

目录

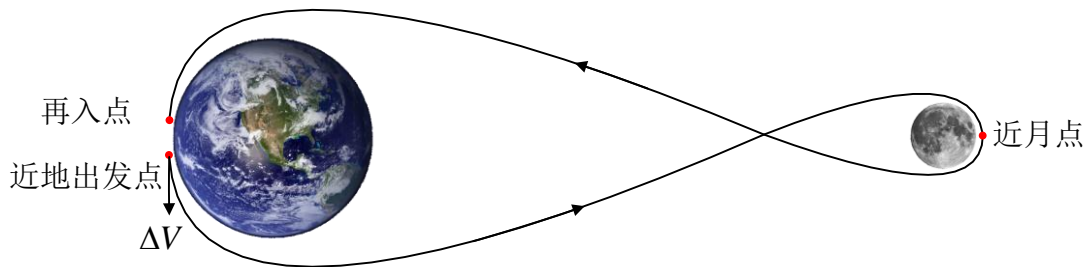
界面操作案例集（上册）	1
1. ATK 特定问题建模—地月自由返回轨道设计	2
2. ATK 特定问题建模—水滴型相对绕飞轨道设计	11
3. ATK 特定问题建模—空间站与空间目标的接近分析	24
4. ATK 特定问题建模—灾害监测星座对地覆盖性能分析	33
5. ATK 特定问题建模—基于二次开发的轨道偏差演化分析	45
二次开发案例集（下册）	74
6. 基于 ATK.Connect 模式的轨道快速转移的 C++实现	75
7. 基于 ATK.Connect 模式的轨道快速转移的 Matlab 实现.....	89
8. 基于 ATK.Connect 模式的轨道快速转移的 Java 实现.....	104
9. 基于 ATK.Connent 模式的轨道快速转移的 Python 实现.....	118
10. 基于 ATK.Component 模式的轨道快速转移的 C++实现	132
11. 基于 ATK.Component 模式的轨道快速转移的 Java 实现.....	149
12. 基于 ATK.Component 模式的轨道快速转移的 Python 实现.....	160
参考文献	171

界面操作案例集（上册）

1. ATK 特定问题建模—地月自由返回轨道设计

内容简介

自由返回轨道是从地球出发，在不施加机动的情况下，能够绕月安全返回地球的轨道，如图所示。此类轨道对保障航天员的人身安全具有重要作用，因此被广泛应用于载人月球探测任务中，美国的阿波罗 8 号、10 号以及 11 号。



地月自由返回轨道设计本质上属于一类非线性方程组的边值问题。通过求解该边值问题，得到某一给定时刻下的满足约束的自由返回轨道。该问题的边界条件为近地出发轨道历元、近地出发点轨道高度、近地出发点轨道偏心率（等于 0）、近地出发点轨道倾角、近月点高度、绕月后返回的近地点高度和近地点轨道倾角，需要求解的参数为近地出发轨道升交点赤经、近地出发点纬度幅角、近地出发速度增量大小（通常施加切向的速度增量）。

在本案例中，边界条件参数设置如下：近地出发轨道历元为：2033-04-04 10:05:00.000（可以任意选择），近地出发点轨道高度为 200km，近地出发点轨道偏心率为 0，近地出发点轨道倾角为 28° ，近月点高度为 100km，绕月后返回的近地点高度为 50km，绕月后返回的近地点轨道倾角为 43° 。需要求解的参数初值设置如下：近地出发轨道升交点赤经为 86° ，近地出发点纬度幅角为 223° ，近地点出发速度增量大小为 3.1645km/s。

1.1 自由返回轨道设计方法

1.1.1. 动力学模型

高精度模型是最接近飞行器真实运动轨迹的轨道动力学模型。该模型主要考虑地球引力、太阳引力、月球引力、太阳光压、大气阻力、地球非球形摄动和月球非球形摄动等多种力的影响，建立飞行器位置和速度的动力学微分方程

组，然后通过数值积分方法计算飞行器在给定初始状态后任意时刻下的位置和速度。在地心 J2000 系下，飞行器的高精度轨道动力学模型可以表示为

$$\begin{cases} \frac{d\mathbf{R}}{dt} = \mathbf{V} \\ \frac{d\mathbf{V}}{dt} = -\frac{\mu_E}{R^3}\mathbf{R} - \mu_M\left(\frac{\mathbf{R}_M}{R_M^3} + \frac{\boldsymbol{\rho}_M}{\rho_M^3}\right) - \mu_S\left(\frac{\mathbf{R}_S}{R_S^3} + \frac{\boldsymbol{\rho}_S}{\rho_S^3}\right) \\ \quad + \mathbf{a}_E^{NS} + \mathbf{a}_M^{NS} + \mathbf{a}_{\text{Drag}} + \mathbf{a}_{\text{SR}} + \mathbf{a}_T + \mathbf{a}_{\text{Other}} \end{cases} \quad (1)$$

其中， $\mathbf{R} = [X, Y, Z]^T$ 和 $\mathbf{V} = [V_x, V_y, V_z]^T$ 分别表示飞行器的位置和速度矢量， μ_E 、 μ_M 和 μ_S 分别表示地球、月球和太阳引力常数， \mathbf{R}_M 为月球位置矢量， $\boldsymbol{\rho}_M$ 为飞行器相对于月球位置矢量， \mathbf{R}_S 为太阳位置矢量， $\boldsymbol{\rho}_S$ 为飞行器相对于太阳位置矢量， \mathbf{a}_E^{NS} 为地球非球形引力摄动项， \mathbf{a}_M^{NS} 为月球非球形引力摄动项， \mathbf{a}_{Drag} 为大气阻力摄动项， \mathbf{a}_{SR} 为太阳光压摄动项， \mathbf{a}_T 为飞行器发动机推力摄动项， $\mathbf{a}_{\text{Other}}$ 为其他高阶小量摄动项。

定义 t 时刻下位置 \mathbf{R} 和速度 \mathbf{V} 组成的向量记为 $\boldsymbol{\eta}(t)$ ，如下式所示

$$\boldsymbol{\eta}(t) = (\mathbf{R}, \mathbf{V}, t) \quad (2)$$

根据动力学方程 (1)，当给定初始状态 t_0 时刻的位置 \mathbf{R}_0 和速度 \mathbf{V}_0 ，对动力学方程 (1) 进行高精度轨道预报获得任意时刻 t 的位置 \mathbf{R} 和速度 \mathbf{V} 。

$$\boldsymbol{\eta}(t) = \mathbf{F}(\boldsymbol{\eta}(t_0), t) \quad (3)$$

式中， \mathbf{F} 是高精度轨道预报函数。

1.1.2. 轨道的计算

基于动力学方程 (1)，通过求解边值问题，得到某一给定时刻下的满足约束的自由返回轨道。近地出发状态由历元时刻 t_0 与半长轴 a_0 、轨道偏心率 e_0 、轨道倾角 i_0 、升交点赤经 Ω_0 、近地点幅角 ω_0 、真近点角 f_0 确定。选择近地出发点的升交点赤经 Ω_0 、纬度幅角 $u_0 = \omega_0 + f_0$ 和近地出发点所施加的切向速度增量 ΔV 作为控制变量，其它参数根据工程要求给定。当给定一组控制变量后，一条地月空间的转移轨道便能确定。

该边值问题的边界条件为

$$\begin{cases} h_{\text{PRL}} - h_{\text{L}} = 0 \\ h_{\text{VCP}} - h_{\text{F}} = 0 \\ i_{\text{VCP}} - i_{\text{F}} = 0 \end{cases} \quad (4)$$

式中， h_{PRL} 是迭代计算过程中的近月点高度， h_{VCP} 是迭代计算过程中的绕月后返回的近地点高度， i_{VCP} 是迭代计算过程中的绕月后返回的近地点轨道倾角， h_{L} 是目标近月点高度， h_{F} 是绕月后返回的目标近地点高度， i_{F} 是绕月后返回的目标近地点轨道倾角。

当给定了控制变量和边界条件后，对于动力学公式（4）中的第一个边界条件，利用公式（3）进行高精度预报获得近月点处的 \mathbf{R}_{PRL} 和 \mathbf{V}_{PRL} ，再利用状态转换获得 t_{PRL} 时刻的轨道根数，进而得到了迭代计算过程中的近月点高度 h_{PRL} 。

$$\boldsymbol{\eta}(t_{\text{PRL}}) = \mathbf{F}(\boldsymbol{\eta}(t_0), t_{\text{PRL}}) \quad (5)$$

式中， t_{PRL} 为近月点时刻。

对于公式（4）中的第二个和第三个边界条件，利用公式（3）进行高精度预报获得绕月后返回近地点的 \mathbf{R}_{VCP} 和 \mathbf{V}_{VCP} ，再利用状态转换获得 t_{VCP} 时刻的轨道根数，进而得到了迭代计算过程中的绕月后返回的近地点高度 h_{VCP} 和绕月后返回的近地点倾角 i_{VCP} 。

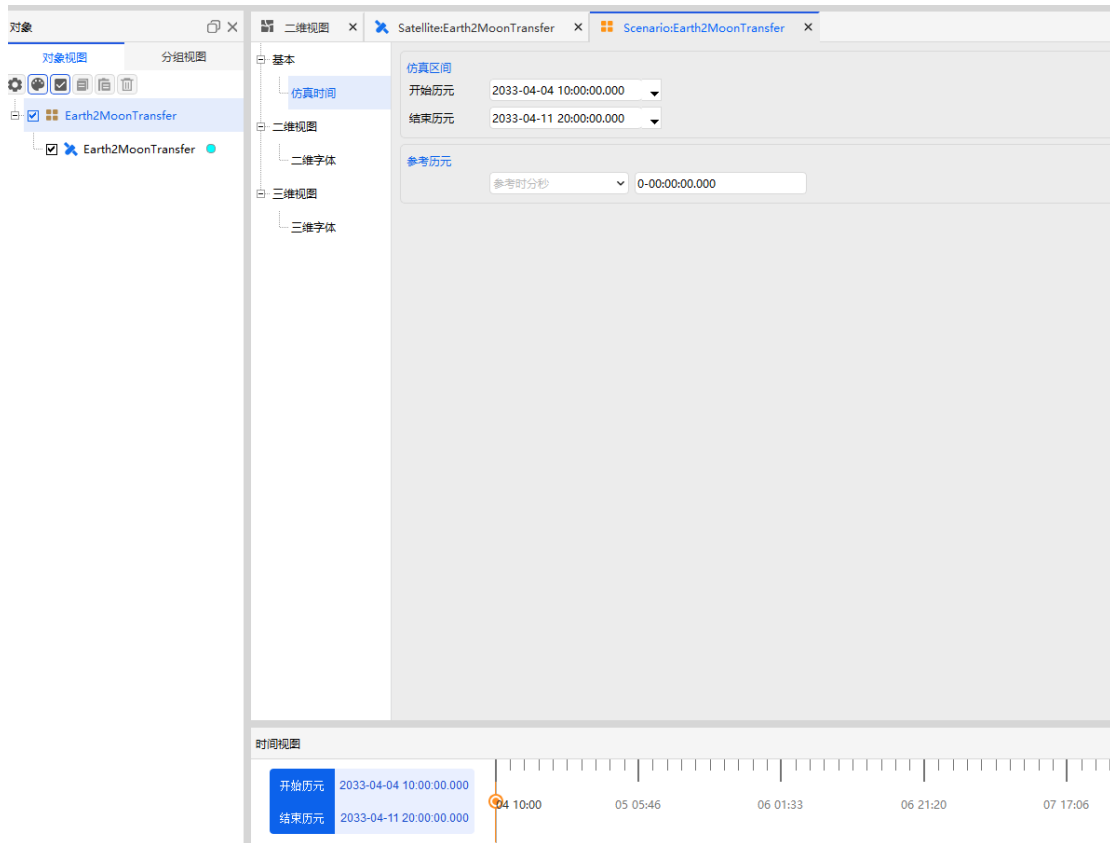
$$\boldsymbol{\eta}(t_{\text{VCP}}) = \mathbf{F}(\boldsymbol{\eta}(t_0), t_{\text{VCP}}) \quad (6)$$

式中， t_{VCP} 为真空近地点时刻。

利用微分修正法对三个控制变量 Ω_0 、 u_0 、 ΔV 进行迭代以获得满足边界条件（4）的三个控制变量的最终解，进而最终获得一条完整的自由返回轨道。

1.2 创建地月自由返回轨道任务场景与对象

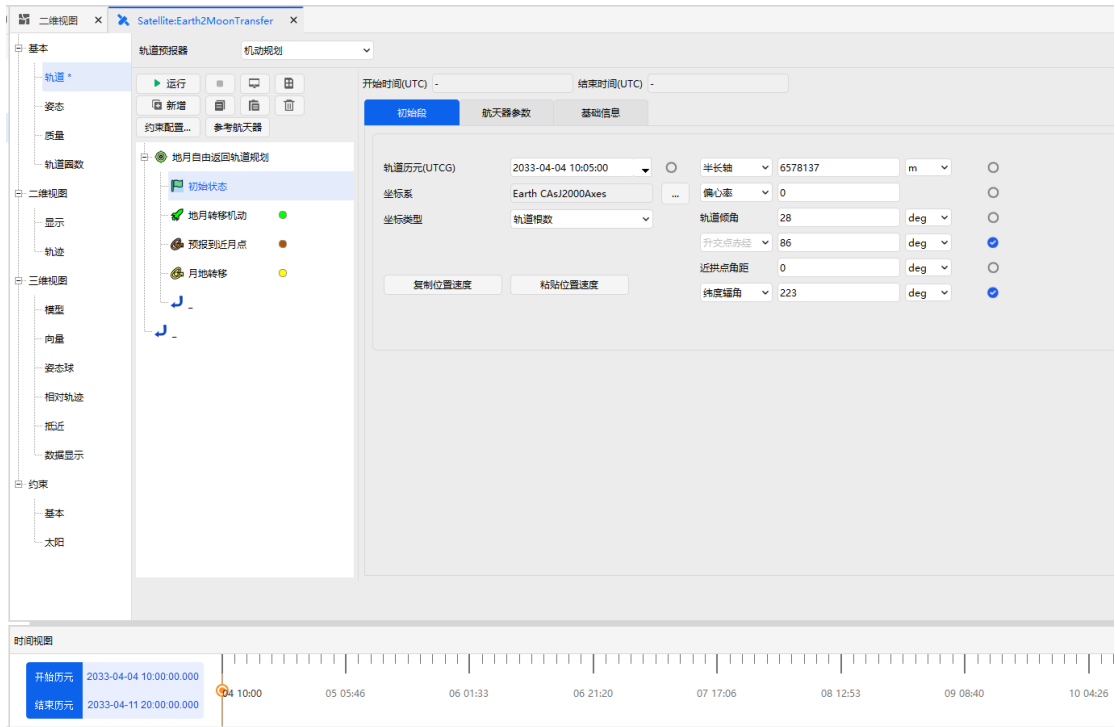
安装并运行 ATK.exe，新建一个想定“Earth2MoonTransfer”。设置开始历元为 2033-04-04 10:00:00.000000，结束历元为 2023-04-11 20:00:00.000000；插入卫星对象，将其命名为“Earth2MoonTransfer”。轨道预报器类型选择机动规划。



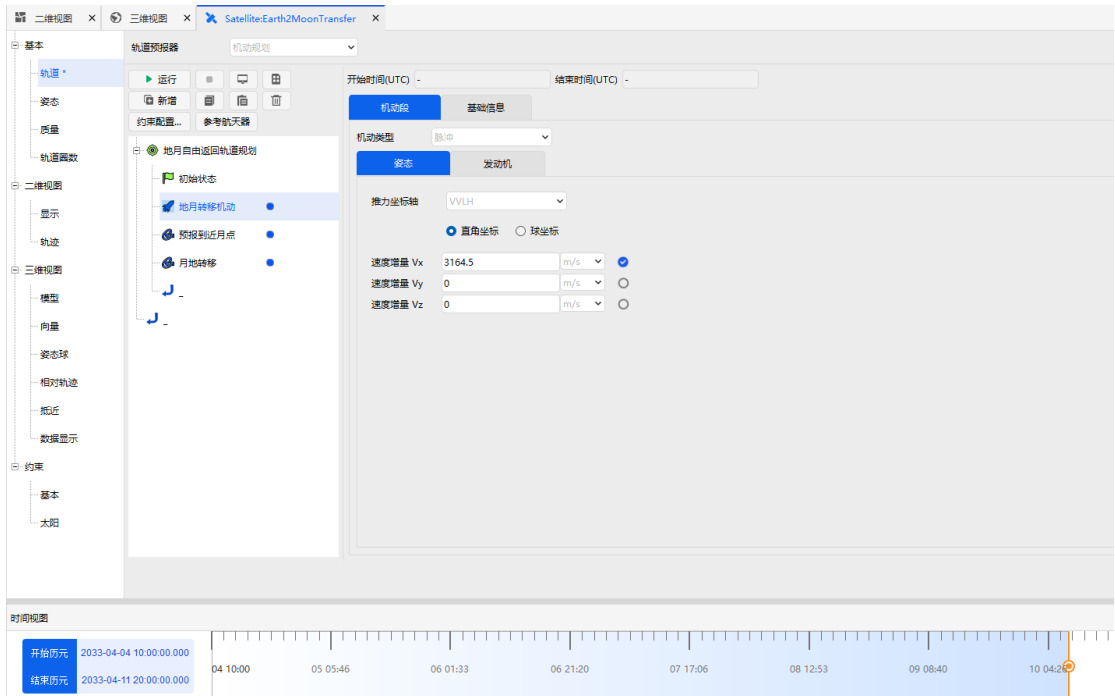
1.3 设计自由返回轨道

1.3.1. 定义瞄准序列段

添加一个瞄准序列段，命名为“地月自由返回轨道规划”，然后嵌入一个初始段，命名为初始状态。设置轨道历元（UTCG）为 2033-04-04 10:05:00.000。坐标系选择 Earth CAsJ2000Axes，坐标类型选择轨道根数，如下图所示。



在初始段后插入一个机动段，将其命名为“地月转移机动”。机动类型为脉冲机动，推力坐标轴为VVLH坐标系。X、Y、Z 三轴的速度增量分别是 3.1645km/s 、0km/s 和 0km/s，仅勾选 X 方向的速度增量。



在机动段后面插入一个预报段，命名为“预报到近月点”，中心天体选择为地球，引力场模型选择为JGM3，引力的次数和阶次均选择为6，忽略大气阻力摄动和太阳光压摄动，考虑太阳和月球的三体摄动，并选择“点质量”模型。停

