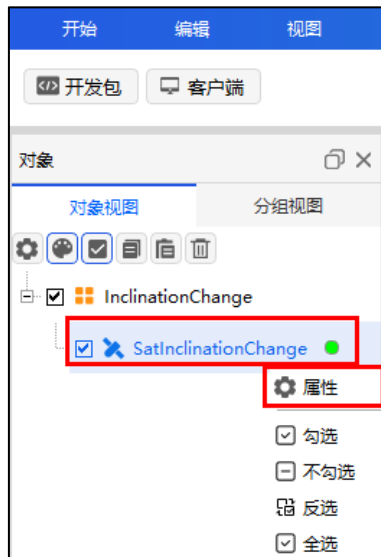


图 1-67 右键点击卫星，属性按钮位置

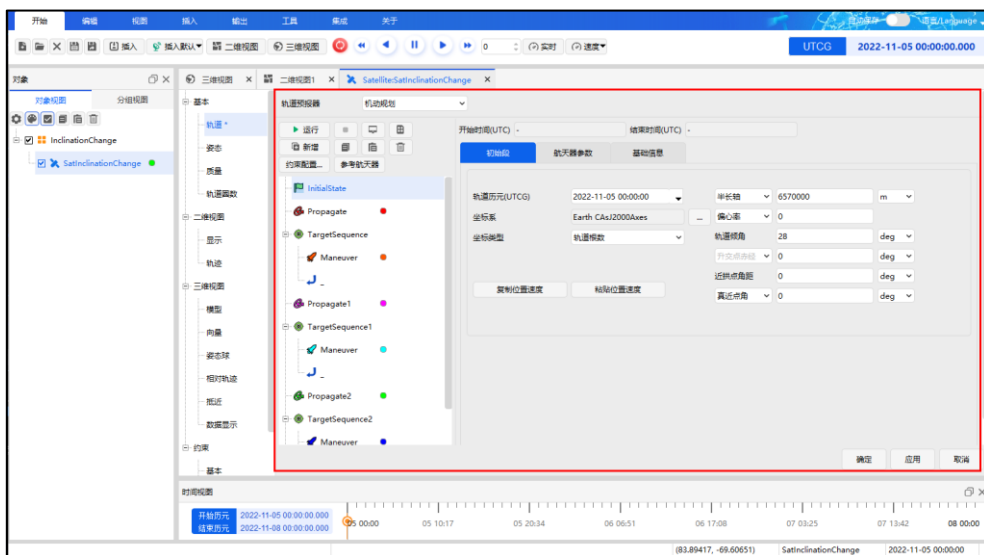


图 1-68 卫星属性界面

1.3.4.1.4. 第四步：开始仿真；属性设置完成后，请关闭连接。

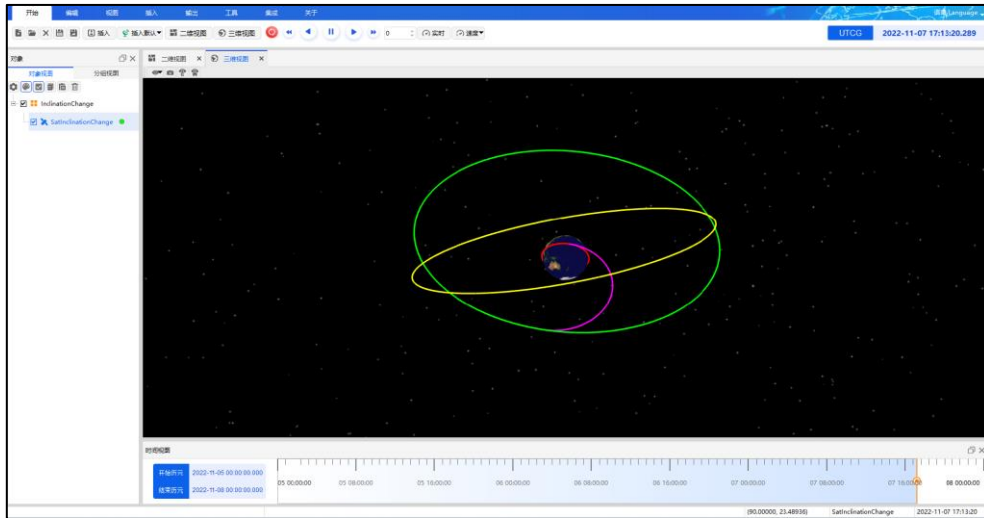


图 1-69 场景仿真界面

使用 [Save](#) 命令保存场景，若保存脚本文件，请在客户端界面点击保存按钮。

```
atkConnect(conID, 'Save', '/ *');  
Client-Sent: success
```

图 1-70 场景保存命令

场景及对象设置完成后，使用 [atkClose](#) 命令与 ATK 断开连接。

```
atkClose(conID)  
Client-Sent: success
```

图 1-71 与 ATK 断开连接

1.3.5. 案例展示

1.3.5.1. InclinationChange 场景案例

通过二次开发 Connect 模式，使用 Java 客户端创建 InclinationChange 场景，首次设置为机动规划，会默认添加一个初始段，必须保持 ATK 为打开状态，具体命令分类包括：

- (1) 与 ATK 进行连接。
- (2) 新建场景和多个卫星。
- (3) 设置场景属性。
- (4) 设置卫星轨道属性。

(5) 保存场景后与 ATK 断开连接。

1.3.5.2. 脚本展示

1.3.5.2.1. 与 ATK 进行连接。

```
//与 ATK 进行连接  
conID = atkOpen();
```

1.3.5.2.2. 新建场景和多个卫星。

```
//新建一个名为 InclinationChange 的场景  
atkConnect(conID, 'New', '/ Scenario InclinationChange');  
//新建一个名为 SatInclinationChange 的卫星  
atkConnect(conID, 'New', '/ Satellite SatInclinationChange');
```

1.3.5.2.3. 设置场景属性。

```
//使用 UTCG 格式设置场景分析时间段，开始时间为 2022-11-05 零点，结束时间为 2022-11-08 零点  
atkConnect(conID, 'SetAnalysisTimePeriod', '* "5 Nov 2022 00:00:00.000" "8 Nov 2022 00:00:00.000" ');  
//仿真重置  
atkConnect(conID, 'Animate', '* Reset');
```

1.3.5.2.4. 设置卫星轨道属性。

```
//设置卫星轨迹线宽为 2  
atkConnect(conID, 'Graphics', '*/Satellite/SatInclinationChange Basic LineWidth 2');  
//设置卫星轨道预报器类型为机动规划；第一级目录默认存在一个初始段，一个预报段  
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetProp');  
//在第一级目录下添加一个预报段，默认段名为 Propagate  
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList.- Propagate');  
//在第一级目录下添加一个瞄准序列段，默认段名为 Target_Sequence  
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList.- Target_Sequence');  
//在瞄准序列段内添加一个机动段，默认段名为 Maneuver  
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList.- Maneuver');  
//在第一级目录下添加一个预报段；由于已经存在一个预报段，所以默认此段段名为 Propagate1
```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Propagate');
//在第一级目录下添加一个瞄准序列段；由于已经存在一个瞄准序列段，所以默
认此段段名为 Target_Sequence1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Target_Sequence');
//在 Target_Sequence1 下添加一个机动段，默认段名为 Maneuver
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.Target_Sequence1.SegmentList.- Maneuver');
//在第一级目录下添加一个预报段；由于已经存在两个预报段，所以默认此段段
名为 Propagate2
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Propagate');
//在第一级目录下添加一个瞄准序列段；由于已经存在两个瞄准序列段，所以默
认此段段名为 Target_Sequence2
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Target_Sequence');
//在 Target_Sequence2 下添加一个机动段，默认段名为 Maneuver
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.Target_Sequence2.SegmentList.- Maneuver');
//在第一级目录下添加一个预报段；由于已经存在三个预报段，所以此段段名为
Propagate3
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange InsertSegment
MainSequence.SegmentList.- Propagate');
//设置初始段轨道历元
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Epoch 5 Nov 2022 00:00:00.000
UTCG');
//设置初始段坐标类型为轨道根数
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.CoordinateType "Modified Keplerian" ');
//设置初始段半长轴为 6570000 米
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.sma 6570000 m');
//设置初始段偏心率为 0
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.ecc 0');
//设置初始段轨道倾角为 28 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.inc 28 deg');
//设置初始段升交点赤经为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.RAAN 0');
//设置初始段近拱点角距为 0 度

```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.w 0');
//设置初始段真近点角为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Initial_State.InitialState.Keplerian.ta 0');
//设置第一个预报段颜色为红色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate.SegmentColor 4278190335');
//设置第一个预报段停止条件为 Duration
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate.StoppingConditions Duration');
//设置第一个预报段停止条件 Duration 的触发值为 7200 秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate.StoppingConditions.Duration.TripValue 7200 sec');
//设置第一个预报段停止条件 Duration 的误差值为 0.0001 秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate.StoppingConditions.Duration.Tolerance 0.0001 sec');
//设置第一个瞄准序列段内的机动段颜色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence.SegmentList.Maneuver.SegmentColor
4278212095');
//设置第一个瞄准序列段内的机动段在脉冲类型下推力坐标轴为 VNC
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence.SegmentList.Maneuver.Impul-
siveMnvr.ThrustAxes "Satellite VNC(Earth)" ');
//设置第一个瞄准序列段控制变量为机动段的直角坐标 X
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange AddMCS-
SegmentControl MainSequence.SegmentList.Target_Sequence.SegmentList.Maneu-
ver ImpulsiveMnvr.Cartesian.X');
//第一个瞄准序列段添加属性页
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence.Profiles Differential_Corrector');
//设置第一个瞄准序列段的控制变量-机动段直角坐标 X 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc-
tor Maneuver ImpulsiveMnvr.Cartesian.X Active true');
//设置第一个瞄准序列段的控制变量-机动段直角坐标 X 的最大步长为 100 米每
秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc-
tor Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');
//设置第一个瞄准序列段的控制变量-机动段直角坐标 X 的累计校正量为
2456.42862556706 米每秒

```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc-
tor Maneuver ImpulsiveMnvr.Cartesian.X Correction 2456.42862556706 m/sec');
//设置第一个瞄准序列段的控制变量-机动段直角坐标 X 的摄动量为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc-
tor Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
//设置第一个瞄准序列段的控制变量-机动段直角坐标 X 的归一化参数为 0.1 米
每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc-
tor Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
//设置第一个瞄准序列段内的机动段的约束为 Radius Of Apoapsis
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence.SegmentList.Maneuver.Results " Radius Of
Apoapsis " ');
//设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Cor-
rector Maneuver "Radius Of Apoapsis" Active true');
//设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 的收敛误差为 0.1 米
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Cor-
rector Maneuver "Radius Of Apoapsis" Tolerance 0.1 m');
//设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 的归一化参数为 1 米
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Cor-
rector Maneuver "Radius Of Apoapsis" Scale 1 m');
//设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 的权重系数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Cor-
rector Maneuver "Radius Of Apoapsis" Weight 1');
//设置第一个瞄准序列段的机动段约束 Radius Of Apoapsis 的期望值为 42160000
米
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Cor-
rector Maneuver "Radius Of Apoapsis" Desired 42160000 m');
//设置第二个预报段的颜色为亮紫色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate1.SegmentColor 4294902015');
//设置第二个预报段的停止条件为 Apoapsis
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate1.StoppingConditions Apoapsis');
//设置第二个预报段停止条件 Apoapsis 的重复次数为 1

```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate1.StoppingConditions.Apoapsis.RepeatCount 1');
//设置第二个预报段停止条件 Apoapsis 的误差值为 0.0001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate1.StoppingConditions.Apoapsis.Tolerance 0.0001');
//设置第二个瞄准序列段的机动段颜色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.SegmentColor -256');
//设置第二个瞄准序列段内的机动段没冲类型下推力坐标轴为 VNC
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.ImpulsiveMnvr.ThrustAxes "Satellite VNC(Earth)" ');
//设置第二个瞄准序列段的控制变量为机动段直角坐标 X
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange AddMCS-SegmentControl MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver ImpulsiveMnvr.Cartesian.X');
//第二个瞄准序列段添加属性页
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence1.Profiles Differential_Corrector');
//设置第二个瞄准序列段的控制变量-机动段直角坐标 X 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCControlValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Active true');
//设置第二个瞄准序列段的控制变量-机动段直角坐标 X 的最大步长为 100 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCControlValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');
//设置第二个瞄准序列段的控制变量-机动段直角坐标 X 的累计校正量为 1480.06329844802 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCControlValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Correction 1480.06329844802 m/sec');
//设置第二个瞄准序列段的控制变量-机动段直角坐标 X 的摄动量为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCControlValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
//设置第二个瞄准序列段的控制变量-机动段直角坐标 X 的归一化参数为 1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCControlValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
//设置第二个瞄准序列段内的机动段约束为 Eccentricity

```



```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.Results "Eccentricity"');
//设置第二个瞄准序列段内的机动段约束 Eccentricity 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corrector Maneuver "Eccentricity" Active true');
//设置第二个瞄准序列段内的机动段约束收敛误差为 0.001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corrector Maneuver "Eccentricity" Tolerance 0.001');
//设置第二个瞄准序列段内的机动段约束归一化参数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corrector Maneuver "Eccentricity" Scale 1');
//设置第二个瞄准序列段内的机动段约束权重系数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corrector Maneuver "Eccentricity" Weight 1');
//设置第二个瞄准序列段内的机动段约束期望值为 0
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corrector Maneuver "Eccentricity" Desired 0');
//设置第三个预报段的颜色为绿色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate2.SegmentColor 4278255360');
//设置第三个预报段的停止条件为 AscendingNode
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate2.StoppingConditions AscendingNode');
//设置第三个预报段的停止条件 AscendingNode 的重复次数为 2
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate2.StoppingConditions.AscendingNode.RepeatCount 2');
//设置第三个预报段的停止条件 AscendingNode 的误差值为 0.000001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Propagate2.StoppingConditions.AscendingNode.Tolerance 0.000001');
//设置第三个瞄准序列段的机动段颜色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence2.SegmentList.Maneuver.SegmentColor -65536');
//设置第三个瞄准序列段内的机动段脉冲属性下推力坐标轴为 VNC

```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence2.SegmentList.Maneuver.ImpulsiveMnvr.ThrustAxes "Satellite VNC(Earth)" ');
//设置第三个瞄准序列段的控制变量为机动段的直角坐标 X
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange AddMCS-SegmentControl MainSequence.SegmentList.Target_Sequence2.SegmentList.Maneuver ImpulsiveMnvr.Cartesian.X');
//第三个瞄准序列段添加属性页
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSequence.SegmentList.Target_Sequence2.Profiles Differential_Corrector');
//设置第三个瞄准序列段控制变量-机动段直角坐标 X 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSControlValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Active true');
//设置第三个瞄准序列段控制变量-机动段直角坐标 X 的最大步长为 100 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSControlValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');
//设置第三个瞄准序列段控制变量-机动段直角坐标 X 的累计校正量为-359.983751789857 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSControlValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Correction -359.983751789857 m/sec');
//设置第三个瞄准序列段控制变量-机动段直角坐标 X 摄动量为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSControlValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
//设置第三个瞄准序列段控制变量-机动段直角坐标 X 归一化参数为 1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSControlValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
//设置第三个瞄准序列段的控制变量为机动段直角坐标 Y
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange AddMCS-SegmentControl MainSequence.SegmentList.Target_Sequence2.SegmentList.Maneuver ImpulsiveMnvr.Cartesian.Y');
//设置第三个瞄准序列段控制变量-机动段直角坐标 Y 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSControlValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.Y Active true');
//设置第三个瞄准序列段控制变量-机动段直角坐标 Y 的最大步长为 100 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSControlValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.Y MaxStep 100 m/sec');

```

```

//设置第三个瞄准序列段控制变量-机动段直角坐标 Y 的累计校正量为-
1444.72488992759 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y Correction -1444.72488992759 m/sec');
//设置第三个瞄准序列段控制变量-机动段直角坐标 Y 的摄动量为 0.1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y Perturbation 0.1 m/sec');
//设置第三个瞄准序列段控制变量-机动段直角坐标 Y 的归一化参数为 1 米每秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
trolValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differential_Cor-
rector Maneuver ImpulsiveMnvr.Cartesian.Y Scale 1 m/sec');
//设置第三个瞄准序列段内的机动段约束为 Inclination 和 Eccentricity
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Target_Sequence2.SegmentList.Maneuver.Results "Inclination"
"Eccentricity" ');
//设置第三个瞄准序列段内机动段约束 Inclination 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Inclination" Active true');
//设置第三个瞄准序列段内机动段约束 Inclination 的收敛误差为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Inclination" Tolerance 0 deg');
//设置第三个瞄准序列段内机动段约束 Inclination 的归一化参数为 1 弧度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Inclination" Scale 1 rad');
//设置第三个瞄准序列段内机动段约束 Inclination 的权重系数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Inclination" Weight 1');
//设置第三个瞄准序列段内机动段约束 Inclination 的期望值为 0 度
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Inclination" Desired 0 deg');
//设置第三个瞄准序列段内机动段约束 Eccentricity 为使用状态
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Active true');
//设置第三个瞄准序列段内机动段约束 Eccentricity 的收敛误差为 0.001

```

```

atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Tolerance 0.001');
//设置第三个瞄准序列段内机动段约束 Eccentricity 的收敛误差为 0.001
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Scale 1');
//设置第三个瞄准序列段内机动段约束 Eccentricity 的权重系数为 1
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Weight 1');
//设置第三个瞄准序列段内机动段约束 Eccentricity 期望值为 0
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetMCSCon-
straintValue MainSequence.SegmentList.Target_Sequence2.Profiles.Differen-
tial_Corrector Maneuver "Eccentricity" Desired 0');
//设置第四个预报段的颜色为黄色
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate3.SegmentColor 4278255615');
//设置第四个预报段的停止条件为 Duration
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate3.StoppingConditions Duration');
//设置第四个预报段停止条件 Duration 的触发值为 129600 秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate3.StoppingConditions.Duration.TripValue 129600
sec');
//设置第四个预报段停止条件 Duration 的误差值为 0.00001 秒
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange SetValue MainSe-
quence.SegmentList.Propagate3.StoppingConditions.Duration.Tolerance 0.00001
sec');
//运行此算例
atkConnect(conID, 'Astrogator', '*/Satellite/SatInclinationChange RunMCS');

```

1.3.5.2.5. 保存场景后与 ATK 断开连接。