止条件选择“CAsStopPeriapsis”（预报到近拱点停止），中心天体选择“Moon”。

轨道预报器

摄动力 积分参数

中心天体 Earth

引力

引力场模型 JGM3

次数 6

阶数 6

太阳光压摄动

采用

大气阻力摄动

采用

大气模型 AsENRLMSISE00

太阳辐射通量/地磁指数

手动输入

每日F10.7 150

平均F10.7 150

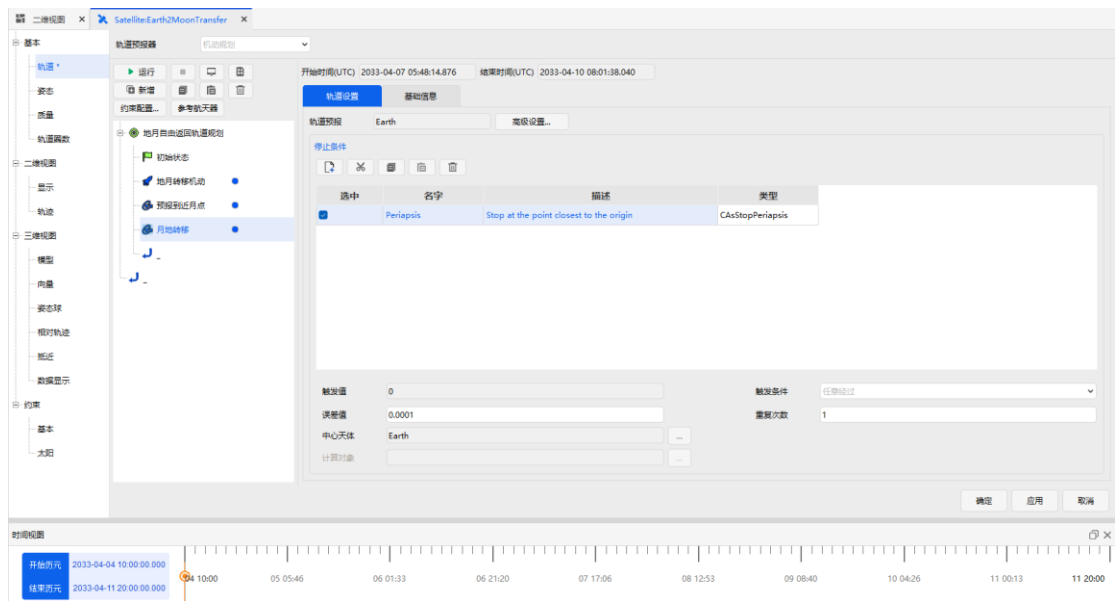
地磁指数Ap 14.9186

三体摄动

选中	名字	类型	模式	阶数	次数
<input checked="" type="checkbox"/>	太阳			0	0
<input checked="" type="checkbox"/>	月球	点质量模型	GLGM2	0	0
<input type="checkbox"/>	火星	点质量模型	GMM1	0	0

确定 取消

在预报到近月点段后插入一个预报段，命名为“月地转移”，摄动力的设置与上一步相同，停止条件选择“CAsStopPeriapsis”（预报到近拱点停止），中心天体选择“Earth”。

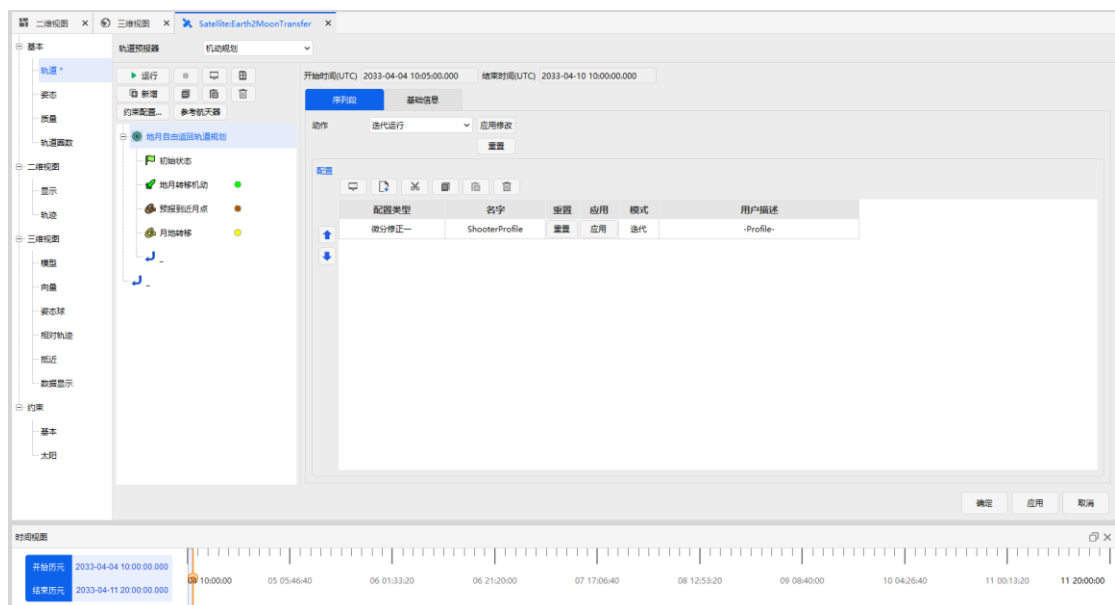


1.3.2. 选择变量

初始状态段中，选择纬度幅角、升交点赤经作为设计变量；地月转移机动段中，选择 X 方向的速度增量为设计变量；预报到近月点选择 CStateCalcAltitudeOfPeriapsis 作为约束条件，中心天体选择月球；月地转移段选择 CStateCalcAltitudeOfPeriapsis 和 CStateCalcInclination 作为约束条件，中心天体选地球。

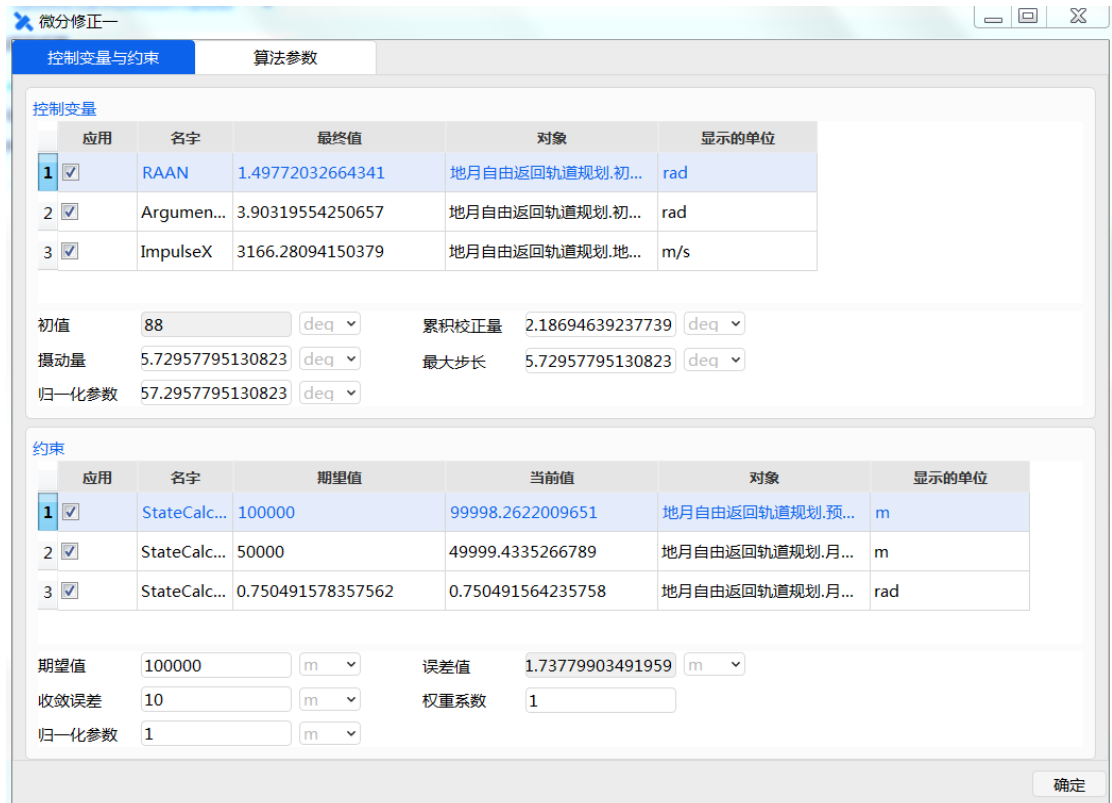
1.3.3. 设置瞄准器

地月自由返回轨道规划配置类型选择“微分修正一”。



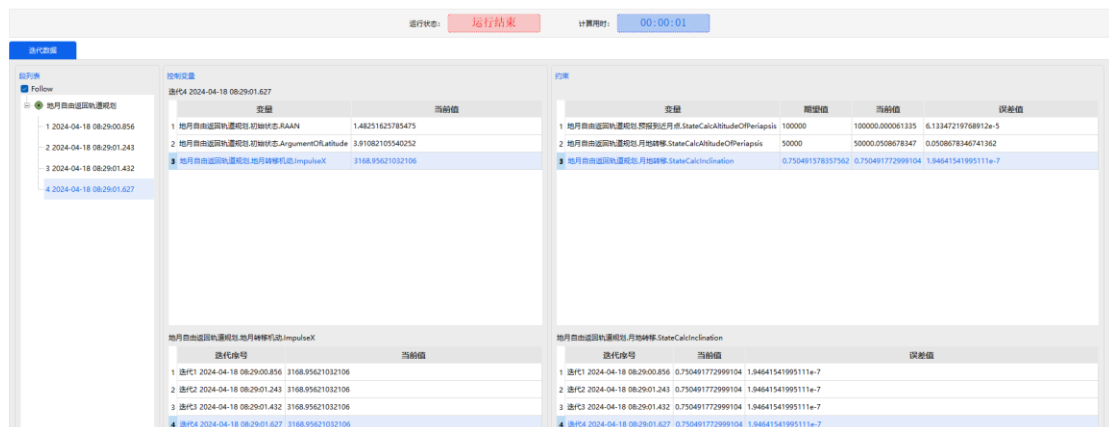
双击“微分修正一”进入瞄准器设置界面。分别点击“控制变量”和“约束”

中“使用”栏下的方框，勾选使用的控制变量和约束，具体的控制变量和约束及其数值如下图所示。近月点高度的期望值为 100km、近地点高度的期望值为 50km、近地点轨道倾角的期望值为 43° ，近月点高度和近地点高度的收敛误差为 0.01km、轨道倾角的收敛误差为 0.001° ，其余参数可以选择默认参数。



1.4 运行地月自由返回轨道并分析

构建完整个任务的控制序列后，点击瞄准序列段，选择迭代运行，点击运行，点击三维视图或者二维视图可以查看运行结果。

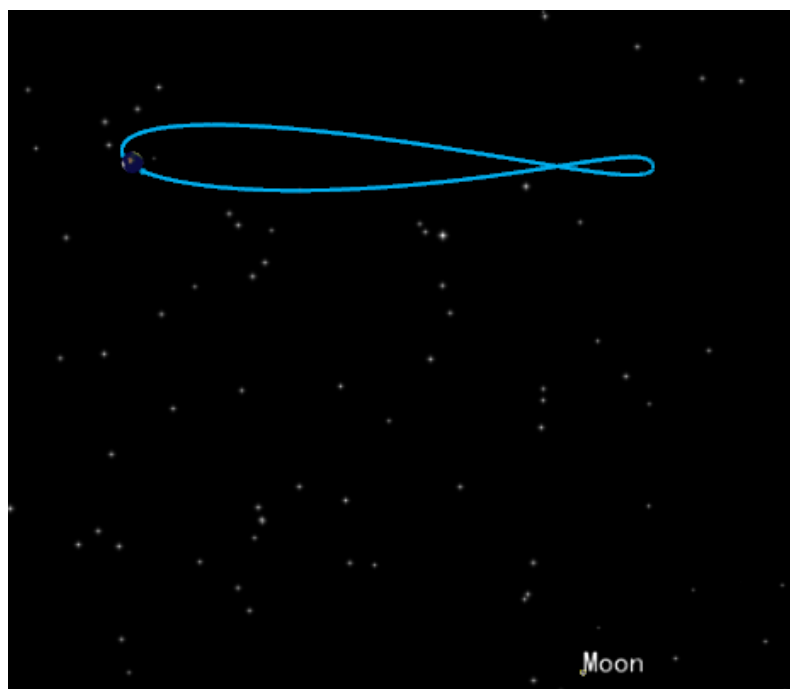


在 ATK 中，针对三个控制变量（近地出发点纬度幅角、近地出轨道升交点赤经和近地出发速度增量大小）设置四种不同的初值，分别计算满足约束条件的自由返回轨道。经过计算发现，这四种工况下，软件均可以得到满足约束条件的解，具体数值如表 1 所示。因此，在一定的初值范围内，通过使用 ATK 可以设计满足约束条件的自由返回轨道。

表 1 四种不同控制变量对应的自由返回轨道计算结果

	纬度幅角初值	升交点赤经初值	速度增量初值	纬度幅角计算结果	升交点赤经计算结果	速度增量计算结果
工况 1	223	86	3164.5	223.6367	85.8130	3166.2811
工况 2	225	88	3169.0	223.6366	85.8131	3166.2809
工况 3	226	88	3168.5	223.6366	85.8131	3166.2809
工况 4	224	88	3167.0	223.6366	85.8131	3166.2808

选取一条自由返回轨道结果展示如下图。



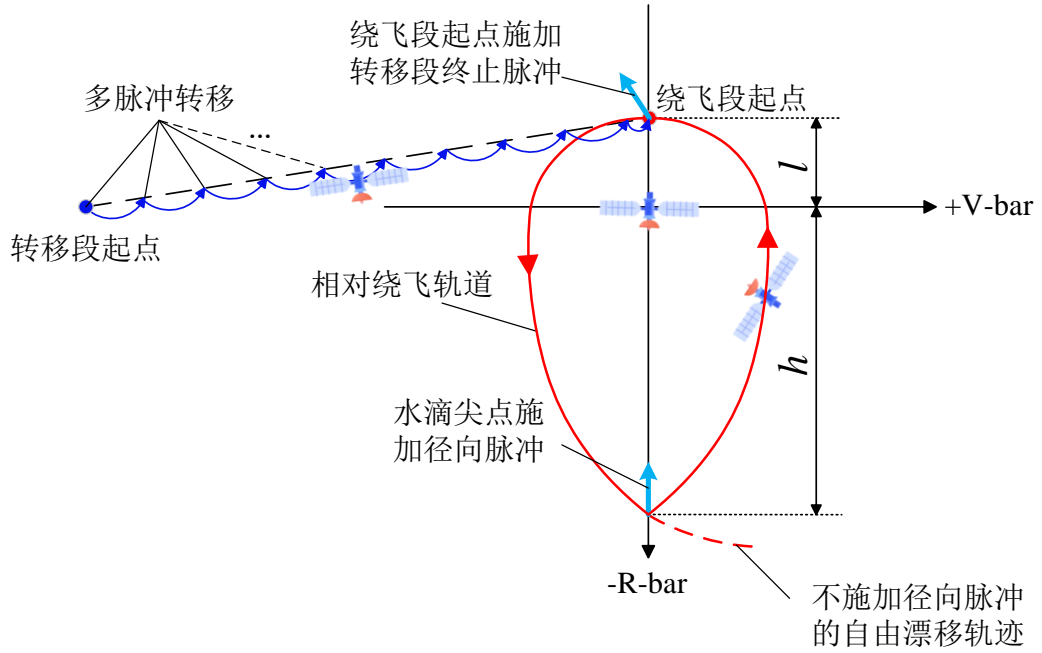
三维动图如下（双击查看）



2. ATK 特定问题建模—水滴型相对绕飞轨道设计

内容简介

随着空间技术水平的不断发展, 多国开展了针对航天器在轨服务技术的尝试和研究, 如美国开展的 Restore-L 在轨补加试验项目, NASA 与加拿大航天局实施机器人燃料加注任务 (RRM) 等。其在任务执行过程中经常用到“绕飞”技术——一个航天器 (伴随航天器) 在一定距离范围内围绕另一目标航天器 (参考航天器) 的周期性相对运动^[1]。要形成水滴形状的封闭绕飞构型, 需要在漂移轨迹自相交点施加径向 ($-\mathbf{R}\text{-bar}$ 方向) 控制。如下图所示, 给出了伴随航天器在参考航天器正后方 (以 $+\mathbf{V}\text{-bar}$ 方向为正) 进行单水滴转移绕飞的轨迹示意图, 其中, 蓝色曲线为伴随航天器受控多脉冲转移的相对转移轨道, 红色曲线为伴随航天器绕参考航天器的水滴状相对轨道 (为便于描述, 下文用 A 点表示水滴尖点, 用 B 点表示绕飞段起点)。



相对绕飞轨道可以分为转移段和绕飞段，其轨道设计可以转化为一个非线性方程组问题进行求解，进而得到整个过程施加的脉冲机动量和机动时刻。针对水滴型相对绕飞轨道，给定任务初始时刻伴随航天器与参考航天器的相对位置和相对速度，多脉冲转移段约束条件为：机动转移时间、最大转移飞行角^[2]；水滴绕飞段约束条件为：水滴绕飞圈数、水滴绕飞构型（由上图中的 l 和 h 决定）。

在本案例中，首先根据需求设置仿真场景的历元，以及参考航天器的轨道根数，给定任务初始时刻伴随航天器的相对位置为 $(-100, 0, 0)$ km，相对速度为 $(0, 0, 0)$ m/s。约束取值为：机动转移时间为 86400s，最大转移飞行角为 5° ，水滴绕飞圈数为 1，水滴绕飞构型为 $l = 2$ km， $h = 10$ km。

2.1 水滴型相对绕飞轨道设计

2.1.1. 动力学模型

当参考航天器运行于圆或近圆轨道，且伴随航天器与参考航天器距离很近时，两航天器之间的引力差可线性化处理。此时，伴随航天器的相对运动模型采用 C-W 方程（也称为 Hill 方程），得到其在参考航天器 VVLH 坐标系中的相对运动方程为

$$\begin{cases} \ddot{X} - 2n\dot{Z} = a_x \\ \ddot{Y} + n^2Y = a_y \\ \ddot{Z} + 2n\dot{X} - 3n^2Z = a_z \end{cases} \quad (1)$$

式中， n 为参考航天器轨道角速度； X 、 Y 、 Z 为伴随航天器在 VVLH 坐标系中

的位置； \dot{X} 和 \ddot{X} 分别表示伴随航天器在 VVLH 坐标系中 X 轴方向的速度和加速度，其他 2 轴的速度和加速度表示方式与此类似； a_x 、 a_y 、 a_z 分别为伴随航天器在 X 、 Y 、 Z 轴 3 个方向的控制加速度。

将 $\mathbf{S} = [X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}]^T$ 作为系统的状态量，伴随航天器状态量在无控情况下有以下解析形式：

$$\mathbf{S}(t) = \Phi(t, t_0) \mathbf{S}(t_0) \quad (2)$$

式中， $\Phi(t, t_0)$ 为状态转移矩阵，具体形式见式(3)； $\mathbf{S}(t_0)$ 为 t_0 时刻的状态量。

注意到平面内变量 X 、 Z 和平面外变量 Y 的运动是解耦的，本次水滴型相对绕飞轨道是在参考航天器轨道平面内的相对运动。

$$\Phi(t, t_0) = \begin{bmatrix} 1 & 0 & 6[n(t-t_0)-s] & \frac{4s-3n(t-t_0)}{n} & 0 & \frac{2(1-c)}{n} \\ 0 & c & 0 & 0 & \frac{s}{n} & 0 \\ 0 & 0 & 4-3c & \frac{2(c-1)}{n} & 0 & \frac{s}{n} \\ 0 & 0 & 6n(1-c) & 4c-3 & 0 & 2s \\ 0 & -ns & 0 & 0 & c & 0 \\ 0 & 0 & 3ns & -2s & 0 & c \end{bmatrix} \quad (3)$$