```

//保存场景
atkConnect(conID, 'Save', '/ *');
//与 ATK 断开连接
atkClose(conID);

```

2. ATK.Connect 模式 MBSE 接口

2.1. 与插件相关的 SysML 语言要素

在联合仿真过程中会用到 SysML 语言模块定义图、状态机图、活动图进行

建模，在这里介绍一下这三种图，便于后续理解联合仿真搭建过程。

2.1.1. 模块定义图

模块定义图的图类型缩写为 BDD，是 SysML 中最常见的一种图，用于定义模块的特征和模块之间的关系。

1) 主要元素

模块是 SysML 结构中的基本单元，可以使用模块为系统中或者系统外部环境中任意一种感兴趣的实体类型创建模型。模块代表的是类型或者类别，而不是一个具体的实例。如下图所示：



图 2-1 模块

模块的总体分为两大类：结构属性和行为属性。结构属性包括组成部分属性、引用属性、值属性、约束属性、流属性、端口；行为属性包括操作、信号接收。它们表示系统的状态和系统可能表现出的行为，在模块定义图中模块的属性分类放置在分隔区内。

2) 模块的属性

结构属性：组成部分属性/部件属性代表模块内部的结构，换句话讲就是模块是由部件构成的，表示的是一种所属关系，它表示模块的内部结构。如下图所示：

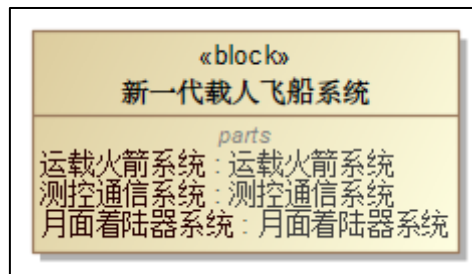


图 2-2 部件属性

引用属性：代表模块外部的一种结构，它可能位于另一个模块的内部，不是所属关系，而是“需要”的关系，为当前模块提供功能、数据或是信息。如下图所示：



图 2-3 引用属性

值属性：可以代表一个数字（某种类型）、一个布尔值或者一个字符串。较为常见的是被赋予一个数字。如下图所示：



图 2-4 值属性

约束属性：一般代表一种数学关系，等式或是不等式，约束属性的类型是一个约束模块，约束模块中包含了可以重复使用的约束关系。如下图所示：



图 2-5 约束属性

端口：一个特殊的部件，代表交换信息，它是专门用来和外界交换信息的。端口有三种：完整端口、代理端口和流端口。如下图所示：

完整端口可以把它当作一个独立的部件来看待，完整端口是自己直接处理输入的信息、输出处理后的信息。

代理端口自己并没有处理信息的功能，而是通过绑定连接器绑定到一个内部的模块。

流端口是具有方向的，流属性的方向有“进”“出”或“进出”。

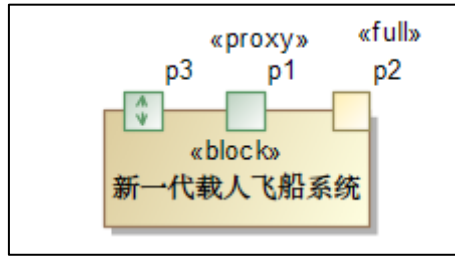


图 2-6 端口

行为属性：

操作：代表客户端调用模块的时候它所执行的行为，是由调用事件触发的，操作代表的是一种同步行动，但是它的实现需要通过对应方法的行为图来描述。

接收：表示模块接收信号并进行处理的一个方法。接收一定对应模型中某个地方定义的信号它的参数和信号的属性对应。接收是异步行为，发送信号的客户端不会等待回复。

3) 模块定义图间的关系

模块之间存在三种主要类型关系：关联、泛化和依赖。关联是表达模块、引用属性、部件属性这些系统结构之间关系的一种记号，关联默认为引用关联。关联关系中需要区分组合关联和聚合关联。

组合关联：表示结构上的分解，组合端的模块实例由一些组成部分端模块实例结合而成，可以理解为“包含”的关系，一个组成部分只能属于一个组合体。组合关联标识中有箭头是单向访问，无箭头则是双向访问。如下图所示：

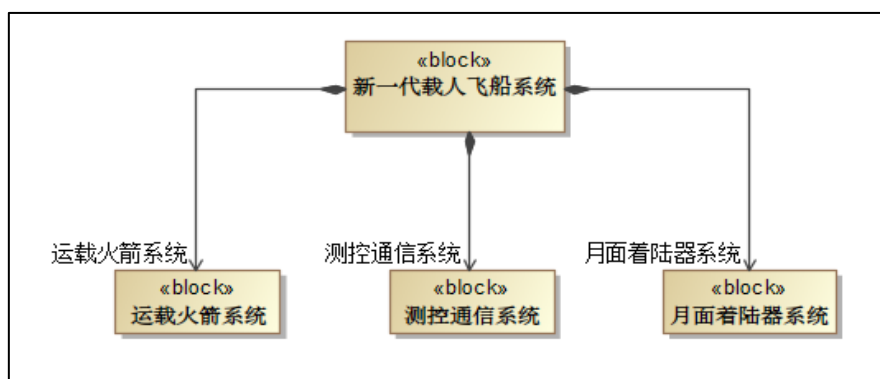


图 2-7 组合关联

泛化关系：表示两种元素之间的继承关系，一个一般化的元素（超类型），以及一个更为具体的元素（子类型），这种从子类型向超类型读作“.....是一种.....”。泛化关系是可传递的，通过泛化关系传递到最终层级。除此之外，泛化表示子类型会继承超类型的所有特性，还可能会拥有超类型所不具备的其他

特性。如下图所示：

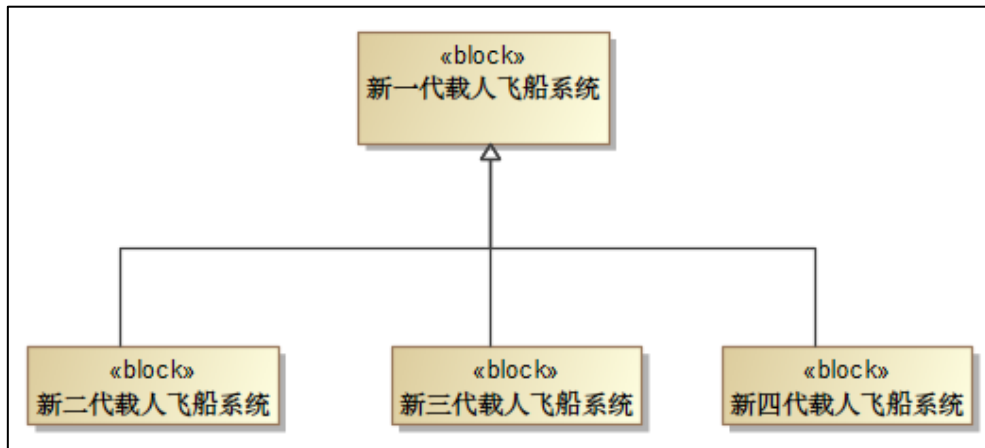


图 2-8 泛化关系

依赖关系：意味着模型中的一种元素、客户端，依赖于模型中的另一种元素、提供者。当提供者元素发生改变时，客户端元素可能也需要改变。建立依赖关系确定二者之间的可追踪性。

2.1.2. 状态机图

状态机图是能够用于说明系统动态行为信息的三种 SysML 图中的一种。可以在状态机图上显示各种各样的状态，并且可以指定四种类型的事件，从而在运行的系统中触发哪些状态之间的转换。SysML 还可以使用正交区域对基于状态的并发行为建模。主要包含元素如下：

状态：是指对象的生命周期中的某个条件或者状况，在此期间对象将满足某些条件、执行某些活动或者等待某些事件。如下图所示：

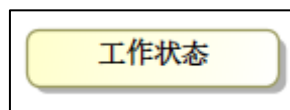


图 2-9 状态

初始和最终状态：状态机图的初始状态，称为初始伪状态，用实心圆圈表示，此状态将转换到第一个真实状态。状态机图的最终状态显示为同心圆。如下图所示：

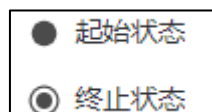


图 2-10 初始状态和终止状态

转换：代表的是从一种状态到另一种状态的改变。如下图所示：

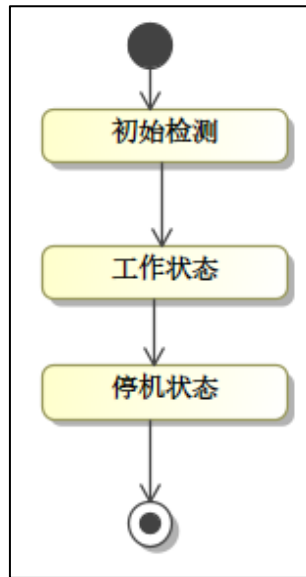


图 2-11 转换

事件是在系统模型中定义的一种元素，它定义了能够在实际系统中触发行为的事件类型。SysML 定义了如下四种类型的事件，如下图所示：

- a) 信号事件：可以理解为发送和接收信号
- b) 调用事件：对应于对操作的过程调用
- c) 时间事件：在指定时间过去后发生的定时事件，相对时间（after）与绝对时间（at）
- d) 改变事件：每当满足指定条件时就会发生更改事件

Trigger	
Event Type	<UNSPECIFIED>
Trigger	<UNSPECIFIED>
Event Element	AnyReceiveEvent
Port	CallEvent
Effect	ChangeEvent
Behavior Type	SignalEvent
Behavior Element	TimeEvent

图 2-12 事件

动作：是一个可执行的原子计算，是不可中断的，包括操作调用、对象的创建或销毁，或者向对象发送信号。动作可能会导致模型状态的改变或者一个值的返回。

入口动作：是进入那个状态执行的第一个行为。如下图所示：



图 2-13 入口动作

出口动作：是离开那种状态之前执行的最后一个行为。如下图所示：

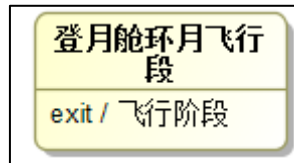


图 2-14 出口动作

复合状态：拥有子状态，如下图所示：

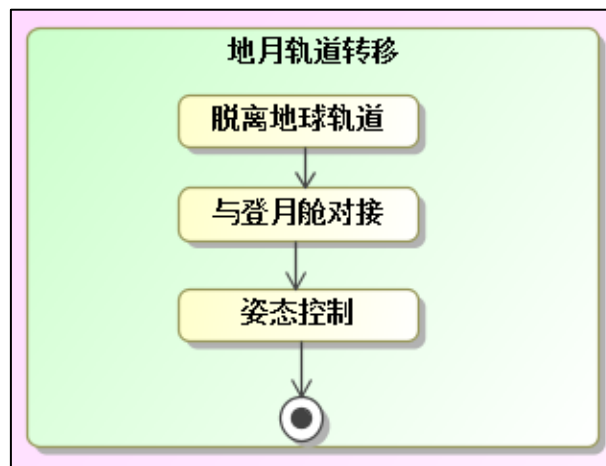


图 2-15 复合状态

历史状态：是一个伪状态，其目的是记住从组合状态中退出时所处的子状态，当再次进入组合状态，可直接进入这个子状态，而不是从初态开始。

状态机图通常用于描述对象的状态相关的行为。一个对象根据它所处的状态对事件做出相应的响应。状态机图非常适合作为详细设计的产出物（即开发的输入），因此，它通常用于系统设计和仿真/代码生成。

2.1.3. 活动图

活动图是 UML 活动图的扩展。它是一个强大的工具，用于表示描述块或其他结构元素行为的动作序列；该序列是使用控制流定义的。动作可以包含输入和输出引脚，这些输入和输出引脚充当从一个动作流到另一个动作的项的缓冲区，因为该动作执行的任务会消耗或产生它们。活动图是一种行为图；它是系统的一种动态视图，随着时间的推移行为和时间的发生序列。与结构图

(BDD、IBD 和参数图) 相对, 结构图都是静态视图, 不会表达任何动态的时间, 或者系统及其环境的变化。

活动图图头命名为 **act**, 活动本身是一种模型元素, 是一种行为。在活动图中, “令牌流” 是贯穿活动图的概念, “令牌流” 分为对象令牌和控制令牌, 要知道令牌并不是一种模型元素, 不需要在系统模型中创建。对象令牌代表的是在活动中流动的事件、能量或者数据的实例。它可以从总体上代表一个活动的输入或者输出, 并且可以代表活动中一个动作的输入或输出。正式的说法是, 一个对象令牌代表模型层级关系中某处创建的模块、值类型或者信号的实例(以定义事件、能量或者数据的类型)。正如你可能期望的, 会有多个对象令牌流动通过活动的单次执行。控制令牌只表示活动的哪个动作在活动执行的特定时刻处于启用状态。可能会有多个控制令牌流过活动的单次执行。那样意味着活动中的多个动作同时处于启用状态。。

接下来详细介绍活动基本动作、对象节点、边、再次阐述动作和控制节点五个方面。

1.基本动作

动作是一种可以存在于活动之中的节点, 它是为活动基本的功能单元建模的节点。一个动作代表某种类型的处理或者转换, 它会在系统操作过程中活动被执行的时候发生。如下图所示:

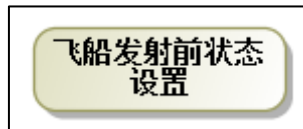


图 2-16 基本动作图例

2.对象节点

对象节点是另一种能够存在于活动之中的节点, 它会对对象令牌通过活动的流建模(其中对象令牌代表的是事件、能量或者数据的实例)。对象节点最常出现在两个动作之间, 以表示第一个动作会产出对象令牌作为输出, 第二个动作会将这些对象令牌作为输入。

- 1) 栓是特殊的对象节点, 是附加在动作上的, 表示动作输入和输出。
如下图所示:

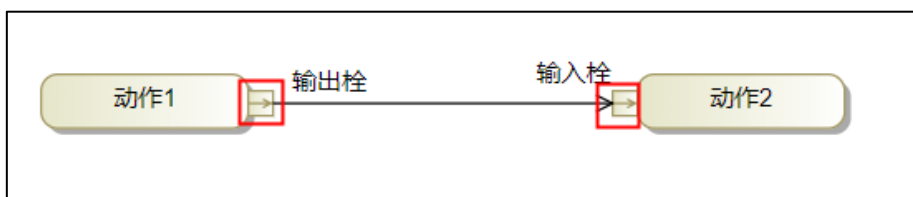


图 2-17 输入栓和输出栓图例

- 2) 活动参数是另一种特殊类型的对象节点。你会把它附加到活动图的外框上，从总体上表示活动的一种输入或者输出。活动参数的标识法是横跨在活动图外框上的矩形，如下图所示：

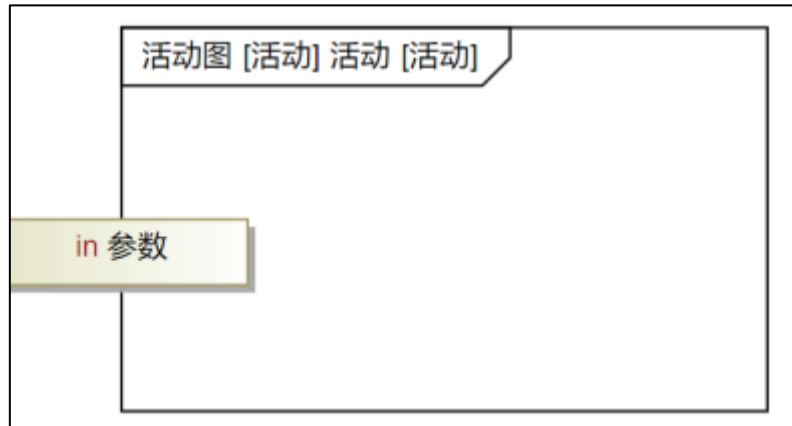


图 2-18 活动参数图例

- 3) 流与非流行为：非流行为意味着在动作完成后才可以进行下一步输入输出，而流行为则是在执行动作过程中同时允许其他输入和输出。

3.边

对象流和控制流是两种类型的边，对象流是用来传输对象令牌的，标识方法是带箭头的实线；控制流是传递控制令牌的边，控制令牌启动下一个动作，标识为带箭头的虚线。如下图所示：

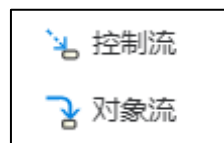


图 2-19 控制流与对象流图例

4.再次阐述动作

再次阐述动作是特别类型的动作，含有调用行为动作、发送信号动作、接收事件动作以及等待时间动作四种动作。

- 1) 调用行为动作是一种特定的动作，它会在启用的时候触发另一种行为。调用行为动作可以把一个高层次的行为分解成系列低层次的行为。如下图所示：

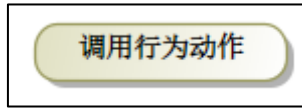


图 2-20 调用行为动作

2) 发送信号动作是异步，不会等待回应就启动发送。如下图所示：

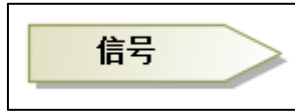


图 2-21 发送信号动作

3) 接收事件动作不仅接收发送信号的动作，而是接收异步的时间事件。如下图所示：



图 2-22 接收信号动作

4) 等待时间动作可以指绝对时间 (at) 或者相对时间 (after)。如下图所示：



图 2-23 等待时间动作

5.控制节点

控制节点有 7 种类型：初始节点、流最终节点、活动最终节点、决定节点、合并节点、分支节点和集合节点。

初始节点标记活动的起点。正式的说法是，它标记了活动中的一个位置，控制令牌的流会在那里开始。初始节点的标识法是一个小的实心圆形，它一般只有一个输出控制流。如下图所示：

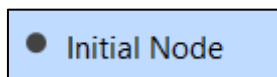


图 2-24 初始节点

流最终节点和活动最终节点是标记控制令牌流结束的控制节点。然而，二者之间有明显的区别：当控制令牌到达流最终节点的时候，那个令牌会被销毁，以此标记单独一个控制流的结束。而当控制令牌到达活动最终节点的时候，整

个活动都会结束，以此标记所有控制流的结束(不管它们当前是否还在执行中)。如下图所示：

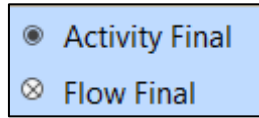


图 2-25 流最终节点和活动最终节点

决定节点标记在活动中可替换序列的开始。决定节点必须拥有单一的输入边，一般拥有两个或多个输出边。每个输出边会带有布尔表达式的标签，那叫做监听，显示为方括号中间的字符串。如下图所示：

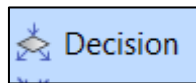


图 2-26 决定节点

合并节点标记活动中可选序列的结尾。其标识法和决定节点相同:空心菱形。可以通过输入边和输出边的数量来区分它们。合并节点拥有两条或多条输入边，而只拥有一个输出边。当一个令牌——可能是对象令牌，也可能是控制令牌——通过任意一条输入边到达合并节点，令牌马上就会提供给输出边。如下图所示：



图 2-27 合并节点

分支节点标记活动中并发序列的起点。分支节点的标识法是一条线段(方向随意)，它必须拥有一条输入边和两条或多条输出边。当一个令牌——可能是对象令牌，也可能是控制令牌——到达分支节点的时候，它会被复制到所有输出边上。原始令牌的每个副本都代表独立、并发、沿着各自路径前进的控制流。如下图所示：

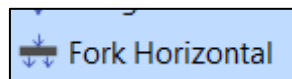


图 2-28 分支节点

集合节点标记活动中并发序列的结束。集合节点的标识法和分支节点一样:一条线。你可以通过输入、输出边的数量来区分它们。集合节点一般拥有两条或多条输入边，而只有一条输出边。如下图所示：

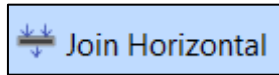


图 2-29 集合节点

6.活动分区

活动图不仅可以传达动作的顺序，还可以传达执行每一个动作的结构，可以用活动分区来表示。如下图所示：

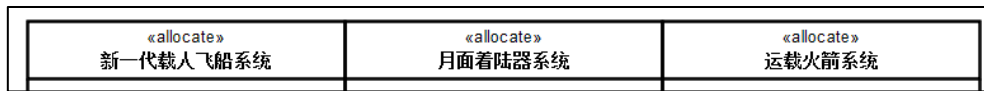


图 2-30 活动分区

活动图是强大的信息沟通媒介，它的优势在于它的可读性而且可以显示行为拥有复杂的控制逻辑。当你需要显示持续的行为，并且需要关注事件、能量与数据的流在一系列动作之间的流动时，可选择活动图。

2.2. 相关软件部署过程

本操作指南使用 ATK v1.5.0.7 版本、Cameo Systems Modeler 19.0 版本进行以下章节内容演示。

2.2.1. MagicDraw 安装

打开 MagicDraw 安装包，按照提示进行安装，如下图所示：

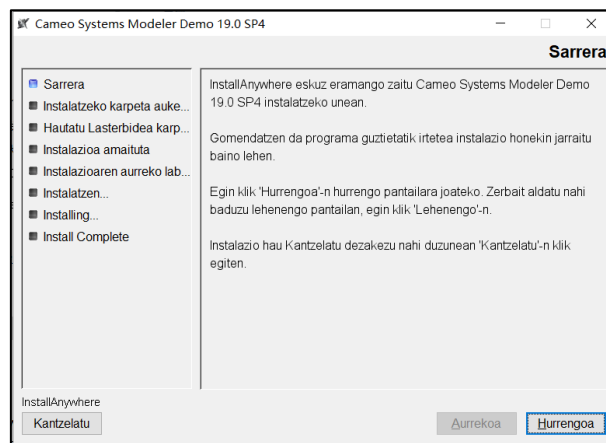


图 2-31 MagicDraw 安装

2.2.2. ATK 安装

将 ATK 安装包解压，如下图所示：

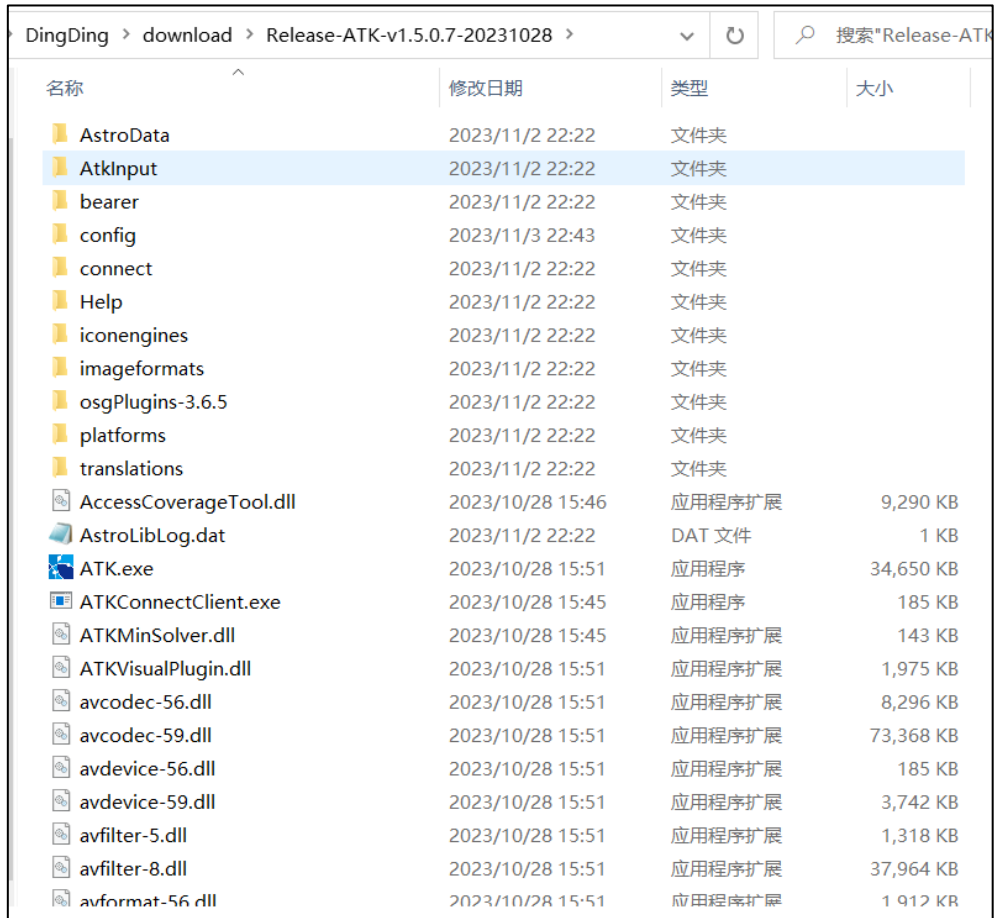


图 2-32 ATK 安装包解压

打开 AtkInput-Register_2015.exe 进行注册激活，如下图所示：

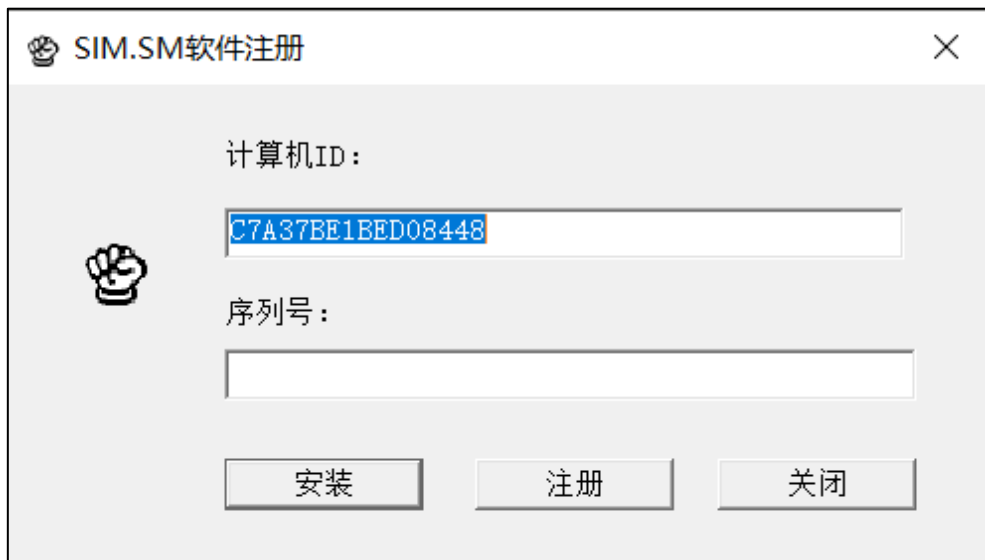


图 2-33 ATK 注册激活

激活完成后就可以打开根目录下 ATK.exe 进行使用，如下图所示：

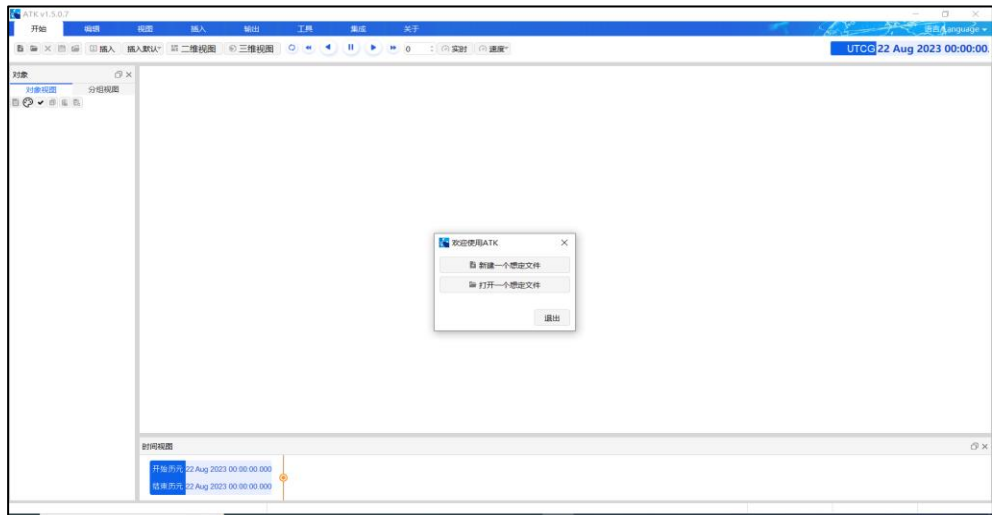


图 2-34 打开 ATK 工具

2.2.3. MagicDraw 与 ATK 联合仿真插件安装

将 HW-MagicDrawToATK-Plug 文件夹复制到 Cameo Systems Modeler Demo\plugins 目录下，如下图所示。

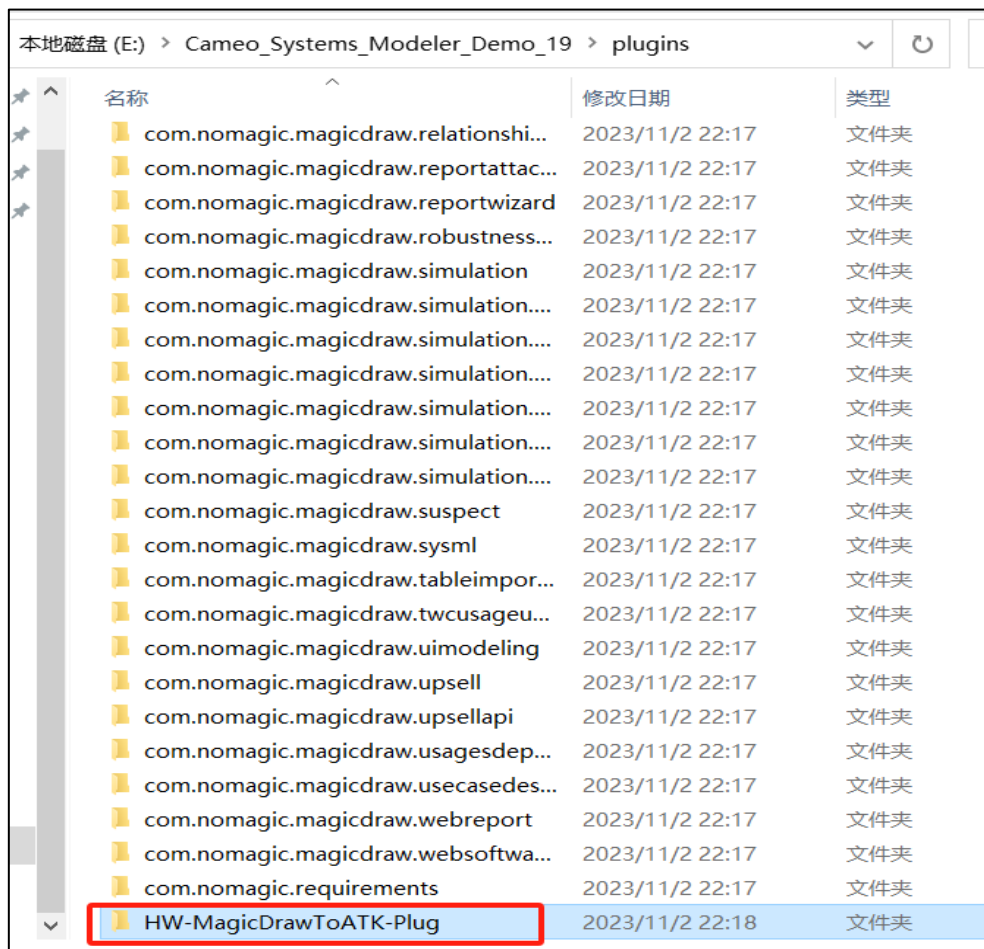


图 2-35 配置联合仿真插件

在菜单栏选中 Options-Environment-Plugins，可以在 CATIA Magic 软件上确认工具安装成功，如下图所示。

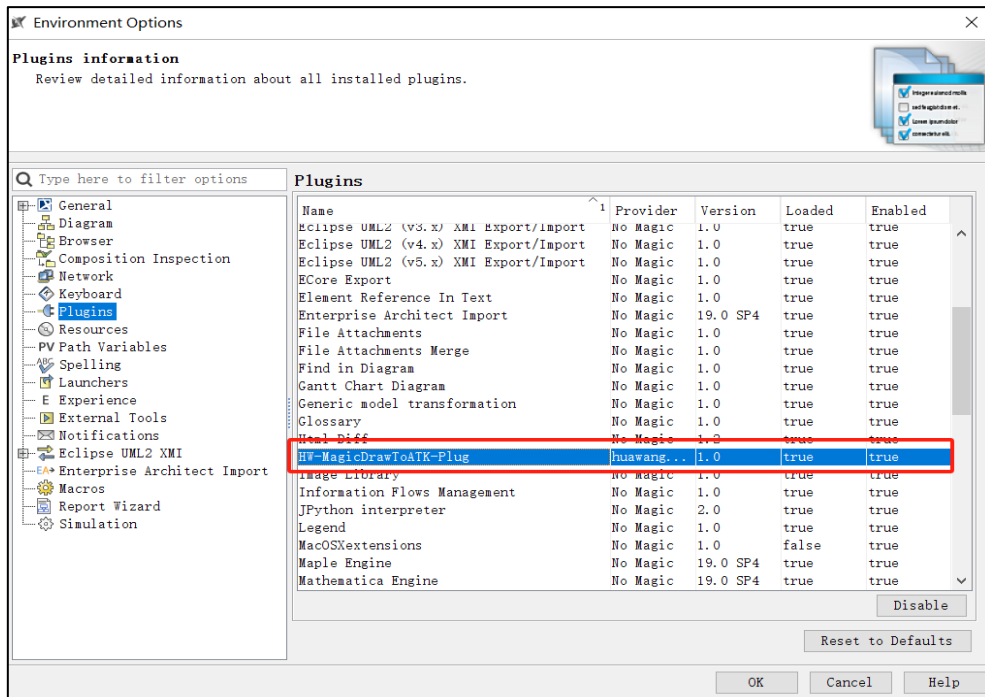


图 2-36 检验联合仿真插件是否安装成功

2.3. MagicDraw 与 ATK 联合仿真使用过程

2.3.1. 载人航天任务建模过程

打开 MagicDraw 软件，选择 File-NewProject 新建项目，如下图所示：

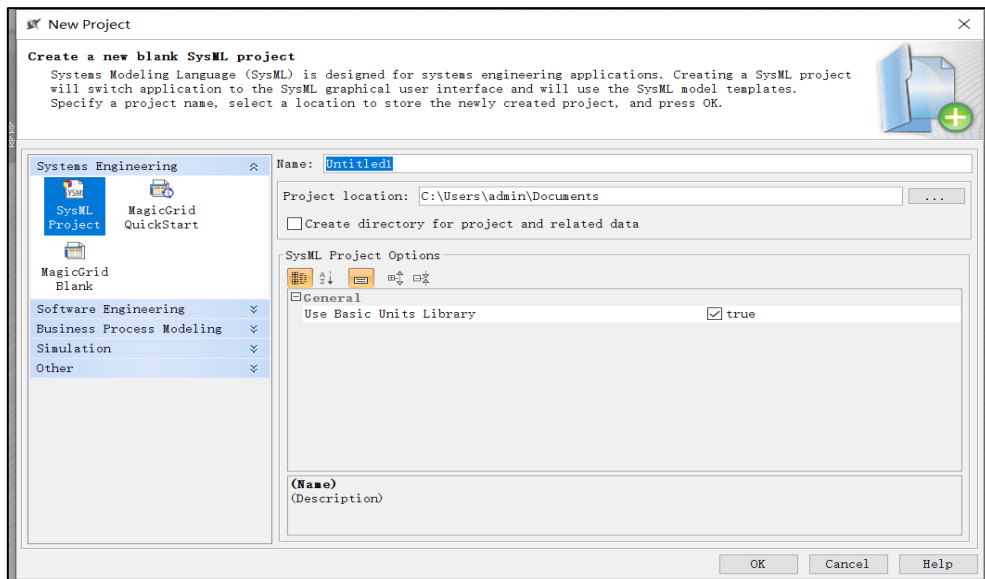


图 2-37 新建 SysML 项目

创建 BDD 图，构建系统架构，如下图所示：

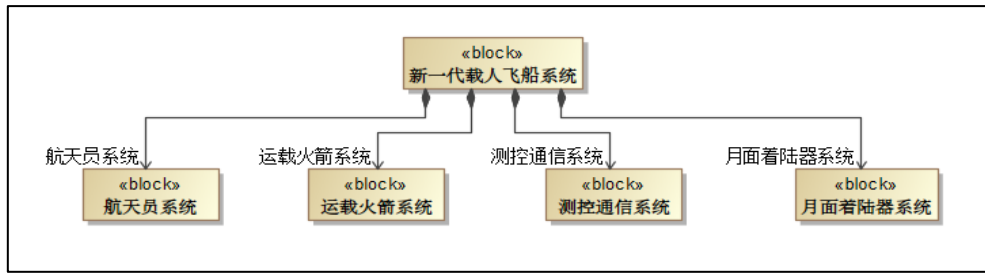


图 2-38 系统架构

右键“新一代载人飞船系统”Block，选择 CreatDiagram，新建状态机图或者活动图，如下图所示：

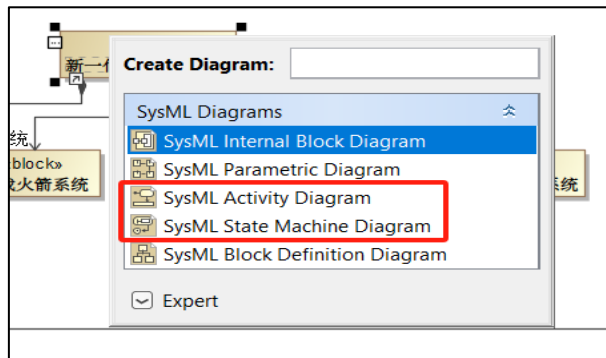


图 2-39 新建状态机或活动图

先以状态机为例，创建如下图所示状态机：

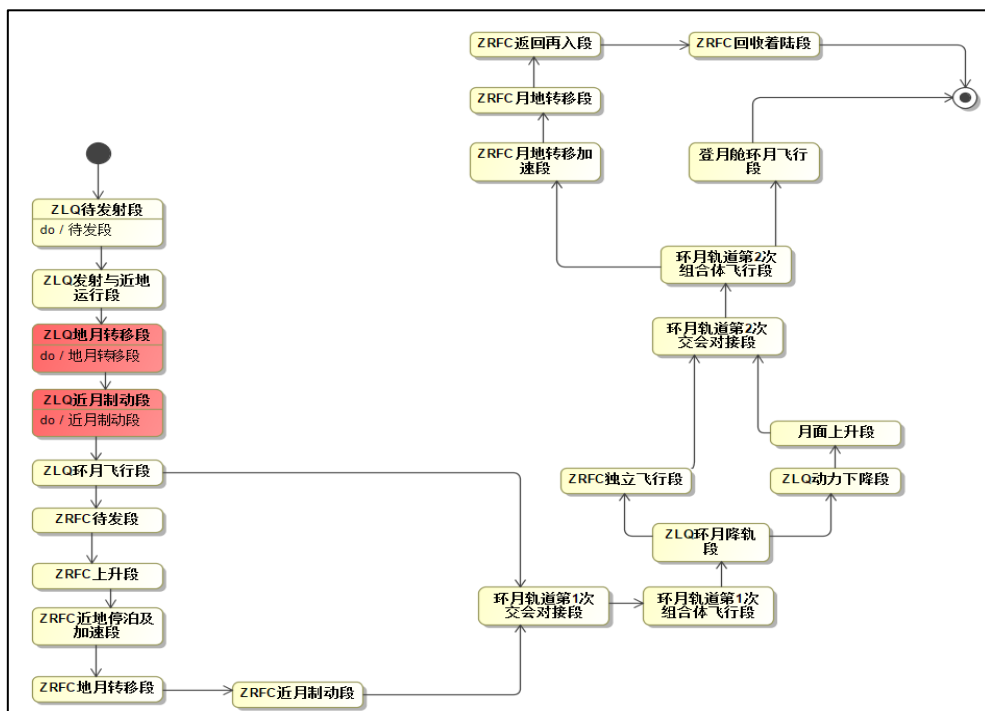


图 2-40 MagicDraw 系统建模

构建完成仿真模型后，打开 ATK 工具，如下图所示：

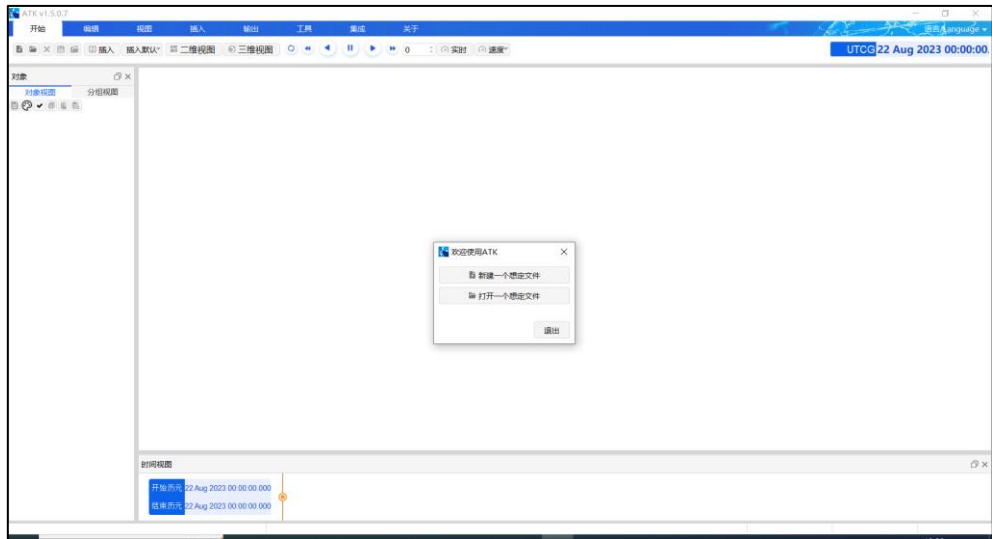


图 2-41 打开 ATK 工具

2.3.2. 脚本编辑过程

在完成 MagicDraw 端系统建模以及启动 ATK 软件后，编辑脚本。右键 State 模型元素，选中 Specification，打开属性面板，如下图所示：

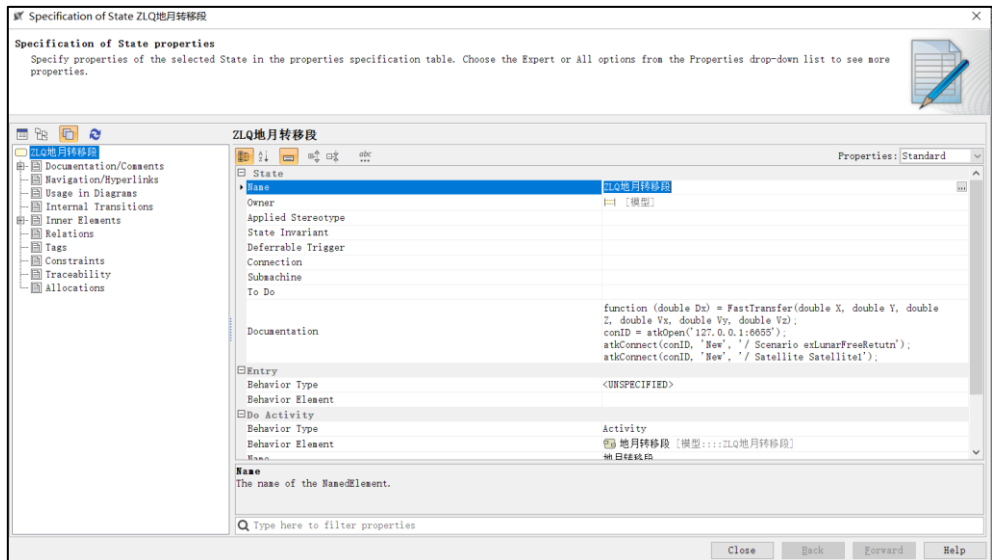


图 2-42 打开 State 属性面板

在 Documentation/Comments 属性内填下 ATK 脚本，实现联合仿真，如下图所示：

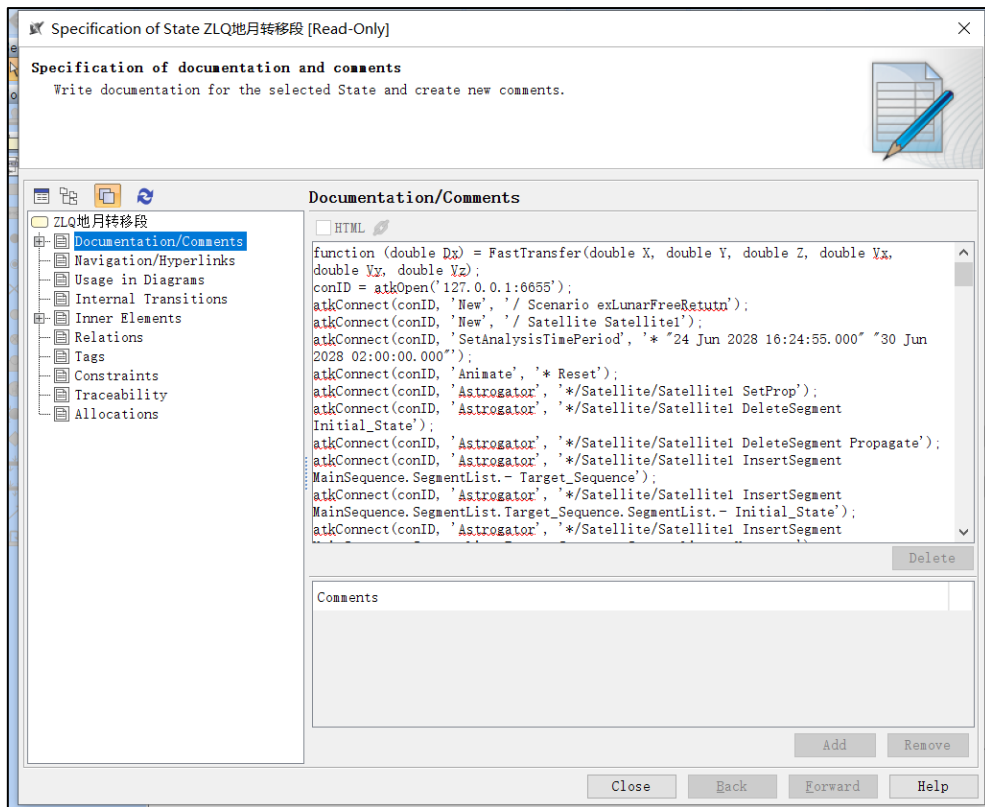


图 2-43 输入脚本

- `function (double Dx) = FastTransfer(double X, double Y, double Z, double Vx, double Vy, double Vz);`//定义输入输出
- `conID = atkOpen('127.0.0.1:6655');`//输入 ATK 所在地址以及端口，与 ATK 进行通信连接
- `atkConnect(conID, 'New', '/ Scenario exLunarFreeRetutn');`//通过命令新建场景
- `atkConnect(conID, 'New', '/ Satellite Satellite1');`// 通过命令新建卫星
- `atkConnect(conID, 'SetAnalysisTimePeriod', '* "24 Jun 2028 16:24:55.000" "30 Jun 2028 02:00:00.000"');`//使用 UTCG 格式设置场景分析时间段
- `atkConnect(conID, 'Animate', '* Reset');`// 重置界面
- `atkConnect(conID, 'Astrogator', '* /Satellite/Satellite1 SetProp');`//设置卫星轨道预报器类型为机动规划；第一级目录默认存在一个初始段，一个预报段
- `atkConnect(conID, 'Astrogator', '* /Satellite/Satellite1 DeleteSegment`

Initial_State');

- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 DeleteSegment Propagate');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 InsertSegment MainSequence.SegmentList.- Target_Sequence');` //在第一级目录下添加一个瞄准序列段，默认段名为 Target_Sequence
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList.- Initial_State');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList.- Maneuver');` //在瞄准序列段内添加一个机动段，默认段名为 Maneuver
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList.- Propagate');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Propagate.SegmentColor - 16776961');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.Epoch 24 Jun 2028 16:24:55.000 UtCG');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.CoordinateType Cartesian');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.Cartesian.X @X m');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue`

MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.Cartesian.Y (@Y m');

- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.Cartesian.Z (@Z m');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.Cartesian.Vx (@Vx');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.Cartesian.Vy (@Vy');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.Cartesian.Vz (@Vz');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.dry-mass 100');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.fuel-mass 100');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.Cd 1');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.dragarea 1');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue`

MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.Cr 1');

- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State.InitialState.srpara 2.2');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Propagate.Propagator Earth JGM3 20 20 true E1976StdAtm false 150 150 15 false true false 0 GMM1 0 0 true 0 GLGM2 0 0');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.ImpulsiveMnvr.ThrustAxes "Satellite VNC(Earth)");`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.ImpulsiveMnvr.coordtype Cartesian');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.ImpulsiveMnvr.Cartesian.X 3160');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Propagate.StoppingConditions Epoch');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Propagate.StoppingConditions.Epoch.TripValue 30 Jun 2028 02:00:00.000 UtCG');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Propagate.StoppingConditions.Epoch.Tolerance 0.0001');`

- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.Profiles Differential_Corrector');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 AddMCSegmentControl MainSequence.SegmentList.Target_Sequence.SegmentList.Initial_State InitialState.Epoch');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Initial_State InitialState.Epoch Active true');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Initial_State InitialState.Epoch MaxStep 60 sec');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Initial_State InitialState.Epoch Correction 1001.71404699199 sec');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Initial_State InitialState.Epoch Perturbation 0.1 sec');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Initial_State InitialState.Epoch Scale 1 sec');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 AddMCSegmentControl MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver ImpulsiveMnvr.Cartesian.X');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Active true');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue`

MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 10 m/sec');

- atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Correction 2.39263205821203 m/sec');
- atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
- atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
- atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 AddMCSegmentControl MainSequence.SegmentList.Target_Sequence.SegmentList.Propagate StoppingConditions.Epoch.TripValue');
- atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate StoppingConditions.Epoch.TripValue Active true');
- atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate StoppingConditions.Epoch.TripValue MaxStep 300 sec');
- atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate StoppingConditions.Epoch.TripValue Correction 0.00989449023243827 sec');
- atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate StoppingConditions.Epoch.TripValue Perturbation 0.1 sec');

- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSCControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate StoppingConditions.Epoch.TripValue Scale 1 sec');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetValue MainSequence.SegmentList.Target_Sequence.SegmentList.Propagate.Results Inclination AltitudeOfPeriapsis');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate Inclination Active true');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate Inclination Desired 0.750491578357562 rad');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate Inclination Scale 1 rad');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate Inclination tolerance 0.00872664625997165 rad');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate Inclination Weight 1');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate AltitudeOfPeriapsis Active true');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate AltitudeOfPeriapsis Desired 50000 m');`

- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate AltitudeOfPeriapsis Scale 1 m');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate AltitudeOfPeriapsis tolerance 100 m');`
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 SetMCSCConstraintValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Propagate AltitudeOfPeriapsis Weight 0.0001');` //设置第四个预报段停止条件 Duration 的误差值为 0.00001 秒
- `atkConnect(conID, 'Save', '/ *');` // 保存此场景
- `atkConnect(conID, 'Astrogator', '*/Satellite/Satellite1 RunMCS');`
- `@Dx = atkConnect(conID, 'Astrogator_RM', '*/Satellite/Satellite1 GetMCSCControlValue MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector Maneuver ImpulsiveMnvr.Cartesian.X finalvalue');`
- `atkClose(conID);` // 与 ATK 断开连接

脚本中定义了输入输出，在模型中也需要同时的定义这些输入输出。右键状态机选择 Create Element-Property，新建属性，如下图所示：

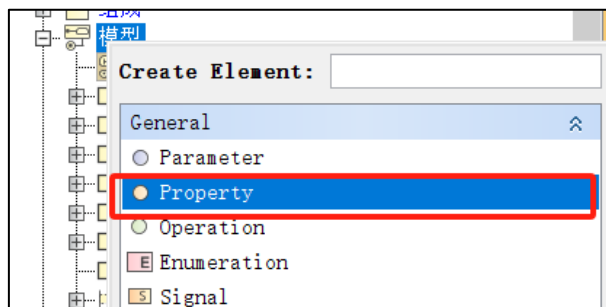


图 2-44 新建属性

将属性的 Stereotype 修改为 ValueProperty，如下图所示：

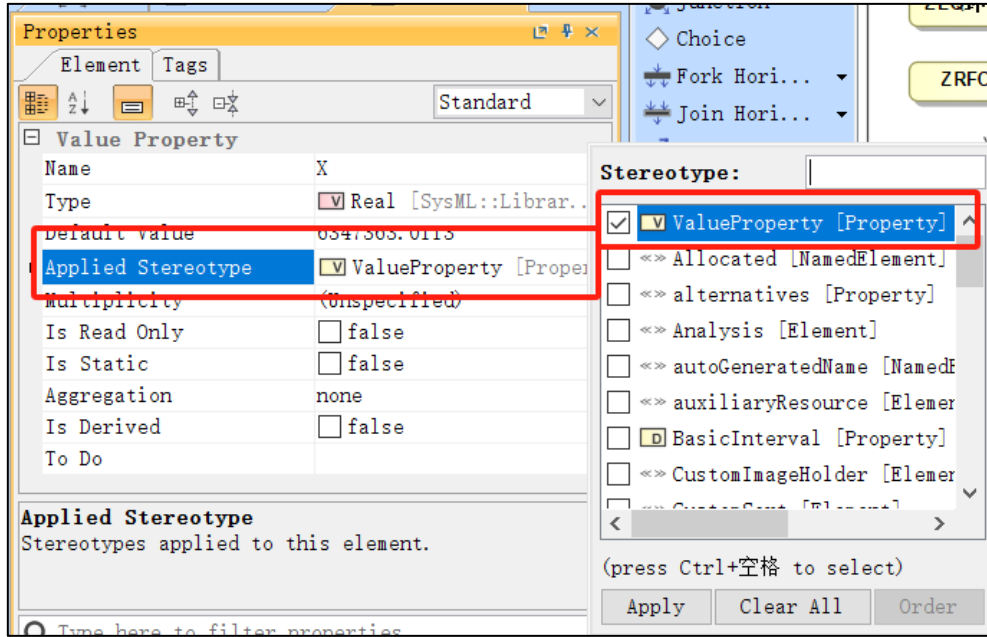


图 2-45 应用构造型

依据脚本中的输入输出创建属性，输入属性包含 X、Y、Z、Vx、Vy、Vz，输出属性包含 Dx，并将这些属性的 Type 修改为 Real，如下图所示：

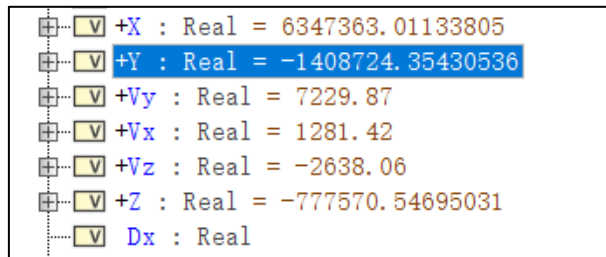


图 2-46 设置输入输出

同时给输入属性赋初值，如下图所示：

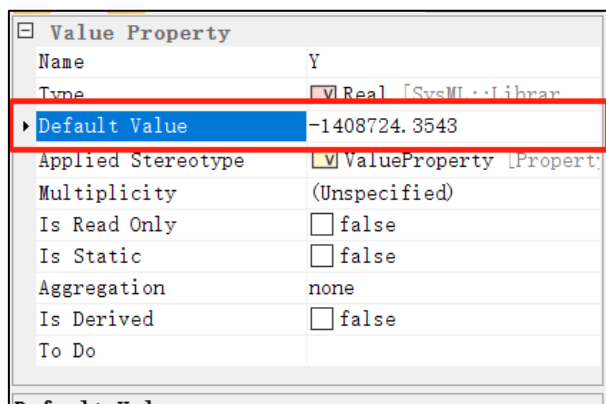


图 2-47 设置初值

2.3.3. 载人航天任务仿真过程

完成模型、脚本、值属性构建后，开始联合仿真。右键状态机选择 Simulation-Run，打开仿真界面，如下图所示：

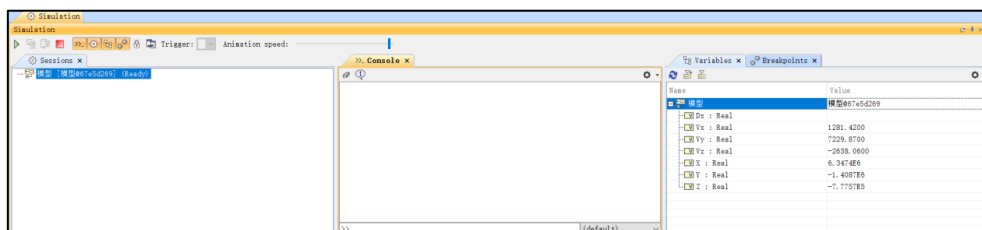


图 2-48 打开仿真

点击开始仿真按钮 MagicDraw 与 ATK 开始联合仿真，如下图所示：

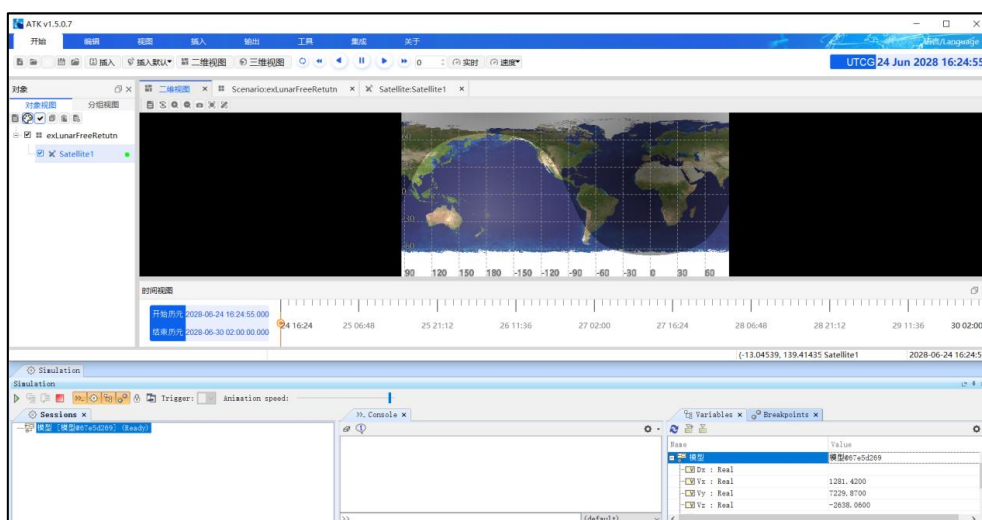


图 2-49 联合仿真

开始仿真后，MagicDraw 将输入参数发送至 ATK 中，如下图所示：

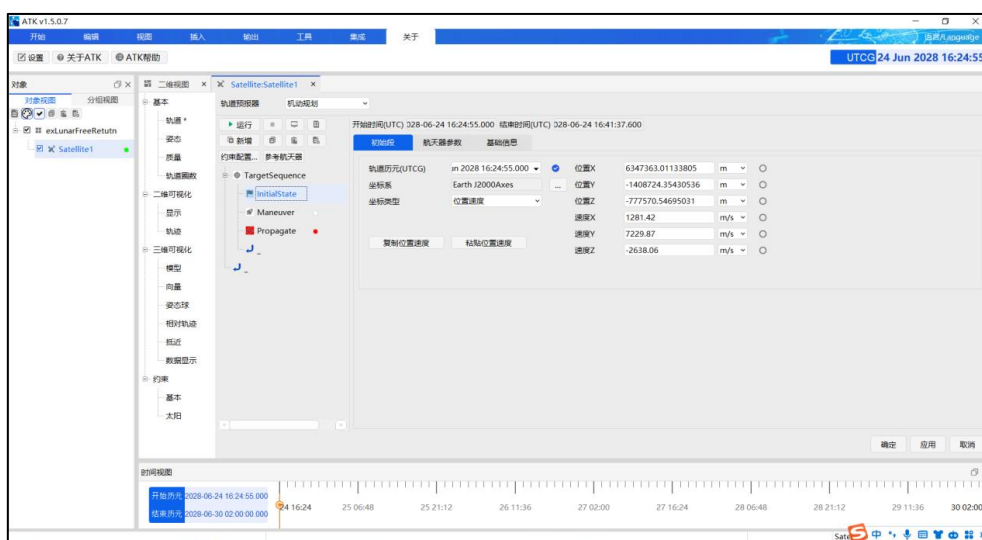


图 2-50 ATK 接收数据

使用活动图与 ATK 联合仿真方式类似，首先构建活动图，如下图所示：

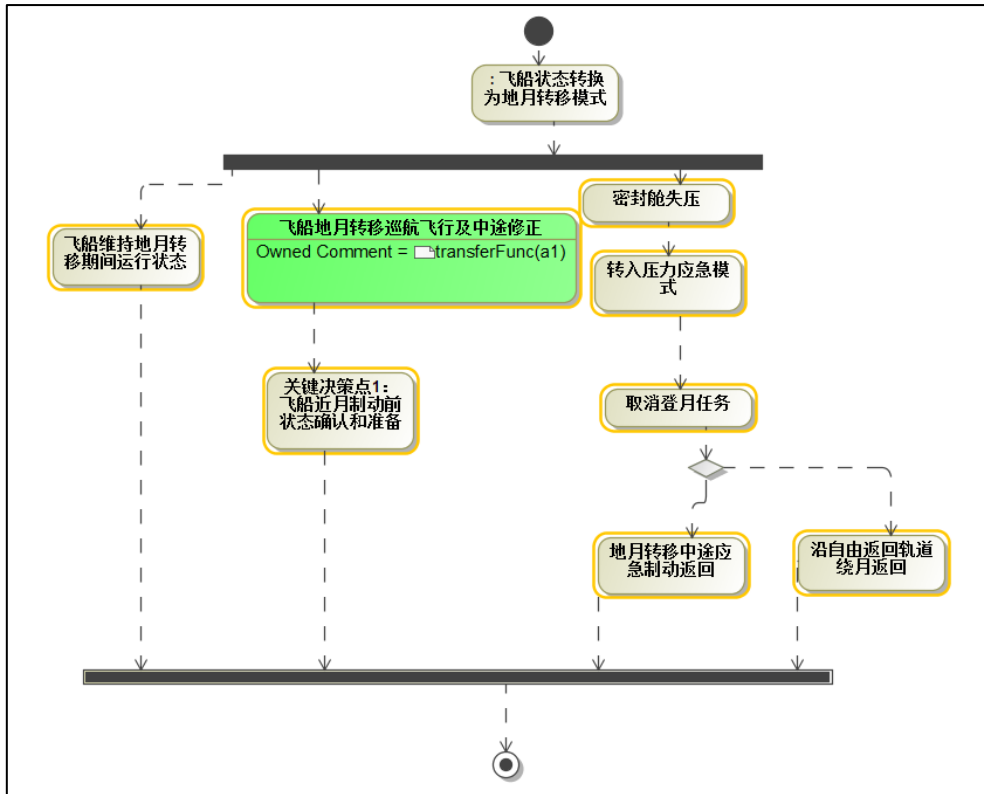


图 2-51 构建活动图

右键 Action 模型元素，选中 Specification，打开属性面板，如下图所示：

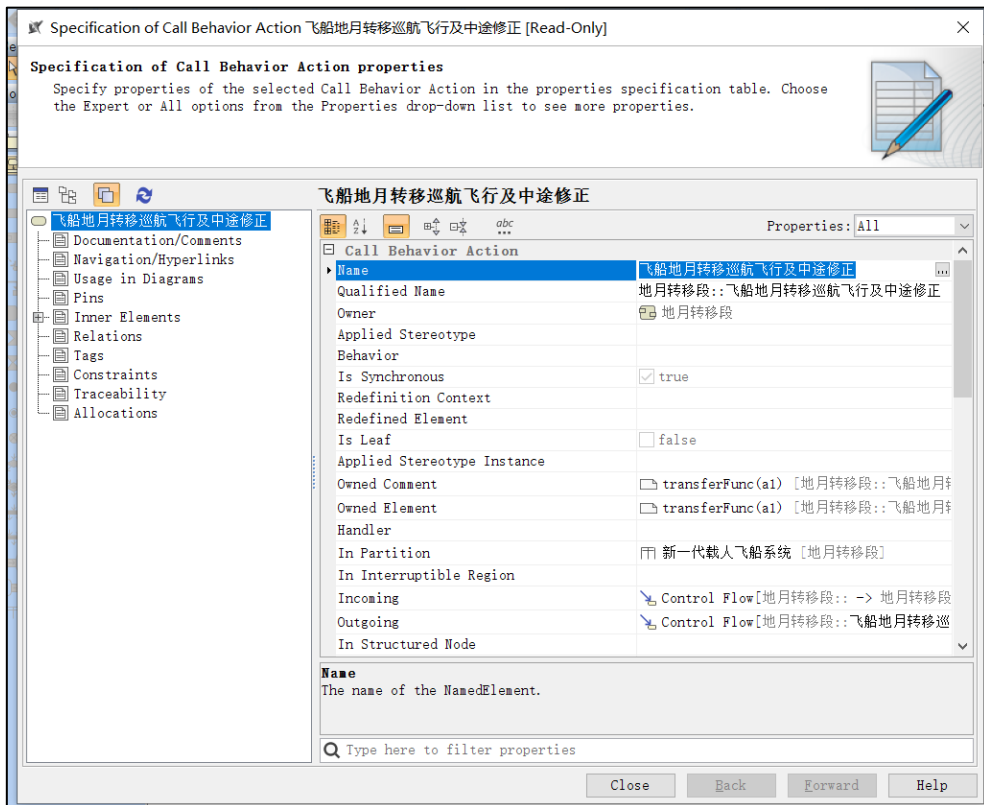


图 2-52 Action 属性面板

在 Documentation/Comments 属性内填下 ATK 脚本，实现联合仿真，如下图所示：

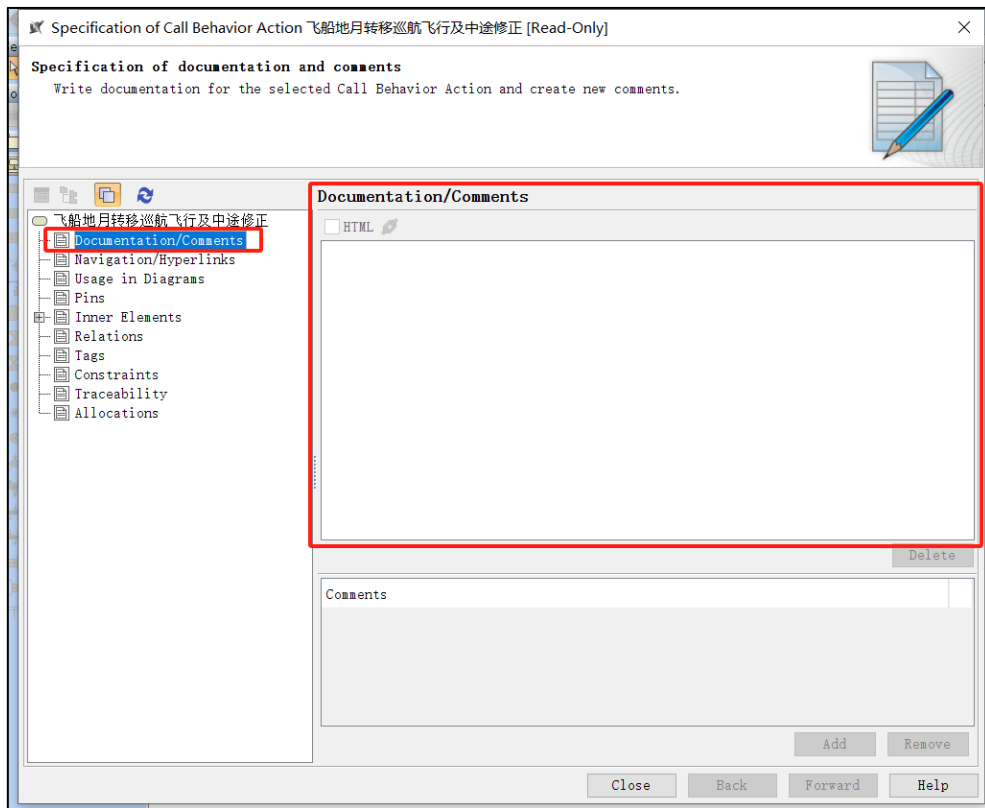


图 2-53 编写脚本

2.4. MagicDraw 与 ATK 联合仿真插件升级

将原先版本 HW-MagicDrawToATK-Plug 文件夹删除，将最新版 HW-MagicDrawToATK-Plug 文件夹复制到 Cameo Systems Modeler Demo\plugins 目录下，如下图所示。

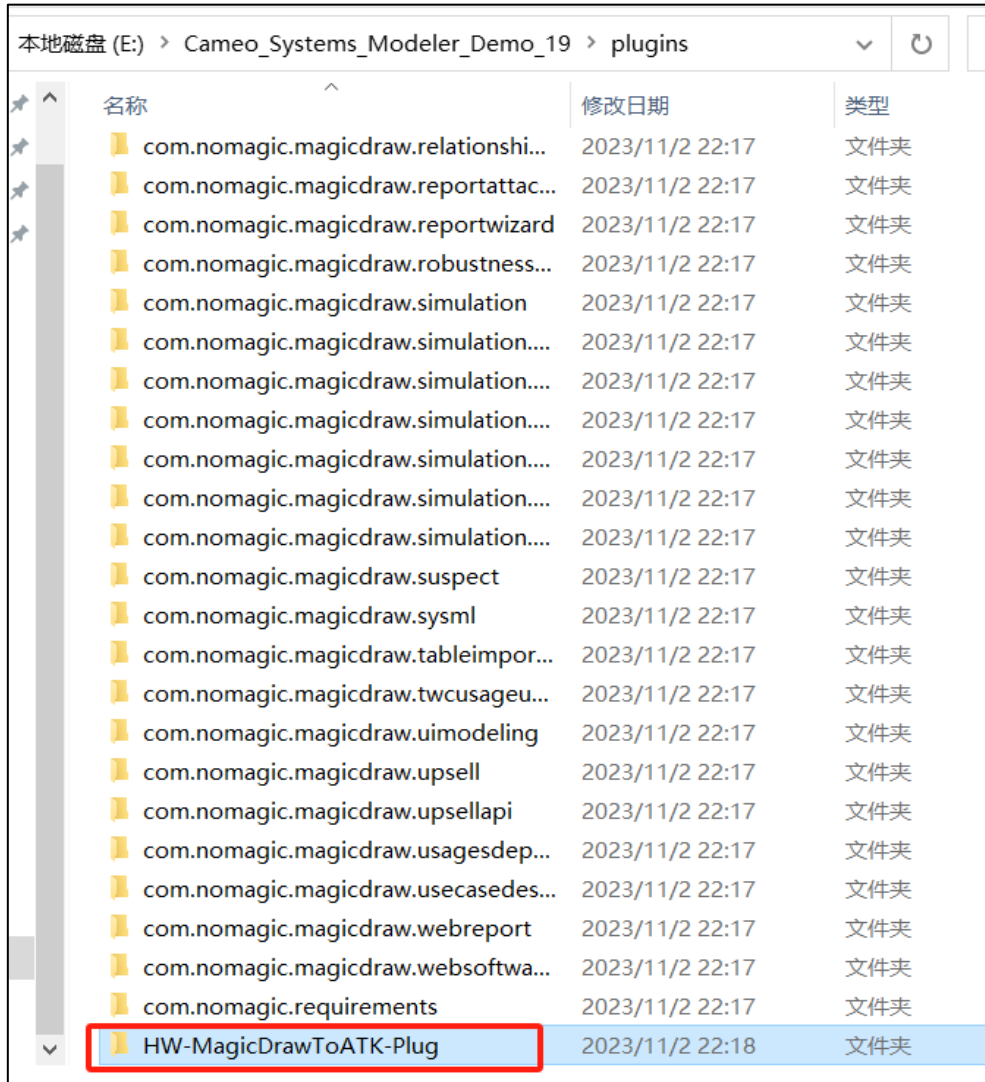


图 2-54 插件升级

3. ATK.Component 模式教程

ATK.Component 模式向用户提供多种语言操作方式,提供不同的操作接口,以方便用户对 ATK 进行操作。具体使用方式及接口介绍如下。

3.1. C++语言操作流程

以 VisualStudio2015 为例,使用动态库进行项目规划。

3.1.1. 新建项目

打开 VS2015,点击新建项目,新建空项目,并设置项目名称与位置,设置完成后点击确定。

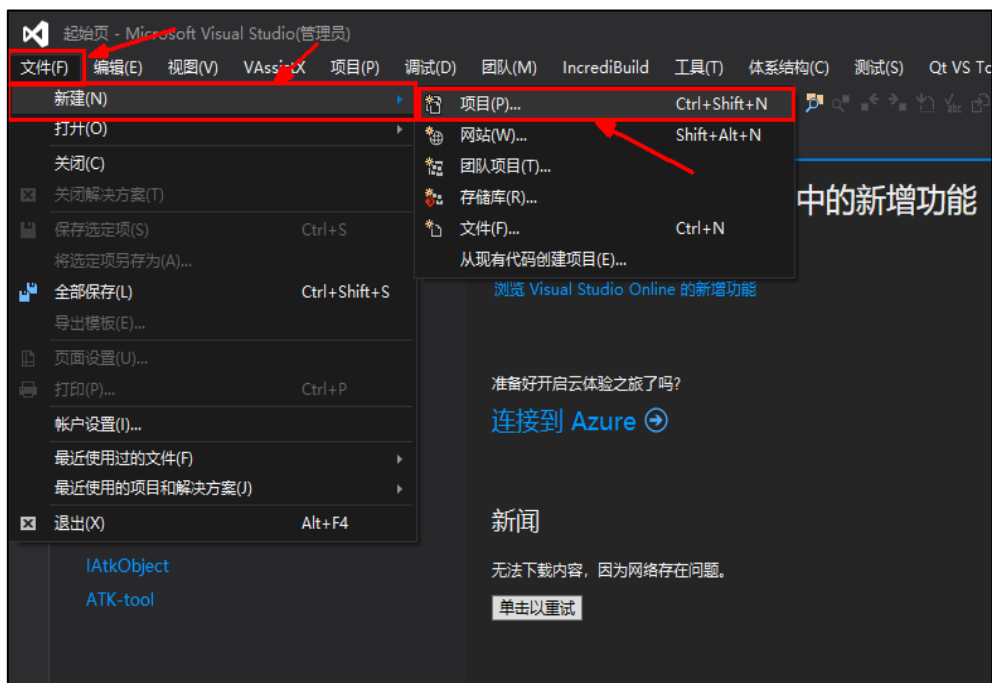


图 3-1 新建项目步骤

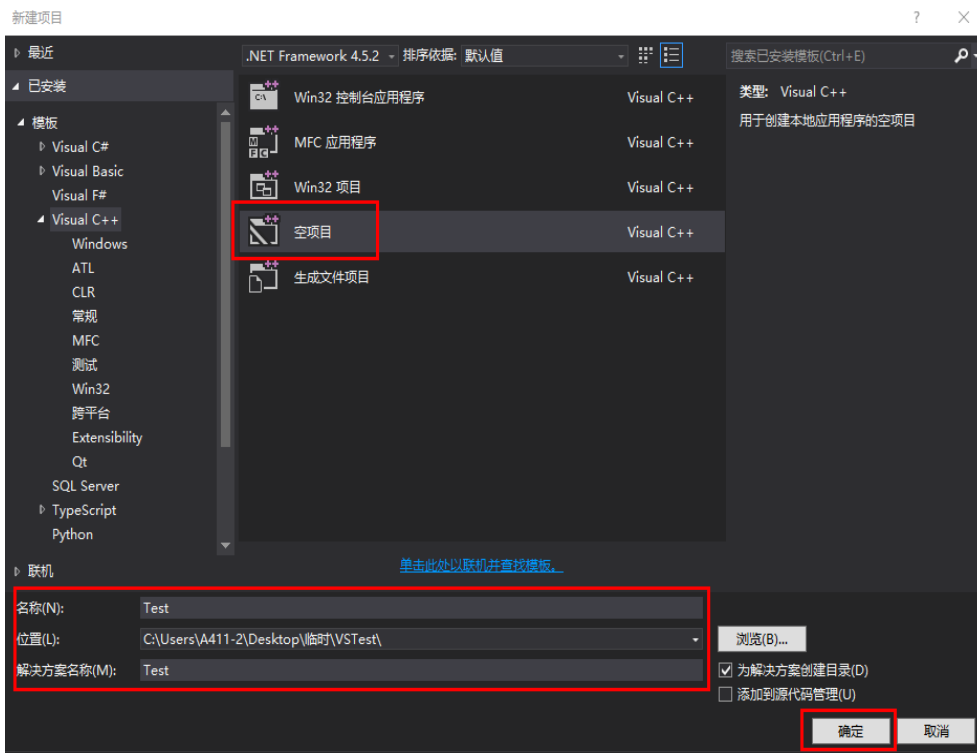


图 3-2 新建项目

3.1.2. 文件配置

在 ATK 安装包目录中，点击 IntegratingWithATK 文件夹，点开 Component 文件夹。



图 3-3 Component 包含文件文件夹



图 3-4 Component 包含动态库文件夹

名称	修改日期	类型
Lib	2024/6/29 09:04	文件夹

图 3-5 Component 包含库文件夹

将所有文件添加至工程 Test 根目录中。

名称	修改日期	类型	大小
Include	2024/6/29 08:57	文件夹	
Lib	2024/6/29 08:57	文件夹	
AtMultiManeu.dll	2024/6/26 17:44	应用程序扩展	1,623 KB
AtMultiManeuD.dll	2024/6/26 16:15	应用程序扩展	4,836 KB
IAtkObj.h	2024/6/18 08:01	C/C++ Header	16 KB
IAtkObject.dll	2024/6/29 08:24	应用程序扩展	10,064 KB
IAtkObjectD.dll	2024/6/29 08:23	应用程序扩展	19,442 KB
IAtkObjectH.h	2024/6/29 08:44	C/C++ Header	20 KB
Test.vcxproj	2024/3/29 11:51	VC++ Project	7 KB
Test.vcxproj.filters	2024/3/29 11:51	VC++ Project Fil...	1 KB

图 3-6 Lib 库与头文件添加

将需要的配置文件添加至此文件夹。（AstroData 文件夹在 ATK 安装目录下）。

名称	修改日期	类型	大小
AstroData	2024/6/29 08:40	文件夹	
Include	2024/6/29 08:57	文件夹	
Lib	2024/6/29 08:57	文件夹	
AtMultiManeu.dll	2024/6/26 17:44	应用程序扩展	1,623 KB
AtMultiManeuD.dll	2024/6/26 16:15	应用程序扩展	4,836 KB
IAtkObj.h	2024/6/18 08:01	C/C++ Header	16 KB
IAtkObject.dll	2024/6/29 08:24	应用程序扩展	10,064 KB
IAtkObjectD.dll	2024/6/29 08:23	应用程序扩展	19,442 KB
IAtkObjectH.h	2024/6/29 08:44	C/C++ Header	20 KB
Test.vcxproj	2024/3/29 11:51	VC++ Project	7 KB
Test.vcxproj.filters	2024/3/29 11:51	VC++ Project Fil...	1 KB

图 3-7 配置文件添加

3.1.3. 项目新建项

项目添加文件，项目右键->添加->新建项。

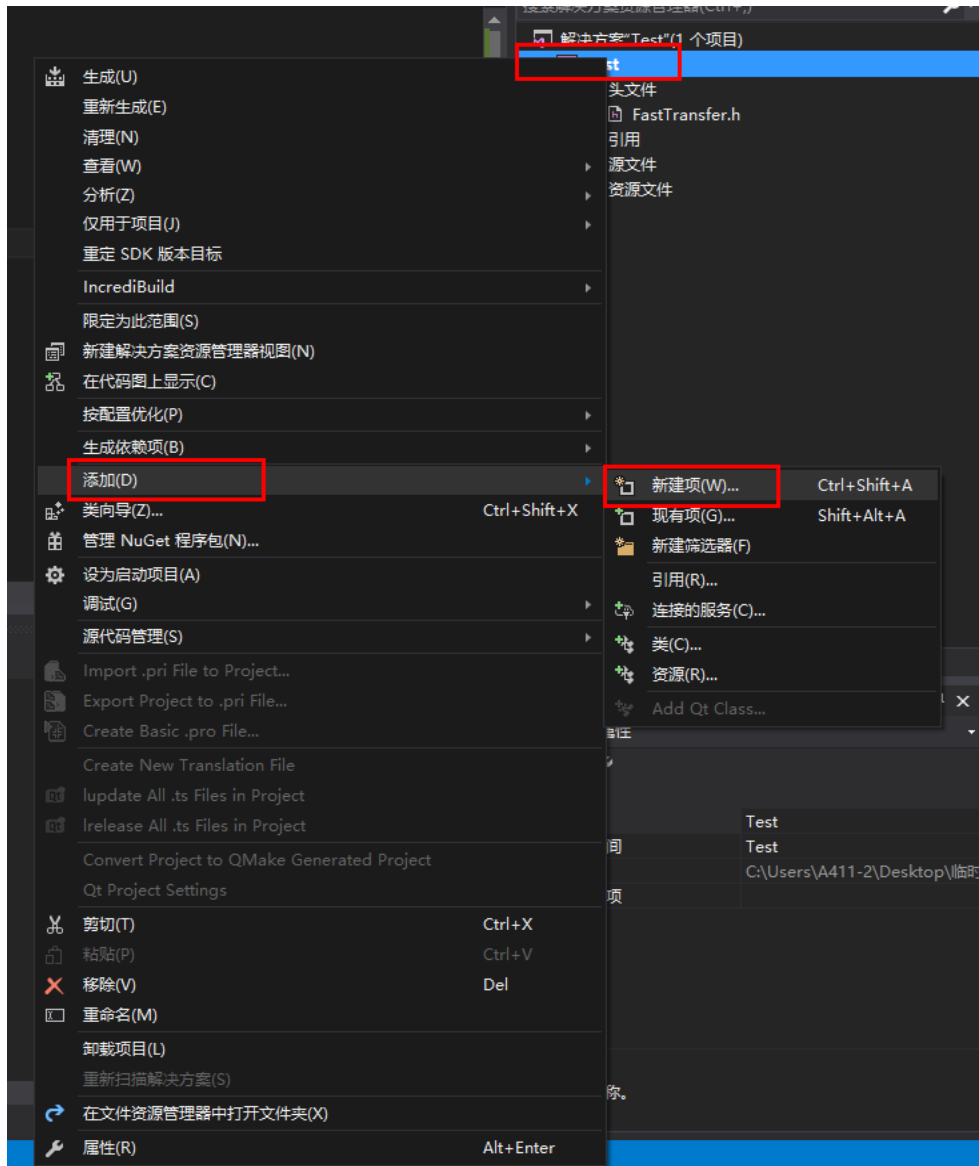


图 3-8 新建文件步骤

3.1.4. 设置添加项的名称与位置

设置文件名称及位置，点击添加，完成后点击生成。

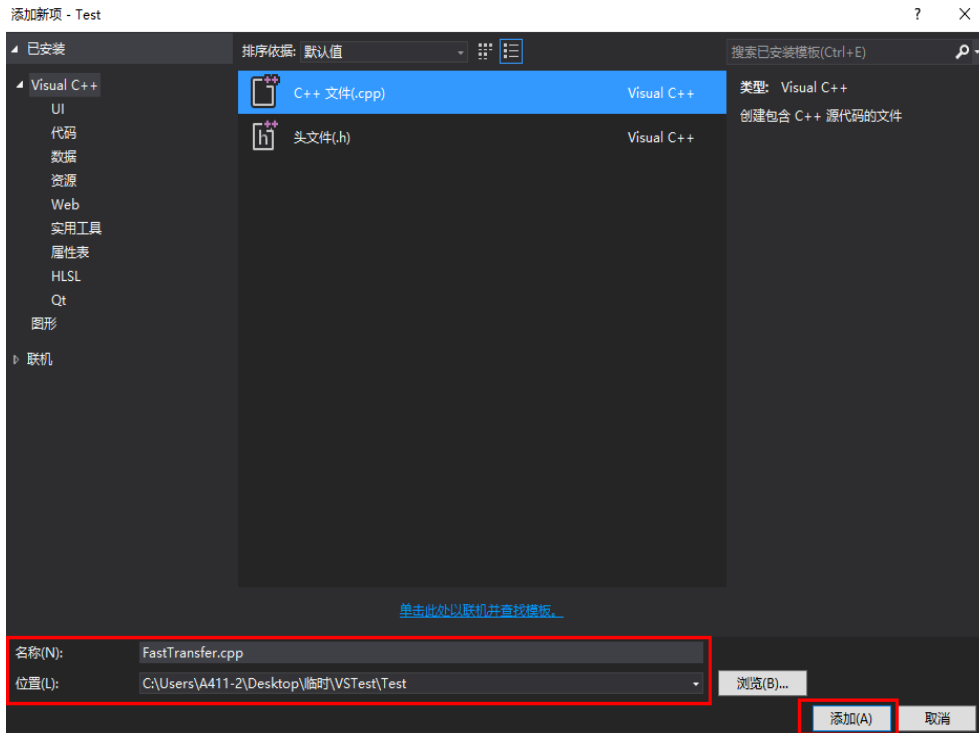


图 3-9 新建文件

3.1.5. Debug 属性配置

打开项目右键->属性。

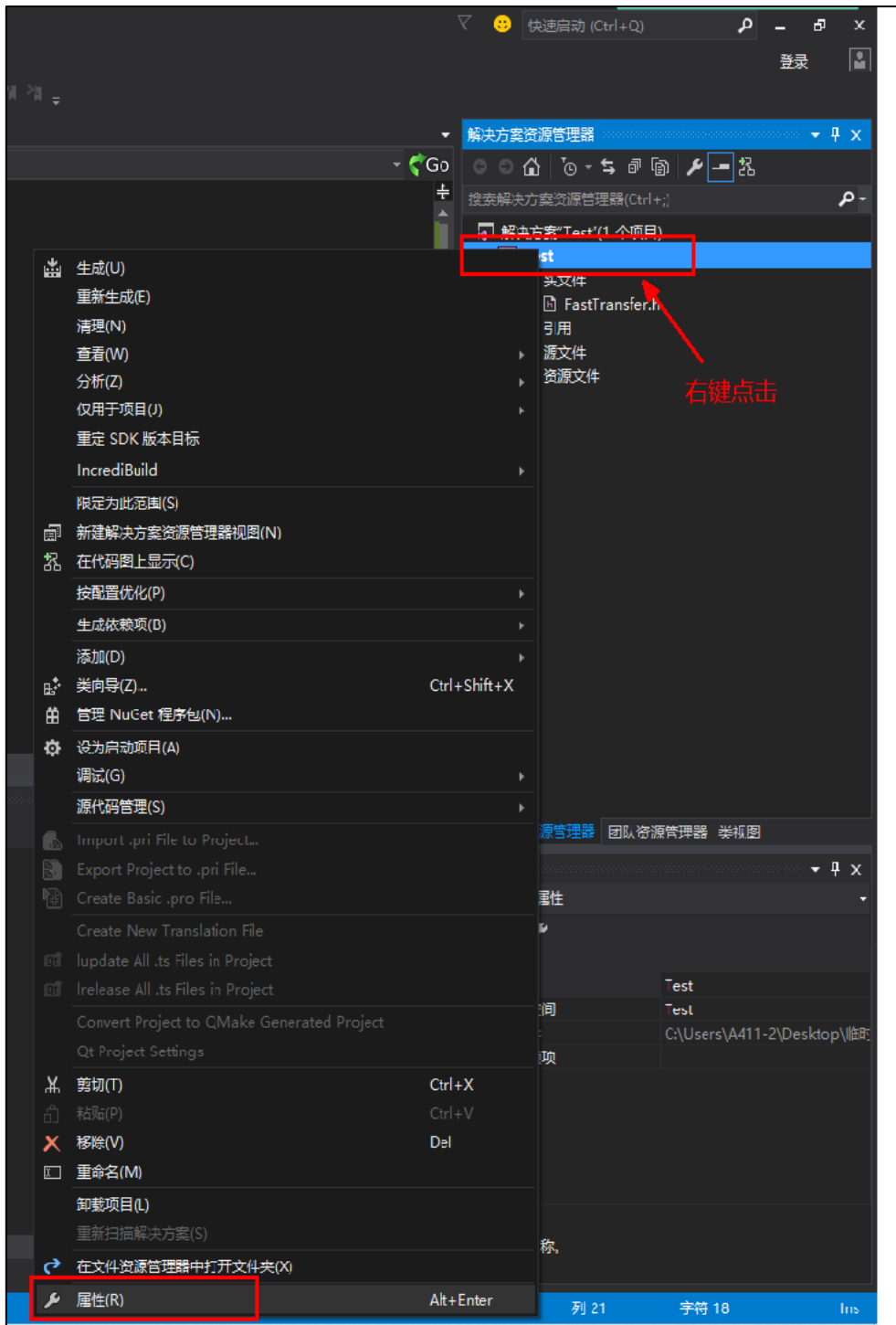


图 3-10 属性设置步骤

配置平台改为 Debug, x64 模式, 点击常规->输出目录, 将输出目录修改为项目目录。

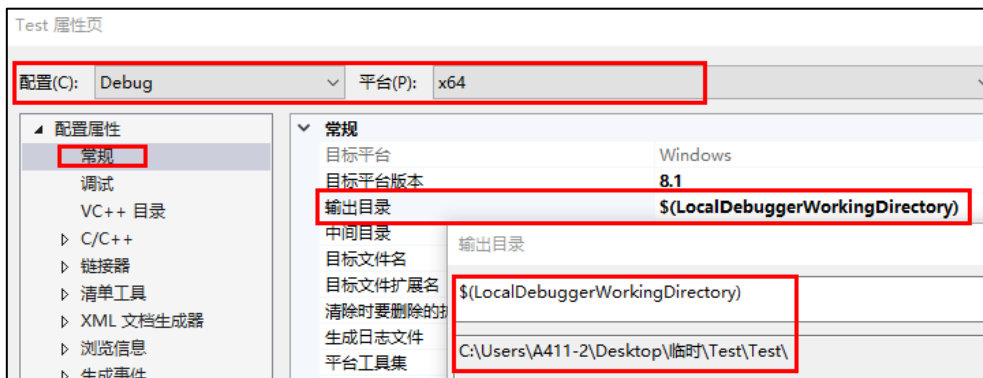


图 3-11 修改输出目录

点击 C/C++->常规->附加包含目录，将头文件目录添加至附加包含目录。



图 3-12 附加包含目录设置

链接器->常规->附加库目录，将 lib 文件夹添加至附加库目录，点击确定。

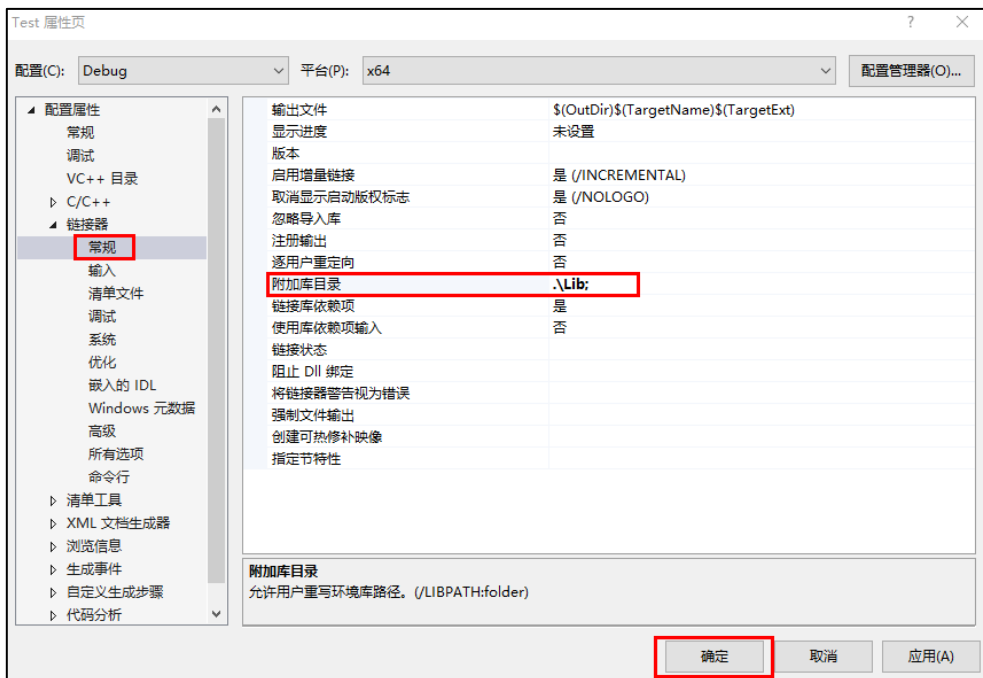


图 3-13 附加库设置

链接器->输入->附加依赖项，将 lib 库全称添加至附加依赖项，点击确定。

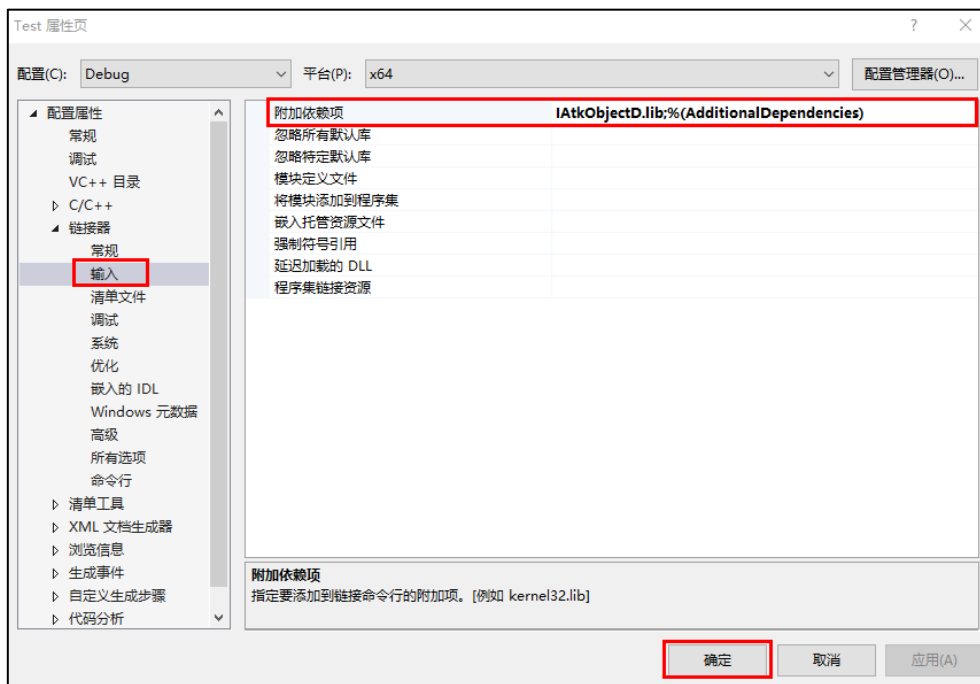


图 3-14 附加库设置

3.1.6. Release 属性配置

打开项目右键->属性。

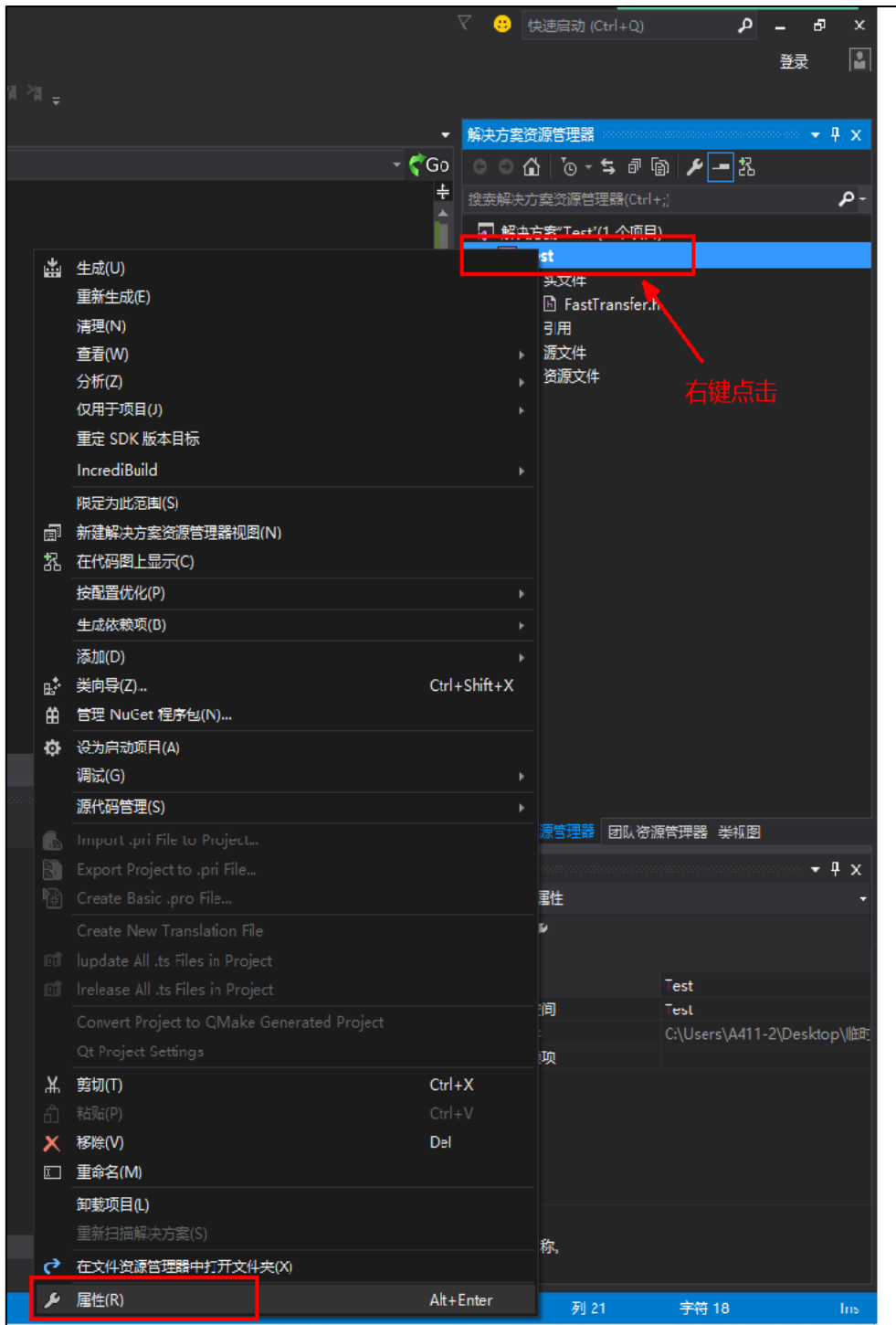


图 3-15 属性设置步骤

配置平台改为 Release，x64 模式，点击常规->输出目录，将输出目录修改为项目目录。

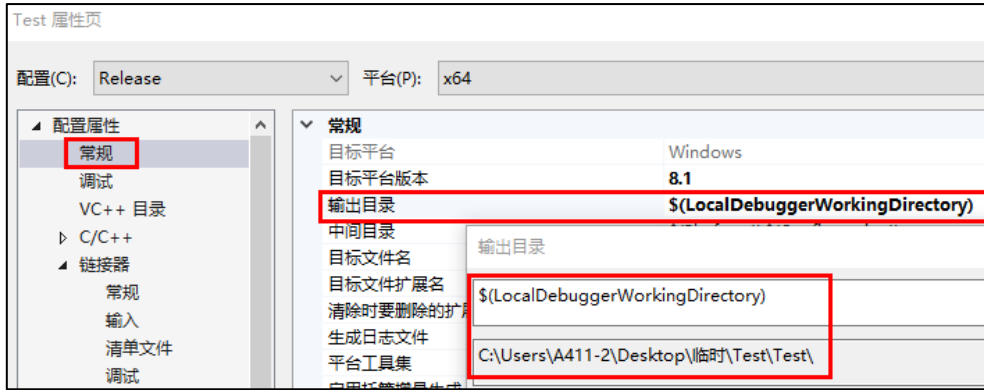


图 3-16 修改输出目录

点击 C/C++->常规->附加包含目录，将头文件目录添加至附加包含目录。



图 3-17 附加包含目录设置

链接器->常规->附加库目录，将 lib 文件夹添加至附加库目录，点击确定。

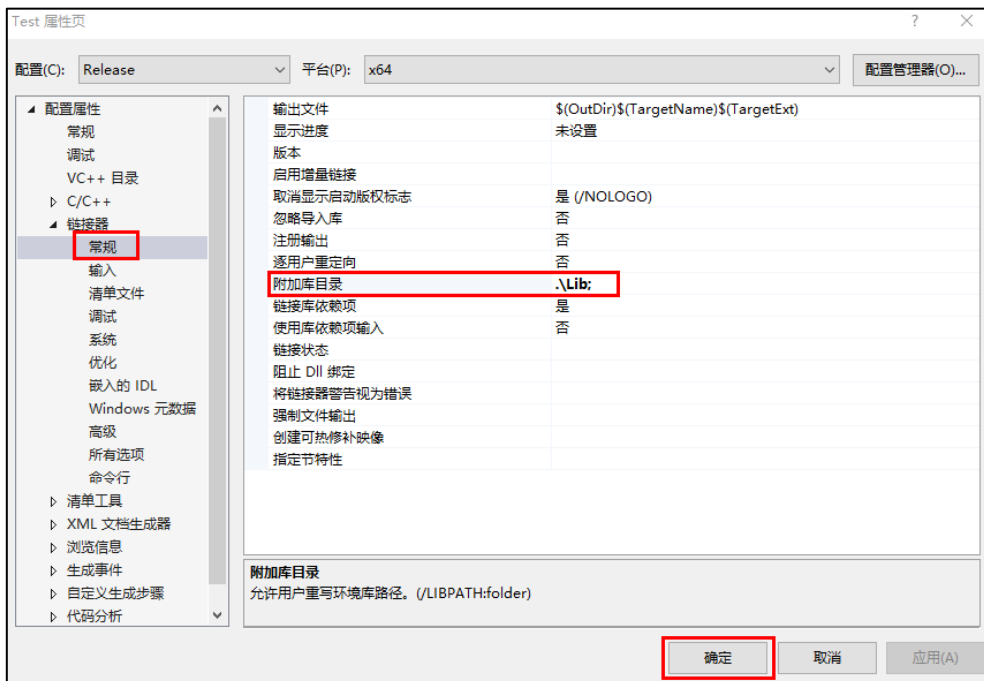


图 3-18 附加库设置

链接器->输入->附加依赖项，将 lib 库全称添加至附加依赖项，点击确定。

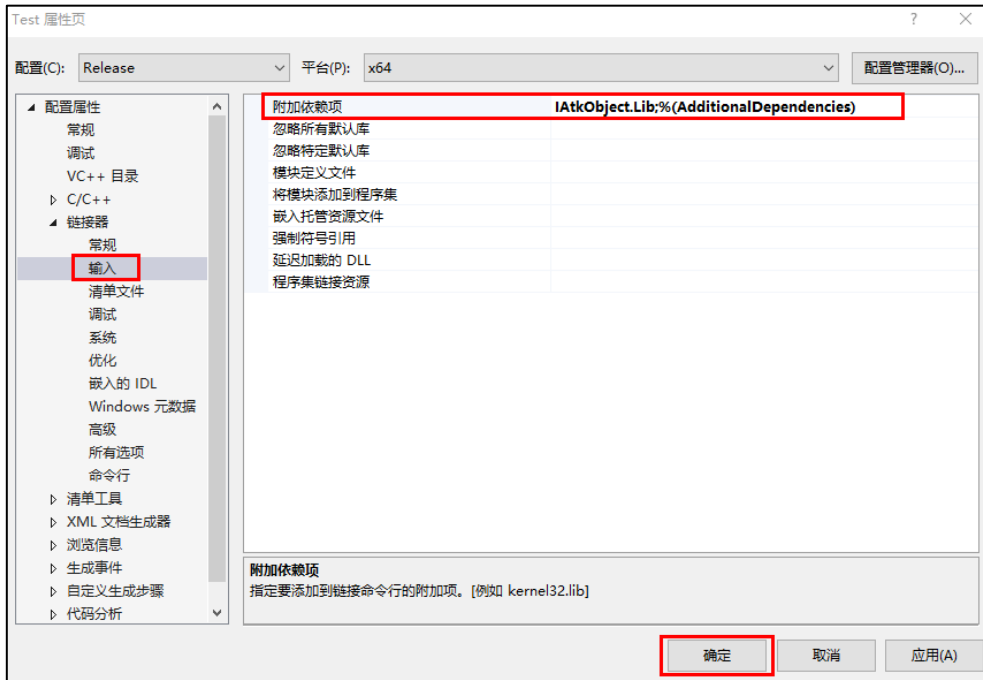


图 3-19 附加库设置

3.1.7. 添加包含文件，编写代码