式中， $s = \sin(n(t-t_0))$ ， $c = \cos(n(t-t_0))$ 。

2.1.2. 转移轨道计算

已知伴随航天器初始相对位置 $(X_0, Y_0, Z_0)^T$ ，抵近目标相对位置 $(X_w, Y_w, Z_w)^T$ （绕飞段起点）。若给定机动抵近任务时长 T_w 和相邻导航点之间的最大转移飞行角 A_m ，则能确定伴随航天器进入绕飞段的规划参数。

当参考航天器轨道为近圆轨道时，拱线抵近段子段数量为 W_N ，其表达式如下：

$$W_N = 1 + \left\lfloor \frac{T_w}{T_0} \cdot \frac{360}{A_m} \right\rfloor \quad (4)$$

其中， T_0 为参考航天器轨道周期， $\lfloor \cdot \rfloor$ 为向下取整符号。

此时，所有段的停止条件均为时间 Δt_1 ，其表达式如下：

$$\Delta t_1 = \frac{T_w}{W_N} \quad (5)$$

第 W_N 段的停止条件为时间，即进入绕飞的剩余时间 Δt_2 ，见下式：

$$\Delta t_2 = T_w - T_r \quad (6)$$

其中， T_r 为前 $(W_N - 1)$ 段的飞行时间。

已知拱线抵近段子段的数量则可以计算出每一段的终端约束，即瞄准序列段的瞄准点 $(X_i, Y_i, Z_i)^T$ ，见下式：

$$\begin{cases} X_i = X_0 + i(X_w - X_0)/W_N \\ Y_i = Y_0 + i(Y_w - Y_0)/W_N \\ Z_i = Z_0 + i(Z_w - Z_0)/W_N \end{cases}, \quad i = 1, 2, \dots, W_N \quad (7)$$

至此，就可以得到每一个拱线抵近段子段的初、末位置矢量；初速度矢量即为伴随航天器机动时刻的相对速度，末速度矢量可以设为零矢量，后面通过迭代计算将其修正为下一个拱线抵近段子段的初始速度矢量。

进一步，对于有脉冲机动的相对运动方程，利用C-W方程可以表示为下式：

$$S(t) - \Phi(t, t_0)S(t_0) = F\Delta V \quad (8)$$

其中， $\Delta V = [(\Delta V_1)^T, (\Delta V_2)^T, \dots, (\Delta V_N)^T]^T$ ， $F = [\Phi_V(t, t_1), \Phi_V(t, t_2), \dots, \Phi_V(t, t_N)]$ ，

其分别表示 N 次速度脉冲的脉冲矢量和其对应的状态转移矩阵。

将式(8)变形可得下式：

$$\Delta V = F^{-1}[S(t) - \Phi(t, t_0)S(t_0)] \quad (9)$$

式(9)中有6个方程，当 $N = 2$ 时， ΔV 中包含6个未知脉冲分量（初始位置速度脉冲和目标位置速度脉冲各有3个方向的速度脉冲分量），联立可以求得速度脉冲的唯一解。

当考虑摄动力影响时，以各拱线抵近段子段的初位置和末位置为约束，利用非线性求解器迭代求解即可得到相应的速度脉冲。上述计算结果可以作为求解器的初值，加快解的收敛速度，减少计算资源消耗。

2.1.3. 绕飞轨道的计算

当伴随航天器的半长轴与参考航天器不同时，即 $\dot{X}_0 \neq 2nZ_0$ 时，伴随航天器 X 轴方向的位置含随时间变化的漂移项，若不加控制，相对距离越来越大。若在漂移轨迹自相交点施加径向控制，则形成水滴形状的封闭绕飞构型。

前文已经提到 l 和 h 参数可以描述水滴的绕飞构型，除此之外，伴随航天器绕飞周期 T 也可以作为描述参数。但这 3 个参数只有 2 个参数是独立的，即任意 1 个参数可以由另外 2 个参数计算表示。若选取 l 和 T 为独立参数，则 h 可表示为

$$h = \frac{3nTl\cos(nT/2) - 8l\sin(nT/2)}{3nT - 8\sin(nT/2)} \quad (10)$$

选 B 点为初始点，伴随航天器的初始状态变量为

$$[X_0, Z_0, \dot{X}_0, \dot{Z}_0]^T = \left[0, -l, \frac{6nl(2\sin(nT/2) - nT)}{3nT - 8\sin(nT/2)}, 0 \right]^T \quad (11)$$

水滴绕飞构型的解析式为

$$\begin{cases} \begin{bmatrix} X(t') \\ Z(t') \end{bmatrix} = \begin{bmatrix} \frac{12nt'\sin(nT/2) - 6nT\sin(nt')}{3nT - 8\sin(nT/2)} l \\ \frac{8\sin(nT/2) - 3nT\cos(nt')}{3nT - 8\sin(nT/2)} l \end{bmatrix}, & 0 \leq t' < T/2 \\ \begin{bmatrix} X(t') \\ Z(t') \end{bmatrix} = \begin{bmatrix} \frac{12n(t'-T)\sin(nT/2) - 6nT\sin[n(t'-T)]}{3nT - 8\sin(nT/2)} l \\ \frac{8\sin(nT/2) - 3nT\cos[n(t'-T)]}{3nT - 8\sin(nT/2)} l \end{bmatrix}, & T/2 \leq t' < T \end{cases} \quad (12)$$

伴随航天器在 A 点处需要施加的径向脉冲为

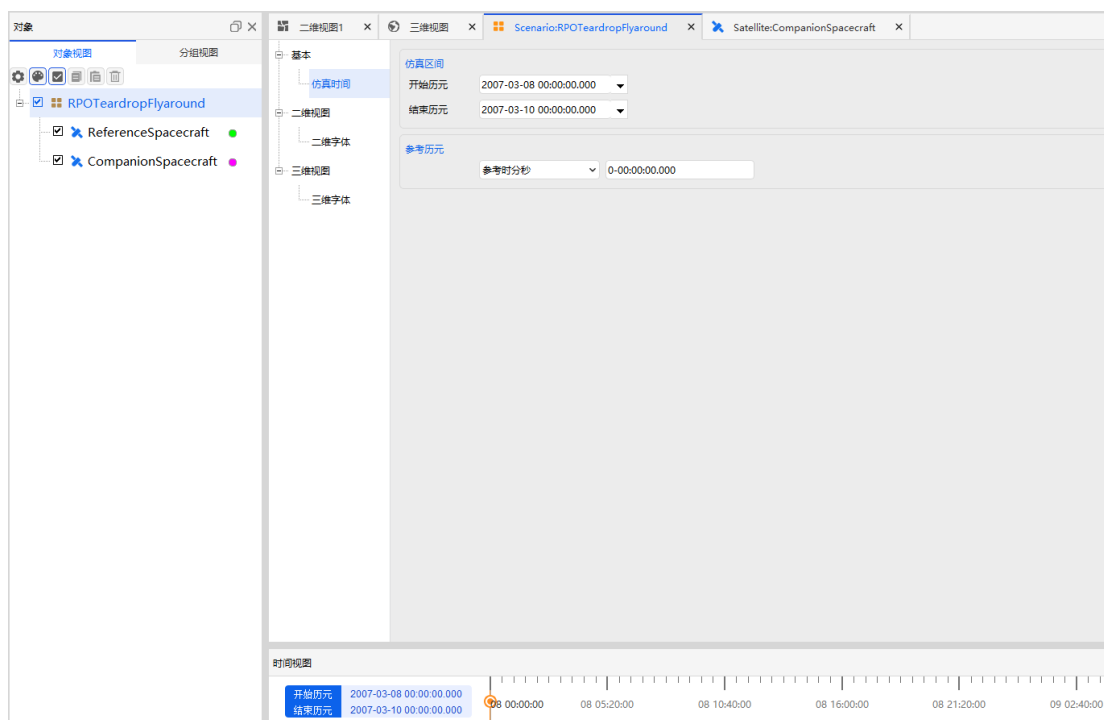
$$\Delta V_z = \frac{6n^2Tl\sin(nT/2)}{3nT - 8\sin(nT/2)} \quad (13)$$

此时的时刻应为 $t' = T/2$ 。

设绕飞周期 $T = kT_0$ ，则 k 为绕飞周期系数。对于水滴绕飞，当 k 取值较小时，A、B 两点位于 X 轴同侧；要形成绕飞构型，要求 A、B 位于 X 轴两侧，即 l 和 h 异号，计算可得 k 的取值范围为 $(0.406, 1.4067)$ 。当 $k < 0.406$ ，A、B 点位于 X 轴同侧，形成悬停构型。当 k 逼近 0.406 时，式(10)和式(13)中分母逼近零，出现奇点，绕飞构型半径和脉冲特别大。注意到，对于慢速绕飞 ($1.4067 > k > 1$)，水滴绕飞构型的形状呈现出倒立的“心”形；随着 k 的不断增大，水滴尖点 A 不断上移，直至逼近甚至超越 X 轴。当 $k > 1.4067$ 时，A、B 两点再次同侧，绕飞构型形状比较复杂，一个完整的构型中伴随航天器实际上绕参考航天器飞行了 2 周，不符合慢速绕飞要求。

2.2 创建水滴型相对绕飞轨道任务场景和对象

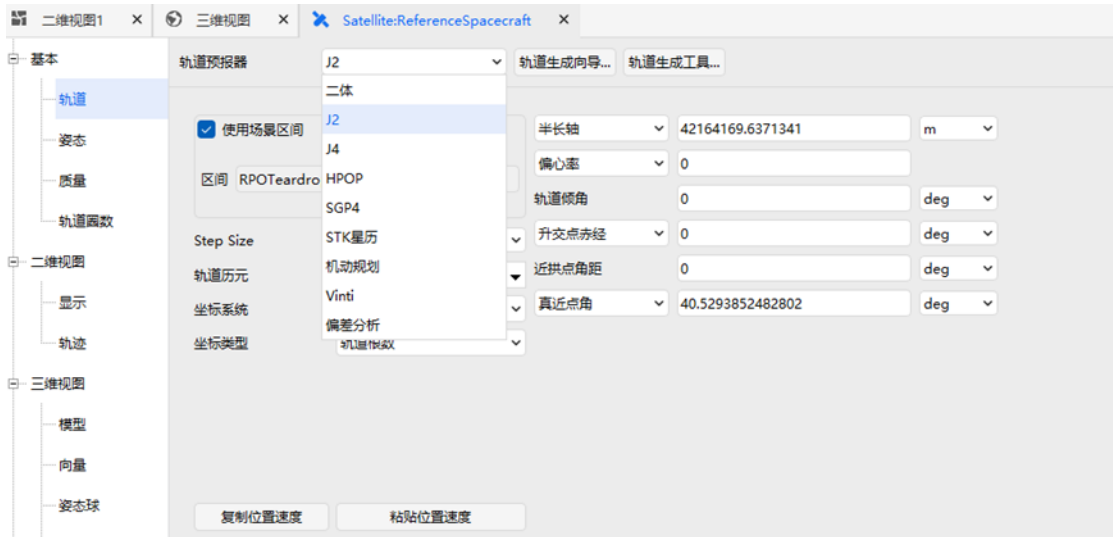
安装并运行 ATK.exe，新建一个想定“RPOTeardropFlyaround”。设置开始历元为 2007-03-08 00:00:00.000，结束历元为 2007-03-10 00:00:00.000；利用卫星轨道生成向导插入地球静止轨道卫星，将其作为参考航天器，命名为“ReferenceSpacecraft”；再插入一颗卫星，将其作为伴随航天器，命名为“CompanionSpacecraft”。



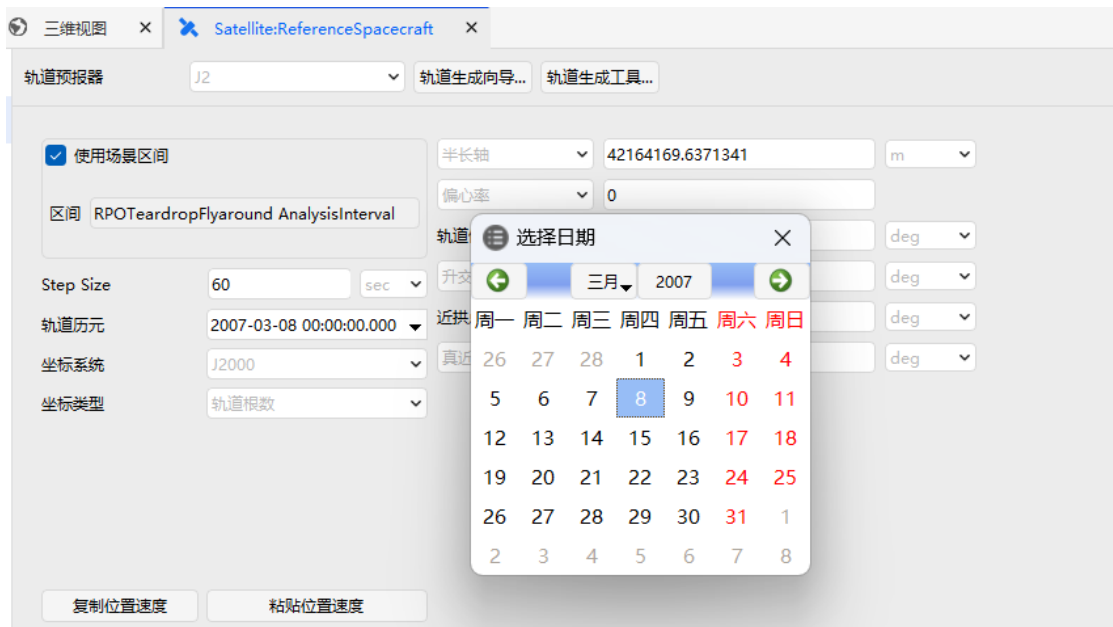
2.3 设计水滴型相对绕飞轨道

2.3.1. 参考航天器参数配置

选择“ReferenceSpacecraft”，在轨道预报器下拉菜单中选择 J2 预报方式。

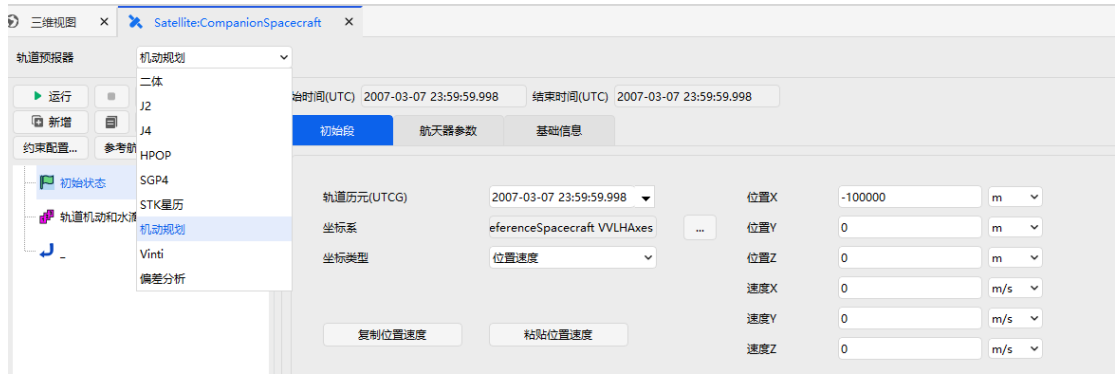


再将参考航天器的轨道历元设置为与场景开始历元相同，并应用。

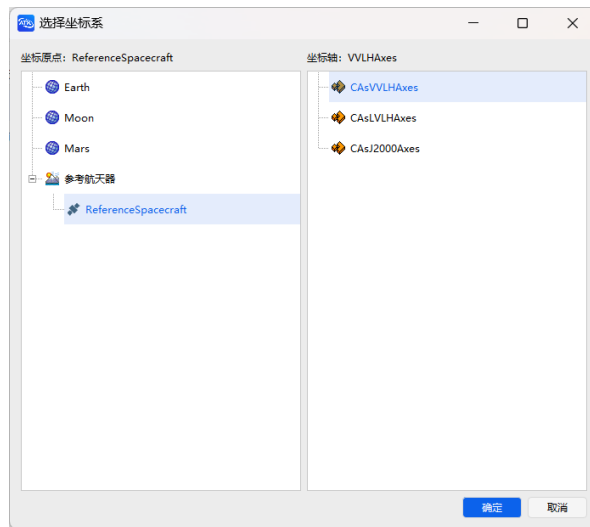


2.3.2. 伴随航天器初始参数配置

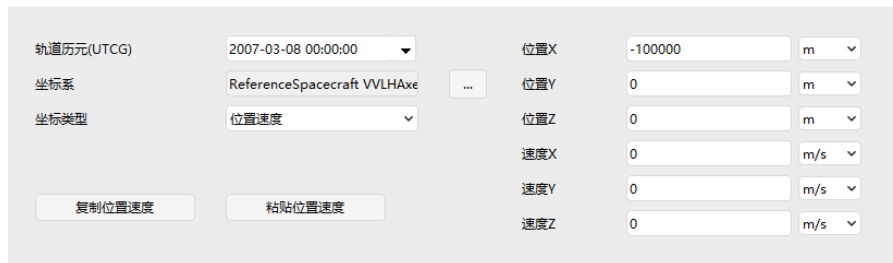
选择“CompanionSpacecraft”，在轨道预报器下拉菜单中选择机动规划预报方式。



将初始段“InitialState”命名为“初始状态”，选择“参考航天器”为“ReferenceSpacecraft”。在右侧“初始段”界面，设置参考系为“ReferenceSpacecraft”的“CASVVLH Axes”坐标系。

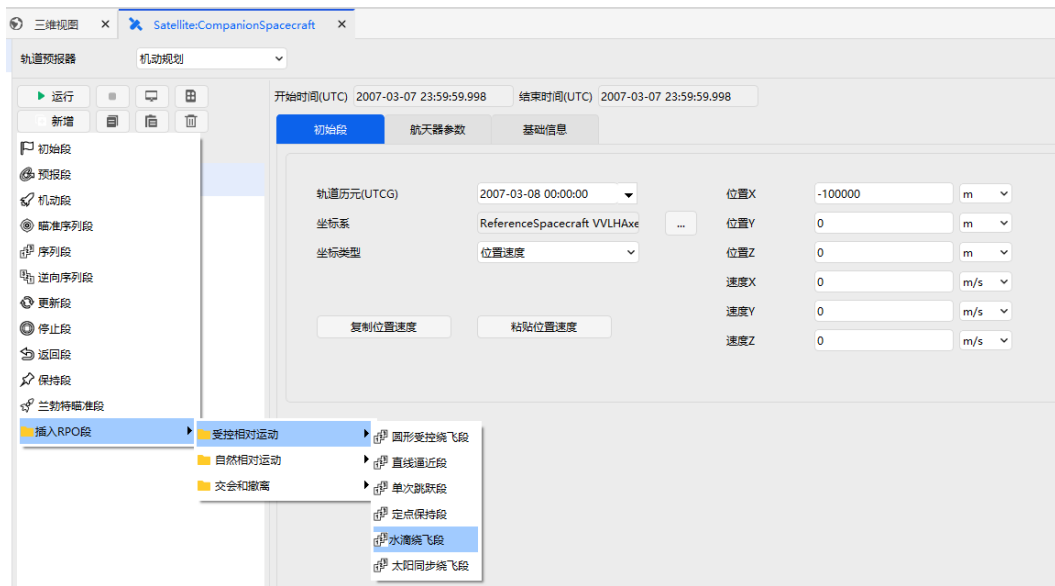


上述步骤表明建立了伴随航天器相对于参考航天器的坐标系。其中，X轴指向参考航天器速度的正方向，Y轴指向参考航天器轨道平面的负法线方向，Z轴正方向指向地心。设置伴随航天器的初始相对位置：