```
#pragma once
#include <iostream>
#include <stdlib.h>
#include "IAtkObjectH.h"
using namespace std;

int main()
{
```

图 3-20 包含文件

```
64 IVADControl* pIVADControl2 = nullptr;
65 IVADResult* pIVADResult = nullptr;
66 IVADResult* pIVADResult1 = nullptr;
67 IVADResult* pIVADResult2 = nullptr;
68 pIScen = (IScenario*)pRoot->Children->New(eScenario, "FastTransfer");
69 if (nullptr == pIScen) return false;
70 pIScen->SetTimePeriod("5 Nov 2022 00:00:00.000", "6 Nov 2022 00:00:00.000");
71 pISatellite = (ISatellite*)pIScen->Children->New(eSatellite, "Satellite1");
72 pISatellite->SetPropagatorType(ePropagatorAstrogator);
73 pIVADriverMCS = (IVADriverMCS*)pISatellite->Propagator;
74 pIVAMCSSegmentCollection = pIVADriverMCS->GetMainSequence();
75
76 if (eVASegmentTypeInitialState == pIVAMCSSegmentCollection->Item(0)->Type)
77 {
78     pIVAMCSInitialState = (IVAMCSInitialState*)pIVAMCSSegmentCollection->Item(0);
79 }
80 //pIVAMCSPropagate = (IVAMCSPropagate*)pIVAMCSSegmentCollection->Item(1);
81 pIVAMCSPropagate = (IVAMCSPropagate*)pIVAMCSSegmentCollection->Insert(eVASegmentTypePropagate, "Propagate", "-");
82 pIVAMCSTargetSequence = (IVAMCSTargetSequence*)pIVAMCSSegmentCollection->Insert(eVASegmentTypeTargetSequence, "TargetSequence", "-");
83 pIVAMCSManeuver = (IVAMCSManeuver*)pIVAMCSTargetSequence->GetSegments()->Insert(eVASegmentTypeManeuver, "Maneuver", "-");
```

图 3-21 设置代码

3.1.8. 查看生成文件



图 3-22 生成文件

3.1.9. 仿真轨迹

使用 ATK 打开生成文件，可以查看仿真轨迹。

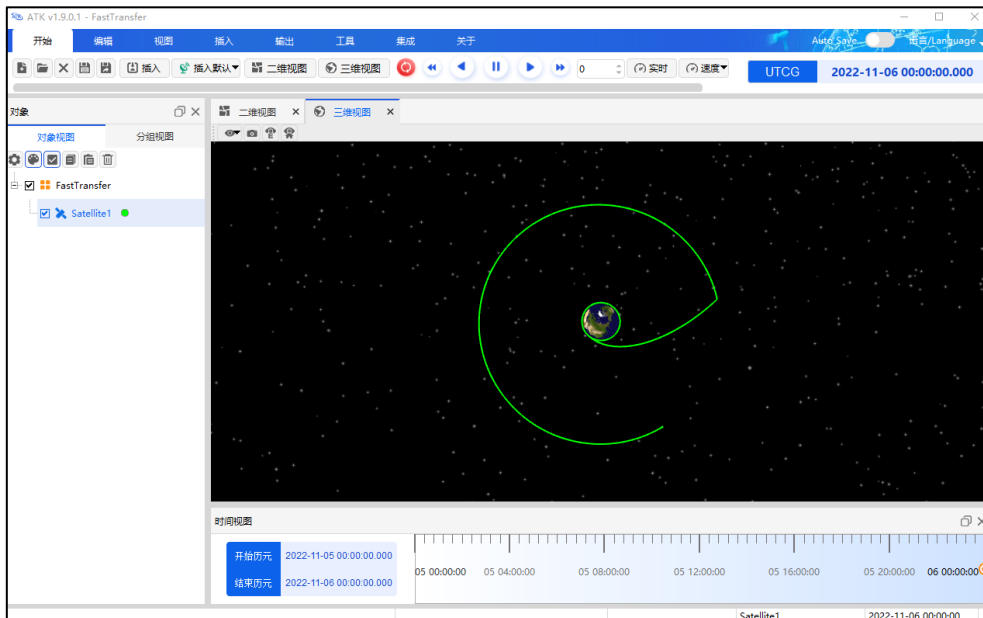


图 3-23 仿真轨迹

3.1.10. 代码示例

- // 代码结构功能流程说明
- // (1) 包含所需头文件，并添加执行函数
- // (2) 添加根节点
- // (3) 想定新建与属性设置
- // (4) 卫星新建与轨道预报设置为机动规划
- // (5) 机动规划添加段，新添加卫星会有默认初始段
- // (6) 初始段属性设置

```

// (7) 第一个预报段属性设置
// (8) 第一个瞄准段中机动段属性设置
// (9) 第一个瞄准段添加属性页，并设置属性页中控制变量与约束条件的属性
// (10) 第二个预报段属性设置
// (11) 第二个瞄准段中机动段属性设置
// (12) 第二个瞄准段添加属性页，并设置属性页中控制变量与约束条件的属性
// (13) 第三个预报段属性设置
// (14) 机动规划运行
// (15) 生成数据到文件
// (16) 想定保存及关闭
#pragma once
#include <iostream>
#include <stdlib.h>
#include "IAtkObjectH.h" //包含头文件
using namespace std;

int main()
{
    IAtkObjectRoot* pRoot = new IAtkObjectRoot();

    //
    //想定新建与属性设置
    //
    IScenario* pIScen = (IScenario*)pRoot->Children->New(eScenario, "FastTransfer");
    if (nullptr == pIScen) return false;
    pIScen->SetTimePeriod("5 Nov 2022 00:00:00.000", "6 Nov 2022
00:00:00.000");

    //
    //卫星新建与轨道预报设置为机动规划
    //
    ISatellite* pISatellite = (ISatellite*)pIScen->Children->New(eSatellite, "Satellite1");
    pISatellite->SetPropagatorType(ePropagatorAstrogator);
    IVADriverMCS* pIVADriverMCS = (IVADriverMCS*)pISatellite->Propagator;
    IVAMCSSegmentCollection* pIVAMCSSegmentCollection = pIVADriverMCS->GetMainSequence();
    //
    //机动规划添加段，新添加卫星会有默认初始段
    //
    IVAMCSInitialState* pIVAMCSInitialState = nullptr;

```

```

    if (eVASegmentTypeInitialState == pIVAMCSSegmentCollection->Item(0)->Type)
    {
        pIVAMCSInitialState = (IVAMCSInitialState*)pIVAMCSSegmentCollection->Item(0);
    }
    IVAMCSPropagate* pIVAMCSPropagate = (IVAMCSPropagate*)pIVAMCSSegmentCollection->Insert(eVASegmentTypePropagate, "Propagate", "-");
    IVAMCSTargetSequence* pIVAMCSTargetSequence = (IVAMCSTargetSequence*)pIVAMCSSegmentCollection->Insert(eVASegmentTypeTargetSequence, "TargetSequence", "-");
    IVAMCSManeuver* pIVAMCSManeuver = (IVAMCSManeuver*)pIVAMCSTargetSequence->GetSegments()->Insert(eVASegmentTypeManeuver, "Maneuver", "-");
    IVAMCSPropagate* pIVAMCSPropagate1 = (IVAMCSPropagate*)pIVAMCSSegmentCollection->Insert(eVASegmentTypePropagate, "Propagate", "-");
    IVAMCSTargetSequence* pIVAMCSTargetSequence1 = (IVAMCSTargetSequence*)pIVAMCSSegmentCollection->Insert(eVASegmentTypeTargetSequence, "TargetSequence1", "-");
    IVAMCSManeuver* pIVAMCSManeuver1 = (IVAMCSManeuver*)pIVAMCSTargetSequence1->GetSegments()->Insert(eVASegmentTypeManeuver, "Maneuver", "-");
    IVAMCSPropagate* pIVAMCSPropagate2 = (IVAMCSPropagate*)pIVAMCSSegmentCollection->Insert(eVASegmentTypePropagate, "Propagate", "-");

    //
    //初始段属性设置
    //
    pIVAMCSInitialState->SetOrbitEpoch("5 Nov 2022 00:00:00.000");
    pIVAMCSInitialState->SetElementType(eVAElementTypeKeplerian);
    IVAElementKeplerian* pIVAElementKeplerian = (IVAElementKeplerian*)pIVAMCSInitialState->GetElement();
    pIVAElementKeplerian->SetSemiMajorAxis(6700);
    pIVAElementKeplerian->SetEccentricity(0);
    pIVAElementKeplerian->SetInclination(0);
    pIVAElementKeplerian->SetRAAN(0);
    pIVAElementKeplerian->SetArgOfPeriapsis(0);
    pIVAElementKeplerian->SetTrueAnomaly(0);

    //
    //第一个预报段属性设置
    //
    IVAStoppingConditionElement* pIVAStoppingConditionElement =
    pIVAMCSPropagate->StoppingConditions->Add("Duration");

```

```

    IVAStoppingCondition* pIVAStoppingCondition = (IVAStoppingCondition*)pI-
VAStoppingConditionElement->Properties;
    pIVAStoppingCondition->SetTrip(7200);
    pIVAStoppingCondition->SetTolerance(0.0001);

    //
    //第一个瞄准段中机动段属性设置
    //
    pIVAMCSManeuver->SetManeuverType(eVAManeuverTypeImpulsive);
    IVAManeuverImpulsive* pIVAManeuverImpulsive = (IVAManeuverImpul-
sive*)pIVAMCSManeuver->Maneuver;
    IVAAttitudeControlImpulsiveThrustVector* pIVAAttitudeControlImpul-
siveThrustVector = (IVAAttitudeControlImpulsiveThrustVector*)pIVAManeuverIm-
pulsive->AttitudeControl;
    pIVAAttitudeControlImpulsiveThrustVector->SetThrustAxesName("Satellite
VNC(Earth)");
    pIVAMCSManeuver->EnableControlParameter(eVAControlManeuverImpul-
siveCartesianX);
    pIVAMCSManeuver->Results->Add("Radius_Of_Apoapsis");

    //
    //第一个瞄准段添加属性页
    //
    IVAProfileDifferentialCorrector1* pIVAProfileDifferentialCorrector1 =
(IVAProfileDifferentialCorrector1*)pIVAMCSTargetSequence->GetPro-
files()->Add("Differential Corrector");
    IVADCCControl* pIVADCCControl = pIVAProfileDifferentialCorrector1->Con-
trolParameters->GetControlByPaths("Maneuver", "ImpulseX");
    IVADCResult* pIVADCResult = pIVAProfileDifferentialCorrec-
tor1->GetResults()->GetResultByPaths("Maneuver", "RadiusOfApoapsis");
    //
    //属性页中控制变量属性设置
    //
    pIVADCCControl->SetEnable(true);
    pIVADCCControl->SetMaxStep(100);
    pIVADCCControl->SetCorrection(2781.50365947627);
    pIVADCCControl->SetPerturbation(0.1);
    pIVADCCControl->SetScalingValue(1);
    //
    //属性页中约束条件属性设置
    //
    pIVADCResult->SetEnable(true);
    pIVADCResult->SetDesiredValue(84328394);
    pIVADCResult->SetScalingValue(1);

```



```

pIVADCResult->SetTolerance(0.1);
pIVADCResult->SetWeight(1);

//
//第二个预报段属性设置
//
IVASToppingConditionElement* pIVASToppingConditionElement1 =
pIVAMCSPropagate1->StoppingConditions->Add("RMagnitude");
IVASToppingCondition* pIVASToppingCondition1 = (IVASToppingCondi-
tion*)pIVASToppingConditionElement1->Properties;
pIVASToppingCondition1->SetTrip(42164.197);
pIVASToppingCondition1->SetTolerance(1e-6);
pIVASToppingCondition1->SetRepeatCount(1);
pIVASToppingCondition1->SetCriterion(eVACriterionCrossEither);

//
//第二个瞄准段中机动段属性设置
//
pIVAMCSManeuver1->SetManeuverType(eVAManeuverTypeImpulsive);
IVAManeuverImpulsive* pIVAManeuverImpulsive1 = (IVAManeuverImpul-
sive*)pIVAMCSManeuver1->Maneuver;
IVAAttitudeControlImpulsiveThrustVector* pIVAAttitudeControlImpul-
siveThrustVector1 = (IVAAttitudeControlImpulsiveThrustVector*)pIVAManeuver-
Impulsive1->AttitudeControl;
pIVAAttitudeControlImpulsiveThrustVector1->SetThrustAxesName("Satellite
VNC(Earth)");
pIVAMCSManeuver1->EnableControlParameter(eVAControlManeuverImpul-
siveCartesianX);
pIVAMCSManeuver1->EnableControlParameter(eVAControlManeuverImpul-
siveCartesianZ);
pIVAMCSManeuver1->Results->Add("Eccentricity");
pIVAMCSManeuver1->Results->Add("Cosine_of_Vertical_FPA");

//
//第二个瞄准段添加属性页
//
IVAProfileDifferentialCorrector1* pIVAProfileDifferentialCorrector11 =
(IVAProfileDifferentialCorrector1*)pIVAMCSTargetSequence1->GetPro-
files()->Add("Differential Corrector");
//
//属性页中控制变量属性设置
//
IVADCCControl* pIVADCCControl1 = pIVAProfileDifferentialCorrector11->Con-
trolParameters->Item(0);

```

```

pIVADCCControl1->SetEnable(true);
pIVADCCControl1->SetMaxStep(300);
pIVADCCControl1->SetCorrection(-1581.97670664023);
pIVADCCControl1->SetPerturbation(0.1);
pIVADCCControl1->SetScalingValue(1);
IVADCCControl* pIVADCCControl2 = pIVAProfileDifferentialCorrector11->ControlParameters->Item(1);
pIVADCCControl2->SetEnable(true);
pIVADCCControl2->SetMaxStep(300);
pIVADCCControl2->SetCorrection(-2771.82057041661);
pIVADCCControl2->SetPerturbation(0.1);
pIVADCCControl2->SetScalingValue(1);
//
//属性页中约束条件属性设置
//
IVADCResult* pIVADCResult1 = pIVAProfileDifferentialCorrector11->GetResults()->Item(0);
pIVADCResult1->SetEnable(true);
pIVADCResult1->SetDesiredValue(0);
pIVADCResult1->SetScalingValue(1);
pIVADCResult1->SetTolerance(0.1);
pIVADCResult1->SetWeight(1);
IVADCResult* pIVADCResult2 = pIVAProfileDifferentialCorrector11->GetResults()->Item(1);
pIVADCResult2->SetEnable(true);
pIVADCResult2->SetDesiredValue(0);
pIVADCResult2->SetScalingValue(1);
pIVADCResult2->SetTolerance(0.1);
pIVADCResult2->SetWeight(1);

//
//第三个预报段属性设置
//
IVASToppingConditionElement* pIVASToppingConditionElement2 =
pIVAMCSPropagate2->StoppingConditions->Add("Duration");
IVASToppingCondition* pIVASToppingCondition2 = (IVASToppingCondition*)pIVASToppingConditionElement2->Properties;
pIVASToppingCondition2->SetTrip(86400);
pIVASToppingCondition2->SetTolerance(0.0001);

//
//机动规划运行
//
pIVADriverMCS->RunMCS();

```

```

pIVADriverMCS->ApplyAllProfileChanges();

//
//生成数据到文件
//
std::string strReportFilePath = pRoot->OutputDataReport(pISatellite, " J2000位
置速度", "5 Nov 2022 00:00:00.000", "6 Nov 2022 00:00:00.000");
cout << "报告文件位置:  " << strReportFilePath << endl;
pRoot->m_Animation->OutputData();
pRoot->m_Animation->OutputPropCortsData(pISatellite);