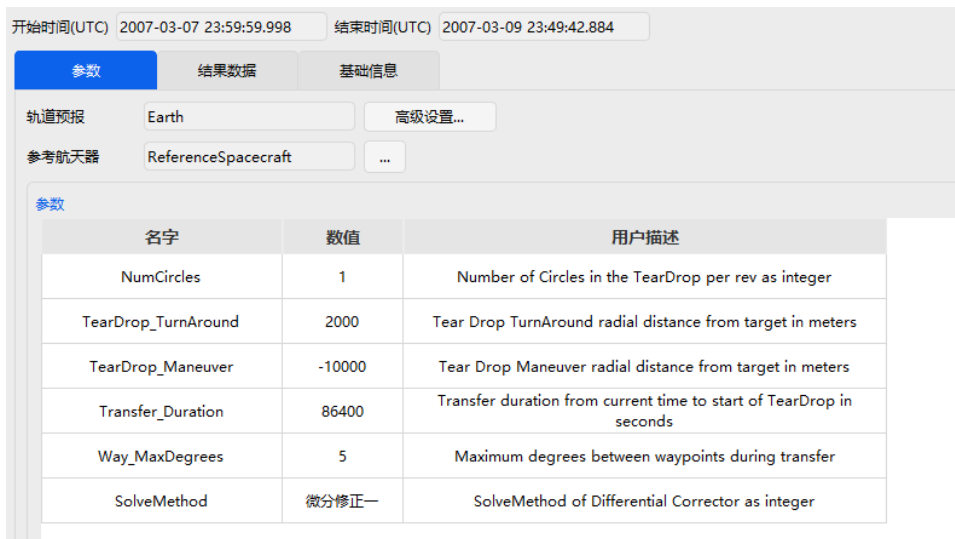


2.3.3. 伴随航天器水滴绕飞轨道参数设置

在机动规划界面选择“新增”、“插入 RPO 段”、“受控相对运动”、“水滴绕飞段”，并重命名为“轨道机动和水滴绕飞”。



随后设置“参考航天器”为“ReferenceSpacecraft”，并设置水滴受控绕飞的参数，“NumCircles”为 1，“TearDrop_TurnAround”为 2000m，“TearDrop_Maneuver”为 -10000m，“Transfer_Duration”为 86400s，“Way_MaxDegrees”为 5°，“SolveMethod”选择微分修正一。

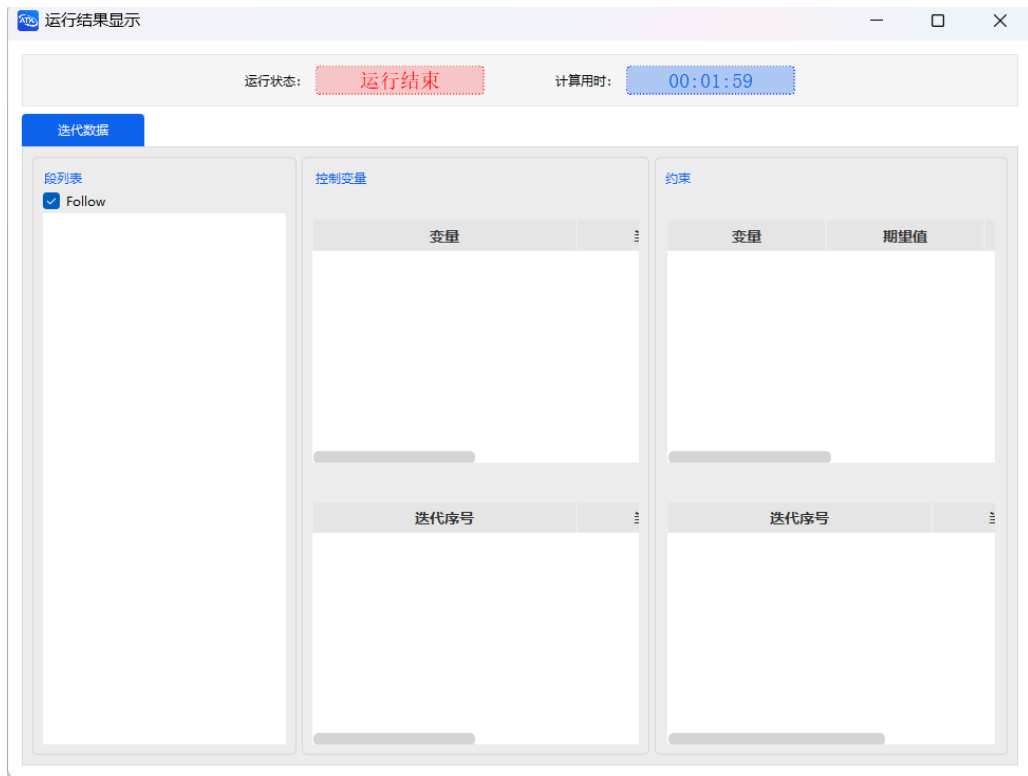


上述参数即对应场景设计中的约束变量。

2.4 水滴型相对绕飞轨道仿真运行与结果显示

2.4.1. 仿真计算和显示设置

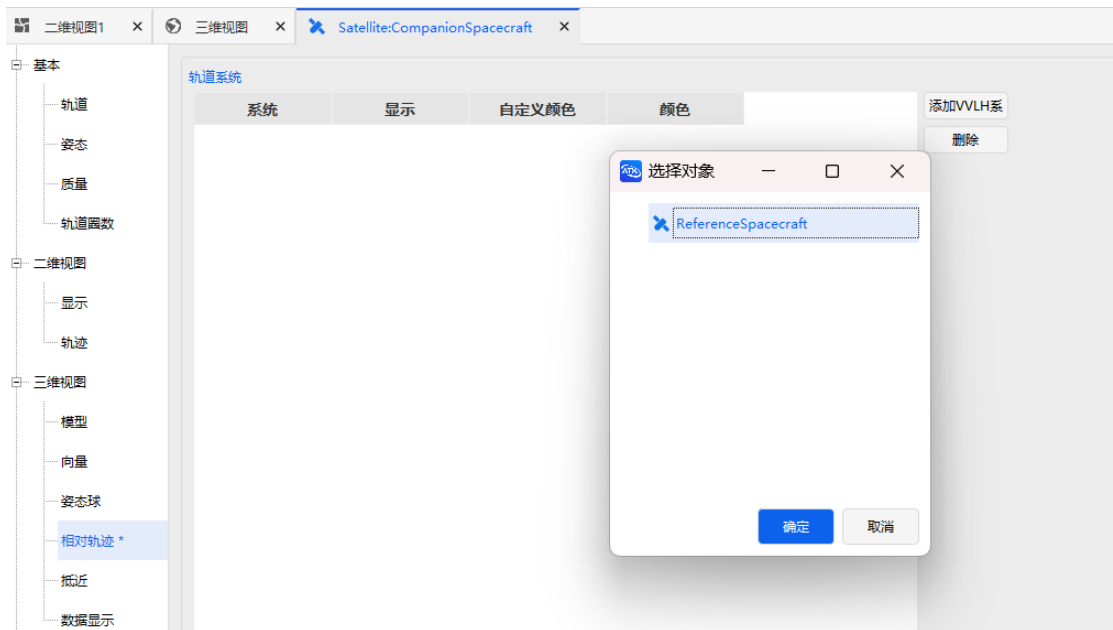
伴随航天器的参数配置完毕后，点击机动规划窗口的“运行”，等待计算结束：



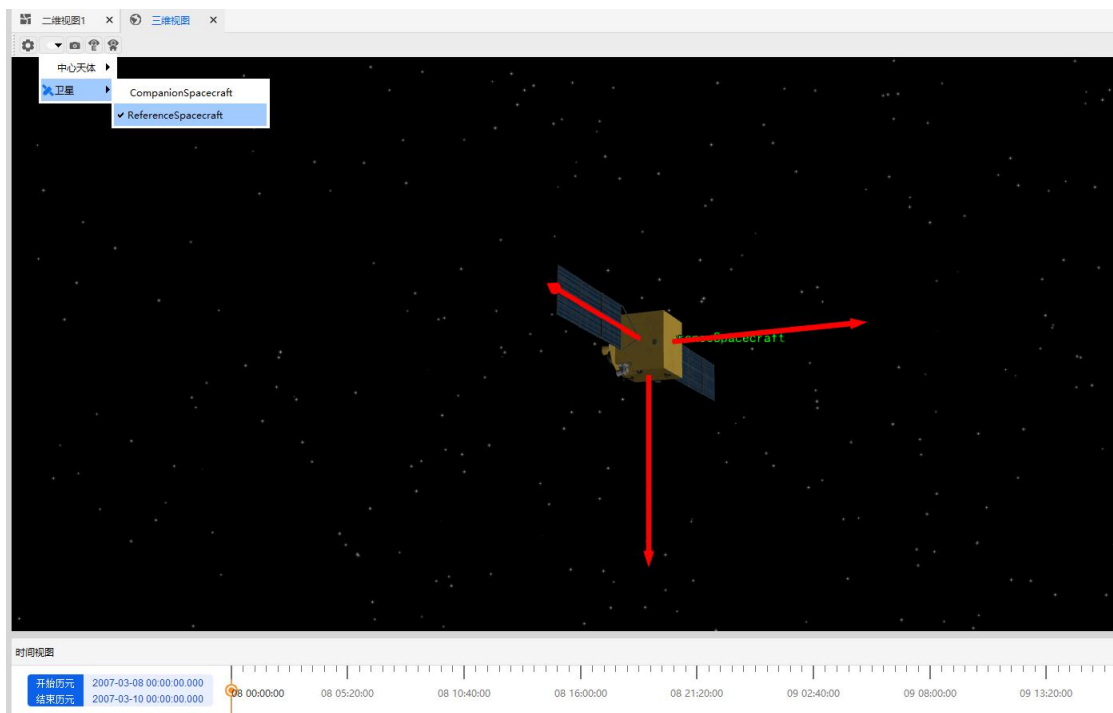
计算结束后点击“结果数据”即可查看轨道转移和绕飞过程施加的脉冲矢量及时间。分析数据可以得到，在轨道转移初始点、转移段和绕飞段的衔接点、水滴尖点施加的脉冲较大（1m/s~2m/s），整个转移段施加脉冲 64 次，但单次脉冲量不大（0.3 m/s 左右）。具体结果如下图所示。

参数	结果数据	基础信息				
名字	UTC	-Vx(m/s)	-Vy(m/s)	-Vz(m/s)	-V(m/s)	
FMW.TargetSequence1.DifferentialCorrector.Maneuver	2007-03-07 23:59:59.998	1.511208714858444	-0.000014419...	0.104995082...	1.514852031326560	
FMW.TargetSequence2.DifferentialCorrector.Maneuver	2007-03-08 00:19:53.059	0.009127332510064	-0.000007812...	0.2561171666...	0.256279752069841	
FMW.TargetSequence3.DifferentialCorrector.Maneuver	2007-03-08 00:39:46.120	0.004013832649726	-0.000022946...	0.2570986889...	0.257130020125932	
FMW.TargetSequence4.DifferentialCorrector.Maneuver	2007-03-08 00:59:39.182	0.004002386606191	-0.000013657...	0.2576056266...	0.257636717387523	
FMW.TargetSequence5.DifferentialCorrector.Maneuver	2007-03-08 01:19:32.243	0.004017282428758	-0.000017100...	0.2581329803...	0.258164239185811	
FMW.TargetSequence6.DifferentialCorrector.Maneuver	2007-03-08 01:39:25.305	0.004024816627798	-0.000015064...	0.2586495588...	0.258680872236247	
FMW.TargetSequence7.DifferentialCorrector.Maneuver	2007-03-08 01:59:18.366	0.004033792520776	-0.000010523...	0.2591631917...	0.259194582427985	
FMW.TargetSequence8.DifferentialCorrector.Maneuver	2007-03-08 02:19:11.427	0.004041382473565	-0.000014371...	0.2597003382...	0.259731782134608	
FMW.TargetSequence9.DifferentialCorrector.Maneuver	2007-03-08 02:39:04.489	0.004049356360204	-0.000009264...	0.2602141850...	0.260245690585778	
FMW.TargetSequence10.DifferentialCorrector.Maneuver	2007-03-08 02:58:57.550	0.004054838976456	-0.000010834...	0.2607344599...	0.260765987874788	
FMW.TargetSequence11.DifferentialCorrector.Maneuver	2007-03-08 03:18:50.612	0.004072971722802	-0.000007136...	0.2612665555...	0.261298301198258	
FMW.TargetSequence12.DifferentialCorrector.Maneuver	2007-03-08 03:38:43.673	0.004068569287671	-0.000007149...	0.2617899079...	0.261821521658167	
FMW.TargetSequence13.DifferentialCorrector.Maneuver	2007-03-08 03:58:36.735	0.004090180764752	-0.000006043...	0.2623189322...	0.262350818214848	
FMW.TargetSequence14.DifferentialCorrector.Maneuver	2007-03-08 04:18:29.796	0.004077974138265	-0.000004646...	0.2628404536...	0.262872086745776	
FMW.TargetSequence15.DifferentialCorrector.Maneuver	2007-03-08 04:38:22.857	0.004109203074768	-0.000003528...	0.2633878566...	0.263419909250540	
FMW.TargetSequence16.DifferentialCorrector.Maneuver	2007-03-08 04:58:15.919	0.004098254628451	-0.000002471...	0.2639017420...	0.263933561965282	
FMW.TargetSequence17.DifferentialCorrector.Maneuver	2007-03-08 05:18:08.980	0.004110921726098	-0.000001174...	0.2644305680...	0.264462520955841	
FMW.TargetSequence18.DifferentialCorrector.Maneuver	2007-03-08 05:38:02.042	0.004136539123588	0.0000000166...	0.2649795987...	0.265011884076597	
FMW.TargetSequence19.DifferentialCorrector.Maneuver	2007-03-08 05:57:55.103	0.004133772441985	0.0000008182...	0.2655002768...	0.265532455842397	
FMW.TargetSequence20.DifferentialCorrector.Maneuver	2007-03-08 06:17:48.164	0.004126884594856	0.0000018638...	0.2660393062...	0.266071313117744	
FMW.TargetSequence21.DifferentialCorrector.Maneuver	2007-03-08 06:37:41.226	0.004150372329030	0.0000026177...	0.2665771244...	0.266609431365571	
FMW.TargetSequence22.DifferentialCorrector.Maneuver	2007-03-08 06:57:34.287	0.00415936919892	0.0000031985...	0.2671174396...	0.267149821004046	
FMW.TargetSequence23.DifferentialCorrector.Maneuver	2007-03-08 07:17:27.349	0.004161792599535	0.0000051252...	0.2676365866...	0.267668943064505	
FMW.TargetSequence24.DifferentialCorrector.Maneuver	2007-03-08 07:37:20.410	0.004164074862652	0.0000039685...	0.2681965567...	0.26822880945546	
FMW.TargetSequence25.DifferentialCorrector.Maneuver	2007-03-08 07:57:13.471	0.004196971240730	0.0000056643...	0.2687273232...	0.268760095375096	
FMW.TargetSequence26.DifferentialCorrector.Maneuver	2007-03-08 08:17:06.533	0.004186595182711	0.0000058710...	0.2692676862...	0.269300231124477	
FMW.TargetSequence27.DifferentialCorrector.Maneuver	2007-03-08 08:36:59.594	0.004184716755807	0.0000064155...	0.2698191999...	0.269851649138150	
FMW.TargetSequence28.DifferentialCorrector.Maneuver	2007-03-08 08:56:52.655	0.004210791421107	0.0000063167...	0.2703535740...	0.270386363914711	
FMW.TargetSequence29.DifferentialCorrector.Maneuver	2007-03-08 09:16:45.717	0.004212116267773	0.0000070822...	0.2709075699...	0.2709400313373849	
FMW.TargetSequence30.DifferentialCorrector.Maneuver	2007-03-08 09:36:38.778	0.004220760704240	0.0000054720...	0.2714316321...	0.27146446675300	
FMW.TargetSequence31.DifferentialCorrector.Maneuver	2007-03-08 09:56:31.840	0.004230613358459	0.0000066748...	0.2719991393...	0.272032038441949	
FMW.TargetSequence32.DifferentialCorrector.Maneuver	2007-03-08 10:16:24.901	0.004237258108969	0.0000075872...	0.2725391403...	0.272572077513093	
FMW.TargetSequence33.DifferentialCorrector.Maneuver	2007-03-08 10:36:17.962	0.004245458476071	0.0000054780...	0.2730904413...	0.273123439276930	
FMW.TargetSequence34.DifferentialCorrector.Maneuver	2007-03-08 10:56:11.024	0.004250483333057	0.0000049346...	0.2736374434...	0.273670453432104	
FMW.TargetSequence35.DifferentialCorrector.Maneuver	2007-03-08 11:16:04.085	0.004269095903219	0.0000054814...	0.2741869527...	0.274220185692509	
FMW.TargetSequence36.DifferentialCorrector.Maneuver	2007-03-08 11:35:57.146	0.004270196597580	0.0000044150...	0.2747505493...	0.274783731231566	
FMW.TargetSequence37.DifferentialCorrector.Maneuver	2007-03-08 11:55:50.208	0.004269792450517	0.0000048225...	0.2752815885...	0.275314700181306	
FMW.TargetSequence38.DifferentialCorrector.Maneuver	2007-03-08 12:15:43.269	0.004291668851582	0.0000017432...	0.2758543160...	0.275887698377722	
FMW.TargetSequence39.DifferentialCorrector.Maneuver	2007-03-08 12:35:36.331	0.004295842044510	0.0000026609...	0.2764021613...	0.276435542337335	
FMW.TargetSequence40.DifferentialCorrector.Maneuver	2007-03-08 12:55:29.392	0.004300201187143	0.0000015833...	0.2769542834...	0.27698766562245	
FMW.TargetSequence41.DifferentialCorrector.Maneuver	2007-03-08 13:15:22.453	0.004312922218847	0.000003340...	0.2775193668...	0.277552878356810	
FMW.TargetSequence42.DifferentialCorrector.Maneuver	2007-03-08 13:35:15.515	0.004319341873614	-0.000000964...	0.2780844192...	0.278117962262486	
FMW.TargetSequence43.DifferentialCorrector.Maneuver	2007-03-08 13:55:08.576	0.004325439630972	-0.000001325...	0.2786254359...	0.27865900408041	
FMW.TargetSequence44.DifferentialCorrector.Maneuver	2007-03-08 14:15:01.637	0.004344295749366	-0.000003341...	0.2791933736...	0.279227170577450	
FMW.TargetSequence45.DifferentialCorrector.Maneuver	2007-03-08 14:34:54.699	0.004327890825811	-0.000003655...	0.2797547303...	0.279788205220308	
FMW.TargetSequence46.DifferentialCorrector.Maneuver	2007-03-08 14:54:47.760	0.004368903562134	-0.000005322...	0.2803192817...	0.280353325406920	
FMW.TargetSequence47.DifferentialCorrector.Maneuver	2007-03-08 15:14:40.821	0.004338953582769	-0.000006668...	0.2808899930...	0.280923503387923	
FMW.TargetSequence48.DifferentialCorrector.Maneuver	2007-03-08 15:34:33.883	0.004383576129706	-0.000007250...	0.2814388453...	0.281472981750360	
FMW.TargetSequence49.DifferentialCorrector.Maneuver	2007-03-08 15:54:26.944	0.004371227132331	-0.000009739...	0.2820129881...	0.282046863530717	
FMW.TargetSequence50.DifferentialCorrector.Maneuver	2007-03-08 16:14:20.006	0.004389286552274	-0.000010573...	0.2825783585...	0.282612446074980	
FMW.TargetSequence51.DifferentialCorrector.Maneuver	2007-03-08 16:34:13.067	0.004387513491614	-0.000009909...	0.2831474524...	0.28318443990759	
FMW.TargetSequence52.DifferentialCorrector.Maneuver	2007-03-08 16:54:06.128	0.004394256294639	-0.000012206...	0.2837164025...	0.283750430285059	
FMW.TargetSequence53.DifferentialCorrector.Maneuver	2007-03-08 17:13:59.190	0.004419716473206	-0.000016501...	0.2842791888...	0.284313544198991	
FMW.TargetSequence54.DifferentialCorrector.Maneuver	2007-03-08 17:33:52.251	0.004425487393292	-0.000011795...	0.2848647889...	0.284899162990010	
FMW.TargetSequence55.DifferentialCorrector.Maneuver	2007-03-08 17:53:45.313	0.004410434455677	-0.000016434...	0.2854267989...	0.285460872550863	
FMW.TargetSequence56.DifferentialCorrector.Maneuver	2007-03-08 18:13:38.374	0.004437857967276	-0.000017015...	0.2860017285...	0.286036157836671	
FMW.TargetSequence57.DifferentialCorrector.Maneuver	2007-03-08 18:33:31.435	0.004440162817107	-0.000017149...	0.2865742344...	0.286608630707590	
FMW.TargetSequence58.DifferentialCorrector.Maneuver	2007-03-08 18:53:24.497	0.004452848378177	-0.000018947...	0.2871547256...	0.287189248846429	
FMW.TargetSequence59.DifferentialCorrector.Maneuver	2007-03-08 19:13:17.558	0.004460243379903	-0.000020027...	0.2877253153...	0.287759884757434	
FMW.TargetSequence60.DifferentialCorrector.Maneuver	2007-03-08 19:33:10.620	0.004458280193896	-0.000017140...	0.2883093899...	0.288343858761431	
FMW.TargetSequence61.DifferentialCorrector.Maneuver	2007-03-08 19:53:03.681	0.004478305987071	-0.000022177...	0.2888768454...	0.288911556633325	
FMW.TargetSequence62.DifferentialCorrector.Maneuver	2007-03-08 20:12:56.742	0.004485665133486	-0.000020510...	0.2894667049...	0.289501459216414	
FMW.TargetSequence63.DifferentialCorrector.Maneuver	2007-03-08 20:32:49.804	0.004488836466376	-0.000020962...	0.2900443269...	0.290079061140288	
FMW.TargetSequence64.DifferentialCorrector.Maneuver	2007-03-08 20:52:42.865	0.004495269663823	-0.000021237...	0.2906258123...	0.290660576444783	
Hop.TargetSequence1.DifferentialCorrector.Maneuver	2007-03-08 21:12:35.926	-1.213622463697835	-0.000103794...	0.5179866381...	1.319541455379173	
TTD.TargetSequence1.DifferentialCorrector.Maneuver	2007-03-08 23:59:59.997	-0.161307961540150	0.0000888408...	-0.352573544...	0.387722027744992	
ITDM.TargetSequence1.DifferentialCorrector1.Maneuver	2007-03-09 07:57:13.898	0.000624649592039	0.0000000000...	-2.040773053...	2.040773154957450	

接下来在伴随航天器的属性设置窗口，选择“三维视图”下的“相对轨迹”设置，在右侧选择“添加 VVLH 系”，添加“ReferenceSpacecraft”对象。可以根据需要设置相对轨迹颜色，这里采用默认颜色。

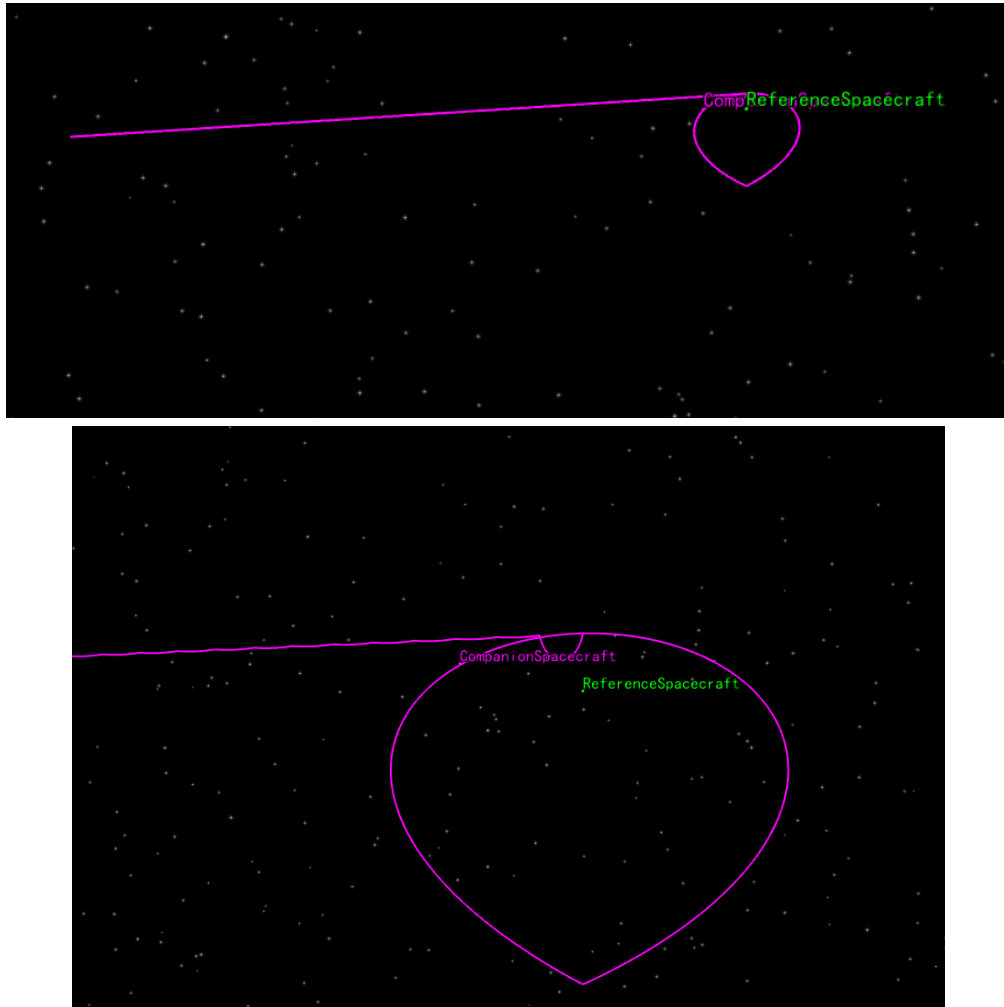


回到“三维视图”界面，选择左上角的“局部视点”，然后依次选择“卫星”，“ReferenceSpacecraft”，即可将视角固定在“ReferenceSpacecraft”上。



2.4.2. 仿真场景显示

点击“开始”按钮，即可观察参考航天器和伴随航天器的相对运动轨迹。为了使相对轨迹更加明显，可以将绝对轨迹显示关闭，RPO 水滴绕飞轨迹展示如下。



图中紫红色轨迹是伴随航天器相对于参考航天器的轨迹，绿色点为相对坐标系的坐标原点，即参考航天器的位置，其相对位置保持不变。图中包含了伴随航天器从初始位置机动至水滴绕飞初始点的转移段轨迹，以及水滴受控绕飞过程。仿真场景三维动图如下（双击查看）



3. ATK 特定问题建模—空间站与空间目标的接近分析

内容简介

随着空间探测与开发活动的日趋频繁，地球附近的太空物体日益增多，在轨航天器与空间目标发生接近与碰撞的风险提高。而通过对航天器与其他空间目标进行接近分析，能够筛选出可能接近航天器的威胁目标，提升对大量空间目标进一步分析时的计算效率，提高碰撞预警与规避能力，保障航天器以及航天员的安全，因此开展航天器与空间目标的接近分析有重要意义。

空间目标接近分析重点研究主目标与威胁目标之间相对距离的变化规律。在已知两目标的轨道数据、距离门限和分析起止时间的情况下，计算相对距离小于距离门限的时间段，两目标最接近时刻，以及对应的最接近距离、相对速度、接近角等信息。如图所示：

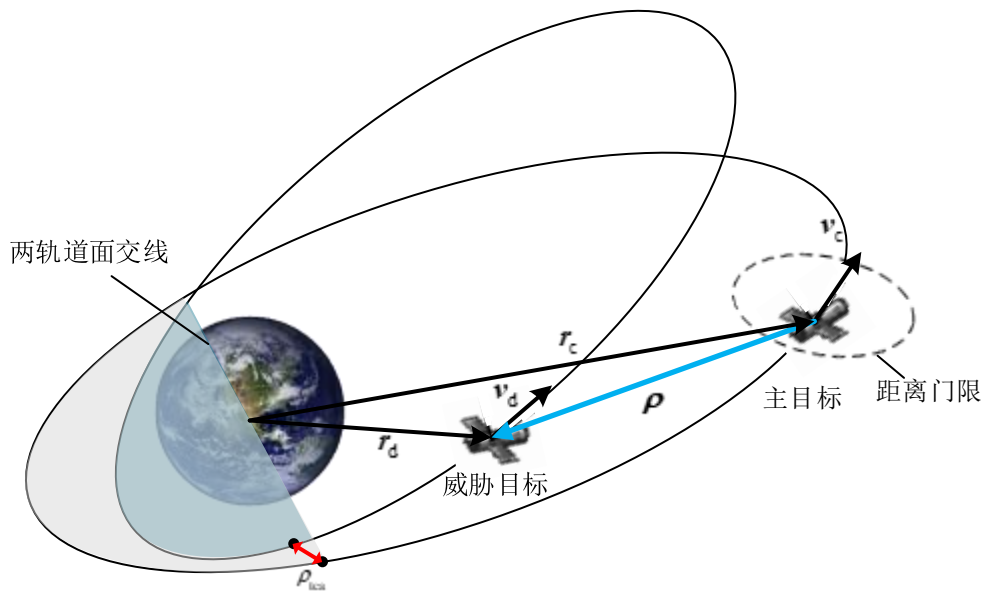


图 0-1 接近分析示意图

式中， r_c 、 v_c 分别为主目标的位置矢量与速度矢量， r_d 、 v_d 为威胁目标的位置速度矢量， ρ 为主目标与威胁目标之间的相对位置矢量， ρ_{ica} 为最小相对距离。

在本案例中，从 space track 官网上下载 2024 年 1 月 1 日至 2024 年 3 月 11 日所有空间目标的 TLE 数据。选择其中的空间站作为主目标，其他作为威胁目标的 TLE 数据。使用基于 ATK 仿真场景的接近分析模块，设置距离门限为 2km，分析 2024 年 3 月 12 日至 19 日的最小接近距离、接近时刻与相对位置、速度等参数。

表 1 接近分析输入输出参数

变量类型	参数
输入变量	距离门限 2km
输入变量	空间站 TLE 数据
输入变量	其他空间目标的 TLE 数据
输入变量	接近分析起止时间（2024 年 3 月 12 日至 2024 年 3 月 19 日）
输出变量	接近距离
输出变量	接近时刻
输出变量	相对位置
输出变量	相对速度
输出变量	...

3.1 接近分析数值方法

数值法以太空目标某时刻或一段时间内的轨道星历为基础，利用轨道模型求取指定时刻的目标位置速度信息，通过筛选、插值、拟合、多项式求根等方法获得接近距离、相对速度等接近信息。

3.1.1. 轨道历元日期筛选

历元日期筛选是去除轨道历元距离分析起始时刻太远的目标，由于轨道预报误差的存在，太长时间的轨道预报结果置信度低，从而使接近分析结果可信度低。提取空间目标 TLE 数据中的历元时刻信息，轨道历元日期的筛选条件为：

$$|t - t_0| \leq t_{\text{tol}} \quad (14)$$

式中， t 为空间目标 TLE 数据对应的儒略日； t_0 为分析起始时刻的儒略日； t_{tol} 为设置的时间阈值，一般取 30.0。

3.1.2. 轨道远地点-近地点高度筛选

远地点-近地点筛选是指当其中一个目标远地点高度低于另一个目标近地点高度的安全距离时所进行的筛除操作。因为运行在这种轨道上的两个目标在自然情况下不可能发生碰撞。考虑目标之间的相对几何关系，计算起始时刻航天器和空间目标的远地点及近地点高度，将不可能发生碰撞的太空目标排除掉，筛选的标准为：

$$q_c - Q_d > D_{\text{tol}} \text{ 或 } q_d - Q_c > D_{\text{tol}} \quad (15)$$

式中， Q_c 、 q_c 分别表示主目标的远地点高度和近地点高度， Q_d 、 q_d 分别表示威胁目标的远地点高度和近地点高度， D_{tol} 为设定的距离阈值，默认取 50000m。

3.1.3. 相对距离筛选

对于通过以上两项筛选的空间目标，在接近分析关注的时间区间内以一定步长对主目标位置 $\mathbf{r}_c(t)$ 和威胁目标位置 $\mathbf{r}_d(t)$ 之间的相对位置矢量 $\boldsymbol{\rho}(t)$ 随时间的变化进行分析，进而识别出最接近时刻与接近距离等信息。如果每个时间段内

两目标间的相对距离大于给定的阈值，则认为两目标在该时间段内不可能发生碰撞。当然，这里的阈值必须保证两目标持续到下一个时间段均不发生碰撞，因此该阈值需随着时间段的增加而增大。

筛选脱靶距离。当相对距离小于距离门限时，需计算它与航天器的脱靶距离 $R_{\text{miss}}(t)$ ，并与航天器的联合包络体半径 R_{cr} 比较，当脱靶距离大于联合包络体半径，则两目标在当前时间段内不会发生碰撞，脱靶距离的计算方法为：

$$R_{\text{miss}}(t) = \sqrt{\rho(t)^2 - \left(\frac{\boldsymbol{\rho}^T(t)\dot{\boldsymbol{\rho}}(t)}{\|\dot{\boldsymbol{\rho}}(t)\|} \right)^2} \quad (16)$$

式中， $\rho(t)$ 为两目标间的相对距离值， $\boldsymbol{\rho}(t)$ 为两目标间的相对距离矢量， $\dot{\boldsymbol{\rho}}(t)$ 为相对速度矢量。

筛选相对距离。本文基于飞行动力学规律，进行多步相对距离筛选，由于环绕地球飞行物体的飞行速度不会超过逃逸中心天体的速度 v_{esc} ，所以两个目标间的相对速度不会超过两倍的逃逸速度，即可定义一个保守的脱靶距离。在时间段 $[t_i, \Delta t + t_{i+1}]$ 内，若 t 时刻的相对距离值 $\rho(t)$ 大于 $2v_{\text{esc}}\Delta t$ ，则两目标在该步长内不发生碰撞。

$$R_{\text{th}} = R_{\text{cr}} + 2v_{\text{esc}}\Delta t \quad (17)$$

式中， $v_{\text{esc}} = \sqrt{2\mu/R}$ ， μ 为中心天体引力常数， R 为中心天体半径， Δt 为仿真分析步长，默认取 600sec。

用类似方式考虑轨道加速度的影响，对环绕地球的物体，其轨道加速度应小于海平面的重力加速度 g_0 。根据运动方程，最大相对加速度 $2g_0$ 在 $[t_i, \Delta t + t_{i+1}]$ 时间段内引起的最大轨迹弯曲效应可表示为 $d_{\text{acc}} = g_0\Delta t^2$ 。考虑加速度影响的安全距离阈值可表示为：

$$R_{\text{acc}} = R_{\text{cr}} + d_{\text{acc}} = R_{\text{cr}} + g_0\Delta t^2 \quad (18)$$

再次，考虑两目标真实的相对速度，在前序筛选方法的基础上进一步精炼距离阈值，表示为：

$$R_{\text{th, fine}} = R_{\text{acc}} + \left| \frac{\boldsymbol{\rho}^T(t)\dot{\boldsymbol{\rho}}(t)}{\rho(t)} \right| \Delta t \quad (19)$$