//
//想定保存及关闭
//
pRoot->SaveScenario();
pRoot->CloseScenario();

delete pRoot;
pRoot = nullptr;
system("Pause");
}

```

3.2. Java 语言操作流程

3.2.1. swigwin-4.1.1 安装

官网地址: <http://www.swig.org/tutorial.html>

下载地址: <https://sourceforge.net/projects/swig/files/swigwin/> (下载后解压即可)

3.2.2. Java 安装与环境配置

下载并安装 JDK(Java Development Kit), 官网 www.oracle.com/java/technologies/download。本章以 1.8.0_301 版本安装包 jdk-8u301-windows-x64.exe 为例, 直接双击安装, 默认安装路径” C:\Program Files\Java\jdk1.8.0_301”。

3.2.2.1. 配置 Java 路径到系统环境变量

安装完 JDK 后, 需要将 Java 路径配置到系统环境变量中。

JAVA_HOME 设置路径: C:\Program Files\Java\jdk1.8.0_301

Path 添加路径: %JAVA_HOME%\bin

Path 添加路径: %JAVA_HOME%\jre\bin

CLASSPATH 添加路

径: .;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar;

3.2.2.1.1. 系统环境变量配置过程（以 JAVA_HOME 为例）

Step1: 右键点击此电脑，再点击弹出框属性。

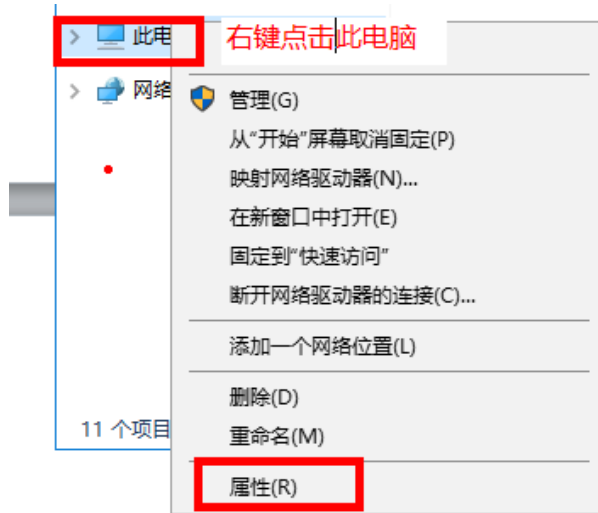


图 3-24 打开电脑属性

Step2: 点击控制面板框下系统高级设置，再点击系统属性框高级选项下的环境变量。

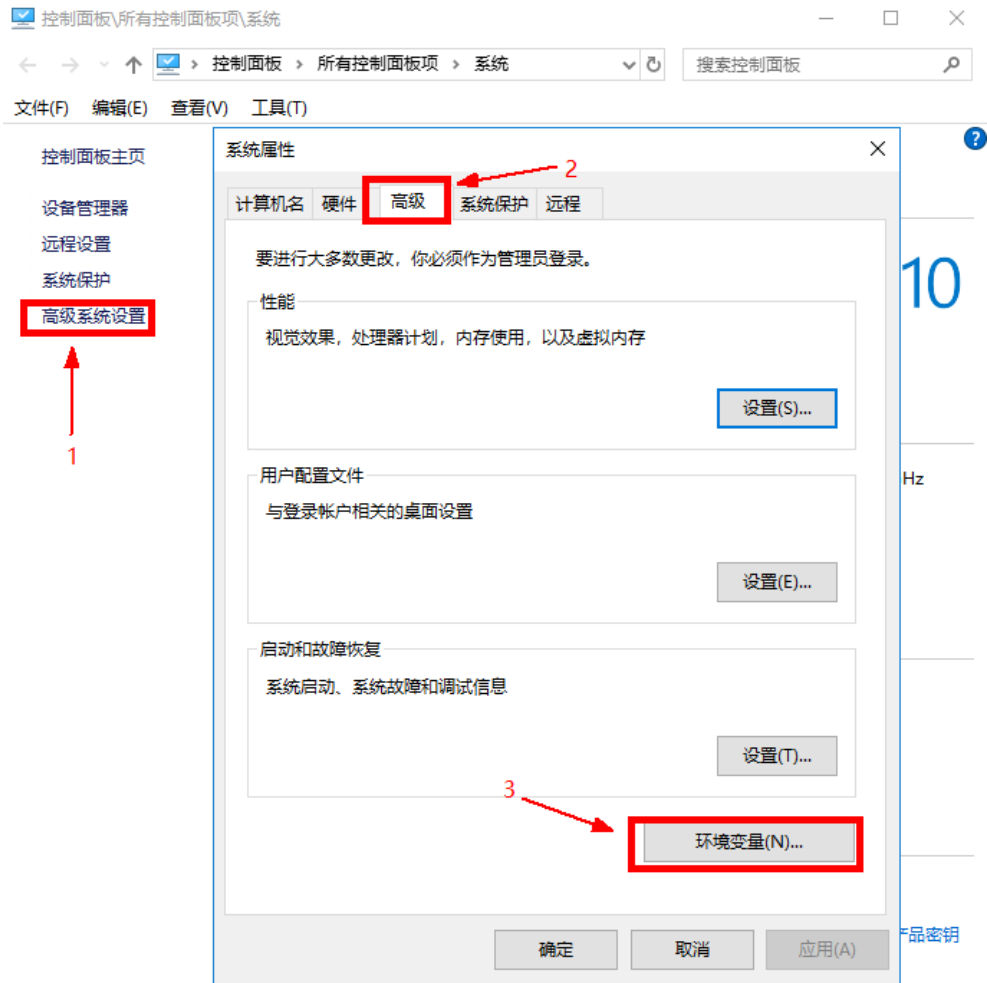


图 3-25 打开系统环境变量

Step3: 在弹出的环境变量框系统变量中新建（已存在则点击编辑）并输入变量名 JAVA_HOME 与 JAVA 路径，最后点击确定。

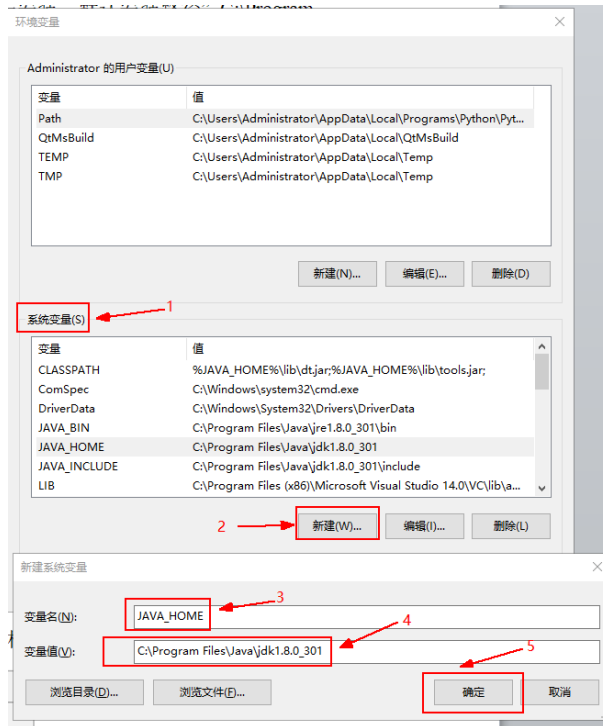


图 3-26 新建系统环境变量

Step4: 新建完变量 JAVA_HOME 后, 再依此点击环境变量框最下方确定, 以及系统属性框最下方的确定, 至此, 新建环境变量 JAVA_HOME 保存成功。

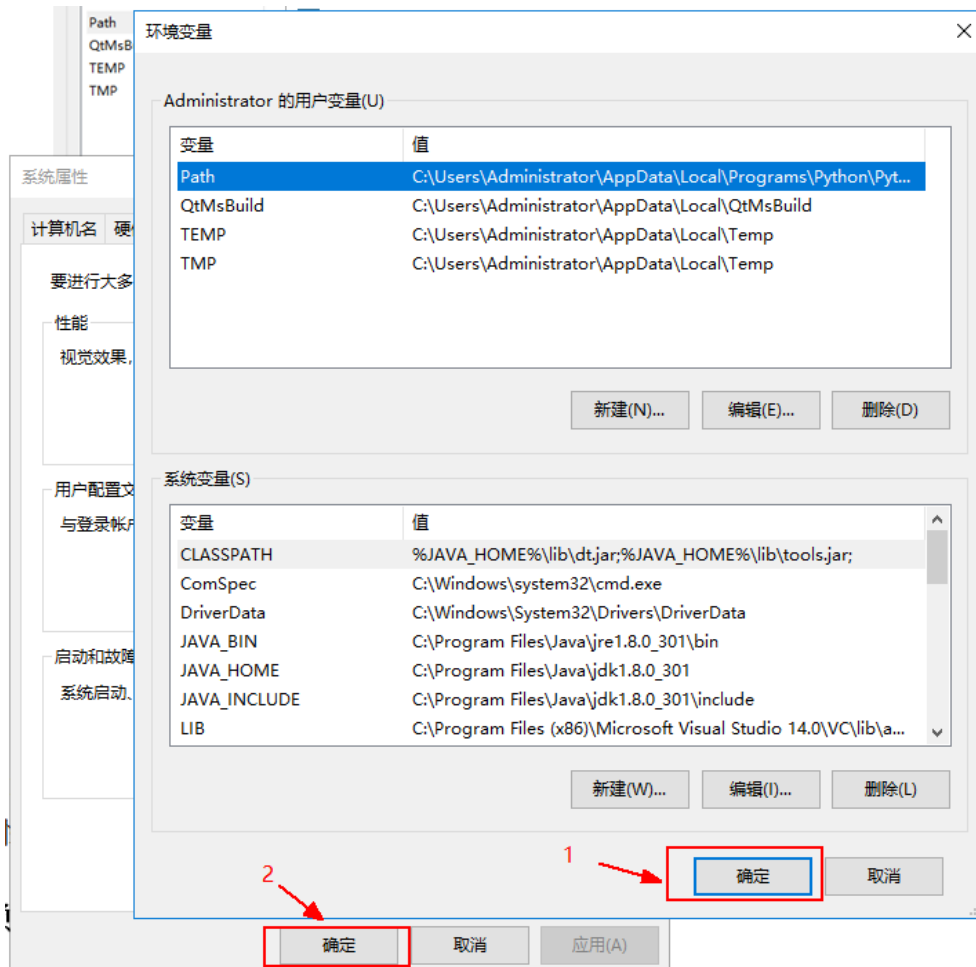


图 3-27 保存系统环境变量

3.2.3. SWIG 将 C++ 转成 Java 封装文件

3.2.3.1. 编写 SWIG 接口文件

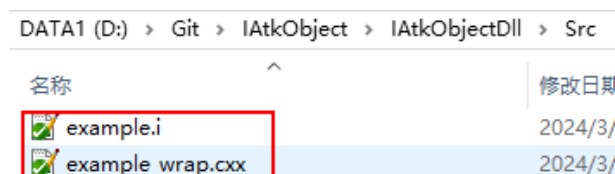


图 3-28 SWIG C++ 转 Java 接口封装与生成文件

```
// example.i
```

```
//Java 模块名
```

```
%module example
```

```
{
```

```
//包含需要调用的 C++ 声明文件，以本接口文件位置为基础
```

```

#include "../IAtkObj.h"
#include "../Include/AsObjManage.h"
#include "Common.h"
#include "IAnimation.h"
#include "IAtkObject.h"
#include "IAtkObjectCollection.h"
#include "IAtkObjectRoot.h"
#include "ICEnum.h"
#include "IScenario.h"
#include "IGreatArcVehicle.h"
include "ICObjectFunction.h"

#include "AccessConstraints/IAccessConstraint.h"
#include "AccessConstraints/IAccessCnstrAngle.h"
#include "AccessConstraints/IAccessCnstrMinMax.h"
#include "AccessConstraints/IAccessConstraintCollection.h"

#include "Aircraft/IAircraft.h"
    include "Satellite/Propagator/IVePropagator.h"
#include "Aircraft/IVePropagatorGreatArc.h"
#include "Aircraft/IVeWaypointsCollection.h"
#include "Aircraft/IVeWaypointsElement.h"

#include "Chain/IChain.h"
#include "Chain/IObjectLinkCollection.h"
#include "Chain/IObjectLink.h"

#include "CoverageDefinition/ICoverageDefinition.h"
#include "CoverageDefinition/ICvGrid.h"
#include "CoverageDefinition/ICvBounds.h"
#include "CoverageDefinition/ICvBoundsGlobal.h"
#include "CoverageDefinition/ICvBoundsLat.h"
#include "CoverageDefinition/ICvBoundsLatLine.h"
#include "CoverageDefinition/ICvBoundsLatLonRegion.h"
#include "CoverageDefinition/ICvBoundsLonLine.h"

```



```
#include "CoverageDefinition/ICvResolution.h"
#include "CoverageDefinition/ICvResolutionArea.h"
#include "CoverageDefinition/ICvResolutionDistance.h"
#include "CoverageDefinition/ICvResolutionLatLon.h"

#include "Facility/IFacility.h"
    #include "Sensor/ILocationData.h"
#include "Facility/IPosition.h"
#include "Facility/ICartesian.h"
#include "Facility/IGeodetic.h"

#include "Graphics/ISnGraphics.h"
#include "Graphics/ISnProjection.h"
#include "Graphics/IDisplayDistance.h"
#include "Graphics/ISnProjConstantAlt.h"
#include "Graphics/ISnProjObjectAlt.h"

#include "GroundVehicle/IGroundVehicle.h"

#include "Planet/IPlanet.h"
#include "Planet/IPCommonTasks.h"
#include "Planet/IPositionSourceData.h"
#include "Planet/IPIPosCentralBody.h"

#include "Receiver/IReceiver.h"
#include "Receiver/IReceiverModel.h"
#include "Receiver/IReceiverModelSimple.h"
#include "Receiver/IPolarization.h"
#include "Receiver/IPolarizationNone.h"
#include "Receiver/IPolarizationLinear.h"
#include "Receiver/IPolarizationCircular.h"
#include "Receiver/IPolarizationVertical.h"
#include "Receiver/IPolarizationHorizontal.h"
#include "Receiver/IPolarizationElliptical.h"
#include "Receiver/ILinkMargin.h"
```

```
#include "Receiver/IDemodulatorModel.h"
#include "Receiver/IDemodulatorModelBpsk.h"
#include "Receiver/IDemodulatorModelQpsk.h"
#include "Receiver/IReceiverModelMedium.h"

#include "Sensor/ISensor.h"
#include "Sensor/ISnCommonTasks.h"
#include "Sensor/ISnPattern.h"
#include "Sensor/ISnRectangularPattern.h"
#include "Sensor/ISnSimpleConicPattern.h"
#include "Sensor/ILocationData.h"
#include "Sensor/ISnPointing.h"
#include "Sensor/ISnPtFixed.h"
#include "Sensor/IOrientation.h"
#include "Sensor/IOrientationEulerAngles.h"
#include "Sensor/IOrientationQuaternion.h"

#include "Ship/IShip.h"

#include "Star/IStar.h"

#include "Submarine/ISubmarine.h"

#include "Transmitter/ITransmitter.h"
#include "Transmitter/ITransmitterModel.h"
#include "Transmitter/ITransmitterModelSimple.h"
#include "Transmitter/ITransmitterModelMedium.h"
#include "Transmitter/IModulatorModel.h"
#include "Transmitter/IModulatorModelBpsk.h"
#include "Transmitter/IModulatorModelQpsk.h"

#include "Satellite/ISatellite.h"

#include "Satellite/Astrogator/IComponentInfo.h"
#include "Satellite/Astrogator/IVAMCSSegment.h"
```

```

#include "Satellite/Astrogator/EndSegment/IVAMCSEnd.h"

#include "Satellite/Astrogator/HoldSegment/IVAMCSHold.h"

#include "Satellite/Astrogator/InitialStateSegment/IVAMCSInitialState.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAEElement.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAEElementCartesian.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAEElementKeplerian.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAFuelTank.h"
#include "Satellite/Astrogator/InitialStateSegment/IVASpacecraftParameters.h"

#include "Satellite/Astrogator/ManeuverSegment/IVAMCSManeuver.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuver.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuverFinite.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuverImpulsive.h"
#include "Satellite/Astrogator/ManeuverSegment/IDirection.h"
#include "Satellite/Astrogator/ManeuverSegment/IDirectionRADec.h"
#include "Satellite/Astrogator/ManeuverSegment/IDirectionXYZ.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControl.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlFinite.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlFiniteThrustVector.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlImpulsive.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlImpulsiveThrustVector.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuverFinitePropagator.h"

#include "Satellite/Astrogator/PropagateSegment/IVAMCSPropagate.h"
#include "Satellite/Astrogator/PropagateSegment/IVASToppingConditionComponent.h"
#include "Satellite/Astrogator/PropagateSegment/IVASToppingCondition.h"
#include "Satellite/Astrogator/PropagateSegment/IVASToppingConditionCollection.h"

```

```
#include "Satellite/Astrogator/PropagateSegment/IVASToppingCondi-  
tionElement.h"
```

```
#include "Satellite/Astrogator/Results/IVACalcObjectCollection.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcAltOfApoapsis.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcAltOfPeriapsis.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcArgOfLat.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcArgOfPeriapsis.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcBetaAngle.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcCartesianElem.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcCosOfVerticalFPA.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcDec.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcDeltaDec.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcDeltaFromMaster.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcDeltaRA.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcDeltaV.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcEccAnomaly.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcEccentricity.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcEquinoctialElem.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcFPA.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcGeodeticElem.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcInclination.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcLonDriftRate.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcMeanAnomaly.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcMeanEarthLon.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcOrbitPeriod.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcPosDifferenceOtherSeg-  
ment.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcPosVelDifferenceOtherSeg-  
ment.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcRA.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcRAAN.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcRadOfApoapsis.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcRadOfPeriapsis.h"  
#include "Satellite/Astrogator/Results/IVASTateCalcRMag.h"
```

```

#include "Satellite/Astrogator/Results/IVASStateCalcSegEpochDifference.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSegStateCalcDifference.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSemiMajorAxis.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSphRMag.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSphVMag.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSunlight.h"
#include "Satellite/Astrogator/Results/IVASStateCalcTrueAnomaly.h"
#include "Satellite/Astrogator/Results/IVASStateCalcTrueLon.h"
#include "Satellite/Astrogator/Results/IVASStateCalcVelAz.h"
#include "Satellite/Astrogator/Results/IVASStateCalcVelDifferenceOtherSegment.h"

#include "Satellite/Astrogator/Results/IVASStateCalcVMag.h"

#include "Satellite/Astrogator/ReturnSegment/IVAMCSReturn.h"

#include "Satellite/Astrogator/SequenceSegment/IVAMCSSequence.h"
#include "Satellite/Astrogator/SequenceSegment/IVAMCSBackwardSequence.h"

#include "Satellite/Astrogator/StopSegment/IVAMCSStop.h"

#include "Satellite/Astrogator/TargetSequenceSegment/IVAMCSTargetSequence.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCControl.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCControlCollection.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCResult.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCResultCollection.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfile.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfileCollection.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfileDifferentialCorrector1.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfileDifferentialCorrector2.h"

```

#include "Satellite/Astrogator/UpdateSegment/IVAMCSUpdate.h"

#include "Satellite/Astrogator/IVAMCSSegmentCollection.h"

#include "Satellite/Astrogator/IComponentInfo.h"

#include "Satellite/Astrogator/IVAMCSSegment.h"

#include "Satellite/Astrogator/IVAMCSSegmentProperties.h"

#include "Satellite/Astrogator/IVASState.h"

#include "Satellite/Attitude/IVeAttitude.h"

#include "Satellite/Attitude/IVeAttitudeStandard.h"

#include "Satellite/Attitude/IVeAttProfile.h"

#include "Satellite/Attitude/IVeOrbitAttitudeStandard.h"

#include "Satellite/Attitude/IVeStandardBasic.h"

#include "Satellite/HPOP/ISRPMModelBase.h"

#include "Satellite/HPOP/ISRPMModelSpherical.h"

#include "Satellite/HPOP/IVeHPOPCentralBodyGravity.h"

#include "Satellite/HPOP/IVeHPOPDragModel.h"

#include "Satellite/HPOP/IVeHPOPDragModelSpherical.h"

#include "Satellite/HPOP/IVeHPOPForceModel.h"

#include "Satellite/HPOP/IVeHPOPForceModelDrag.h"

#include "Satellite/HPOP/IVeHPOPSolarRadiationPressure.h"

#include "Satellite/HPOP/IVeHPOPSRPMModel.h"

#include "Satellite/HPOP/IVeInitialState.h"

#include "Satellite/HPOP/IVeIntegrator.h"

#include "Satellite/HPOP/IVeSolarFluxGeoMag.h"

#include "Satellite/HPOP/IVeSolarFluxGeoMagEnterManually.h"

#include "Satellite/HPOP/IVeStepSizeControl.h"

#include "Satellite/HPOP/IVeThirdBodyGravityCollection.h"

#include "Satellite/HPOP/IVeThirdBodyGravityElement.h"

#include "Satellite/Mass/IVeInertia.h"

#include "Satellite/Mass/IVeMassProperties.h"

#include "Satellite/OrbitState/IOrbitState.h"

```

#include "Satellite/OrbitState/IClassicalLocation.h"
#include "Satellite/OrbitState/IClassicalLocationArgumentOfLatitude.h"
#include "Satellite/OrbitState/IClassicalLocationEccentricAnomaly.h"
#include "Satellite/OrbitState/IClassicalLocationMeanAnomaly.h"
#include "Satellite/OrbitState/IClassicalLocationTimePastAN.h"
#include "Satellite/OrbitState/IClassicalLocationTimePastPerigee.h"
#include "Satellite/OrbitState/IClassicalLocationTrueAnomaly.h"
#include "Satellite/OrbitState/IClassicalOrientation.h"
#include "Satellite/OrbitState/IClassicalSizeShape.h"
#include "Satellite/OrbitState/IClassicalSizeShapeAltitude.h"
#include "Satellite/OrbitState/IClassicalSizeShapeMeanMotion.h"
#include "Satellite/OrbitState/IClassicalSizeShapePeriod.h"
#include "Satellite/OrbitState/IClassicalSizeShapeRadius.h"
#include "Satellite/OrbitState/IClassicalSizeShapeSemimajorAxis.h"
#include "Satellite/OrbitState/IFlightPathAngle.h"
#include "Satellite/OrbitState/IOrbitStateCartesian.h"
#include "Satellite/OrbitState/IOrbitStateClassical.h"
#include "Satellite/OrbitState/IOrbitStateSpherical.h"
#include "Satellite/OrbitState/IOrientationAscNode.h"
#include "Satellite/OrbitState/IOrientationAscNodeLAN.h"
#include "Satellite/OrbitState/IOrientationAscNodeRAAN.h"
#include "Satellite/OrbitState/ISphericalFPAHorizontal.h"

#include "Satellite/PassBreak/IVePassBreak.h"
#include "Satellite/PassBreak/IVeBreakAngle.h"
#include "Satellite/PassBreak/IVeBreakAngleBreakByLatOrLon.h"
#include "Satellite/PassBreak/IVeDefinition.h"

// #include "Satellite/Propagator/IVePropagator.h"
#include "Satellite/Propagator/IVADriverMCS.h"
#include "Satellite/Propagator/IVePropagatorHPOP.h"
#include "Satellite/Propagator/IVePropagatorJ2Perturbation.h"
#include "Satellite/Propagator/IVePropagatorSGP4.h"
#include "Satellite/Propagator/IVePropagatorTwoBody.h"

```

```

#include "Satellite/SGP4/IVePropagatorSGP4CommonTasks.h"
#include "Satellite/SGP4/IVeSGP4Segment.h"
#include "Satellite/SGP4/IVeSGP4SegmentCollection.h"
%}

//swig 系统自带的常用封装类型接口
#include "std_string.i"
#include "cpointer.i"
%pointer_functions(double, double_p);
#include "std_vector.i"
%template(vector_string)std::vector<std::string>;
%template(vector_IVeWaypointsElement)std::vector<IVeWaypointsElement*>;
%template(vector_IObjectLink)std::vector<IObjectLink*>;

//要忽略的文件/类/函数/变量
%ignore IAsGObjManage;

//包含需要调用的 C++声明文件，以本接口文件位置为基础
#include "../IAtkObj.h"
#include "Common.h"
#include "IAnimation.h"
#include "IAtkObject.h"
#include "IAtkObjectCollection.h"
#include "IAtkObjectRoot.h"
#include "ICEnum.h"
#include "IScenario.h"
#include "IGreatArcVehicle.h"
//%include "IObjectFunction.h"

#include "AccessConstraints/IAccessConstraint.h"
#include "AccessConstraints/IAccessCnstrAngle.h"
#include "AccessConstraints/IAccessCnstrMinMax.h"
#include "AccessConstraints/IAccessConstraintCollection.h"

#include "Aircraft/IAircraft.h"

```



```

    %include "Satellite/Propagator/IVePropagator.h"
%include "Aircraft/IVePropagatorGreatArc.h"
%include "Aircraft/IVeWaypointsCollection.h"
%include "Aircraft/IVeWaypointsElement.h"

%include "Chain/IChain.h"
%include "Chain/IObjectLinkCollection.h"
%include "Chain/IObjectLink.h"

%include "CoverageDefinition/ICoverageDefinition.h"
%include "CoverageDefinition/ICvGrid.h"
%include "CoverageDefinition/ICvBounds.h"
%include "CoverageDefinition/ICvBoundsGlobal.h"
%include "CoverageDefinition/ICvBoundsLat.h"
%include "CoverageDefinition/ICvBoundsLatLine.h"
%include "CoverageDefinition/ICvBoundsLatLonRegion.h"
%include "CoverageDefinition/ICvBoundsLonLine.h"
%include "CoverageDefinition/ICvResolution.h"
%include "CoverageDefinition/ICvResolutionArea.h"
%include "CoverageDefinition/ICvResolutionDistance.h"
%include "CoverageDefinition/ICvResolutionLatLon.h"

%include "Facility/IFacility.h"
    %include "Sensor/ILocationData.h"
%include "Facility/IPosition.h"
%include "Facility/ICartesian.h"
%include "Facility/IGeodetic.h"

%include "Graphics/ISnGraphics.h"
%include "Graphics/ISnProjection.h"
%include "Graphics/IDisplayDistance.h"
%include "Graphics/ISnProjConstantAlt.h"
%include "Graphics/ISnProjObjectAlt.h"

%include "GroundVehicle/IGroundVehicle.h"

```

```

#include "Planet/IPlanet.h"
#include "Planet/IPCommonTasks.h"
#include "Planet/IPositionSourceData.h"
#include "Planet/IPIPosCentralBody.h"

#include "Receiver/IRceiver.h"
#include "Receiver/IRceiverModel.h"
#include "Receiver/IRceiverModelSimple.h"
#include "Receiver/IPolarization.h"
#include "Receiver/IPolarizationNone.h"
#include "Receiver/IPolarizationLinear.h"
#include "Receiver/IPolarizationCircular.h"
#include "Receiver/IPolarizationVertical.h"
#include "Receiver/IPolarizationHorizontal.h"
#include "Receiver/IPolarizationElliptical.h"
#include "Receiver/ILinkMargin.h"
#include "Receiver/IDemodulatorModel.h"
#include "Receiver/IDemodulatorModelBpsk.h"
#include "Receiver/IDemodulatorModelQpsk.h"
#include "Receiver/IRceiverModelMedium.h"

#include "Sensor/ISensor.h"
#include "Sensor/ISnCommonTasks.h"
#include "Sensor/ISnPattern.h"
#include "Sensor/ISnRectangularPattern.h"
#include "Sensor/ISnSimpleConicPattern.h"
#include "Sensor/ILocationData.h"
#include "Sensor/ISnPointing.h"
#include "Sensor/ISnPtFixed.h"
#include "Sensor/IOrientation.h"
#include "Sensor/IOrientationEulerAngles.h"
#include "Sensor/IOrientationQuaternion.h"

#include "Ship/IShip.h"

```

```

#include "Star/IStar.h"

#include "Submarine/ISubmarine.h"

#include "Transmitter/ITransmitter.h"
#include "Transmitter/ITransmitterModel.h"
#include "Transmitter/ITransmitterModelSimple.h"
#include "Transmitter/ITransmitterModelMedium.h"
#include "Transmitter/IModulatorModel.h"
#include "Transmitter/IModulatorModelBpsk.h"
#include "Transmitter/IModulatorModelQpsk.h"

#include "Satellite/ISatellite.h"

    #include "Satellite/Astrogator/IComponentInfo.h"
    #include "Satellite/Astrogator/IVAMCSSegment.h"
#include "Satellite/Astrogator/EndSegment/IVAMCSEnd.h"

#include "Satellite/Astrogator/HoldSegment/IVAMCSHold.h"

#include "Satellite/Astrogator/InitialStateSegment/IVAMCSInitialState.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAEElement.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAEElementCartesian.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAEElementKeplerian.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAFuelTank.h"
#include "Satellite/Astrogator/InitialStateSegment/IVASpacecraftParameters.h"

#include "Satellite/Astrogator/ManeuverSegment/IVAMCSManeuver.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuver.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuverFinite.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuverImpulsive.h"
#include "Satellite/Astrogator/ManeuverSegment/IDirection.h"
#include "Satellite/Astrogator/ManeuverSegment/IDirectionRADec.h"
#include "Satellite/Astrogator/ManeuverSegment/IDirectionXYZ.h"

```

```

#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControl.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlFinite.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlFiniteThrustVector.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlImpulsive.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlImpulsiveThrustVector.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuverFinitePropagator.h"

#include "Satellite/Astrogator/PropagateSegment/IVAMCSPropagate.h"
#include "Satellite/Astrogator/PropagateSegment/IVAStoppingConditionComponent.h"
#include "Satellite/Astrogator/PropagateSegment/IVAStoppingCondition.h"
#include "Satellite/Astrogator/PropagateSegment/IVAStoppingConditionCollection.h"
#include "Satellite/Astrogator/PropagateSegment/IVAStoppingConditionElement.h"

#include "Satellite/Astrogator/Results/IVACalcObjectCollection.h"
#include "Satellite/Astrogator/Results/IVAStateCalcAltOfApoapsis.h"
#include "Satellite/Astrogator/Results/IVAStateCalcAltOfPeriapsis.h"
#include "Satellite/Astrogator/Results/IVAStateCalcArgOfLat.h"
#include "Satellite/Astrogator/Results/IVAStateCalcArgOfPeriapsis.h"
#include "Satellite/Astrogator/Results/IVAStateCalcBetaAngle.h"
#include "Satellite/Astrogator/Results/IVAStateCalcCartesianElem.h"
#include "Satellite/Astrogator/Results/IVAStateCalcCosOfVerticalFPA.h"
#include "Satellite/Astrogator/Results/IVAStateCalcDec.h"
#include "Satellite/Astrogator/Results/IVAStateCalcDeltaDec.h"
#include "Satellite/Astrogator/Results/IVAStateCalcDeltaFromMaster.h"
#include "Satellite/Astrogator/Results/IVAStateCalcDeltaRA.h"
#include "Satellite/Astrogator/Results/IVAStateCalcDeltaV.h"
#include "Satellite/Astrogator/Results/IVAStateCalcEccAnomaly.h"
#include "Satellite/Astrogator/Results/IVAStateCalcEccentricity.h"

```

```

#include "Satellite/Astrogator/Results/IVASStateCalcEquinoctialElem.h"
#include "Satellite/Astrogator/Results/IVASStateCalcFPA.h"
#include "Satellite/Astrogator/Results/IVASStateCalcGeodeticElem.h"
#include "Satellite/Astrogator/Results/IVASStateCalcInclination.h"
#include "Satellite/Astrogator/Results/IVASStateCalcLonDriftRate.h"
#include "Satellite/Astrogator/Results/IVASStateCalcMeanAnomaly.h"
#include "Satellite/Astrogator/Results/IVASStateCalcMeanEarthLon.h"
#include "Satellite/Astrogator/Results/IVASStateCalcOrbitPeriod.h"
#include "Satellite/Astrogator/Results/IVASStateCalcPosDifferenceOtherSegment.h"
#include "Satellite/Astrogator/Results/IVASStateCalcPosVelDifferenceOtherSegment.h"
#include "Satellite/Astrogator/Results/IVASStateCalcRA.h"
#include "Satellite/Astrogator/Results/IVASStateCalcRAAN.h"
#include "Satellite/Astrogator/Results/IVASStateCalcRadOfApoapsis.h"
#include "Satellite/Astrogator/Results/IVASStateCalcRadOfPeriapsis.h"
#include "Satellite/Astrogator/Results/IVASStateCalcRMag.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSegEpochDifference.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSegStateCalcDifference.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSemiMajorAxis.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSphRMag.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSphVMag.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSunlight.h"
#include "Satellite/Astrogator/Results/IVASStateCalcTrueAnomaly.h"
#include "Satellite/Astrogator/Results/IVASStateCalcTrueLon.h"
#include "Satellite/Astrogator/Results/IVASStateCalcVelAz.h"
#include "Satellite/Astrogator/Results/IVASStateCalcVelDifferenceOtherSegment.h"
#include "Satellite/Astrogator/Results/IVASStateCalcVMag.h"

#include "Satellite/Astrogator/ReturnSegment/IVAMCSReturn.h"

#include "Satellite/Astrogator/SequenceSegment/IVAMCSSequence.h"
#include "Satellite/Astrogator/SequenceSegment/IVAMCSBackwardSequence.h"

```

```

#include "Satellite/Astrogator/StopSegment/IVAMCSStop.h"

#include "Satellite/Astrogator/TargetSequenceSegment/IVAMCSTargetSequence.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCControl.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCControlCollection.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCResult.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCResultCollection.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfile.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfileCollection.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfileDifferentialCorrector1.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfileDifferentialCorrector2.h"

#include "Satellite/Astrogator/UpdateSegment/IVAMCSUpdate.h"

#include "Satellite/Astrogator/IVAMCSSegmentCollection.h"
#include "Satellite/Astrogator/IComponentInfo.h"
#include "Satellite/Astrogator/IVAMCSSegment.h"
#include "Satellite/Astrogator/IVAMCSSegmentProperties.h"
#include "Satellite/Astrogator/IVASState.h"

#include "Satellite/Attitude/IVeAttitude.h"
#include "Satellite/Attitude/IVeAttitudeStandard.h"
#include "Satellite/Attitude/IVeAttProfile.h"
#include "Satellite/Attitude/IVeOrbitAttitudeStandard.h"
#include "Satellite/Attitude/IVeStandardBasic.h"

#include "Satellite/HPOP/ISRPMModelBase.h"
#include "Satellite/HPOP/ISRPMModelSpherical.h"
#include "Satellite/HPOP/IVeHPOPcentralBodyGravity.h"

```

```

#include "Satellite/HPOP/IVeHPOPDragModel.h"
#include "Satellite/HPOP/IVeHPOPDragModelSpherical.h"
#include "Satellite/HPOP/IVeHPOPForceModel.h"
#include "Satellite/HPOP/IVeHPOPForceModelDrag.h"
#include "Satellite/HPOP/IVeHPOPSolarRadiationPressure.h"
#include "Satellite/HPOP/IVeHPOPSRPMModel.h"
#include "Satellite/HPOP/IVeInitialState.h"
#include "Satellite/HPOP/IVeIntegrator.h"
#include "Satellite/HPOP/IVeSolarFluxGeoMag.h"
#include "Satellite/HPOP/IVeSolarFluxGeoMagEnterManually.h"
#include "Satellite/HPOP/IVeStepSizeControl.h"
#include "Satellite/HPOP/IVeThirdBodyGravityCollection.h"
#include "Satellite/HPOP/IVeThirdBodyGravityElement.h"

#include "Satellite/Mass/IVeInertia.h"
#include "Satellite/Mass/IVeMassProperties.h"

#include "Satellite/OrbitState/IOrbitState.h"
#include "Satellite/OrbitState/IClassicalLocation.h"
#include "Satellite/OrbitState/IClassicalLocationArgumentOfLatitude.h"
#include "Satellite/OrbitState/IClassicalLocationEccentricAnomaly.h"
#include "Satellite/OrbitState/IClassicalLocationMeanAnomaly.h"
#include "Satellite/OrbitState/IClassicalLocationTimePastAN.h"
#include "Satellite/OrbitState/IClassicalLocationTimePastPerigee.h"
#include "Satellite/OrbitState/IClassicalLocationTrueAnomaly.h"
#include "Satellite/OrbitState/IClassicalOrientation.h"
#include "Satellite/OrbitState/IClassicalSizeShape.h"
#include "Satellite/OrbitState/IClassicalSizeShapeAltitude.h"
#include "Satellite/OrbitState/IClassicalSizeShapeMeanMotion.h"
#include "Satellite/OrbitState/IClassicalSizeShapePeriod.h"
#include "Satellite/OrbitState/IClassicalSizeShapeRadius.h"
#include "Satellite/OrbitState/IClassicalSizeShapeSemimajorAxis.h"
#include "Satellite/OrbitState/IFlightPathAngle.h"
#include "Satellite/OrbitState/IOrbitStateCartesian.h"
#include "Satellite/OrbitState/IOrbitStateClassical.h"

```

```

#include "Satellite/OrbitState/IOrbitStateSpherical.h"
#include "Satellite/OrbitState/IOrientationAscNode.h"
#include "Satellite/OrbitState/IOrientationAscNodeLAN.h"
#include "Satellite/OrbitState/IOrientationAscNodeRAAN.h"
#include "Satellite/OrbitState/ISphericalFPAHorizontal.h"

#include "Satellite/PassBreak/IVePassBreak.h"
#include "Satellite/PassBreak/IVeBreakAngle.h"
#include "Satellite/PassBreak/IVeBreakAngleBreakByLatOrLon.h"
#include "Satellite/PassBreak/IVeDefinition.h"

//%include "Satellite/Propagator/IVePropagator.h"
#include "Satellite/Propagator/IVADriverMCS.h"
#include "Satellite/Propagator/IVePropagatorHPOP.h"
#include "Satellite/Propagator/IVePropagatorJ2Perturbation.h"
#include "Satellite/Propagator/IVePropagatorSGP4.h"
#include "Satellite/Propagator/IVePropagatorTwoBody.h"

#include "Satellite/SGP4/IVePropagatorSGP4CommonTasks.h"
#include "Satellite/SGP4/IVeSGP4Segment.h"
#include "Satellite/SGP4/IVeSGP4SegmentCollection.h"

```

3.2.3.2. 编译 SWIG 接口文件

编译 SWIG 接口文件：`swig -c++ -java -v -outdir D:\work\cppToJava\CppToJavaTest\javaJNI D:\work\cppToJava\CppToJavaTest\example.i`

swig 命令格式（空格分开）：`swig -源语言 -目标语言 -v(输出编译信息) -outdir java 文件输出目录 swig 编译文件目录`

3.2.4. 依赖文件

3.2.4.1. IAtkObjectDll.Dll

用途：提供 ATK 功能接口调用支持。

来源：由二次开发 Component 接口项目编译生成。

位置：放在 java 执行目录。

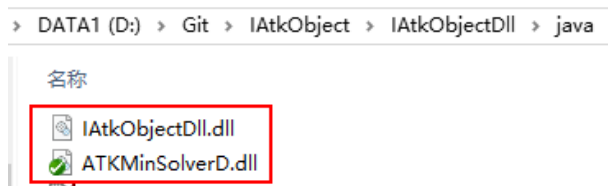


图 3-29 依赖的 ATK 动态库

3.2.4.2. ATKMinSolverD.dll

用途：提供 ATK 功能接口调用支持。

来源：由二次开发 Component 接口项目编译生成。

位置：放在 java 执行目录。

3.2.4.3. C++类型 Java 封装文件（.java）

用途：帮助 Java 能支持 C++接口项目中的各种数据结构与自定义类。

来源：编译 SWIG 接口文件后自动生成。比如接口文件中声明有 c++类文件 Ianimation.h，经由 SWIG 编译会生成 Ianimation.java 类封装文件。

位置：放在 java 执行目录。

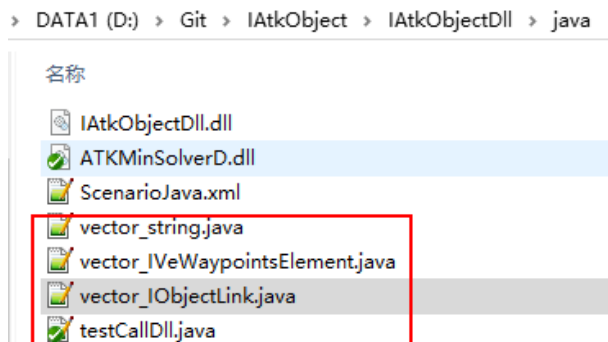


图 3-30 SWIG 生成的 java 接口封装文件

3.2.4.4. ATK 默认配置文件 AstroData

用途：用来配置 ATK 对象的默认值

来源：由 ATK 安装包提供，位置 D:\Release-ATK-v1.8.0.1-20231227\

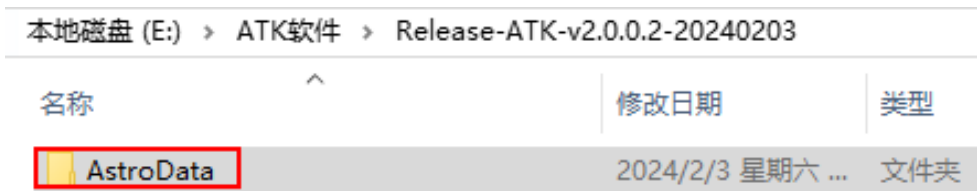


图 3-31 ATK 对象默认值数据库

位置：放在 java 安装目录 C:\Program Files\Java\jdk1.8.0_301\jre\bin\



图 3-32 ATK 对象默认值数据库放置位置

3.2.5. 调用动态库时 Java 与 C++的语法不同

3.2.5.1. 调用符号语法不同

C++: 因为项目中成员实现方式为指针, 只支持指针调用”->”。

举例: pIScen->Children;

Java: java 将所有类型进行了封装, 只支持点调用”.”。

举例: pIScenario.GetChildren();

3.2.5.2. 类成员调用方式不同

C++: 直接调用变量名获取

举例: pIScen->Children;

Java: 只能通过函数获取, java 会将 c++类中的所有公有成员自动封装 get 和 set 函数。

举例: pIScenario.GetChildren();

3.2.5.3. 枚举值使用不同

C++: 支持单独使用枚举值

举例: pISatellite->SetPropagatorType(ePropagatorAstrogator);

Java: 必须前置添加枚举类型。

举例:

pISatellite.SetPropagatorType(EVePropagatorType.ePropagatorAstrogator);

3.2.5.4. 类型转换方式不同

C++: 支持直接强制

举例: pIVADriverMCS = (IVADriverMCS*)pISatellite->Propagator;

Java: 必须调用接口函数实现 (变量实际指向在 C++中实现, java 无法获取变量的当前实际类型), 接口函数在 C++接口项目中实现, 因为没有封装成类, 对应的 java 封装函数实现在模板 java 文件 example.java 中。

举例: IVADriverMCS pIVADriverMCS = example.IVePropagator2IVADriverMCS(pISatellite.GetPropagator());

3.2.6. Java 测试代码示例 (快速转移)