3.1.4. 最接近距离计算

对于通过前面距离筛选的目标，计算航天器与接近目标的最接近时刻与最接近距离。显然，相对距离的极小值点，可通过计算相对距离变化率的零点获得。根据时间段的初始与终端时刻速度位置，采用三次多项式插值函数计算时间段内任意时刻的位置速度。基于相对距离平方值 $\rho^2(t)$ 的多项式插值函数 $\gamma(t)$ ，通过牛顿迭代方法得到最接近时刻与最接近距离：

$$t_{k+1} = t_k - \frac{d\gamma(t_k)/dt_k}{d\gamma^2(t_k)/dt_k^2} \quad (20)$$

式中， $\gamma(t)$ 是相对距离平方值的插值函数。 k 表示迭代次数。

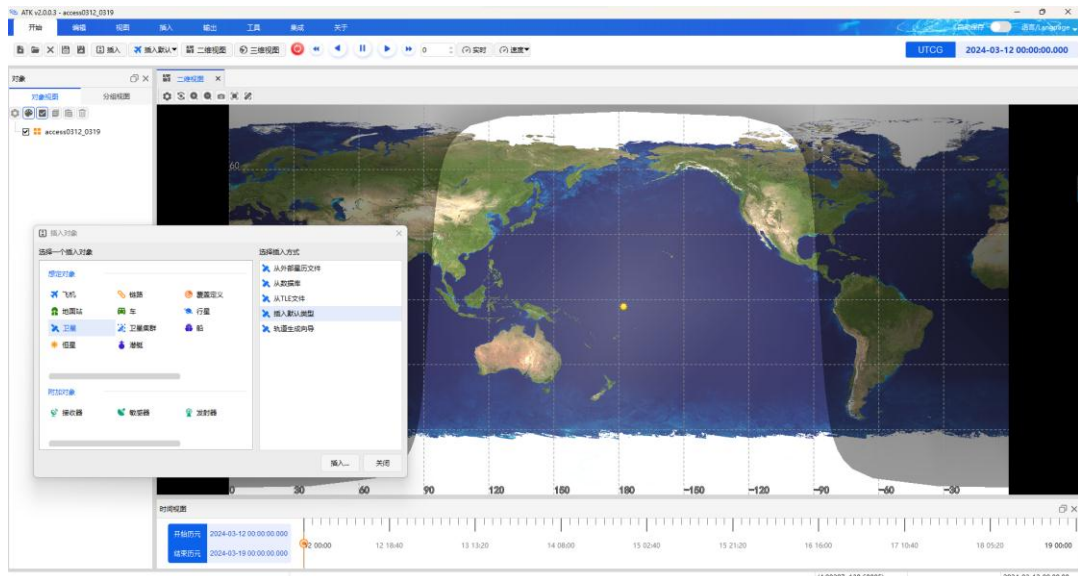
当式(20)迭代收敛时，得到最接近时刻 t_{tca} ，通过 t_{tca} 求解最接近距离值或其他参数。

$$\rho_{tca} = \sqrt{\gamma(t_{tca})} \quad (21)$$

式中， ρ_{tca} 是最小相对距离。

3.2 仿真场景创建

打开 ATK 软件，新建一个想定文件，命名该文件，设置文件保存路径，将本文场景时间设定为 2024 年 3 月 12 日至 19 日，创建 ATK 仿真场景。



3.3 基于仿真场景的接近分析

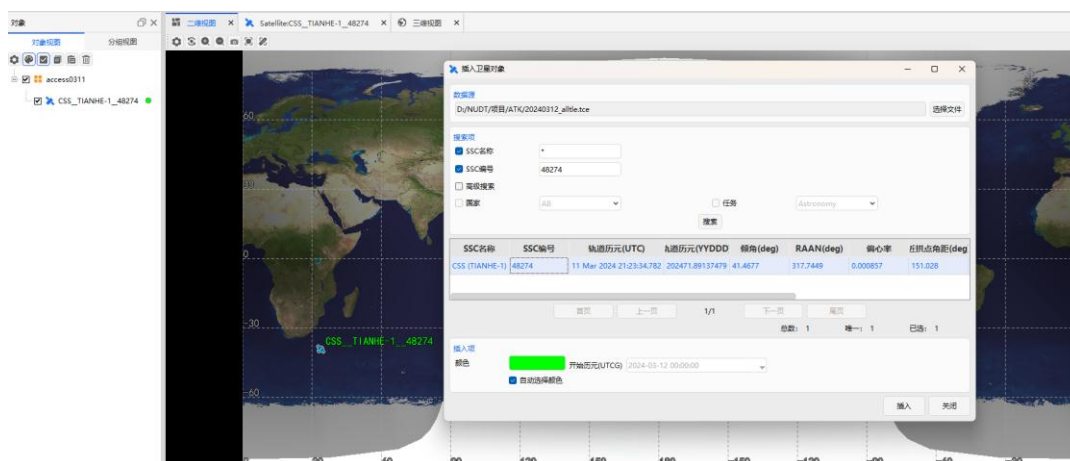
在 ATK 软件中，空间站数据的载入方式有三种，本文将介绍第一种类型，即在 ATK 场景中插入空间站，分析场景中的空间站与目标 TLE 数据库之间的接近关系。

3.3.1. 基于 TLE 数据插入空间站

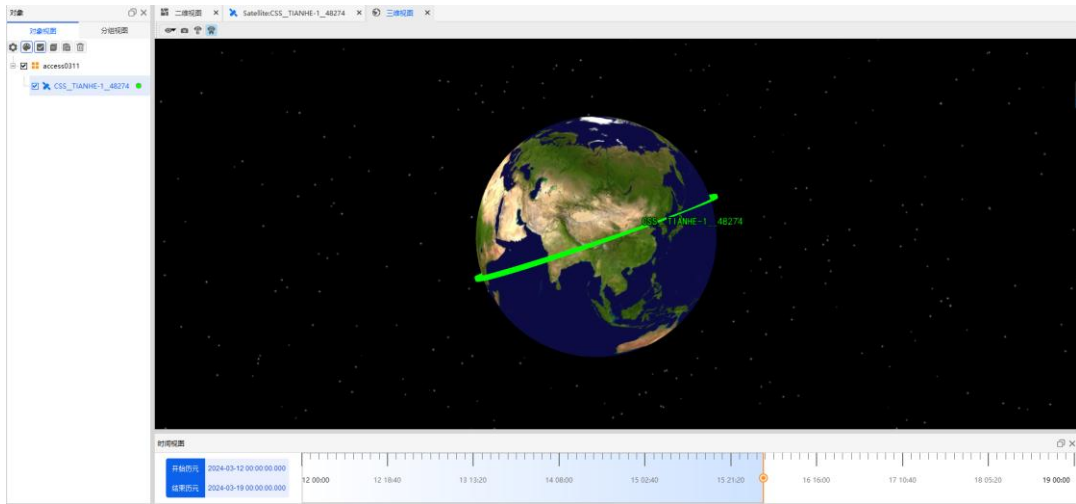
打开插入对象窗口，想定对象选择卫星，插入方式选择 TLE 文件，通过插入按钮，打开插入卫星对象窗口。



打开插入卫星对象窗口后，在 SSC 编号栏中搜索 48274，选择空间站最新的 TLE 数据行，将空间站插入至仿真场景中。

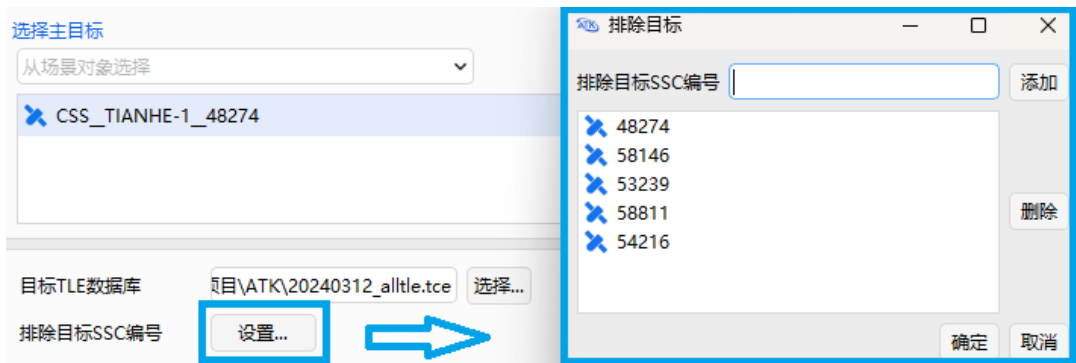


点击开始按钮进行仿真，可以对空间站的未来态势进行分析。



3.3.2. 配置接近分析参数

打开工具内的接近分析模块，从场景对象中选择空间站为主目标，载入目标 TLE 数据，在设置排除目标 SSC 编号中，应当筛除目标 TLE 数据库中的空间站，防止与仿真场景中的空间站进行接近分析。



设置仿真时间为 2024 年 3 月 12 日至 2024 年 3 月 19 日，接近距离门限为 20km，即当卫星与空间站的相对距离小于 20km 时，对应的预测时刻与预测结果会被记录下来。

仿真分析时间

开始历元 2024-03-12 00:00:00.000

结束历元 2024-03-19 00:00:00.000

接近约束

相对距离门限 20000 m

计算 高级设置... 结果...

刷新 应用 关闭

实际上，在接近分析之前首先需要进行筛选，从大量目标中快速排除与所关心航天器轨道不可能接近的目标，再进一步进行接近分析。常用的筛选方法有轨道历元时刻筛选、近地点-远地点筛选、轨道面交线高度差筛选等。用户可以通过高级设置约束 TLE 数据的轨道历元过期门限、近地点门限与仿真分析步长，同时，还可以设置等效距离红黄告警门限。

高级设置

过滤器 等效距离告警门限

名称	使用	值	
轨道历元过期门限	<input checked="" type="checkbox"/>	2592000	sec
远地点门限	<input checked="" type="checkbox"/>	50000	m
仿真分析步长		600	sec

确定 取消

高级设置

过滤器 等效距离告警门限

名称	值	
黄色告警	2000	m
红色告警	500	m

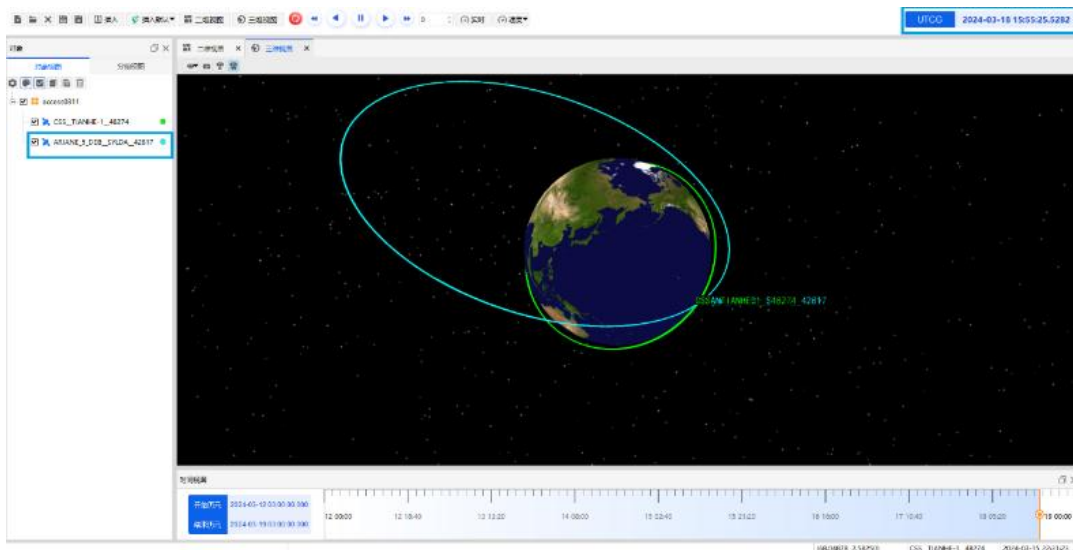
确定 取消

3.4 接近分析结果

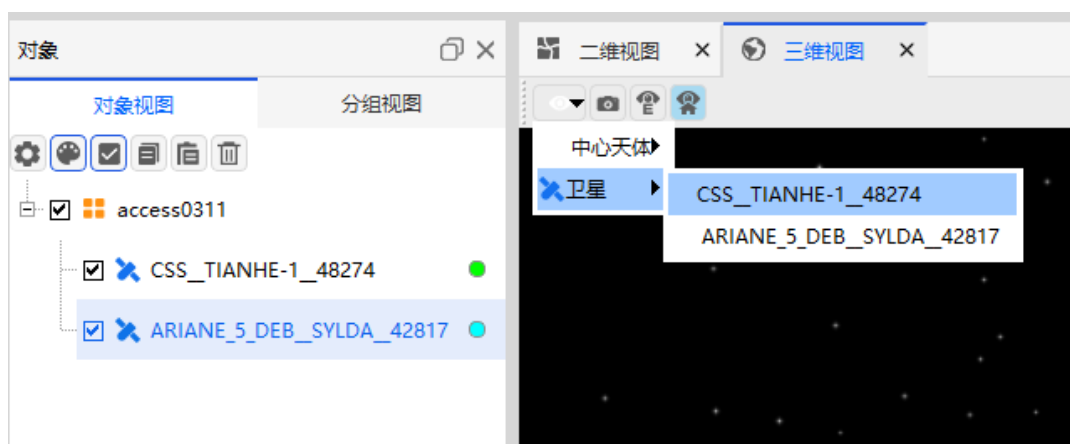
计算后得到本次接近分析结果，按照等效距离排序后，接近结果如下所示，在 2024 年 3 月 12 日至 19 日内共发生 52 次接近事件，出现一次黄色预警，预测在 2024 年 3 月 18 号 15 点 55 分 25 秒，与法国卫星 ARIANE 5 发生一次近距离的接近事件，等效距离约为 1.1676km，相对距离约为 2.0545km。

主目标(ID)	次目标(ID)	接近时刻(UTC)	等效距离(km)	相对距离(km)	径向距离(km)	横向距离(km)	法向距离(km)	接近夹角(deg)	相对速度(km/s)
CSS_TIANHE...	42817	2024-03-18 15:55:25.5282	1.16700090225024	2.05449646533132	-1.16700090225024	-1.28047100960694	-1.10365658276454	42.7776911471694	6.58512233259885
CSS_TIANHE...	53381	2024-03-17 08:02:19.0306	2.04718764873214	9.48330177887748	2.04718764873214	6.95367688264515	-6.11460564268987	82.6249530965201	10.1466673048528
CSS_TIANHE...	47499	2024-03-16 00:29:22.4367	2.26363315211769	19.3084846050258	2.26363315211769	7.95091927015042	-17.4452274774125	131.013044082543	13.9806382618034
CSS_TIANHE...	43669	2024-03-14 22:10:48.8549	2.36047302170755	4.89404972347433	-2.36047302170755	1.52939101945815	-4.00510335936923	138.163418573684	14.3630519965663
CSS_TIANHE...	14819	2024-03-18 06:30:02.3074	2.45282458329171	2.52996208654121	-2.45282458329171	-0.368862268140074	-0.498297451406673	107.135590155928	12.3494812636563
CSS_TIANHE...	14819	2024-03-18 08:02:08.6473	2.46497014513587	17.9213379049332	-2.46497014513587	10.5231224999356	14.2955296277449	107.31328537447	12.3637015051712
CSS_TIANHE...	14819	2024-03-18 04:57:55.9470	2.49144417186737	19.4215411621114	-2.49144417186737	-11.4675801471848	-15.4752567868178	106.957790344265	12.3352185801102
CSS_TIANHE...	43157	2024-03-14 21:32:53.3182	2.68542398451636	2.85332180942062	2.68542398451636	-0.353632982287479	-0.897154995158642	137.41720435384	14.3136730316135
CSS_TIANHE...	44413	2024-03-14 04:19:42.6154	2.74772310319817	2.81934696645682	-2.74772310319817	-0.311571307938783	-0.549234465655129	120.80774787711	13.3278226174363
CSS_TIANHE...	50069	2024-03-15 10:47:35.6409	3.33625016926506	16.4568062764584	3.33625016926506	-13.7826313987113	-8.35074723340474	62.2648869081434	7.94146557722614
CSS_TIANHE...	50069	2024-03-15 12:19:42.9057	3.45242349650045	16.3800334801484	3.45242349650045	13.6918323774564	8.30180672861223	62.4627749275732	7.96420442528385
CSS_TIANHE...	14819	2024-03-18 07:16:07.2577	3.63235033178956	12.2844485570873	-3.63235033178956	6.96418481086795	-9.44530768895056	107.224491215916	12.3761859119325
CSS_TIANHE...	14819	2024-03-18 05:44:00.9107	3.6334171765147	7.74694922147238	-3.6334171765147	-4.07131560447227	5.49889908165808	107.046725215603	12.3620168259031
CSS_TIANHE...	44878	2024-03-18 09:48:56.2589	3.92416733680606	19.1473346197405	-3.92416733680606	-16.5572710332896	8.77941397153555	55.9788445686218	7.20619215660392
CSS_TIANHE...	37852	2024-03-18 07:15:10.3301	4.81182865637218	18.8152996446637	-4.81182865637218	-13.9359148370807	-11.6898282003557	80.2023904216603	9.89956255835973
CSS_TIANHE...	45378	2024-03-17 20:11:52.1788	5.02301741994583	8.82486214681176	-5.02301741994583	-5.58238197268778	-4.6351347586436	79.3257164847277	9.80532213745198
CSS_TIANHE...	37852	2024-03-18 05:43:04.2551	5.11591547228829	17.692064228175	-5.11591547228829	-12.9503343727043	-10.9144576214041	80.4866597926111	9.92849061753989
CSS_TIANHE...	43157	2024-03-14 23:51:03.2805	5.26669961915287	14.0079004189914	-5.26669961915287	-4.73808004607442	12.0844423423257	137.270358368565	14.3133116266379
CSS_TIANHE...	43157	2024-03-14 22:18:56.9223	5.33364981469799	11.7144698610765	-5.33364981469799	3.78919365408825	-9.71714954253862	137.368858566522	14.3181422265229
CSS_TIANHE...	37852	2024-03-18 04:10:58.1667	5.41864218206674	16.8403121851343	-5.41864218206674	-12.1681488144068	-10.3039111907062	80.7720859835197	9.95748059695348
CSS_TIANHE...	50069	2024-03-15 14:37:51.8694	5.70548025268903	11.3978311942974	-5.70548025268903	-8.42352348212787	5.1383171531072	62.7614145682354	7.9992992738908
CSS_TIANHE...	37852	2024-03-18 02:38:52.0645	5.71986361906574	16.2519006702638	-5.71986361906574	-11.585883410022	-9.857724949743	81.0586506888652	9.98652194855564
CSS_TIANHE...	58792	2024-03-12 23:04:52.6716	5.81396822118937	16.0362091647956	-5.81396822118937	14.8639870348731	-1.55522798984722	12.2926523903056	14.6309515995196
CSS_TIANHE...	37852	2024-03-18 01:06:45.9486	6.01948734038112	15.9171379825614	-6.01948734038112	-11.1998440911812	-9.57520475228199	81.3463360248232	10.0156143816365
CSS_TIANHE...	37852	2024-03-17 23:34:39.8187	6.31744611979547	15.8259888467132	-6.31744611979547	-11.0064196842958	-9.45571378795921	81.6351244325095	10.0447552376675
CSS_TIANHE...	37852	2024-03-17 22:02:33.6747	6.61370199630301	15.9689653741025	-6.61370199630301	-11.0019863591754	-9.49858395635367	81.9249987600165	10.0739419766483
CSS_TIANHE...	37852	2024-03-17 20:30:27.5165	6.90824262830426	16.3378618816656	-6.90824262830426	-11.1827543852073	-9.70298505682607	82.2159421975098	10.1031703809255
CSS_TIANHE...	40380	2024-03-14 12:46:08.7398	7.02339028145103	11.0296793582402	7.02339028145103	4.0433978885182	-7.48176110385361	123.440107033433	13.559394265029
CSS_TIANHE...	37852	2024-03-17 18:58:21.3438	7.20109259586164	16.9270290095768	-7.20109259586164	-11.5453647254896	-10.06842241229	82.5079382764996	10.1324435406488
CSS_TIANHE...	37852	2024-03-17 17:26:15.1567	7.4922868105415	17.7324771708733	-7.4922868105415	-12.0860489905076	-10.5940457221582	82.8009708772413	10.1617538918038

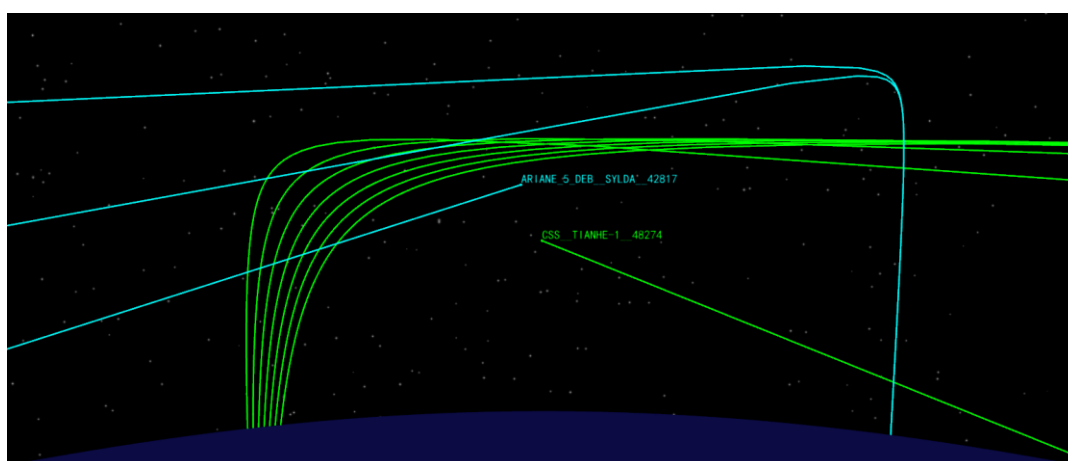
从插入对象窗口中，将 ARIANE 5 卫星插入至仿真场景中，并在 ATK 页面的 UTCG 内输入接近时间“2024-03-18 15:55:25.5282”。



在三维视图中，选择空间站为局部视点，观察接近时刻，空间站与 ARIANE 5 之间的相对运动状态。



基于空间站视角，将空间站与 ARIANE 5 的接近关系可视化，更直观地分析两者的动态变化与接近情况。



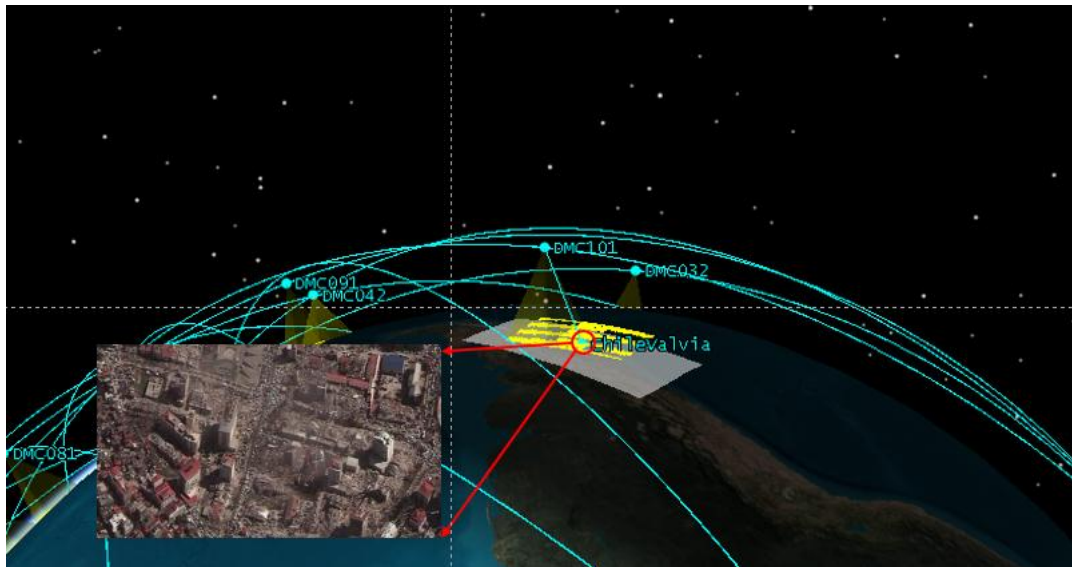
小结

基于 ATK 仿真场景的接近分析模块能够将航天器插入 ATK 仿真场景中，对航天器与海量空间目标进行接近分析。根据本文的场景可知，ATK 能够快速计算空间站与海量空间目标的接近时刻、等效距离、接近姿态等参数，从而高效预测未来 7 天内可能对空间站造成威胁的空间目标，并提供三维可视化仿真结果。

4. ATK 特定问题建模—灾害监测星座对地覆盖性能分析

内容简介

灾害监测星座（Disaster Monitoring Constellation, DMC）由多颗对地观测卫星组成，采用卫星对目标接力观测、共同观测等方式实现对地面目标的长时间、高精度的监测。其监测对象包括已受灾地区或灾害频发区域，作用是通过快速获取相关地区的图像信息，为灾害预警、应急响应和灾后评估提供数据支持。分析评估灾害监测星座的覆盖性能，对星座任务调度、星座效能评估和星座布局优化等具有重要意义。

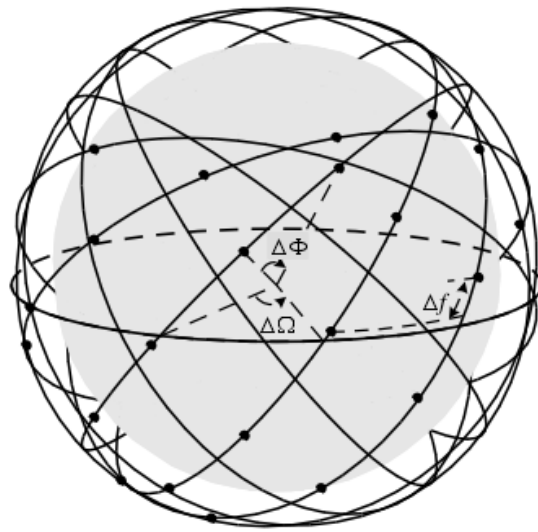


Walker 星座指由具有相同轨道高度和轨道倾角的多颗圆轨卫星组成的以地球为球心的均匀分布的卫星星座，即所有轨道面的升交点赤经在 $[0, 2\pi]$ 内均匀分布，每个轨道面内有相同数量的卫星，且卫星在轨道面内也是均匀分布的。由于该星座围绕地球均匀分布的性质，是全球覆盖最有效的星座，常用于对地观测、导航和互联网通信等^[3]。

鉴于 Walker 星座良好的对地覆盖性能，本文参考国际 DMC 星座和我国“36 天罡星群”的设计，基于 ATK 仿真平台的星座设计模块，设计由 36 颗卫星组成的 Walker 星座作为本次案例的对地观测星座命名为 DMC 星座，并通过覆盖分析模块，计算对地覆盖重数、覆盖时间等品质因子生成相应图表，分析该星座对全球灾害频发地区以及我国周边区域的覆盖性能。

4.1 星座设计及对地覆盖分析方法

4.1.1. Walker 星座模型



Walker 星座参数定义如上图所示，常用的描述方式为 $(N/P/F, h, i)$ 。其中 N 为星座中的卫星总数； P 为星座的轨道面数； F 为相位因子，可取 $0 \sim (P-1)$ 之间的一个整数，代表 Walker 星座中相邻两个轨道面对应卫星之间的相位关系；相邻两个轨道面升交点赤经差为 $\Delta\Omega = 2\pi/P$ ，同一轨道面内相邻卫星相位差为 $\Delta\Phi = 2\pi/(N/P)$ ，两相邻轨道异轨卫星相位差为 $\Delta f = 2\pi F/N$ ； h 为星座中卫星的轨道高度； i 为星座中的卫星轨道倾角。

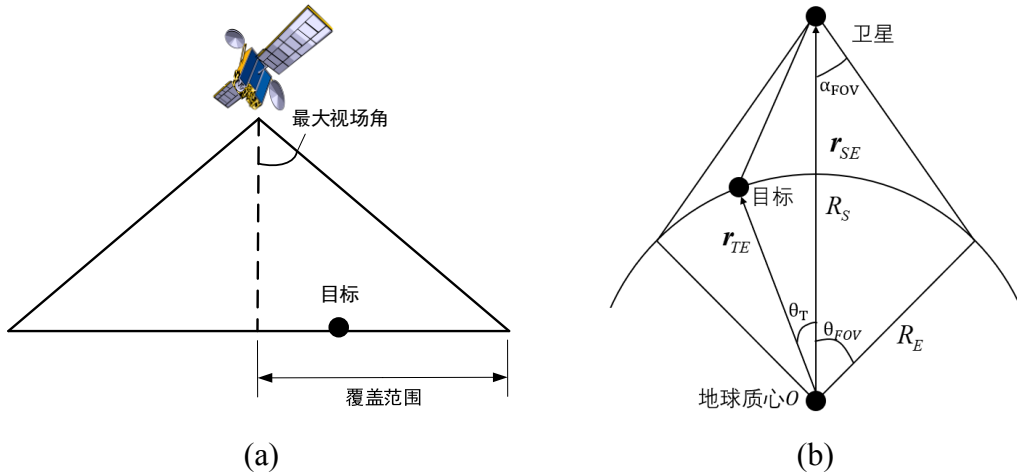
标识符为 $N/P/F$ 的 Walker 星座任意一颗编号为 m 的卫星的升交点赤经和纬度幅角分别为：

$$\begin{cases} \Omega_m = \frac{360}{P}(P_m - 1) & (P_m = 1, 2, \dots, P) \\ u_m = \frac{360}{S}(N_m - 1) + \frac{360}{N}F(P_m - 1) & (N_m = 1, 2, \dots, S - 1) \end{cases}$$

其中， S 为每个轨道平面上的卫星数， $S=N/P$ 。

4.1.2. 卫星对地覆盖模型

卫星的运动和地球自转使得卫星的星下点在地球表面移动，形成星下点轨迹。卫星对地面目标的观测只能在卫星运动到目标的上空时进行，如下图 a 所示，当目标处于卫星的覆盖范围内时，卫星的成像载荷能够观测到地面目标，实现对地面目标的覆盖。



根据覆盖的定义可以获得图 b 所示的星地空间关系图，只要 $\theta_T < \theta_{FOV}$ 也即目标在卫星的视场内，即可认为目标被卫星覆盖，相应观测方程如下式所示。式中 θ_T 为目标角度， θ_{FOV} 为观测范围角度， α_{FOV} 为视场角， R_S 卫星轨道半径， R_E 为地球半径。

$$\begin{cases} \theta_T < \theta_{FOV} \\ \theta_{FOV} = \arcsin\left(\frac{\sin(\alpha_{FOV})R_S}{R_E}\right) - \alpha_{FOV} \\ \theta_T = \arccos\left(\frac{\mathbf{r}_{SE} \cdot \mathbf{r}_{TE}}{|\mathbf{r}_{SE}| |\mathbf{r}_{TE}|}\right) \end{cases}$$

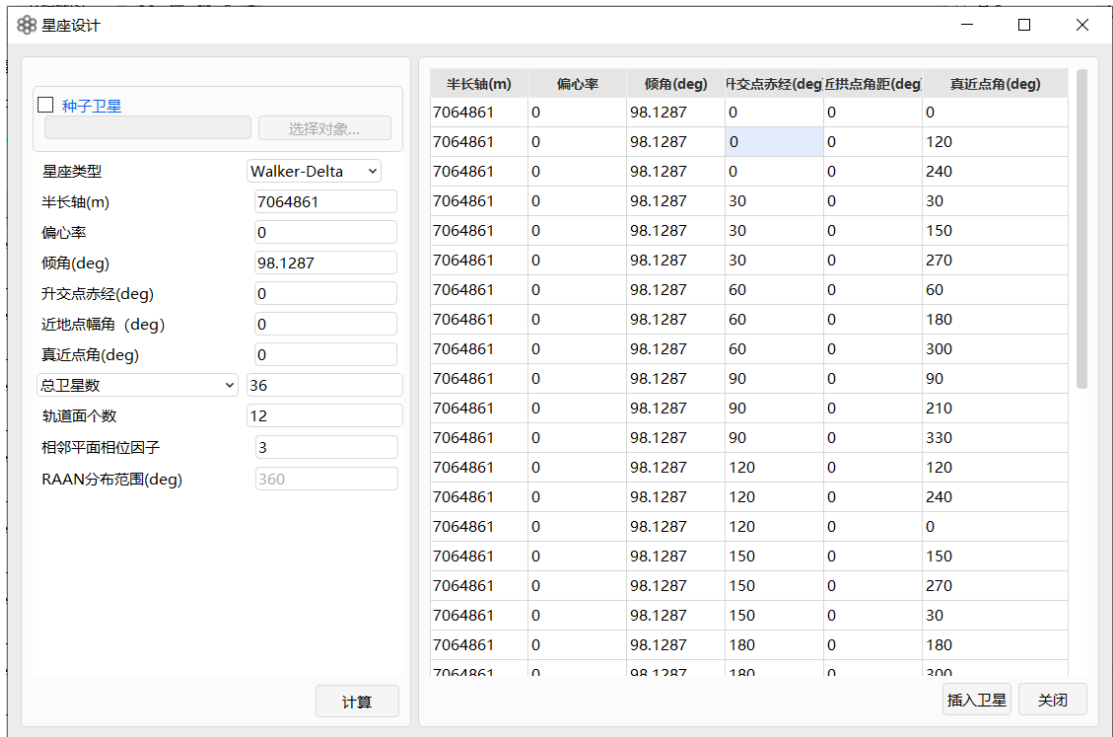
4.2 场景创建与星座生成

4.2.1. 新建想定

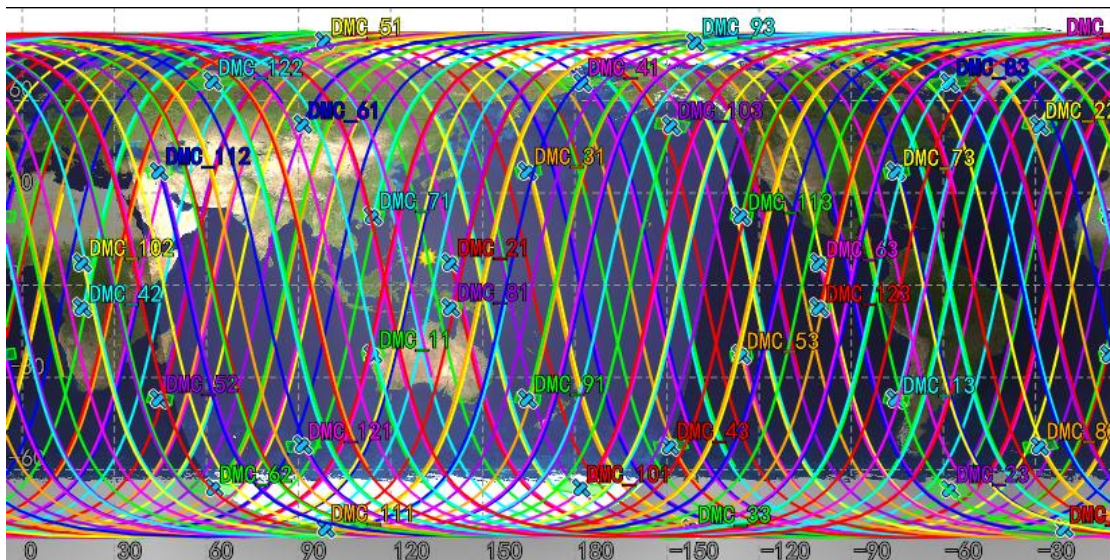
安装并运行 ATK.exe，新建一个想定“灾害监测星座覆盖性分析”。设置开始历元为 2033-04-13 00:00:00.000000，结束历元为 2023-04-14 00:00:00.000000。

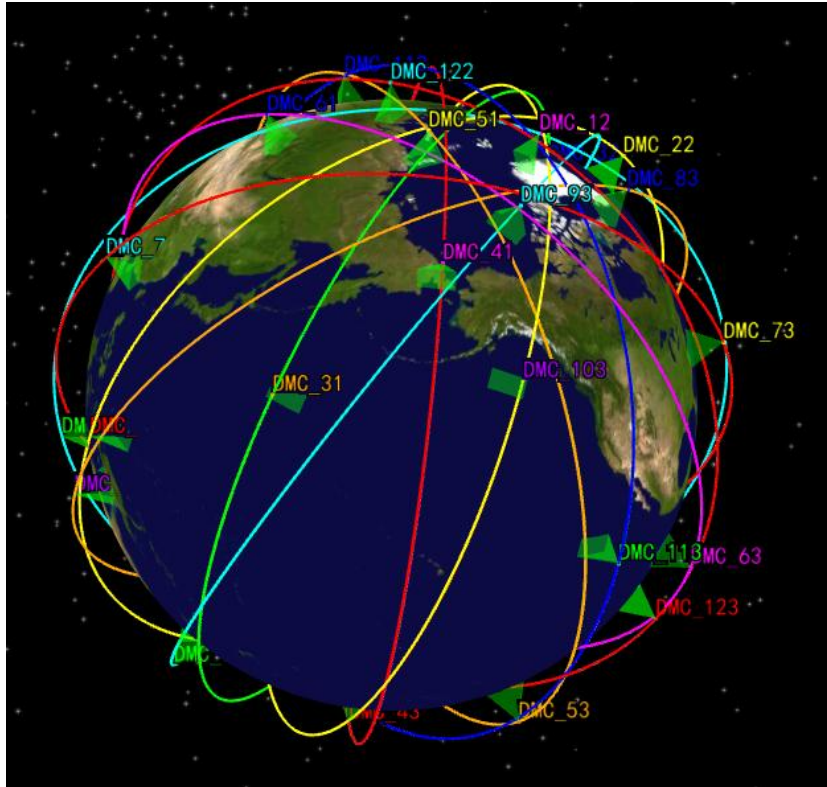
4.2.2. 导入 DMC 星座和传感器

点击“工具”选项卡的【星座设计】模块，星座类型选择为 Walker-Delta 星座；设置基准卫星参数，半长轴为 7064861m，偏心率为 0，倾角为 98.1287° ，升交点赤经、近地点幅角、真近点角均为 0；设置星座参数，总卫星数为 36 颗，轨道面个数为 12，相邻平面相位因子为 3。点击【计算】，在右侧窗口中显示每颗卫星的轨道根数如下：



点击【插入卫星】，在对象视图中全选所有卫星，右键选择“批量添加传感器”，设置传感器参数为：矩形，垂直半锥角为 26.7° ，水平半锥角为 15° [4]。可以看到星座的二维、三维视图如下：