```
public class testCallDll {
    static {
        try {
            System.loadLibrary("IAtkObjectDll");
        } catch (UnsatisfiedLinkError e) {
            System.err.println("Native code library failed to load. See the chapter on
Dynamic Linking Problems in the SWIG Java documentation for help.\n" + e);
            System.exit(1);
        }
    }

    public static void main(String []argv) {
        //root
        IAtkObjectRoot pIAtkObjectRoot = new IAtkObjectRoot();

        //new IScenario
        IAtkObject pIAtkObjectIScenario = pIAtkObjectRoot.Children().New(EATKObjectType.eScenario, "FastTransfer");
        IScenario pIScenario = example.IAtkObject2IScenario(pIAtkObjectIScenario);
        pIScenario.SetTimePeriod("5 Nov 2022 00:00:00.000", "6 Nov 2022
00:00:00.000");

        NewISatelliteFastTransfer(pIAtkObjectRoot, pIScenario);

        pIAtkObjectRoot.getM_Animation().PlayForward();
        pIAtkObjectRoot.SaveScenario();
        pIAtkObjectRoot.CloseScenario();
    }

    public static void NewISatelliteFastTransfer(IAtkObjectRoot
```

```

pIAtkObjectRoot,IScenario pIScenario){
    //new ISatellite
    IAtkObject pIAtkObjectSatellite = pIScenario.GetChildren().New(EATKOb-
jectType.eSatellite,"Satellite1");
    ISatellite pISatellite = example.IAtkObject2ISatellite(pIAtkObjectSatellite);
    //set ISatellite IVePropagator FastTransfer
    pISatellite.SetPropagatorType(EVePropagatorType.ePropagatorAstrogator);
    IVADriverMCS pIVADriverMCS = example.IVePropaga-
tor2IVADriverMCS(pISatellite.GetPropagator());
    IVAMCSSegmentCollection pIVAMCSSegmentCollection = pIVADriv-
erMCS.GetMainSequence();
    IVAMCSInitialState pIVAMCSInitialState = example.IVAMCS-
Segment2IVAMCSInitialState(pIVAMCSSegmentCollection.Item(0));
    //pIVAMCSPropagate = (IVAMCSPropagate*)pIVAMCS-
SegmentCollection.Item(1);
    IVAMCSPropagate pIVAMCSPropagate = example.IVAMCS-
Segment2IVAMCSPropagate(pIVAMCSSegmentCollection.Insert(EVASeg-
mentType.eVASegmentTypePropagate, "Propagate", "-"));
    IVAMCSTargetSequence pIVAMCSTargetSequence = exam-
ple.IVAMCSSegment2IVAMCSTargetSequence(pIVAMCSSegmentCollection.In-
sert(EVASegmentType.eVASegmentTypeTargetSequence, "TargetSequence", "-"));
    IVAMCSManeuver pIVAMCSManeuver = exam-
ple.IVAMCSSegment2IVAMCSManeuver(pIVAMCSTargetSequence.GetSeg-
ments().Insert(EVASegmentType.eVASegmentTypeManeuver, "Maneuver", "-"));
    IVAMCSPropagate pIVAMCSPropagate1 = example.IVAMCS-
Segment2IVAMCSPropagate(pIVAMCSSegmentCollection.Insert(EVASeg-
mentType.eVASegmentTypePropagate, "Propagate", "-"));
    IVAMCSTargetSequence pIVAMCSTargetSequence1 = exam-
ple.IVAMCSSegment2IVAMCSTargetSequence(pIVAMCSSegmentCollection.In-
sert(EVASegmentType.eVASegmentTypeTargetSequence, "TargetSequence1", "-"));
    IVAMCSManeuver pIVAMCSManeuver1 = example.IVAMCS-
Segment2IVAMCSManeuver(pIVAMCSTargetSequence1.GetSegments().In-
sert(EVASegmentType.eVASegmentTypeManeuver, "Maneuver", "-"));
    IVAMCSPropagate pIVAMCSPropagate2 =

```

```
example.IVAMCSSegment2IVAMCSPropagate(pIVAMCSSegmentCollection.Insert(EVASegmentType.eVASegmentTypePropagate, "Propagate", "-"));
```

```
    pIVAMCSInitialState.SetOrbitEpoch("5 Nov 2022 00:00:00.000");  
    pIVAMCSInitialState.SetElementType(EVAElementType.eVAElementTypeKeplerian);
```

```
    IVAElementKeplerian pIVAElementKeplerian = example.IVAElement2IVAElementKeplerian(pIVAMCSInitialState.GetElement());
```

```
    pIVAElementKeplerian.SetSemiMajorAxis(6700);  
    pIVAElementKeplerian.SetEccentricity(0);  
    pIVAElementKeplerian.SetInclination(0);  
    pIVAElementKeplerian.SetRAAN(0);  
    pIVAElementKeplerian.SetArgOfPeriapsis(0);  
    pIVAElementKeplerian.SetTrueAnomaly(0);
```

```
    IVAStoppingConditionElement pIVAStoppingConditionElement =  
pIVAMCSPropagate.GetStoppingConditions().Add("Duration");
```

```
    IVAStoppingCondition pIVAStoppingCondition = example.IVAStoppingConditionComponent2IVAStoppingCondition(pIVAStoppingConditionElement.GetProperties());
```

```
    pIVAStoppingCondition.SetTrip(7200);  
    pIVAStoppingCondition.SetTolerance(0.0001);
```

```
    pIVAMCSManeuver.SetManeuverType(EVAManeuverType.eVAManeuverTypeImpulsive);
```

```
    IVAManeuverImpulsive pIVAManeuverImpulsive = example.IVAManeuver2IVAManeuverImpulsive(pIVAMCSManeuver.GetManeuver());
```

```
    IVAAttitudeControlImpulsiveThrustVector pIVAAttitudeControlImpulsiveThrustVector = example.IVAAttitudeControl2IVAAttitudeControlImpulsiveThrustVector(pIVAManeuverImpulsive.GetAttitudeControl());
```

```
    pIVAAttitudeControlImpulsiveThrustVector.SetThrustAxesName("Satellite VNC(Earth)");
```

```
    pIVAMCSManeuver.EnableControlParameter(EVAControlManeuver.eVAControlManeuverImpulsiveCartesianX);
```

```
    pIVAMCSManeuver.getResults().Add("Radius_Of_Apoapsis");
```

```
IVAProfileDifferentialCorrector1 pIVAProfileDifferentialCorrector1 = exam-
ple.IVAProfile2IVAProfileDifferentialCorrector1(pIVAMCSTargetSequence.GetPro-
files().Add("Differential Corrector"));
```

```
IVADCCControl pIVADCCControl = pIVAProfileDifferentialCorrec-
tor1.GetControlParameters().GetControlByPaths("Maneuver", "ImpulseX");
```

```
IVADCCResult pIVADCCResult = pIVAProfileDifferentialCorrec-
tor1.getResults().GetResultByPaths("Maneuver", "RadiusOfApoapsis");
```

```
pIVADCCControl = pIVAProfileDifferentialCorrector1.GetControlParame-
ters().Item(0);
```

```
    pIVADCCControl.SetEnable(true);
```

```
    pIVADCCControl.SetMaxStep(100);
```

```
    pIVADCCControl.SetCorrection(2781.50365947627);
```

```
    pIVADCCControl.SetPerturbation(0.1);
```

```
    pIVADCCControl.SetScalingValue(1);
```

```
    pIVADCCResult = pIVAProfileDifferentialCorrector1.getResults().Item(0);
```

```
    pIVADCCResult.SetEnable(true);
```

```
    pIVADCCResult.SetDesiredValue(84328394);
```

```
    pIVADCCResult.SetScalingValue(1);
```

```
    pIVADCCResult.SetTolerance(0.1);
```

```
    pIVADCCResult.SetWeight(1);
```

```
IVASToppingConditionElement pIVASToppingConditionElement1 =
pIVAMCSPropagate1.GetStoppingConditions().Add("RMagnitude");
```

```
IVASToppingCondition pIVASToppingCondition1 = exam-
ple.IVASToppingConditionComponent2IVASToppingCondition(pIVASToppingCondi-
tionElement1.GetProperties());
```

```
    pIVASToppingCondition1.SetTrip(42164197);
```

```
    pIVASToppingCondition1.SetTolerance(1e-6);
```

```
    pIVASToppingCondition1.SetRepeatCount(1);
```

```
    pIVASToppingCondition1.SetCriterion(EVACriterion.eVACriterion-
CrossEither);
```

```

    pIVAMCSManeuver1.SetManeuverType(EVAManeuverType.eVAManeuverTypeImpulsive);

    IVAManeuverImpulsive pIVAManeuverImpulsive1 = example.IVAManeuver2IVAManeuverImpulsive(pIVAMCSManeuver1.GetManeuver());

    IVAAttitudeControlImpulsiveThrustVector pIVAAttitudeControlImpulsiveThrustVector1 = example.IVAAttitudeControl2IVAAttitudeControlImpulsiveThrustVector(pIVAManeuverImpulsive1.GetAttitudeControl());

    pIVAAttitudeControlImpulsiveThrustVector1.SetThrustAxesName("Satellite VNC(Earth)");

    pIVAMCSManeuver1.EnableControlParameter(EVAControlManeuver.eVAControlManeuverImpulsiveCartesianX);

    pIVAMCSManeuver1.EnableControlParameter(EVAControlManeuver.eVAControlManeuverImpulsiveCartesianZ);

    pIVAMCSManeuver1.getResults().Add("Eccentricity");
    pIVAMCSManeuver1.getResults().Add("Cosine_of_Vertical_FPA");

    IVAProfileDifferentialCorrector1 pIVAProfileDifferentialCorrector11 = example.IVAProfile2IVAProfileDifferentialCorrector1(pIVAMCSTargetSequence1.GetProfiles().Add("Differential Corrector"));

    IVADCCControl pIVADCCControl1 = pIVAProfileDifferentialCorrector11.GetControlParameters().Item(0);
    pIVADCCControl1.SetEnable(true);
    pIVADCCControl1.SetMaxStep(300);
    pIVADCCControl1.SetCorrection(-1581.97670664023);
    pIVADCCControl1.SetPerturbation(0.1);
    pIVADCCControl1.SetScalingValue(1);

    IVADCCControl pIVADCCControl2 = pIVAProfileDifferentialCorrector11.GetControlParameters().Item(1);
    pIVADCCControl2.SetEnable(true);
    pIVADCCControl2.SetMaxStep(300);
    pIVADCCControl2.SetCorrection(-2771.82057041661);
    pIVADCCControl2.SetPerturbation(0.1);
    pIVADCCControl2.SetScalingValue(1);

    IVADCCResult pIVADCCResult1 = pIVAProfileDifferentialCorrector11.getResults().Item(0);

```

```

    pIVADCResult1.SetEnable(true);
    pIVADCResult1.SetDesiredValue(0);
    pIVADCResult1.SetScalingValue(1);
    pIVADCResult1.SetTolerance(0.1);
    pIVADCResult1.SetWeight(1);
    IVADCResult pIVADCResult2 = pIVAProfileDifferentialCorrec-
tor11.getResults().Item(1);
    pIVADCResult2.SetEnable(true);
    pIVADCResult2.SetDesiredValue(0);
    pIVADCResult2.SetScalingValue(1);
    pIVADCResult2.SetTolerance(0.1);
    pIVADCResult2.SetWeight(1);

    IVASToppingConditionElement pIVASToppingConditionElement2 =
pIVAMCSPropagate2.GetStoppingConditions().Add("Duration");
    IVASToppingCondition pIVASToppingCondition2 = exam-
ple.IVASToppingConditionComponent2IVASToppingCondition(pIVASToppingCondi-
tionElement2.GetProperties());
    pIVASToppingCondition2.SetTrip(86400);
    pIVASToppingCondition2.SetTolerance(0.0001);

    pIVADriverMCS.RunMCS();
    pIVADriverMCS.ApplyAllProfileChanges();

    //javac -encoding utf-8 testCallDll.java
    String strStyle = new String("地固系位置速度");
    String strStartTime = pIScenario.GetStartTime();
    String strStopTime = pIScenario.GetStopTime();
    String strReportFilePath = pIAtkObjectRoot.OutputDataReport(pIAtkOb-
jectSatellite, strStyle, strStartTime, strStopTime);
    System.out.println("ATK " + strReportFilePath);
}
}

```

3.2.7. CMD 编译执行测试代码

编译测试代码：javac -encoding utf-8 testCallDll.java（会将.java 文件生成对应的.class 文件）

执行测试代码：java testCallDll

3.2.8. 查看生成文件判断是否正确

快速转移想定文件：FastTransfer.xml

地固系位置速度数据报告：

C:\Users\ADMINI~1\AppData\Local\Temp\astBC19.tmp

3.2.9. 打开生成的想定文件查看效果

Step1: 打开 ATK, 点击打开, 选择生成的快速转移想定文件 FastTransfer.xml, 再点击打开

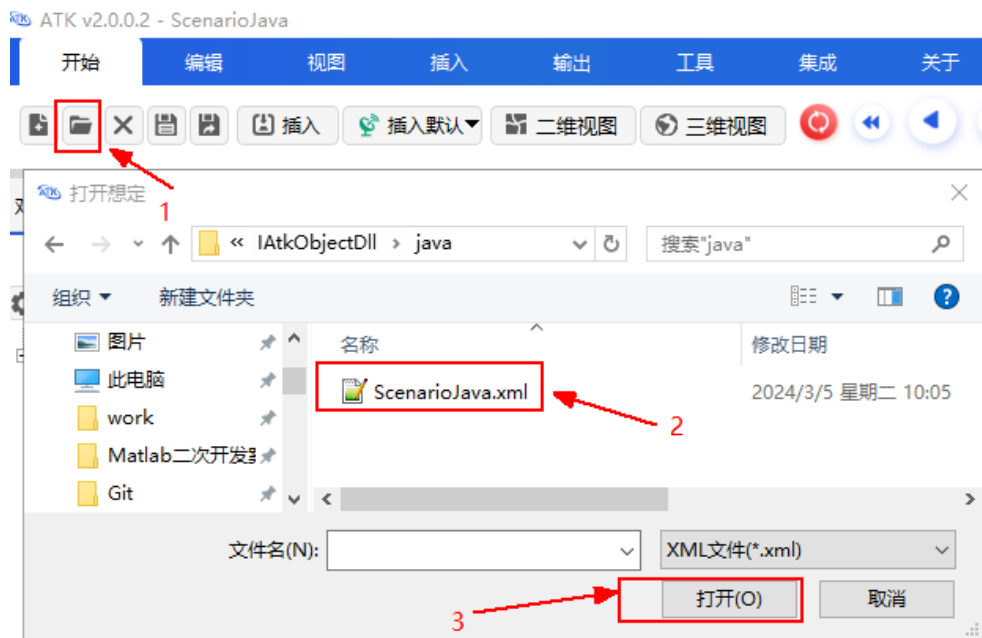


图 3-33 打开生成的快速转移想定文件

Step2: 点击运行, 打开 3 维视图, 查看运行效果

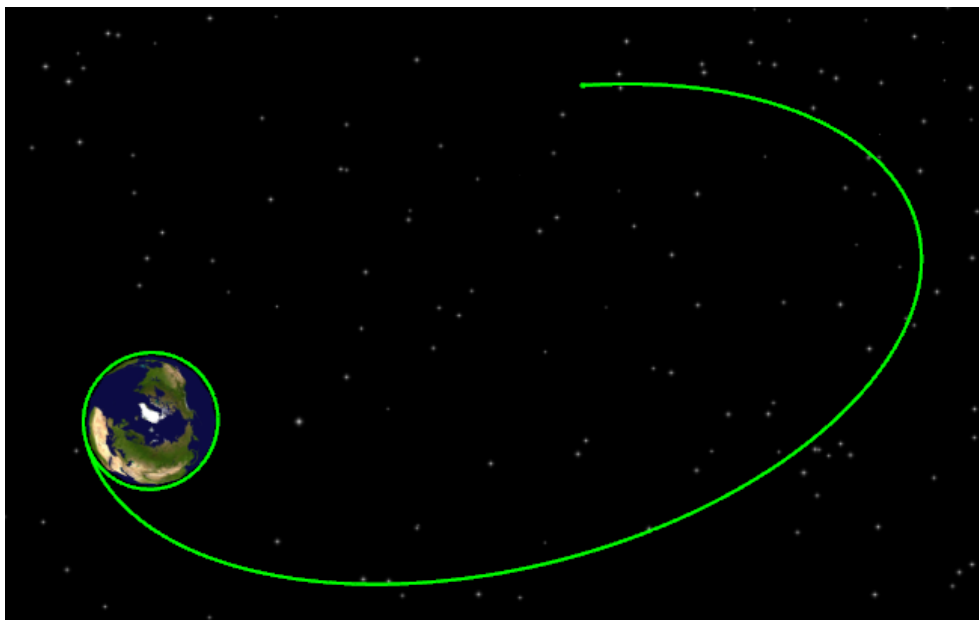


图 3-34 打开三维视图查看仿真运行轨迹

3.3. Python 语言操作流程

3.3.1. swigwin-4.1.1 安装

官网地址: <http://www.swig.org/tutorial.html>

下载地址: <https://sourceforge.net/projects/swig/files/swigwin/> (下载后解压即可)

3.3.2. Python 安装与环境配置

下载并安装 Python (Python-3.11.0.exe 为例)

3.3.2.1. 配置 Python 路径到系统环境变量

安装完 Python 后, 需要将 Python 路径配置到系统环境变量中。

PYTHONHOME 设置路径:

C:\Users\Administrator\AppData\Local\Programs\Python\Python311

Path 添加路径:

C:\Users\Administrator\AppData\Local\Programs\Python\Python311

Path 添加路径:

C:\Users\Administrator\AppData\Local\Programs\Python\Python311\Scripts

3.3.2.1.1. 系统环境变量配置过程 (以 PYTHONHOME 为例)

Step1: 右键点击此电脑, 再点击弹出框属性。



图 3-35 打开电脑属性

Step2: 点击控制面板框下系统高级设置, 再点击系统属性框高级选项下的环境变量。

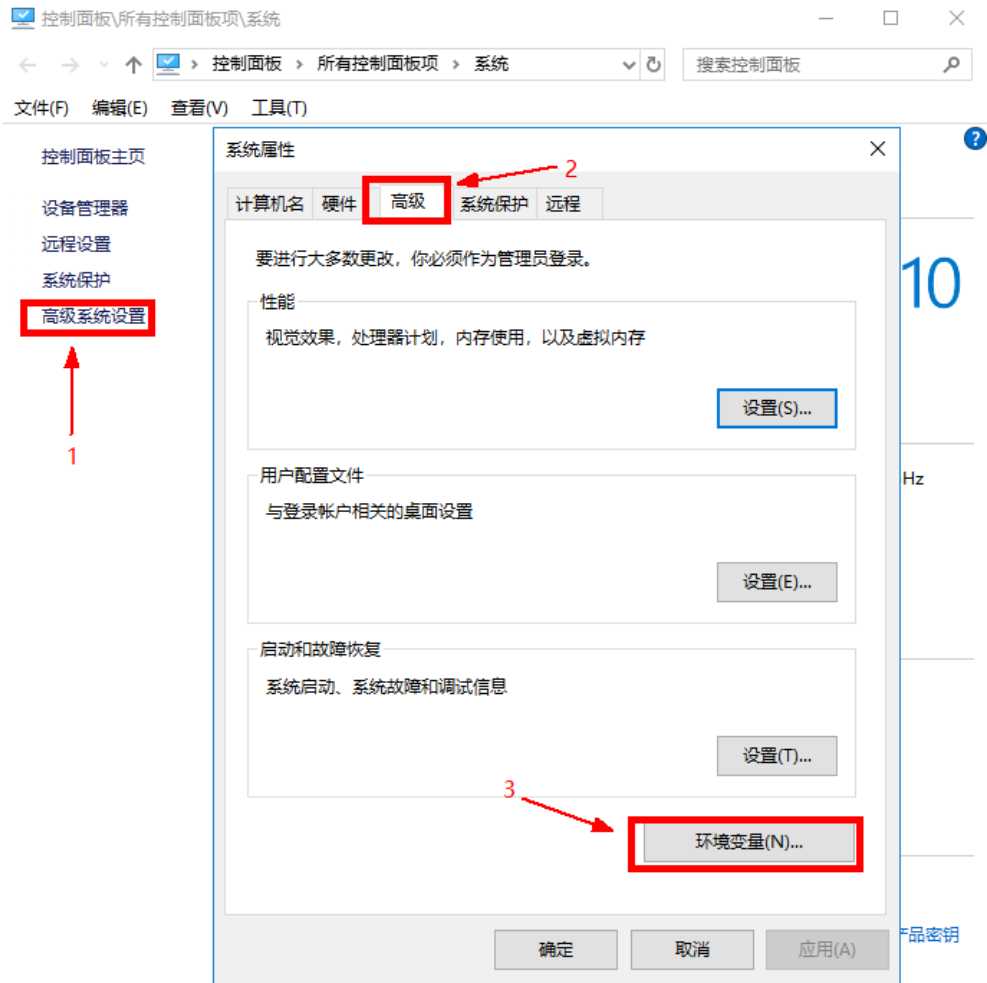


图 3-36 打开系统环境变量

Step3: 在弹出的环境变量框系统变量中新建（已存在则点击编辑）并输入变量名 PYTHONHOME 与 Python 路径，最后点击确定。

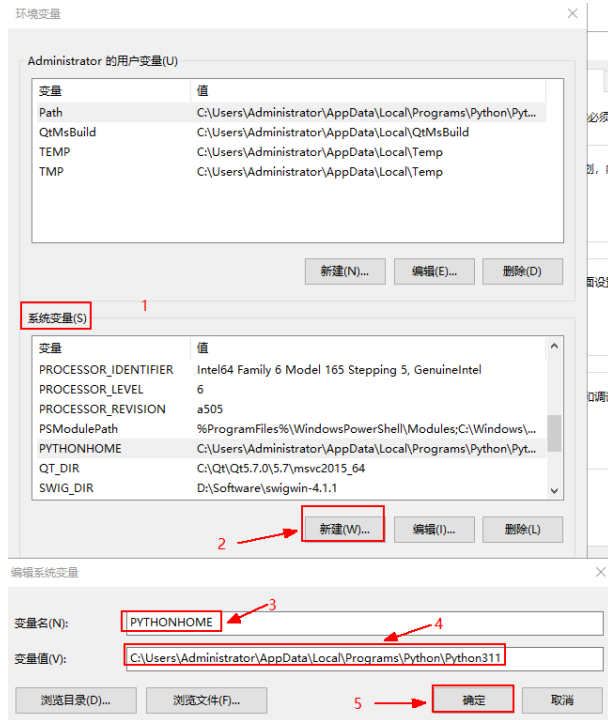


图 3-37 新建系统环境变量

Step4: 新建完变量 PYTHONHOME 后，再依此点击环境变量框最下方确定，以及系统属性框最下方的确定，至此，新建环境变量 PYTHONHOME 保存成功。

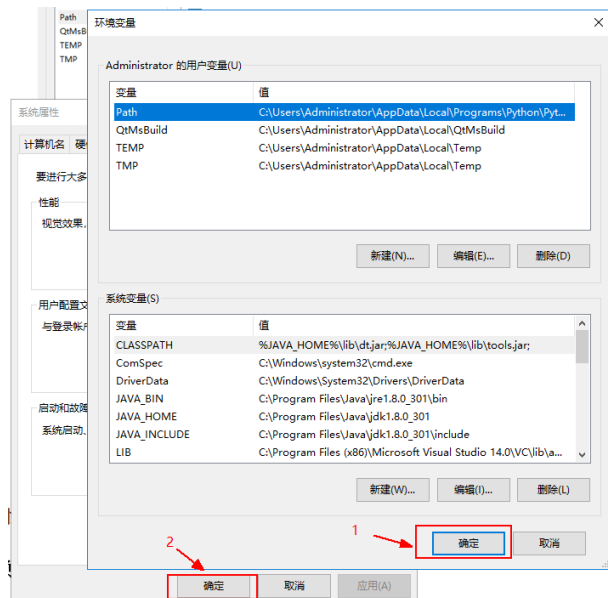


图 3-38 保存系统环境变量

3.3.2.2. 配置项目依赖默认数据库

将保存项目对象默认数值的文件夹 AstroData，整体复制到 Python 安装目录：
C:\Users\Administrator\AppData\Local\Programs\Python\Python311

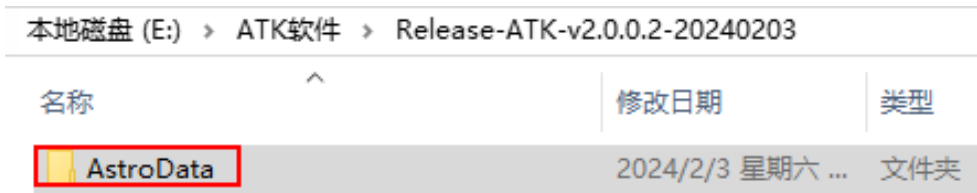


图 3-39 ATK 对象默认值数据库

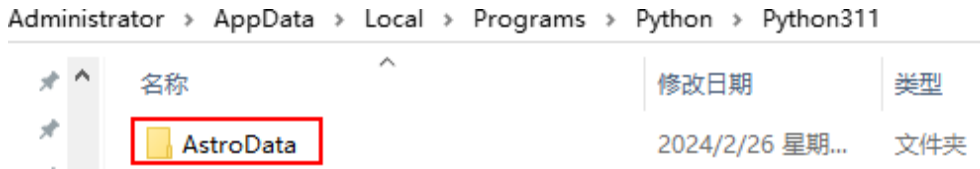


图 3-40 ATK 对象默认值数据库放置位置

3.3.2.3. 配置项目依赖动态库

将项目依赖动态库 ATKMinSolver.dll，复制到 Python 代码运行目录：

D:\Git\IAtkObject\IAtkObjectDll\Python

注意，通过 cmd 执行 Python 测试代码时，依赖动态库的调试版本和系统版本，必须和编译环境一致。

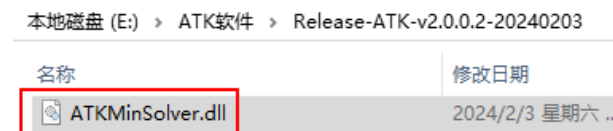


图 3-41 依赖的 ATK 动态库

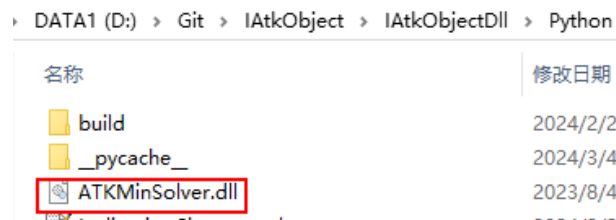


图 3-42 依赖的 ATK 动态库放置位置

3.3.3. 编写 SWIG 接口文件

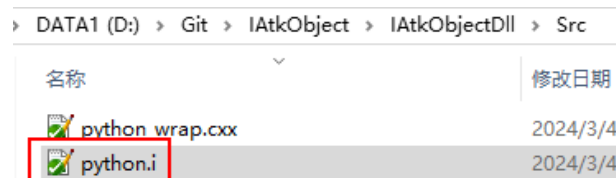


图 3-43 SWIG C++转 Python 接口封装与生成文件

```
//Python.i
```

```
//Python 模块名
```

```

%module python
%{
//包含需要调用的 C++声明文件，以本接口文件位置为基础
#include "../IAtkObj.h"
#include "../Include/AsObjManage.h"
#include "Common.h"
#include "IAnimation.h"
#include "IAtkObject.h"
#include "IAtkObjectCollection.h"
#include "IAtkObjectRoot.h"
#include "ICEnum.h"
#include "IScenario.h"
#include "IGreatArcVehicle.h"
//#include "ICObjectFunction.h"

#include "AccessConstraints/IAccessConstraint.h"
#include "AccessConstraints/IAccessCnstrAngle.h"
#include "AccessConstraints/IAccessCnstrMinMax.h"
#include "AccessConstraints/IAccessConstraintCollection.h"

#include "Aircraft/IAircraft.h"
    #include "Satellite/Propagator/IVePropagator.h"
#include "Aircraft/IVePropagatorGreatArc.h"
#include "Aircraft/IVeWaypointsCollection.h"
#include "Aircraft/IVeWaypointsElement.h"

#include "Chain/IChain.h"
#include "Chain/IObjectLinkCollection.h"
#include "Chain/IObjectLink.h"

#include "CoverageDefinition/ICoverageDefinition.h"
#include "CoverageDefinition/ICvGrid.h"
#include "CoverageDefinition/ICvBounds.h"
#include "CoverageDefinition/ICvBoundsGlobal.h"
#include "CoverageDefinition/ICvBoundsLat.h"

```

```

#include "CoverageDefinition/ICvBoundsLatLine.h"
#include "CoverageDefinition/ICvBoundsLatLonRegion.h"
#include "CoverageDefinition/ICvBoundsLonLine.h"
#include "CoverageDefinition/ICvResolution.h"
#include "CoverageDefinition/ICvResolutionArea.h"
#include "CoverageDefinition/ICvResolutionDistance.h"
#include "CoverageDefinition/ICvResolutionLatLon.h"

#include "Facility/IFacility.h"
    #include "Sensor/ILocationData.h"
#include "Facility/IPosition.h"
// #include "Facility/ICartesian.h"
// #include "Facility/IGeodetic.h"

#include "Graphics/ISnGraphics.h"
#include "Graphics/ISnProjection.h"
#include "Graphics/IDisplayDistance.h"
#include "Graphics/ISnProjConstantAlt.h"
#include "Graphics/ISnProjObjectAlt.h"

#include "GroundVehicle/IGroundVehicle.h"

#include "Planet/IPlanet.h"
#include "Planet/IPCommonTasks.h"
#include "Planet/IPositionSourceData.h"
#include "Planet/IPIPosCentralBody.h"

#include "Receiver/IReceiver.h"
#include "Receiver/IReceiverModel.h"
#include "Receiver/IReceiverModelSimple.h"
#include "Receiver/IPolarization.h"
#include "Receiver/IPolarizationNone.h"
#include "Receiver/IPolarizationLinear.h"
#include "Receiver/IPolarizationCircular.h"
#include "Receiver/IPolarizationVertical.h"

```



```
#include "Receiver/IPolarizationHorizontal.h"
#include "Receiver/IPolarizationElliptical.h"
#include "Receiver/ILinkMargin.h"
#include "Receiver/IDemodulatorModel.h"
///
```

```

#include "Satellite/Astrogator/IComponentInfo.h"
#include "Satellite/Astrogator/IVAMCSSegment.h"
#include "Satellite/Astrogator/EndSegment/IVAMCSEnd.h"

#include "Satellite/Astrogator/HoldSegment/IVAMCSHold.h"

#include "Satellite/Astrogator/InitialStateSegment/IVAMCSInitialState.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAEElement.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAEElementCartesian.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAEElementKeplerian.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAFuelTank.h"
#include "Satellite/Astrogator/InitialStateSegment/IVASpacecraftParameters.h"

#include "Satellite/Astrogator/ManeuverSegment/IVAMCSManeuver.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuver.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuverFinite.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuverImpulsive.h"
#include "Satellite/Astrogator/ManeuverSegment/IDirection.h"
#include "Satellite/Astrogator/ManeuverSegment/IDirectionRADec.h"
#include "Satellite/Astrogator/ManeuverSegment/IDirectionXYZ.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControl.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlFinite.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlFiniteThrustVector.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlImpulsive.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlImpulsiveThrustVector.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuverFinitePropagator.h"

#include "Satellite/Astrogator/PropagateSegment/IVAMCSPropagate.h"
#include "Satellite/Astrogator/PropagateSegment/IVASToppingConditionComponent.h"

```

```

#include "Satellite/Astrogator/PropagateSegment/IVASToppingCondition.h"
#include "Satellite/Astrogator/PropagateSegment/IVASToppingConditionCollec-
tion.h"
#include "Satellite/Astrogator/PropagateSegment/IVASToppingCondi-
tionElement.h"

#include "Satellite/Astrogator/Results/IVACalcObjectCollection.h"
#include "Satellite/Astrogator/Results/IVASTateCalcAltOfApoapsis.h"
#include "Satellite/Astrogator/Results/IVASTateCalcAltOfPeriapsis.h"
#include "Satellite/Astrogator/Results/IVASTateCalcArgOfLat.h"
#include "Satellite/Astrogator/Results/IVASTateCalcArgOfPeriapsis.h"
#include "Satellite/Astrogator/Results/IVASTateCalcBetaAngle.h"
#include "Satellite/Astrogator/Results/IVASTateCalcCartesianElem.h"
#include "Satellite/Astrogator/Results/IVASTateCalcCosOfVerticalFPA.h"
#include "Satellite/Astrogator/Results/IVASTateCalcDec.h"
#include "Satellite/Astrogator/Results/IVASTateCalcDeltaDec.h"
#include "Satellite/Astrogator/Results/IVASTateCalcDeltaFromMaster.h"
#include "Satellite/Astrogator/Results/IVASTateCalcDeltaRA.h"
#include "Satellite/Astrogator/Results/IVASTateCalcDeltaV.h"
#include "Satellite/Astrogator/Results/IVASTateCalcEccAnomaly.h"
#include "Satellite/Astrogator/Results/IVASTateCalcEccentricity.h"
#include "Satellite/Astrogator/Results/IVASTateCalcEquinoctialElem.h"
#include "Satellite/Astrogator/Results/IVASTateCalcFPA.h"
#include "Satellite/Astrogator/Results/IVASTateCalcGeodeticElem.h"
#include "Satellite/Astrogator/Results/IVASTateCalcInclination.h"
#include "Satellite/Astrogator/Results/IVASTateCalcLonDriftRate.h"
#include "Satellite/Astrogator/Results/IVASTateCalcMeanAnomaly.h"
#include "Satellite/Astrogator/Results/IVASTateCalcMeanEarthLon.h"
#include "Satellite/Astrogator/Results/IVASTateCalcOrbitPeriod.h"
#include "Satellite/Astrogator/Results/IVASTateCalcPosDifferenceOtherSeg-
ment.h"
#include "Satellite/Astrogator/Results/IVASTateCalcPosVelDifferenceOtherSeg-
ment.h"
#include "Satellite/Astrogator/Results/IVASTateCalcRA.h"
#include "Satellite/Astrogator/Results/IVASTateCalcRAAN.h"

```

```

#include "Satellite/Astrogator/Results/IVASStateCalcRadOfApoapsis.h"
#include "Satellite/Astrogator/Results/IVASStateCalcRadOfPeriapsis.h"
#include "Satellite/Astrogator/Results/IVASStateCalcRMag.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSegEpochDifference.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSegStateCalcDifference.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSemiMajorAxis.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSphRMag.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSphVMag.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSunlight.h"
#include "Satellite/Astrogator/Results/IVASStateCalcTrueAnomaly.h"
#include "Satellite/Astrogator/Results/IVASStateCalcTrueLon.h"
#include "Satellite/Astrogator/Results/IVASStateCalcVelAz.h"
#include "Satellite/Astrogator/Results/IVASStateCalcVelDifferenceOtherSegment.h"
#include "Satellite/Astrogator/Results/IVASStateCalcVMag.h"

#include "Satellite/Astrogator/ReturnSegment/IVAMCSReturn.h"

#include "Satellite/Astrogator/SequenceSegment/IVAMCSSequence.h"
#include "Satellite/Astrogator/SequenceSegment/IVAMCSBackwardSequence.h"

#include "Satellite/Astrogator/StopSegment/IVAMCSStop.h"

#include "Satellite/Astrogator/TargetSequenceSegment/IVAMCSTargetSequence.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCControl.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCControlCollection.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCResult.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCResultCollection.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfile.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfileCollection.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfileDifferentialCorrector1.h"

```

```
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfileDifferentialCorrector2.h"
```

```
#include "Satellite/Astrogator/UpdateSegment/IVAMCSUpdate.h"
```

```
#include "Satellite/Astrogator/IVAMCSSegmentCollection.h"
```

```
#include "Satellite/Astrogator/IComponentInfo.h"
```

```
#include "Satellite/Astrogator/IVAMCSSegment.h"
```

```
#include "Satellite/Astrogator/IVAMCSSegmentProperties.h"
```

```
#include "Satellite/Astrogator/IVASState.h"
```

```
#include "Satellite/Attitude/IVeAttitude.h"
```

```
#include "Satellite/Attitude/IVeAttitudeStandard.h"
```

```
#include "Satellite/Attitude/IVeAttProfile.h"
```

```
#include "Satellite/Attitude/IVeOrbitAttitudeStandard.h"
```

```
#include "Satellite/Attitude/IVeStandardBasic.h"
```

```
#include "Satellite/HPOP/ISRPMModelBase.h"
```

```
#include "Satellite/HPOP/ISRPMModelSpherical.h"
```

```
#include "Satellite/HPOP/IVeHPOPCentralBodyGravity.h"
```

```
#include "Satellite/HPOP/IVeHPOPDragModel.h"
```

```
#include "Satellite/HPOP/IVeHPOPDragModelSpherical.h"
```

```
#include "Satellite/HPOP/IVeHPOPForceModel.h"
```

```
#include "Satellite/HPOP/IVeHPOPForceModelDrag.h"
```

```
#include "Satellite/HPOP/IVeHPOPSolarRadiationPressure.h"
```

```
#include "Satellite/HPOP/IVeHPOPSRPMModel.h"
```

```
#include "Satellite/HPOP/IVeInitialState.h"
```

```
#include "Satellite/HPOP/IVeIntegrator.h"
```

```
#include "Satellite/HPOP/IVeSolarFluxGeoMag.h"
```

```
#include "Satellite/HPOP/IVeSolarFluxGeoMagEnterManually.h"
```

```
#include "Satellite/HPOP/IVeStepSizeControl.h"
```

```
#include "Satellite/HPOP/IVeThirdBodyGravityCollection.h"
```

```
#include "Satellite/HPOP/IVeThirdBodyGravityElement.h"
```

```
#include "Satellite/Mass/IVeInertia.h"
```

```

#include "Satellite/Mass/IVeMassProperties.h"

#include "Satellite/OrbitState/IOrbitState.h"
#include "Satellite/OrbitState/IClassicalLocation.h"
#include "Satellite/OrbitState/IClassicalLocationArgumentOfLatitude.h"
#include "Satellite/OrbitState/IClassicalLocationEccentricAnomaly.h"
#include "Satellite/OrbitState/IClassicalLocationMeanAnomaly.h"
#include "Satellite/OrbitState/IClassicalLocationTimePastAN.h"
#include "Satellite/OrbitState/IClassicalLocationTimePastPerigee.h"
#include "Satellite/OrbitState/IClassicalLocationTrueAnomaly.h"
#include "Satellite/OrbitState/IClassicalOrientation.h"
#include "Satellite/OrbitState/IClassicalSizeShape.h"
#include "Satellite/OrbitState/IClassicalSizeShapeAltitude.h"
#include "Satellite/OrbitState/IClassicalSizeShapeMeanMotion.h"
#include "Satellite/OrbitState/IClassicalSizeShapePeriod.h"
#include "Satellite/OrbitState/IClassicalSizeShapeRadius.h"
#include "Satellite/OrbitState/IClassicalSizeShapeSemimajorAxis.h"
#include "Satellite/OrbitState/IFlightPathAngle.h"
#include "Satellite/OrbitState/IOrbitStateCartesian.h"
#include "Satellite/OrbitState/IOrbitStateClassical.h"
#include "Satellite/OrbitState/IOrbitStateSpherical.h"
#include "Satellite/OrbitState/IOrientationAscNode.h"
#include "Satellite/OrbitState/IOrientationAscNodeLAN.h"
#include "Satellite/OrbitState/IOrientationAscNodeRAAN.h"
#include "Satellite/OrbitState/ISphericalFPAHorizontal.h"

#include "Satellite/PassBreak/IVePassBreak.h"
#include "Satellite/PassBreak/IVeBreakAngle.h"
#include "Satellite/PassBreak/IVeBreakAngleBreakByLatOrLon.h"
#include "Satellite/PassBreak/IVeDefinition.h"

// #include "Satellite/Propagator/IVePropagator.h"
#include "Satellite/Propagator/IVADriverMCS.h"
#include "Satellite/Propagator/IVePropagatorHPOP.h"
#include "Satellite/Propagator/IVePropagatorJ2Perturbation.h"

```

```

#include "Satellite/Propagator/IVePropagatorSGP4.h"
#include "Satellite/Propagator/IVePropagatorTwoBody.h"

#include "Satellite/SGP4/IVePropagatorSGP4CommonTasks.h"
#include "Satellite/SGP4/IVeSGP4Segment.h"
#include "Satellite/SGP4/IVeSGP4SegmentCollection.h"
%}

//要忽略的文件/类/函数/变量
%ignore CGlobalMemory::Allocate;

//swig 系统自带的常用封装类型接口
#include "std_string.i"
#include "cpointer.i"
//%pointer_functions(double, double_p);
#include "std_vector.i"
%template(vector_string)std::vector<std::string>;
%template(vector_double)std::vector<double>;
%template(vector_int)std::vector<int>;
//%template(vector_EVePropagatorType)std::vector<EVePropagatorType>;
%template(vector_IVeWaypointsElement)std::vector<IVeWaypointsElement*>;
//%template(vector_IVAStoppingConditionElement)std::vec-
tor<IVAStoppingConditionElement*>;
//%template(vector_IComponentInfoP)std::vector<IComponentInfo*>;
//%template(vector_IVADCResultP)std::vector<IVADCResult*>;
//%template(vector_IVAProfileP)std::vector<IVAProfile*>;
//%template(vector_IVAMCSSegmentP)std::vector<IVAMCSSegment*>;
//%template(vector_IVeThirdBodyGravityElementP)std::vector<IVeThirdBody-
GravityElement*>;
//%template(vector_IOrbitStateP)std::vector<IOrbitState*>;
//%template(vector_IVeSGP4SegmentP)std::vector<IVeSGP4Segment*>;
//%template(vector_IAtkObjectP)std::vector<IAtkObject*>;
//%template(vector_IVeWaypointsElementP)std::vector<IVeWaypointsEle-
ment*>;

```

//包含需要调用的 C++声明文件，以本接口文件位置为基础

```
%include "../IAtkObj.h"
```

```
%include "Common.h"
```

```
%include "IAnimation.h"
```

```
%include "IAtkObject.h"
```

```
%include "IAtkObjectCollection.h"
```

```
%include "IAtkObjectRoot.h"
```

```
%include "ICEnum.h"
```

```
%include "IScenario.h"
```

```
%include "IGreatArcVehicle.h"
```

```
//%include "IObjectFunction.h"
```

```
%include "AccessConstraints/IAccessConstraint.h"
```

```
%include "AccessConstraints/IAccessCnstrAngle.h"
```

```
%include "AccessConstraints/IAccessCnstrMinMax.h"
```

```
%include "AccessConstraints/IAccessConstraintCollection.h"
```

```
%include "Aircraft/IAircraft.h"
```

```
    %include "Satellite/Propagator/IVePropagator.h"
```

```
%include "Aircraft/IVePropagatorGreatArc.h"
```

```
%include "Aircraft/IVeWaypointsCollection.h"
```

```
%include "Aircraft/IVeWaypointsElement.h"
```

```
%include "Chain/IChain.h"
```

```
%include "Chain/IObjectLinkCollection.h"
```

```
%include "Chain/IObjectLink.h"
```

```
%include "CoverageDefinition/ICoverageDefinition.h"
```

```
%include "CoverageDefinition/ICvGrid.h"
```

```
%include "CoverageDefinition/ICvBounds.h"
```

```
%include "CoverageDefinition/ICvBoundsGlobal.h"
```

```
%include "CoverageDefinition/ICvBoundsLat.h"
```

```
%include "CoverageDefinition/ICvBoundsLatLine.h"
```

```
%include "CoverageDefinition/ICvBoundsLatLonRegion.h"
```



```

#include "CoverageDefinition/ICvBoundsLonLine.h"
#include "CoverageDefinition/ICvResolution.h"
#include "CoverageDefinition/ICvResolutionArea.h"
#include "CoverageDefinition/ICvResolutionDistance.h"
#include "CoverageDefinition/ICvResolutionLatLon.h"

#include "Facility/IFacility.h"
    #include "Sensor/ILocationData.h"
#include "Facility/IPosition.h"
//%include "Facility/ICartesian.h"
//%include "Facility/IGeodetic.h"

#include "Graphics/ISnGraphics.h"
#include "Graphics/ISnProjection.h"
#include "Graphics/IDisplayDistance.h"
#include "Graphics/ISnProjConstantAlt.h"
#include "Graphics/ISnProjObjectAlt.h"

#include "GroundVehicle/IGroundVehicle.h"

#include "Planet/IPlanet.h"
#include "Planet/IPCommonTasks.h"
#include "Planet/IPositionSourceData.h"
#include "Planet/IPIPosCentralBody.h"

#include "Receiver/IReceiver.h"
#include "Receiver/IReceiverModel.h"
#include "Receiver/IReceiverModelSimple.h"
#include "Receiver/IPolarization.h"
#include "Receiver/IPolarizationNone.h"
#include "Receiver/IPolarizationLinear.h"
#include "Receiver/IPolarizationCircular.h"
#include "Receiver/IPolarizationVertical.h"
#include "Receiver/IPolarizationHorizontal.h"
#include "Receiver/IPolarizationElliptical.h"

```

```
%include "Receiver/ILinkMargin.h"
%include "Receiver/IDemodulatorModel.h"
//%include "Receiver/IDemodulatorModelBpsk.h"
//%include "Receiver/IDemodulatorModelQpsk.h"
%include "Receiver/IReceiverModelMedium.h"

%include "Sensor/ISensor.h"
%include "Sensor/ISnCommonTasks.h"
%include "Sensor/ISnPattern.h"
%include "Sensor/ISnRectangularPattern.h"
%include "Sensor/ISnSimpleConicPattern.h"
%include "Sensor/ILocationData.h"
%include "Sensor/ISnPointing.h"
%include "Sensor/ISnPtFixed.h"
%include "Sensor/IOrientation.h"
%include "Sensor/IOrientationEulerAngles.h"
%include "Sensor/IOrientationQuaternion.h"

%include "Ship/IShip.h"

%include "Star/IStar.h"

%include "Submarine/ISubmarine.h"

%include "Transmitter/ITransmitter.h"
%include "Transmitter/ITransmitterModel.h"
%include "Transmitter/ITransmitterModelSimple.h"
%include "Transmitter/ITransmitterModelMedium.h"
%include "Transmitter/IModulatorModel.h"
%include "Transmitter/IModulatorModelBpsk.h"
%include "Transmitter/IModulatorModelQpsk.h"

%include "Satellite/ISatellite.h"

%include "Satellite/Astrogator/IComponentInfo.h"
```

```

#include "Satellite/Astrogator/IVAMCSSegment.h"
#include "Satellite/Astrogator/EndSegment/IVAMCSEnd.h"

#include "Satellite/Astrogator/HoldSegment/IVAMCSHold.h"

#include "Satellite/Astrogator/InitialStateSegment/IVAMCSInitialState.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAEElement.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAEElementCartesian.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAEElementKeplerian.h"
#include "Satellite/Astrogator/InitialStateSegment/IVAFuelTank.h"
#include "Satellite/Astrogator/InitialStateSegment/IVASpacecraftParameters.h"

#include "Satellite/Astrogator/ManeuverSegment/IVAMCSManeuver.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuver.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuverFinite.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuverImpulsive.h"
#include "Satellite/Astrogator/ManeuverSegment/IDirection.h"
#include "Satellite/Astrogator/ManeuverSegment/IDirectionRADec.h"
#include "Satellite/Astrogator/ManeuverSegment/IDirectionXYZ.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControl.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlFinite.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlFiniteThrustVector.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlImpulsive.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlImpulsiveThrustVector.h"
#include "Satellite/Astrogator/ManeuverSegment/IVAManeuverFinitePropagator.h"

#include "Satellite/Astrogator/PropagateSegment/IVAMCSPropagate.h"
#include "Satellite/Astrogator/PropagateSegment/IVASToppingConditionComponent.h"
#include "Satellite/Astrogator/PropagateSegment/IVASToppingCondition.h"
#include

```

```

"Satellite/Astrogator/PropagateSegment/IVAStoppingConditionCollection.h"
    %include          "Satellite/Astrogator/PropagateSegment/IVAStoppingCondi-
tionElement.h"

    %include "Satellite/Astrogator/Results/IVACalcObjectCollection.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcAltOfApoapsis.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcAltOfPeriapsis.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcArgOfLat.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcArgOfPeriapsis.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcBetaAngle.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcCartesianElem.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcCosOfVerticalFPA.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcDec.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcDeltaDec.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcDeltaFromMaster.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcDeltaRA.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcDeltaV.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcEccAnomaly.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcEccentricity.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcEquinoctialElem.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcFPA.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcGeodeticElem.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcInclination.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcLonDriftRate.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcMeanAnomaly.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcMeanEarthLon.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcOrbitPeriod.h"
    %include      "Satellite/Astrogator/Results/IVASStateCalcPosDifferenceOtherSeg-
ment.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcPosVelDifferenceOtherSeg-
ment.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcRA.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcRAAN.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcRadOfApoapsis.h"
    %include "Satellite/Astrogator/Results/IVASStateCalcRadOfPeriapsis.h"

```

```

#include "Satellite/Astrogator/Results/IVASStateCalcRMag.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSegEpochDifference.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSegStateCalcDifference.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSemiMajorAxis.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSphRMag.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSphVMag.h"
#include "Satellite/Astrogator/Results/IVASStateCalcSunlight.h"
#include "Satellite/Astrogator/Results/IVASStateCalcTrueAnomaly.h"
#include "Satellite/Astrogator/Results/IVASStateCalcTrueLon.h"
#include "Satellite/Astrogator/Results/IVASStateCalcVelAz.h"
#include "Satellite/Astrogator/Results/IVASStateCalcVelDifferenceOtherSegment.h"
#include "Satellite/Astrogator/Results/IVASStateCalcVMag.h"

#include "Satellite/Astrogator/ReturnSegment/IVAMCSReturn.h"

#include "Satellite/Astrogator/SequenceSegment/IVAMCSSequence.h"
#include "Satellite/Astrogator/SequenceSegment/IVAMCSBackwardSequence.h"

#include "Satellite/Astrogator/StopSegment/IVAMCSStop.h"

#include "Satellite/Astrogator/TargetSequenceSegment/IVAMCSTargetSequence.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCControl.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCControlCollection.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCResult.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVADCCResultCollection.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfile.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfileCollection.h"
#include "Satellite/Astrogator/TargetSequenceSegment/IVAProfileDifferentialCorrector1.h"
#include

```

"Satellite/Astrogator/TargetSequenceSegment/IVAProfileDifferentialCorrector2.h"

%include "Satellite/Astrogator/UpdateSegement/IVAMCSUpdate.h"

%include "Satellite/Astrogator/IVAMCSSegmentCollection.h"

%include "Satellite/Astrogator/IComponentInfo.h"

%include "Satellite/Astrogator/IVAMCSSegment.h"

%include "Satellite/Astrogator/IVAMCSSegmentProperties.h"

%include "Satellite/Astrogator/IVASState.h"

%include "Satellite/Attitude/IVeAttitude.h"

%include "Satellite/Attitude/IVeAttitudeStandard.h"

%include "Satellite/Attitude/IVeAttProfile.h"

%include "Satellite/Attitude/IVeOrbitAttitudeStandard.h"

%include "Satellite/Attitude/IVeStandardBasic.h"

%include "Satellite/HPOP/ISRPMModelBase.h"

%include "Satellite/HPOP/ISRPMModelSpherical.h"

%include "Satellite/HPOP/IVeHPOPcentralBodyGravity.h"

%include "Satellite/HPOP/IVeHPOPDragModel.h"

%include "Satellite/HPOP/IVeHPOPDragModelSpherical.h"

%include "Satellite/HPOP/IVeHPOPForceModel.h"

%include "Satellite/HPOP/IVeHPOPForceModelDrag.h"

%include "Satellite/HPOP/IVeHPOPSolarRadiationPressure.h"

%include "Satellite/HPOP/IVeHPOPSRPMModel.h"

%include "Satellite/HPOP/IVeInitialState.h"

%include "Satellite/HPOP/IVeIntegrator.h"

%include "Satellite/HPOP/IVeSolarFluxGeoMag.h"

%include "Satellite/HPOP/IVeSolarFluxGeoMagEnterManually.h"

%include "Satellite/HPOP/IVeStepSizeControl.h"

%include "Satellite/HPOP/IVeThirdBodyGravityCollection.h"

%include "Satellite/HPOP/IVeThirdBodyGravityElement.h"

%include "Satellite/Mass/IVeInertia.h"

%include "Satellite/Mass/IVeMassProperties.h"

```

#include "Satellite/OrbitState/IOrbitState.h"
#include "Satellite/OrbitState/IClassicalLocation.h"
#include "Satellite/OrbitState/IClassicalLocationArgumentOfLatitude.h"
#include "Satellite/OrbitState/IClassicalLocationEccentricAnomaly.h"
#include "Satellite/OrbitState/IClassicalLocationMeanAnomaly.h"
#include "Satellite/OrbitState/IClassicalLocationTimePastAN.h"
#include "Satellite/OrbitState/IClassicalLocationTimePastPerigee.h"
#include "Satellite/OrbitState/IClassicalLocationTrueAnomaly.h"
#include "Satellite/OrbitState/IClassicalOrientation.h"
#include "Satellite/OrbitState/IClassicalSizeShape.h"
#include "Satellite/OrbitState/IClassicalSizeShapeAltitude.h"
#include "Satellite/OrbitState/IClassicalSizeShapeMeanMotion.h"
#include "Satellite/OrbitState/IClassicalSizeShapePeriod.h"
#include "Satellite/OrbitState/IClassicalSizeShapeRadius.h"
#include "Satellite/OrbitState/IClassicalSizeShapeSemimajorAxis.h"
#include "Satellite/OrbitState/IFlightPathAngle.h"
#include "Satellite/OrbitState/IOrbitStateCartesian.h"
#include "Satellite/OrbitState/IOrbitStateClassical.h"
#include "Satellite/OrbitState/IOrbitStateSpherical.h"
#include "Satellite/OrbitState/IOrientationAscNode.h"
#include "Satellite/OrbitState/IOrientationAscNodeLAN.h"
#include "Satellite/OrbitState/IOrientationAscNodeRAAN.h"
#include "Satellite/OrbitState/ISphericalFPAHorizontal.h"

#include "Satellite/PassBreak/IVePassBreak.h"
#include "Satellite/PassBreak/IVeBreakAngle.h"
#include "Satellite/PassBreak/IVeBreakAngleBreakByLatOrLon.h"
#include "Satellite/PassBreak/IVeDefinition.h"

//%include "Satellite/Propagator/IVePropagator.h"
#include "Satellite/Propagator/IVADriverMCS.h"
#include "Satellite/Propagator/IVePropagatorHPOP.h"
#include "Satellite/Propagator/IVePropagatorJ2Perturbation.h"
#include "Satellite/Propagator/IVePropagatorSGP4.h"

```

```
%include "Satellite/Propagator/IVePropagatorTwoBody.h"
```

```
%include "Satellite/SGP4/IVePropagatorSGP4CommonTasks.h"
```

```
%include "Satellite/SGP4/IVeSGP4Segment.h"
```

```
%include "Satellite/SGP4/IVeSGP4SegmentCollection.h"
```

3.3.4. 编译 SWIG 接口文件

3.3.4.1. cmd 编译 SWIG 接口文件

跳转命令：cd: D:\Software\swigwin-4.1.1

SWIG 命令：swig -c++ -python -v -outdir D:/Git/IAtkObject/IAtkObjectDll/Python D:/Git/IAtkObject/IAtkObjectDll/Src/python.i

SWIG 命令格式（空格分开）：SWIG -源语言 -目标语言 -v(输出编译信息) -outdir Python 接口文件生成目录 SWIG 接口文件全路径

3.3.4.2. 编译生成文件

3.3.4.2.1. Python 接口模块 python.py

Python 接口模块 python.py，可被 Python 代码直接导入调用，包含所有 C++ 声明文件中类与函数的对外接口封装，以及对内 Python 共享库 _python.cp311-win_amd64.pyd 的调用实现。

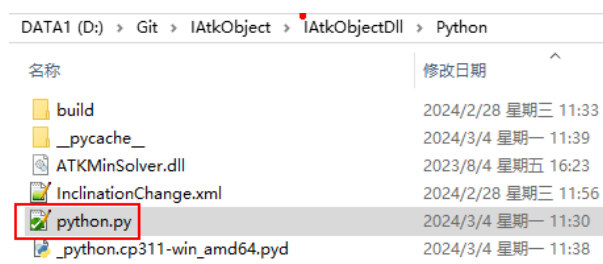


图 3-44 Python 对外接口封装文件

3.3.4.2.2. C++ 接口封装库 python_wrap.cxx

C++ 接口封装库 python_wrap.cxx，可直接调用 C++ 源码，用于和类定义文件.cpp 一起编译生成 Python 共享库 _python.cp311-win_amd64.pyd，共享库相当于 C++ 动态库。

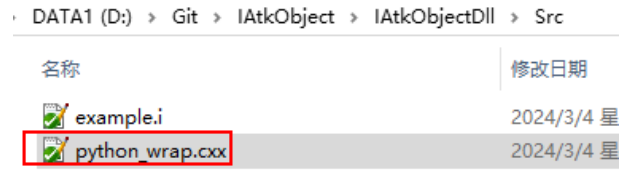


图 3-45 Python 对内接口封装实现文件

3.3.5. 编写 SWIG 编译文件

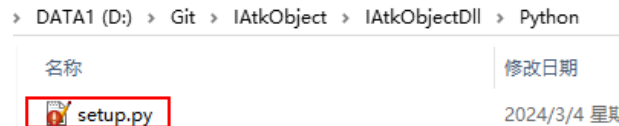


图 3-46 Python 编译文件

```
# setup.py

#!/usr/bin/env python

from distutils.core import setup, Extension

#定义 C++扩展模块
python_module = Extension(
    '_python',
    sources = [
        '../Src/Common.cpp',
        '../Src/IAnimation.cpp',
        '../Src/IAtkObject.cpp',
        '../Src/IAtkObjectCollection.cpp',
        '../Src/IAtkObjectRoot.cpp',
        '../Src/IScenario.cpp',
        '../Src/IGreatArcVehicle.cpp',
        '../Src/ICObjectFunction.cpp',

        '../Src/AccessConstraints/IAccessConstraint.cpp',
        '../Src/AccessConstraints/IAccessCnstrAngle.cpp',
        '../Src/AccessConstraints/IAccessCnstrMinMax.cpp',
        '../Src/AccessConstraints/IAccessConstraintCollection.cpp',
```

'../Src/Aircraft/IAircraft.cpp',
 '../Src/Satellite/Propagator/IVePropagator.cpp',
'../Src/Aircraft/IVePropagatorGreatArc.cpp',
'../Src/Aircraft/IVeWaypointsCollection.cpp',
'../Src/Aircraft/IVeWaypointsElement.cpp',

'../Src/Chain/IChain.cpp',
'../Src/Chain/IObjectLinkCollection.cpp',
'../Src/Chain/IObjectLink.cpp',

'../Src/CoverageDefinition/ICoverageDefinition.cpp',
'../Src/CoverageDefinition/ICvGrid.cpp',
'../Src/CoverageDefinition/ICvBounds.cpp',
'../Src/CoverageDefinition/ICvBoundsGlobal.cpp',
'../Src/CoverageDefinition/ICvBoundsLat.cpp',
'../Src/CoverageDefinition/ICvBoundsLatLine.cpp',
'../Src/CoverageDefinition/ICvBoundsLatLonRegion.cpp',
'../Src/CoverageDefinition/ICvBoundsLonLine.cpp',
'../Src/CoverageDefinition/ICvResolution.cpp',
'../Src/CoverageDefinition/ICvResolutionArea.cpp',
'../Src/CoverageDefinition/ICvResolutionDistance.cpp',
'../Src/CoverageDefinition/ICvResolutionLatLon.cpp',

'../Src/Facility/IFacility.cpp',
 '../Src/Sensor/ILocationData.cpp',
'../Src/Facility/IPosition.cpp',
'../Src/Facility/ICartesian.cpp',
'../Src/Facility/IGeodetic.cpp',

'../Src/Graphics/ISnGraphics.cpp',
'../Src/Graphics/ISnProjection.cpp',
'../Src/Graphics/IDisplayDistance.cpp',
'../Src/Graphics/ISnProjConstantAlt.cpp',
'../Src/Graphics/ISnProjObjectAlt.cpp',

'../Src/GroundVehicle/IGroundVehicle.cpp',

'../Src/Planet/IPlanet.cpp',
'../Src/Planet/IPCommonTasks.cpp',
'../Src/Planet/IPositionSourceData.cpp',
'../Src/Planet/IPIPosCentralBody.cpp',

'../Src/Receiver/IReceiver.cpp',
'../Src/Receiver/IReceiverModel.cpp',
'../Src/Receiver/IReceiverModelSimple.cpp',
'../Src/Receiver/IPolarization.cpp',
'../Src/Receiver/IPolarizationNone.cpp',
'../Src/Receiver/IPolarizationLinear.cpp',
'../Src/Receiver/IPolarizationCircular.cpp',
'../Src/Receiver/IPolarizationVertical.cpp',
'../Src/Receiver/IPolarizationHorizontal.cpp',
'../Src/Receiver/IPolarizationElliptical.cpp',
'../Src/Receiver/ILinkMargin.cpp',
'../Src/Receiver/IDemodulatorModel.cpp',
'../Src/Receiver/IDemodulatorModelBpsk.cpp',
'../Src/Receiver/IDemodulatorModelQpsk.cpp',
'../Src/Receiver/IReceiverModelMedium.cpp',

'../Src/Sensor/ISensor.cpp',
'../Src/Sensor/ISnCommonTasks.cpp',
'../Src/Sensor/ISnPattern.cpp',
'../Src/Sensor/ISnRectangularPattern.cpp',
'../Src/Sensor/ISnSimpleConicPattern.cpp',
'../Src/Sensor/ILocationData.cpp',
'../Src/Sensor/ISnPointing.cpp',
'../Src/Sensor/ISnPtFixed.cpp',
'../Src/Sensor/IOrientation.cpp',
'../Src/Sensor/IOrientationEulerAngles.cpp',
'../Src/Sensor/IOrientationQuaternion.cpp',

```

'../Src/Ship/IShip.cpp',

'../Src/Star/IStar.cpp',

'../Src/Submarine/ISubmarine.cpp',

'../Src/Transmitter/ITransmitter.cpp',
'../Src/Transmitter/ITransmitterModel.cpp',
'../Src/Transmitter/ITransmitterModelSimple.cpp',
'../Src/Transmitter/ITransmitterModelMedium.cpp',
'../Src/Transmitter/IModulatorModel.cpp',
'../Src/Transmitter/IModulatorModelBpsk.cpp',
'../Src/Transmitter/IModulatorModelQpsk.cpp',

'../Src/Satellite/ISatellite.cpp',

    '../Src/Satellite/Astrogator/IComponentInfo.cpp',
    '../Src/Satellite/Astrogator/IVAMCSSegment.cpp',
'../Src/Satellite/Astrogator/EndSegment/IVAMCSEnd.cpp',

'../Src/Satellite/Astrogator/HoldSegment/IVAMCSHold.cpp',
'../Src/Satellite/Astrogator/InitialStateSegment/IVAMCSInitialState.cpp',
'../Src/Satellite/Astrogator/InitialStateSegment/IVAElement.cpp',
'../Src/Satellite/Astrogator/InitialStateSegment/IVAElementCarte-
sian.cpp',
'../Src/Satellite/Astrogator/InitialStateSegment/IVAElementKep-
lerian.cpp',
'../Src/Satellite/Astrogator/InitialStateSegment/IVAFuelTank.cpp',
'../Src/Satellite/Astrogator/InitialStateSegment/IVASpacecraftParame-
ters.cpp',

'../Src/Satellite/Astrogator/ManeuverSegment/IVAMCSManeuver.cpp',
'../Src/Satellite/Astrogator/ManeuverSegment/IVAManeuver.cpp',
'../Src/Satellite/Astrogator/ManeuverSegment/IVAManeuverFinite.cpp',

```

'../Src/Satellite/Astrogator/ManeuverSegment/IVAManeuverImpulsive.cpp',
 '../Src/Satellite/Astrogator/ManeuverSegment/IDirection.cpp',
 '../Src/Satellite/Astrogator/ManeuverSegment/IDirectionRADec.cpp',
 '../Src/Satellite/Astrogator/ManeuverSegment/IDirectionXYZ.cpp',
 '../Src/Satellite/Astrogator/ManeuverSegment/IVAAttitudeControl.cpp',
 '../Src/Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlFi-
 nite.cpp',
 '../Src/Satellite/Astrogator/ManeuverSegment/IVAAttitudeCon-
 trolFiniteThrustVector.cpp',
 '../Src/Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlImpul-
 sive.cpp',
 '../Src/Satellite/Astrogator/ManeuverSegment/IVAAttitudeControlImpul-
 siveThrustVector.cpp',
 '../Src/Satellite/Astrogator/ManeuverSegment/IVAManeuverFinitePropa-
 gator.cpp',

 '../Src/Satellite/Astrogator/PropagateSegment/IVAMCSPropagate.cpp',
 '../Src/Satellite/Astrogator/PropagateSegment/IVAStoppingCondi-
 tionComponent.cpp',
 '../Src/Satellite/Astrogator/PropagateSegment/IVAStoppingCondi-
 tion.cpp',
 '../Src/Satellite/Astrogator/PropagateSegment/IVAStoppingCondition-
 Collection.cpp',
 '../Src/Satellite/Astrogator/PropagateSegment/IVAStoppingCondi-
 tionElement.cpp',

 '../Src/Satellite/Astrogator/Results/IVACalcObjectCollection.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcAltOfApoapsis.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcAltOfPeriapsis.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcArgOfLat.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcArgOfPeriapsis.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcBetaAngle.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcCartesianElem.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcCosOfVerticalFPA.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcDec.cpp',

'../Src/Satellite/Astrogator/Results/IVASStateCalcDeltaDec.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcDeltaFromMaster.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcDeltaRA.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcDeltaV.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcEccAnomaly.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcEccentricity.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcEquinoctialElem.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcFPA.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcGeodeticElem.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcInclination.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcLonDriftRate.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcMeanAnomaly.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcMeanEarthLon.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcOrbitPeriod.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcPosDifferenceOtherSeg-
 ment.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcPosVelDifference-
 OtherSegment.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcRA.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcRAAN.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcRadOfApoapsis.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcRadOfPeriapsis.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcRMag.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcSegEpochDiffer-
 ence.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcSegStateCalcDiffer-
 ence.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcSemiMajorAxis.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcSphRMag.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcSphVMag.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcSunlight.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcTrueAnomaly.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcTrueLon.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcVelAz.cpp',

'../Src/Satellite/Astrogator/Results/IVASStateCalcVelDifferenceOtherSegment.cpp',
 '../Src/Satellite/Astrogator/Results/IVASStateCalcVMag.cpp',
 '../Src/Satellite/Astrogator/ReturnSegment/IVAMCSReturn.cpp',
 '../Src/Satellite/Astrogator/SequenceSegment/IVAMCSSequence.cpp',
 '../Src/Satellite/Astrogator/SequenceSegment/IVAMCSBackwardSequence.cpp',
 '../Src/Satellite/Astrogator/StopSegment/IVAMCSStop.cpp',
 '../Src/Satellite/Astrogator/TargetSequenceSegment/IVAMCSTargetSequence.cpp',
 '../Src/Satellite/Astrogator/TargetSequenceSegment/IVADCCControl.cpp',
 '../Src/Satellite/Astrogator/TargetSequenceSegment/IVADCCControlCollection.cpp',
 '../Src/Satellite/Astrogator/TargetSequenceSegment/IVADDCResult.cpp',
 '../Src/Satellite/Astrogator/TargetSequenceSegment/IVADDCResultCollection.cpp',
 '../Src/Satellite/Astrogator/TargetSequenceSegment/IVAProfile.cpp',
 '../Src/Satellite/Astrogator/TargetSequenceSegment/IVAProfileCollection.cpp',
 '../Src/Satellite/Astrogator/TargetSequenceSegment/IVAProfileDifferentialCorrector1.cpp',
 '../Src/Satellite/Astrogator/TargetSequenceSegment/IVAProfileDifferentialCorrector2.cpp',
 '../Src/Satellite/Astrogator/UpdateSegment/IVAMCSUpdate.cpp',
 '../Src/Satellite/Astrogator/IVAMCSSegmentCollection.cpp',
 '../Src/Satellite/Astrogator/IComponentInfo.cpp',
 '../Src/Satellite/Astrogator/IVAMCSSegment.cpp',
 '../Src/Satellite/Astrogator/IVAMCSSegmentProperties.cpp',
 '../Src/Satellite/Astrogator/IVASState.cpp',

'../Src/Satellite/Attitude/IVeAttitude.cpp',
 '../Src/Satellite/Attitude/IVeAttitudeStandard.cpp',
 '../Src/Satellite/Attitude/IVeAttProfile.cpp',
 '../Src/Satellite/Attitude/IVeOrbitAttitudeStandard.cpp',
 '../Src/Satellite/Attitude/IVeStandardBasic.cpp',

 '../Src/Satellite/HPOP/ISRPMModelBase.cpp',
 '../Src/Satellite/HPOP/ISRPMModelSpherical.cpp',
 '../Src/Satellite/HPOP/IVeHPOPCentralBodyGravity.cpp',
 '../Src/Satellite/HPOP/IVeHPOPDragModel.cpp',
 '../Src/Satellite/HPOP/IVeHPOPDragModelSpherical.cpp',
 '../Src/Satellite/HPOP/IVeHPOPForceModel.cpp',
 '../Src/Satellite/HPOP/IVeHPOPForceModelDrag.cpp',
 '../Src/Satellite/HPOP/IVeHPOPSolarRadiationPressure.cpp',
 '../Src/Satellite/HPOP/IVeHPOPSRPMModel.cpp',
 '../Src/Satellite/HPOP/IVeInitialState.cpp',
 '../Src/Satellite/HPOP/IVeIntegrator.cpp',
 '../Src/Satellite/HPOP/IVeSolarFluxGeoMag.cpp',
 '../Src/Satellite/HPOP/IVeSolarFluxGeoMagEnterManually.cpp',
 '../Src/Satellite/HPOP/IVeStepSizeControl.cpp',
 '../Src/Satellite/HPOP/IVeThirdBodyGravityCollection.cpp',
 '../Src/Satellite/HPOP/IVeThirdBodyGravityElement.cpp',

 '../Src/Satellite/Mass/IVeInertia.cpp',
 '../Src/Satellite/Mass/IVeMassProperties.cpp',

 '../Src/Satellite/OrbitState/IOrbitState.cpp',
 '../Src/Satellite/OrbitState/IClassicalLocation.cpp',
 '../Src/Satellite/OrbitState/IClassicalLocationArgumentOfLatitude.cpp',
 '../Src/Satellite/OrbitState/IClassicalLocationEccentricAnomaly.cpp',
 '../Src/Satellite/OrbitState/IClassicalLocationMeanAnomaly.cpp',
 '../Src/Satellite/OrbitState/IClassicalLocationTimePastAN.cpp',
 '../Src/Satellite/OrbitState/IClassicalLocationTimePastPerigee.cpp',
 '../Src/Satellite/OrbitState/IClassicalLocationTrueAnomaly.cpp',


```

'../Src/Satellite/OrbitState/IClassicalOrientation.cpp',
'../Src/Satellite/OrbitState/IClassicalSizeShape.cpp',
'../Src/Satellite/OrbitState/IClassicalSizeShapeAltitude.cpp',
'../Src/Satellite/OrbitState/IClassicalSizeShapeMeanMotion.cpp',
'../Src/Satellite/OrbitState/IClassicalSizeShapePeriod.cpp',
'../Src/Satellite/OrbitState/IClassicalSizeShapeRadius.cpp',
'../Src/Satellite/OrbitState/IClassicalSizeShapeSemimajorAxis.cpp',
'../Src/Satellite/OrbitState/IFlightPathAngle.cpp',
'../Src/Satellite/OrbitState/IOrbitStateCartesian.cpp',
'../Src/Satellite/OrbitState/IOrbitStateClassical.cpp',
'../Src/Satellite/OrbitState/IOrbitStateSpherical.cpp',
'../Src/Satellite/OrbitState/IOrientationAscNode.cpp',
'../Src/Satellite/OrbitState/IOrientationAscNodeLAN.cpp',
'../Src/Satellite/OrbitState/IOrientationAscNodeRAAN.cpp',
'../Src/Satellite/OrbitState/ISphericalFPAHorizontal.cpp',

'../Src/Satellite/PassBreak/IVePassBreak.cpp',
'../Src/Satellite/PassBreak/IVeBreakAngle.cpp',
'../Src/Satellite/PassBreak/IVeBreakAngleBreakByLatOrLon.cpp',
'../Src/Satellite/PassBreak/IVeDefinition.cpp',

#
'../Src/Satellite/Propagator/IVePropagator.cpp',
'../Src/Satellite/Propagator/IVADriverMCS.cpp',
'../Src/Satellite/Propagator/IVePropagatorHPOP.cpp',
'../Src/Satellite/Propagator/IVePropagatorJ2Perturbation.cpp',
'../Src/Satellite/Propagator/IVePropagatorSGP4.cpp',
'../Src/Satellite/Propagator/IVePropagatorTwoBody.cpp',

'../Src/Satellite/SGP4/IVePropagatorSGP4CommonTasks.cpp',
'../Src/Satellite/SGP4/IVeSGP4Segment.cpp',
'../Src/Satellite/SGP4/IVeSGP4SegmentCollection.cpp',

'../Src/python_wrap.cxx'
],
include_dirs=[

```

'../',
'../Include',
'../Include/AstroObj',
'../Include/APlanning',
'../Include/APlanning/Include',
'../Include/APlanning/Include/Base',
'../Src',
'../Src/AccessConstraints',
'../Src/Aircraft',
'../Src/Chain',
'../Src/CoverageDefinition',
'../Src/Facility',
'../Src/Graphics',
'../Src/GroundVehicle',
'../Src/Planet',
'../Src/Receiver',
'../Src/Satellite',
'../Src/Satellite/Attitude',
'../Src/Satellite/HPOP',
'../Src/Satellite/Mass',
'../Src/Satellite/SGP4',
'../Src/Satellite/OrbitState',
'../Src/Satellite/PassBreak',
'../Src/Satellite/Propagator',
'../Src/Satellite/Astrogator',
'../Src/Satellite/Astrogator/EndSegment',
'../Src/Satellite/Astrogator/InitialStateSegment',
'../Src/Satellite/Astrogator/ManeuverSegment',
'../Src/Satellite/Astrogator/PropagateSegment',
'../Src/Satellite/Astrogator/Results',
'../Src/Satellite/Astrogator/TargetSequenceSegment',
'../Src/Satellite/Astrogator/SequenceSegment',
'../Src/Satellite/Astrogator/UpdateSegment',
'../Src/Satellite/Astrogator/StopSegment',
'../Src/Satellite/Astrogator/ReturnSegment',

```

        './Src/Satellite/Astrogator/HoldSegment',
        './Src/Sensor',
        './Src/Ship',
    './Src/Star',
    './Src/Submarine',
    './Src/Transmitter'
],
library_dirs=[
    'D://Git//IAtkObject//IAtkObjectDll/',
    'D://Git//IAtkObject//IAtkObjectDll//Lib/',
    'C://Program Files (x86)//Windows Kits//8.1//Lib//winv6.3//um//x64/'
],
libraries=[
    'AdvAPI32',
    # 'AstroLib',
    # 'AstroLibD',
    # 'AstroLib64',
    # 'AstroLibD64',
    # 'AstroObj',
    # 'AstroObjD',
    'AstroObj64',
    # 'AstroObjD64',
    # 'ATKMinSolver',
    # 'ATKMinSolverD',
    # 'APlanning64',
    # 'APlanningD64',
]
)

```

#配置 setup 函数

```

setup (
    name = 'python',
    version = '0.1',
    author = 'SWIG-Python',

```

```
description = 'SWIG : c++ conver to Python',
ext_modules = [python_module],
py_modules = ["python"])
```

3.3.6. 编译 SWIG 编译文件

3.3.6.1. cmd 编译 C++接口库和 C++源码

Python 编译命令：`python setup.py build_ext -inplace`

Python 编译命令格式（空格分开）：目标语言 SWIG 编译文件名
`build_ext(distutils 扩展) -inplace(扩展库存放当前目录)`

3.3.6.2. 编译生成文件

3.3.6.2.1. Python 共享库 `_python.cp311-win_amd64.pyd`

Python 共享库 `_python.cp311-win_amd64.pyd`，可被 Python 接口模块导入，支持所有接口调用，包含所含类与函数的实现源码。

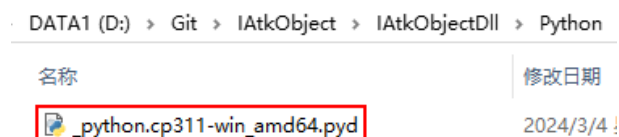


图 3-47 Python 共享库

3.3.7. Python 调用测试

3.3.7.1. Python 测试依赖文件

`python.py`: Python 接口模块，通过编译 SWIG 接口文件生成。

`_python.cp311-win_amd64.pyd`: Python 共享库，通过编译 SWIG 编译文件生成。

`ATKMinSolver.dll`: 项目依赖库，ATK 项目自带

`testCallPyd.py`: Python 测试文件，可参考项目示例文件 `testCallPyd.py`

`AstroData`: 项目依赖默认数据库，存放在 Python 安装目录

只要正确放置这些文件并安装好 Python 环境，就能通过 Python 调用 ATK 接口，完成场景对象的创建与设置。

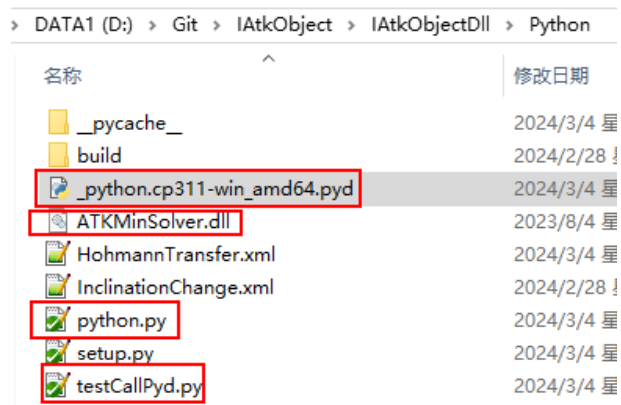


图 3-48 Python 测试依赖文件

3.3.7.2. 编写 Python 测试代码（卫星倾角改变）

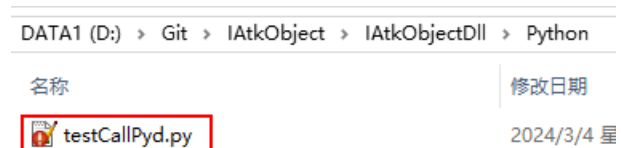


图 3-49 Python 测试脚本

```
# testCallPyd.py

import python;

def TestInclinationChange(pIAtkObjectRoot):
    #新建场景
    pIAtkObject = pIAtkObjectRoot.GetChildren().New(python.eScenario, "In-
clinationChange");
    pIScenario = python.IAtkObject2IScenario(pIAtkObject);
    pIScenario.SetTimePeriod("5 Nov 2022 00:00:00.000", "8 Nov 2022
00:00:00.000");
    #新建卫星
    pIAtkObject = pIScenario.GetChildren().New(python.eSatellite, "SatInclina-
tionChange");
    pISatellite = python.IAtkObject2ISatellite(pIAtkObject);

    pISatellite.SetPropagatorType(python.ePropagatorAstrogator);
    pIVADriverMCS = python.IVePropagator2IVADriverMCS(pISatel-
lite.GetPropagator());
    pIVAMCSSegmentCollection = pIVADriverMCS.GetMainSequence();
```

```

        if (python.eVASegmentTypeInitialState != pIVAMCSSegmentCollection.Item(0).Type):
            return;
        pIVAMCSInitialState = python.IVAMCSSegment2IVAMCSInitialState(pIVAMCSSegmentCollection.Item(0));

        pIVAMCSSegment00 = pIVAMCSSegmentCollection.Insert(python.eVASegmentTypePropagate, "Propagate", "-");
        pIVAMCSPropagate = python.IVAMCSSegment2IVAMCSPropagate(pIVAMCSSegment00);
        pIVAMCSSegment01 = pIVAMCSSegmentCollection.Insert(python.eVASegmentTypeTargetSequence, "TargetSequence", "-");
        pIVAMCSTargetSequence = python.IVAMCSSegment2IVAMCSTargetSequence(pIVAMCSSegment01);
        pIVAMCSSegmentCollection0 = pIVAMCSTargetSequence.GetSegments();
        pIVAMCSSegment02 = pIVAMCSSegmentCollection0.Insert(python.eVASegmentTypeManeuver, "Maneuver", "-");
        pIVAMCSManeuver = python.IVAMCSSegment2IVAMCSManeuver(pIVAMCSSegment02);

        pIVAMCSSegment10 = pIVAMCSSegmentCollection.Insert(python.eVASegmentTypePropagate, "Propagate1", "-");
        pIVAMCSPropagate1 = python.IVAMCSSegment2IVAMCSPropagate(pIVAMCSSegment10);
        pIVAMCSSegment11 = pIVAMCSSegmentCollection.Insert(python.eVASegmentTypeTargetSequence, "TargetSequence1", "-");
        pIVAMCSTargetSequence1 = python.IVAMCSSegment2IVAMCSTargetSequence(pIVAMCSSegment11);
        pIVAMCSSegmentCollection1 = pIVAMCSTargetSequence1.GetSegments();
        pIVAMCSSegment12 = pIVAMCSSegmentCollection1.Insert(python.eVASegmentTypeManeuver, "Maneuver", "-");
        pIVAMCSManeuver1 = python.IVAMCSSegment2IVAMCSManeuver(pIVAMCSSegment12);

```

```

    pIVAMCSSegment20 = pIVAMCSSegmentCollection.Insert(python.eVASe-
gmentTypePropagate, "Propagate2", "-");
    pIVAMCSPropagate2 = python.IVAMCSSegment2IVAMCSPropa-
gate(pIVAMCSSegment20);
    pIVAMCSSegment21 = pIVAMCSSegmentCollection.Insert(python.eVASe-
gmentTypeTargetSequence, "TargetSequence2", "-");
    pIVAMCSTargetSequence2 = python.IVAMCSSegment2IVAMCSTar-
getSequence(pIVAMCSSegment21);
    pIVAMCSSegmentCollection2 = pIVAMCSTargetSequence2.GetSegments();
    pIVAMCSSegment22 = pIVAMCSSegmentCollection2.Insert(py-
thon.eVASegmentTypeManeuver, "Maneuver", "-");
    pIVAMCSManeuver2 = python.IVAMCSSegment2IVAMCSManeu-
ver(pIVAMCSSegment22);

    pIVAMCSSegment30 = pIVAMCSSegmentCollection.Insert(python.eVASe-
gmentTypePropagate, "Propagate3", "-");
    pIVAMCSPropagate3 = python.IVAMCSSegment2IVAMCSPropa-
gate(pIVAMCSSegment30)

    pIVAMCSInitialState.SetOrbitEpoch("5 Nov 2022 00:00:00.000");
    pIVAMCSInitialState.SetElementType(python.eVAElementTypeKeplerian);
    pIVAElementKeplerian = python.IVAElement2IVAElementKep-
lerian(pIVAMCSInitialState.GetElement());
    pIVAElementKeplerian.SetSemiMajorAxis(6570);
    pIVAElementKeplerian.SetEccentricity(0);
    pIVAElementKeplerian.SetInclination(28);
    pIVAElementKeplerian.SetRAAN(0);
    pIVAElementKeplerian.SetArgOfPeriapsis(0);
    pIVAElementKeplerian.SetTrueAnomaly(0);

    pIVASToppingConditionElement = pIVAMCSPropagate.GetStoppingCondi-
tions().Add("Duration");
    pIVASToppingCondition = python.IVASToppingConditionCompo-
nent2IVASToppingCondition(pIVASToppingConditionElement.GetProperties());
    pIVASToppingCondition.SetTrip(7200);

```

```

pIVASToppingCondition.SetTolerance(0.0001);

pIVAMCSManeuver.SetManeuverType(python.eVAManeuverTypeImpulsive);
pIVAManeuverImpulsive = python.IVAManeuver2IVAManeuverImpulsive(pIVAMCSManeuver.GetManeuver());
pIVAAttitudeControlImpulsiveThrustVector = python.IVAAttitudeControl2IVAAttitudeControlImpulsiveThrustVector(pIVAManeuverImpulsive.GetAttitudeControl());
pIVAAttitudeControlImpulsiveThrustVector.SetThrustAxesName("Satellite VNC(Earth)");
pIVAMCSManeuver.EnableControlParameter(python.eVAControlManeuverImpulsiveCartesianX);
pIVAMCSManeuver.Results.Add("Radius_Of_Apoapsis");

pIVAProfileDifferentialCorrector1 = python.IVAProfile2IVAProfileDifferentialCorrector1(pIVAMCSTargetSequence.GetProfiles().Add("Differential Corrector"));
pIVADCCControl = pIVAProfileDifferentialCorrector1.GetControlParameters().GetControlByPaths("Maneuver", "ImpulseX");
pIVADCCResult = pIVAProfileDifferentialCorrector1.Results.GetResultByPaths("Maneuver", "RadiusOfApoapsis");
pIVADCCControl = pIVAProfileDifferentialCorrector1.GetControlParameters().Item(0);
pIVADCCControl.SetEnable(True);
pIVADCCControl.SetMaxStep(100);
//pIVADCCControl.SetCorrection(2456.42862556706);
pIVADCCControl.SetPerturbation(0.1);
pIVADCCControl.SetScalingValue(1);
pIVADCCResult = pIVAProfileDifferentialCorrector1.Results.Item(0);
pIVADCCResult.SetEnable(True);
pIVADCCResult.SetDesiredValue(42160000);
pIVADCCResult.SetScalingValue(1);
pIVADCCResult.SetTolerance(0.1);
pIVADCCResult.SetWeight(1);

```



```

    pIVASToppingConditionElement1 = pIVAMCSPropagate1.GetStoppingCon-
ditions().Add("Apoapsis");
    pIVASToppingCondition1 = python.IVASToppingConditionCompo-
nent2IVASToppingCondition(pIVASToppingConditionElement1.GetProperties());
    pIVASToppingCondition1.SetTolerance(0.0001);
    pIVASToppingCondition1.SetRepeatCount(1);

    pIVAMCSManeuver1.SetManeuverType(python.eVAManeuverTypeImpul-
sive);
    pIVAManeuverImpulsive1 = python.IVAManeuver2IVAManeuverImpul-
sive(pIVAMCSManeuver1.GetManeuver());
    pIVAAttitudeControlImpulsiveThrustVector1 = python.IVAAttitudeCon-
trol2IVAAttitudeControlImpulsiveThrustVector(pIVAManeuverImpulsive1.GetAtti-
tudeControl());
    pIVAAttitudeControlImpulsiveThrustVector1.SetThrustAxesName("Satellite
VNC(Earth)");
    pIVAMCSManeuver1.EnableControlParameter(python.eVAControlManeu-
verImpulsiveCartesianX);
    pIVAMCSManeuver1.Results.Add("Eccentricity");

    pIVAProfileDifferentialCorrector11 = python.IVAProfile2IVAProfileDiffer-
entialCorrector1(pIVAMCSTargetSequence1.GetProfiles().Add("Differential Correc-
tor"));
    pIVADCCControl1 = pIVAProfileDifferentialCorrector11.GetControlParame-
ters().Item(0);
    pIVADCCControl1.SetEnable(True);
    pIVADCCControl1.SetMaxStep(100);
    pIVADCCControl1.SetCorrection(1480.06329844802);
    pIVADCCControl1.SetPerturbation(0.1);
    pIVADCCControl1.SetScalingValue(1);
    pIVADCCResult1 = pIVAProfileDifferentialCorrector11.Results.Item(0);
    pIVADCCResult1.SetEnable(True);
    pIVADCCResult1.SetDesiredValue(0);
    pIVADCCResult1.SetScalingValue(1);
    pIVADCCResult1.SetTolerance(0.001);

```

```

pIVADCResult1.SetWeight(1);

pIVASToppingConditionElement2 = pIVAMCSPropagate2.GetStoppingCon-
ditions().Add("AscendingNode");
pIVASToppingCondition2 = python.IVASToppingConditionCompo-
nent2IVASToppingCondition(pIVASToppingConditionElement2.GetProperties());
pIVASToppingCondition2.SetRepeatCount(2);
pIVASToppingCondition2.SetTolerance(1e-6);

pIVAMCSManeuver2.SetManeuverType(python.eVAManeuverTypeImpul-
sive);
pIVAManeuverImpulsive2 = python.IVAManeuver2IVAManeuverImpul-
sive(pIVAMCSManeuver2.GetManeuver());
pIVAAttitudeControlImpulsiveThrustVector2 = python.IVAAttitudeCon-
trol2IVAAttitudeControlImpulsiveThrustVector(pIVAManeuverImpulsive2.GetAtti-
tudeControl());
pIVAAttitudeControlImpulsiveThrustVector2.SetThrustAxesName("Satellite
VNC(Earth)");
pIVAMCSManeuver2.EnableControlParameter(python.eVAControlManeu-
verImpulsiveCartesianX);
pIVAMCSManeuver2.EnableControlParameter(python.eVAControlManeu-
verImpulsiveCartesianY);
pIVAMCSManeuver2.Results.Add("Eccentricity");
pIVAMCSManeuver2.Results.Add("Inclination");

pIVAProfileDifferentialCorrector12 = python.IVAProfile2IVAProfileDiffer-
entialCorrector1(pIVAMCSTargetSequence2.GetProfiles().Add("Differential Correc-
tor"));
pIVADCCControl2 = pIVAProfileDifferentialCorrector12.GetControlParame-
ters().Item(0);
pIVADCCControl2.SetEnable(True);
pIVADCCControl2.SetMaxStep(100);
pIVADCCControl2.SetCorrection(-359.983751789857);
pIVADCCControl2.SetPerturbation(0.1);
pIVADCCControl2.SetScalingValue(1);

```

```

    pIVADCCControl3 = pIVAProfileDifferentialCorrector12.GetControlParameters().Item(1);
    pIVADCCControl3.SetEnable(True);
    pIVADCCControl3.SetMaxStep(100);
    pIVADCCControl3.SetCorrection(-1444.72488992759);
    pIVADCCControl3.SetPerturbation(0.1);
    pIVADCCControl3.SetScalingValue(1);
    pIVADCResult2 = pIVAProfileDifferentialCorrector12.Results.Item(0);
    pIVADCResult2.SetEnable(True);
    pIVADCResult2.SetDesiredValue(0);
    pIVADCResult2.SetScalingValue(1);
    pIVADCResult2.SetTolerance(0);
    pIVADCResult2.SetWeight(1);
    pIVADCResult3 = pIVAProfileDifferentialCorrector12.Results.Item(1);
    pIVADCResult3.SetEnable(True);
    pIVADCResult3.SetDesiredValue(0);
    pIVADCResult3.SetScalingValue(1);
    pIVADCResult3.SetTolerance(0.001);
    pIVADCResult3.SetWeight(1);

    pIVASToppingConditionElement3 = pIVAMCSPropagate3.GetStoppingConditions().Add("Duration");
    pIVASToppingCondition3 = python.IVASToppingConditionComponent2IVASToppingCondition(pIVASToppingConditionElement3.GetProperties());
    pIVASToppingCondition3.SetTrip(129600);
    pIVASToppingCondition3.SetTolerance(1e-5);

    pIVADriverMCS.RunMCS();
    pIVADriverMCS.ApplyAllProfileChanges();

    print(pIVAProfileDifferentialCorrector12);

#新建根节点
pIAtkObjectRoot = python.IAtkObjectRoot();

```

```
#倾角改变
TestInclinationChange(pIAtkObjectRoot);
#霍曼转移
#TestHohmannTransfer(pIAtkObjectRoot);
#仿真运行
pIAtkObjectRoot.m_Animation.PlayForward();
#输出数据报告
#pIAtkObjectRoot.OutputDataReport(pISate1, "地固系位置速度");
#保存场景
pIAtkObjectRoot.SaveScenario();
#关闭场景
pIAtkObjectRoot.CloseScenario();
```

3.3.7.3. cmd 编译并执行 Python 测试代码

Python 执行命令：python testCallPyd.py

3.3.7.4. 编译生成文件

3.3.7.4.1. 想定文件 InclinationChange.xml

倾角改变想定文件 InclinationChange.xml，保存想定内所有对象的所有基本信息与运算信息。可通过 ATK 软件打开该想定文件，查看数值设定与仿真运行效果。

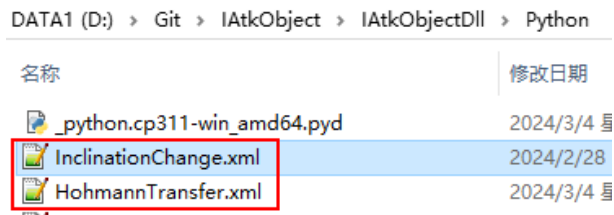


图 3-50 Python 测试生成文件

3.3.7.5. 打开生成的想定文件查看效果

Step1: 打开 ATK,点击打开，选择生成的倾角改变想定文件 Inclination-Change.xml，再点击打开

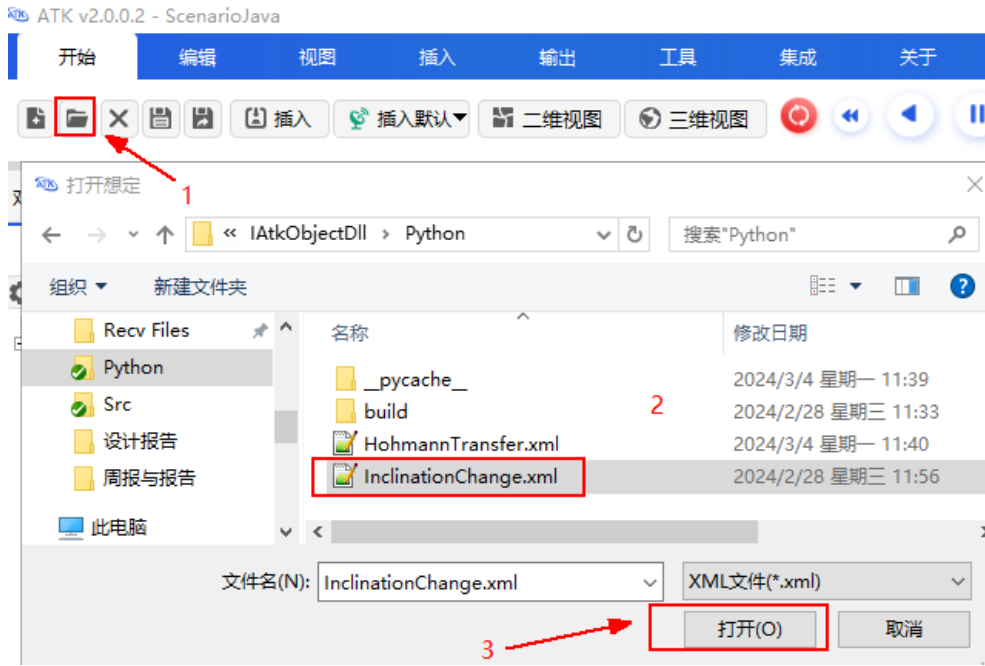


图 3-51 打开生成的倾角改变想定文件

Step2: 点击运行，打开 3 维视图，查看运行效果

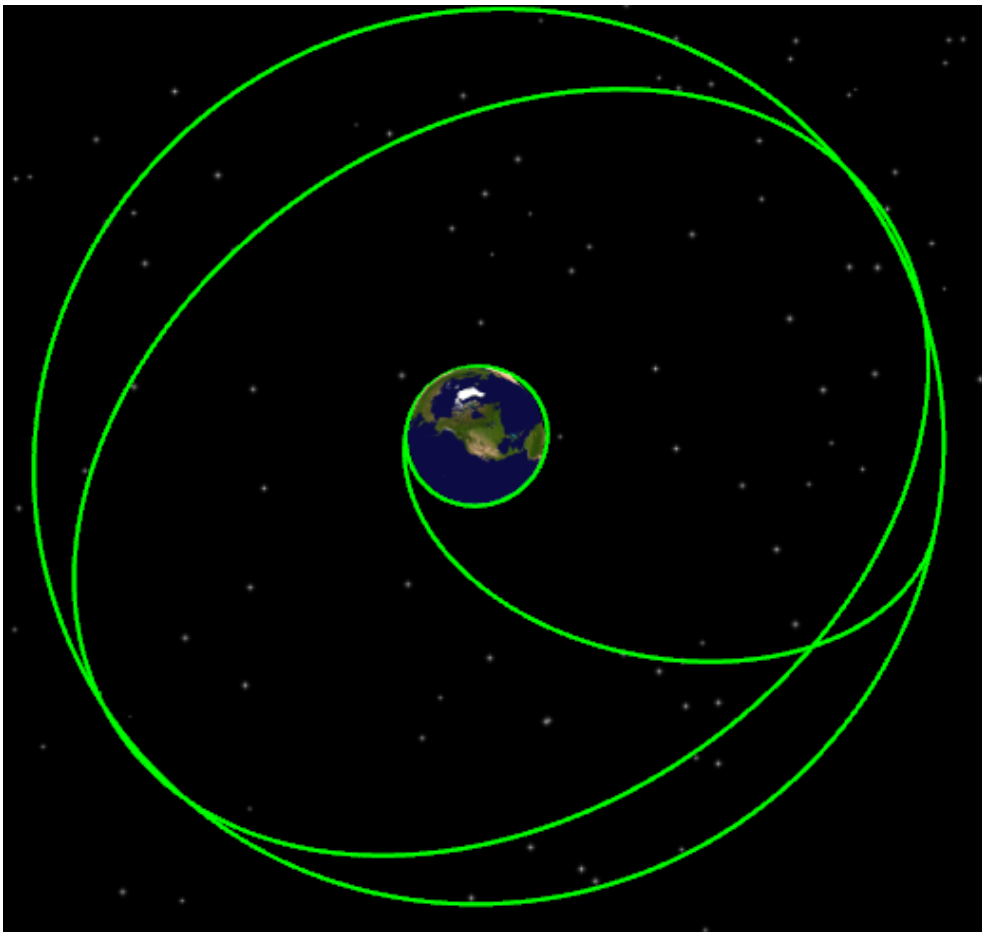


图 3-52 打开三维视图查看仿真运行轨迹

4. ATK 发布说明

4.1. 重要声明

4.1.1. 权利声明

本声明涉及由 ATK 开发组开发的所有软件“ATK, Aerospace Tool Kit”。本软件的一切版权等知识产权, 以及与软件相关的所有信息内容, 包括但不限于: 文字表述及其组合、图标、图饰、图表、色彩、界面设计、版面框架、有关数据、印刷材料、或电子文档等均为国防科技大学所有, 受著作权法和国际著作权条约以及其他知识产权法律法规的保护。

4.1.2. 技术声明

4.1.2.1. Windows 平台兼容性

ATK V1.1 支持 Windows 7/8/x/10 平台的使用。

4.1.2.2. 开发环境信息

ATK V1.1 基于 Visual Studio 2015 开发。

4.1.2.3. 软件授权信息

ATK V1.1 的安装及使用需要许可证文件。您向 ATK 开发组提交计算机 ID 后, 将获得序列号授权使用软件。

4.1.3. 使用条款

4.1.3.1. 接受条款

在您使用 ATK 和任何相关服务时, 需遵守以下使用条款。ATK 保留随时更新使用条款的权利, 恕不另行通知。

4.1.3.2. 服务说明

通过 ATK 软件和相关服务, ATK 让您能够获取各种资源, 包括下载、授权、文档、视频、开发人员工具、和产品信息等(以下统称为“服务”)。服务

（包括任何更新、增强、新功能和/或增加任何新 Web 资产）都须受使用条款约束。

4.1.3.3. 个人和非商业用途限制

除非另有说明，服务只适用于个人与非商业用途。未经 ATK 事先书面许可，对于从服务取得的任何信息、软件、产品或服务，您不得对其修改、复制、分发、传送、公开展示、执行、复制、公布、许可、转让、销售或基于以上内容创建衍生作品（除非用于您自己的个人非商业用途）。

4.1.3.4. 隐私与个人信息保护

为保护用户权益和隐私，本软件不含有任何旨在破坏用户计算机数据和获取用户隐私信息的恶意代码，不含有任何跟踪、监视用户计算机和或操作行为的功能代码，不会监控用户网上、网下的行为或泄漏用户隐私。

4.1.3.5. 有关 ATK 提供的软件的声明

从 ATK 下载使用的任何软件均是 ATK 和/或其供应商的版权作品。软件的使用受软件使用条款或软件使用协议的约束。如果最终用户没有事先同意许可协议条款，则最终用户将无法安装或包含许可协议的任何软件。某些第三方无权向您授予许可。

本软件保留为用户提供软件的修改、升级的权利。

用户无权实施包括但不限于下列行为：

- （1）不得删除本软件及其他副本上所有关于版权的信息、内容；
- （2）不得对本软件进行规避、破坏著作权人为保护本软件著作权而采取的技术措施等；
- （3）对于本软件相关信息等，未经 ATK 书面同意，用户不得擅自实施包括但不限于下列行为：使用、复制、修改、链接、转载、汇编、发表、出版，建立镜像站点、擅自借助"软件"发展与之有关的衍生产品、作品、服务等。
- （4）用户不得利用本软件发表、传送、传播、储存违反国家法律、危害国家安全、祖国统一、社会稳定的内容，或任何不当的、侮辱诽谤的、淫秽的、暴力的及任何违反国家法律法规政策的内容。
- （5）用户不得利用本软件发表、传送、传播、储存侵害他人知识产权、商业秘密权等合法权利的内容。
- （6）用户不得利用本软件误导、欺骗他人。
- （7）用户不得利用本软件传送或散布以其他方式实现传送含有受到知识产

权法律保护的图像、相片、软件或其他资料的文件，作为举例（但不限于此），包括版权或商标法（或隐私权或公开权），除非您拥有或控制着相应的权利或已得到所有必要的认可。

(8) 用户不得利用本软件使用任何包含有通过侵犯商标、版权、专利、商业机密或任何一方的其他专有权利的方式利用本"软件"获得的图像或相片的资料或信息。

(9) 用户不得进行任何危害计算机网络安全的行为，包括但不限于：使用未经许可的数据或进入未经许可的服务器/帐户；未经允许进入公众计算机网络或者他人计算机系统并删除、修改、增加存储信息；未经许可，企图探查、扫描、测试本软件系统或网络的弱点或其它实施破坏网络安全的行为；企图干涉、破坏本软件系统或网站的正常运行，故意传播恶意程序或病毒以及其他破坏干扰正常网络信息服务的行为；伪造 TCP/IP 数据包名称或部分名称。

4.1.3.6. 有关 ATK 提供的文档的声明

某些文档可能受独立于本使用条款的明确许可条款的约束。如果条款间存在冲突，应以明确的许可条款为准。如果：

- (1) 所有副本包含以下版权声明，而且同时包含版权声明和该许可声明；
- (2) 此类服务文档的使用仅限于获得信息以及非商业或个人用途，同时不会在任何网络计算机上复制或公布，或者通过任何介质传播；

(3) 没有对任何文档做出修改，则授予对服务文档（比如白皮书、新闻稿、数据表和常见问题解答）的使用许可。经认可的教育机构和科研院所，可下载和复制文档，并在课堂上分发。如需在课堂之外分发文档，必须获得明确的书面许可。将文档用于任何其他目的是法律明确禁止的，并可能导致严重的民事和刑事处罚。违约者将受到最大可能的起诉。

4.1.3.7. 服务的免责声明和责任限制

(1) 免责声明

对于因 ATK 提供的服务（包括使用或执行服务提供的信息）引起的或与其相关的使用中断、数据或利润损失，无论是由合同行为、过失或其他侵权行为导致，ATK 和/或其供应商对由此造成的任何特别的、间接的、后果性损害或任何其他损害概不负责。

本服务发布的文档和相关图形可能会有技术错误或排印错误。此处会定期添加变更信息。ATK 和/或其供应商可能会随时对此处所述的产品和/或程序进行改进和/或更改。

对于因使用或执行服务、软件、文档、提供或未能提供服务，或者服务提

供的信息引起的或与其相关的使用中断、数据或利润损失，无论是由合同行为、过失或其他侵权行为导致，ATK 和/或其供应商对由此造成的任何特别的、间接的、后果性损害或任何其他损害概不负责。

(2) 责任限制

使用 ATK 必须遵守国家有关法律和政策，维护国家利益，保护国家安全，并遵守本使用条款，对于用户违法或违反本使用条款的使用而引起的一切责任，由用户负全部责任，一概与 ATK 及合作单位无关，导致 ATK 及合作单位损失的，ATK 及合作单位有权要求用户赔偿，并有权立即停止向其提供服务，保留相关记录，保留配合司法机关追究法律责任的权利。

ATK 保留在任何时候根据适用法律、法规、法律程序或政府要求的需要而披露任何信息，或 ATK 自主决定全部或部分地编辑、拒绝张贴或删除任何信息或资料的权利。

ATK 保留在任何时候通过为您提供软件替换、修改、升级版本的权利。同时，为提高用户体验，ATK 有权将其他相关功能产品绑定在 ATK 上供用户下载和安全的权利。

本软件可能使用的第三方软件或技术，是受到合法授权的。因第三方软件或技术引发的任何纠纷，由该第三方负责解决，ATK 不承担任何责任。ATK 不对第三方软件或技术提供客服支持，若您需要获取支持，请与该软件或技术提供商联系。

[附录一 Connect 模式功能介绍](#)

[附录二 Component 模式功能介绍](#)