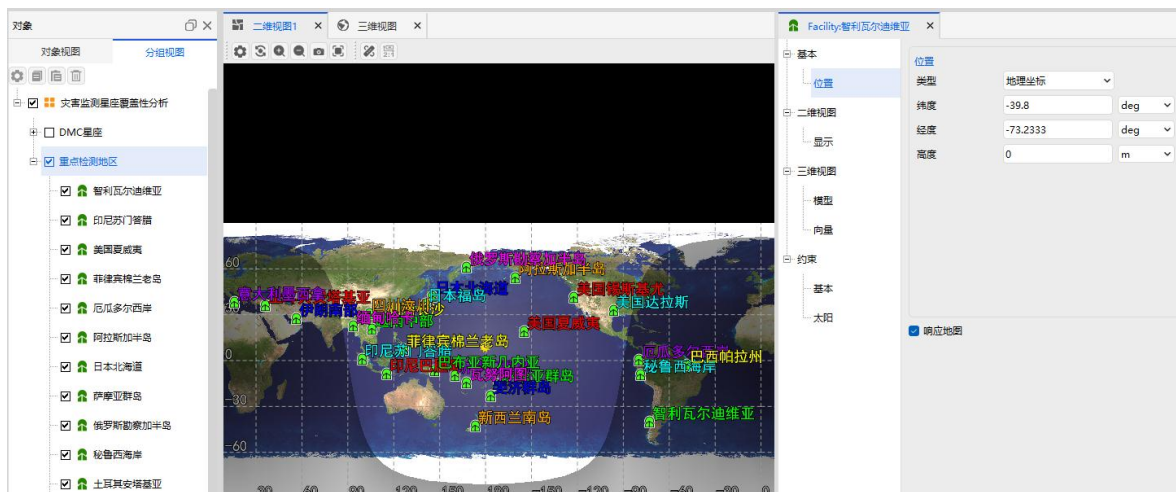




4.3 观测目标与区域导入

4.3.1. 重点监测地区（点目标）

创建新的分组，并命名为重点监测地区，然后选择【插入对象】——“地面站”，将收集到的灾害频发地点的经纬度输入，并插入到该分组中。



4.3.2. 灾害频发地区（大区域）

此处选择墨西哥、印度尼西亚、智利及周边海域作为重点观测区域。选择【插

入对象】——“覆盖定义”，修改覆盖定义属性，“网格区域配置参数”类型为“经纬度区域”，“网格定义”经纬度间隔为1°。墨西哥、印度尼西亚、智利的经纬度区域参数如下：

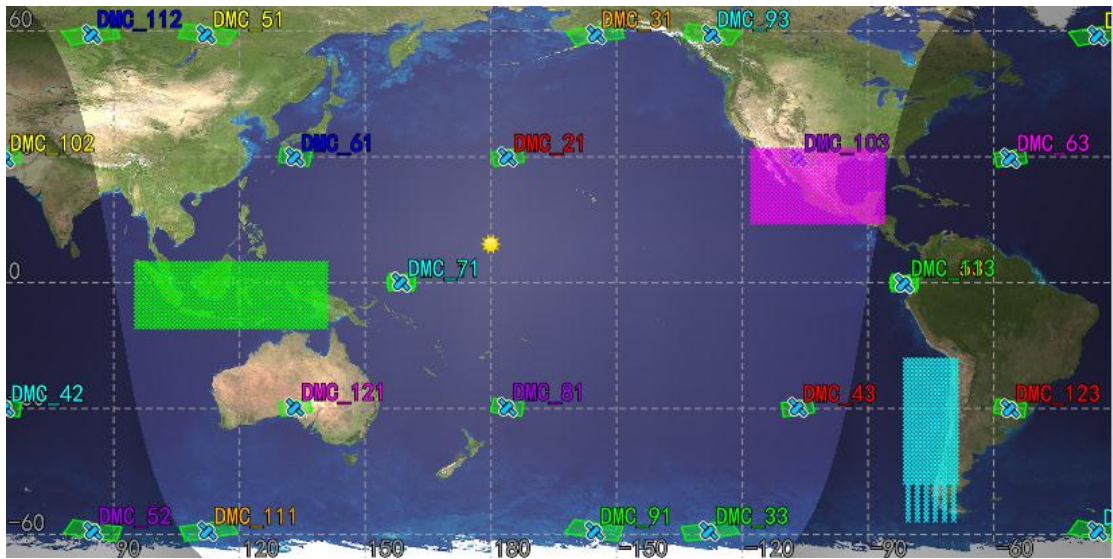
网格区域配置参数		网格区域配置参数		网格区域配置参数	
类型	经纬度区域	类型	经纬度区域	类型	经纬度区域
最小纬度(deg)	14	最小纬度(deg)	-11	最小纬度(deg)	-11
最小经度(deg)	-118	最小经度(deg)	95	最小经度(deg)	95
最大纬度(deg)	32	最大纬度(deg)	5	最大纬度(deg)	5
最大经度(deg)	-86	最大经度(deg)	141	最大经度(deg)	141

(a)墨西哥

(b)印度尼西亚

(c)智利

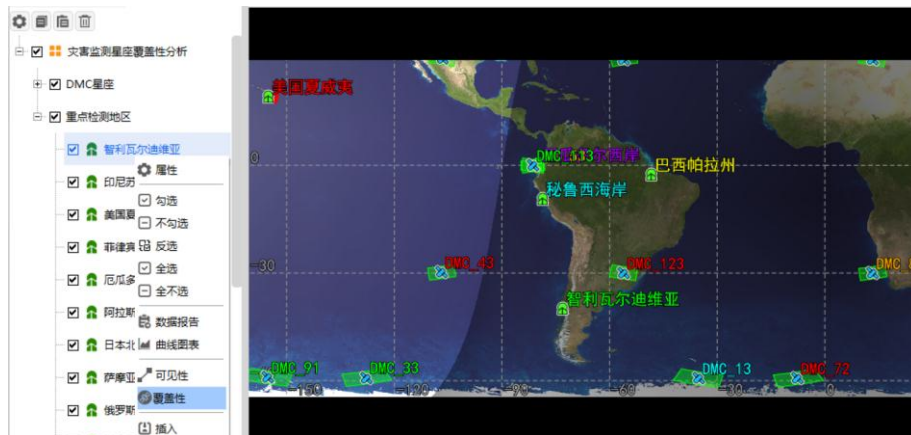
可以在二维视图中看到区域网格点显示如下：



4.4 星座覆盖性能分析

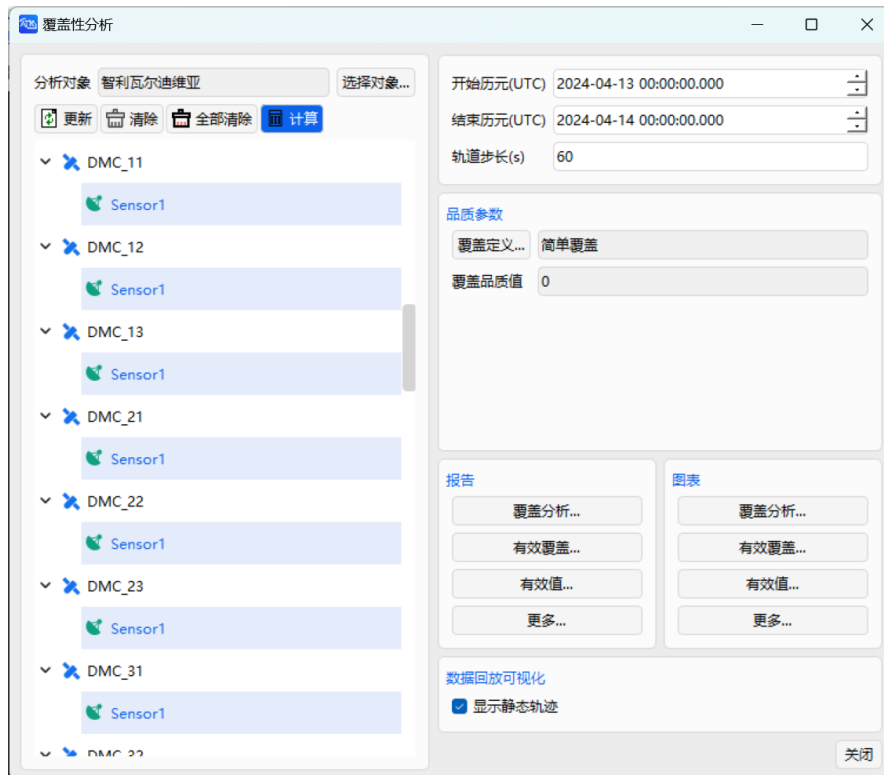
4.4.1. 点目标覆盖分析操作与结果

选择任意一个地点演示星座覆盖性分析操作，本文以智利瓦尔迪维亚为例展开覆盖性分析。右键该地点选择覆盖性，如图所示：

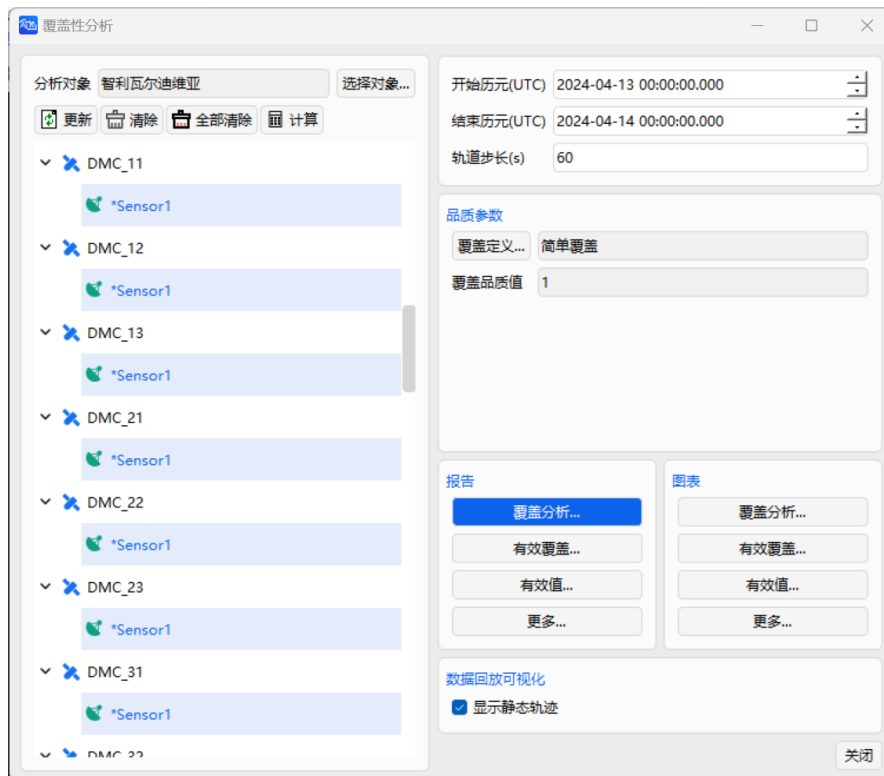


然后选择 DMC 星座的所有 36 颗卫星的传感器 Sensor，点击计算，如下图

所示。

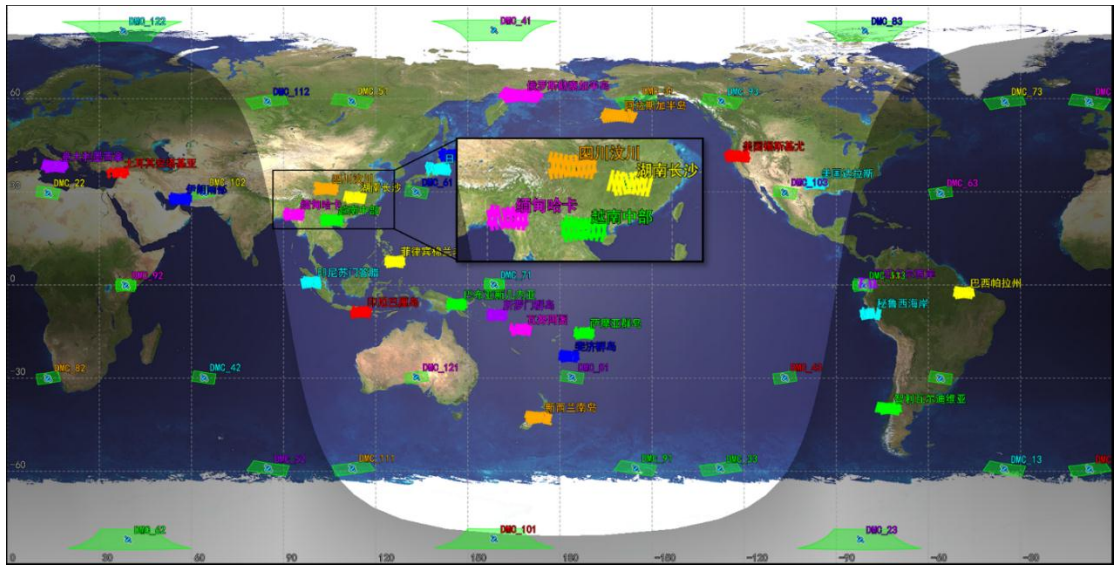


最后可点击覆盖分析报告、有效覆盖分析报告及覆盖分析曲线等报告与图表，展开 DMC 星座对智利瓦尔迪维亚的覆盖性能分析，如下图所示。

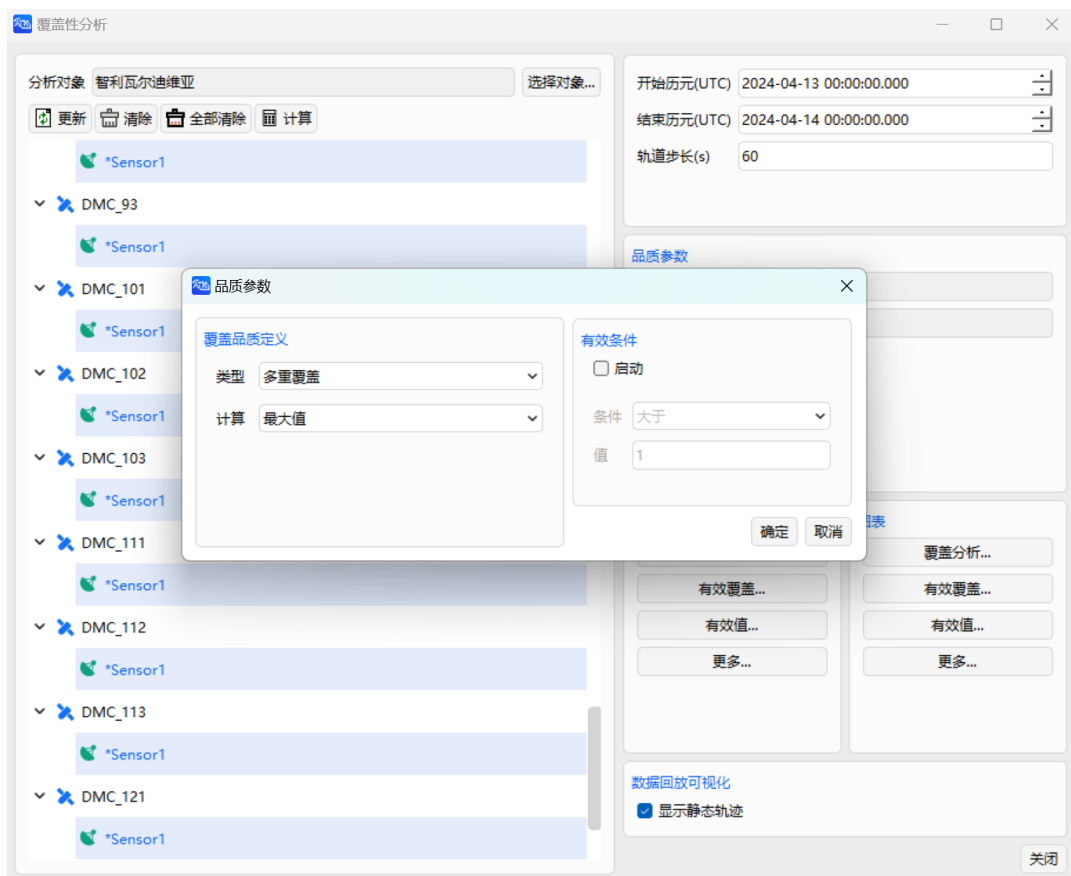


对覆盖结果进行分析，首先计算星群对所有目标的覆盖，得到以下覆盖弧段

详情



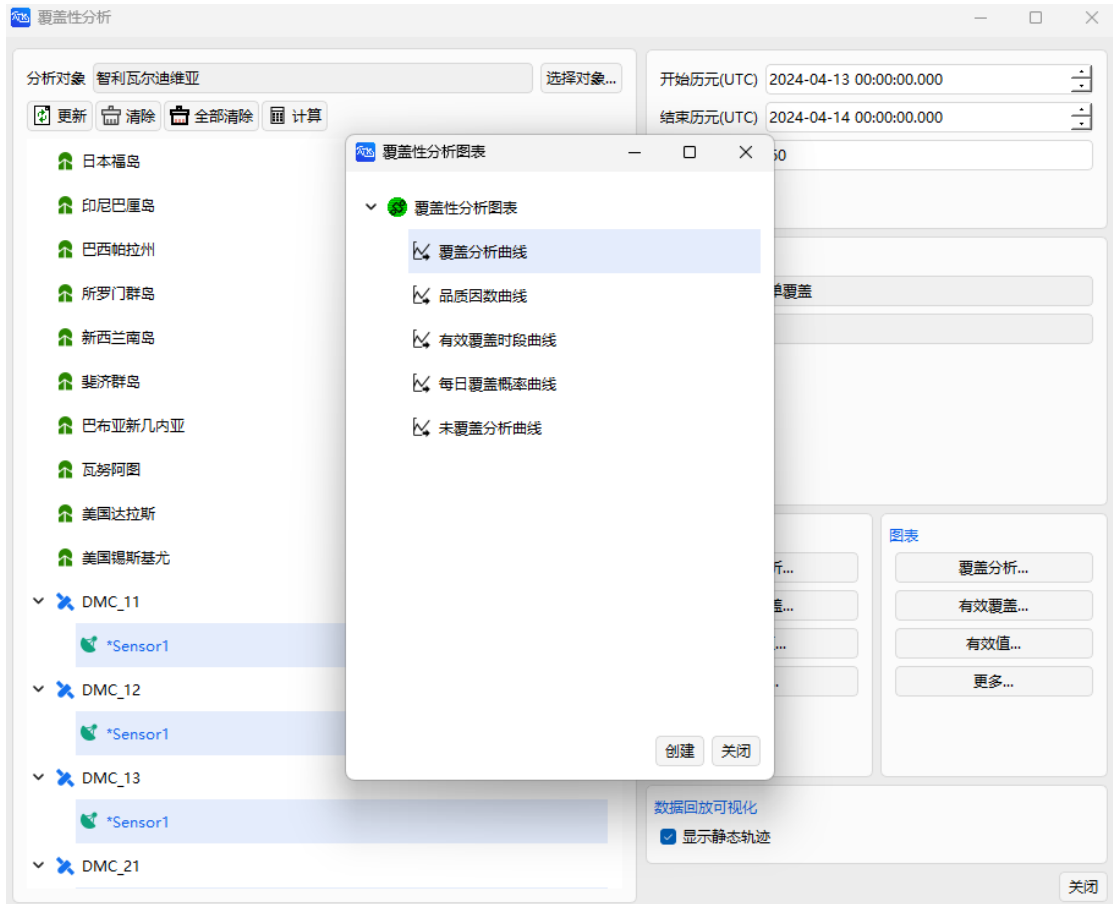
然后选择智利瓦尔迪维亚作为分析对象做进一步分析，在覆盖分析界面中，点击“覆盖定义”，求解不同的品质因子如图所示：



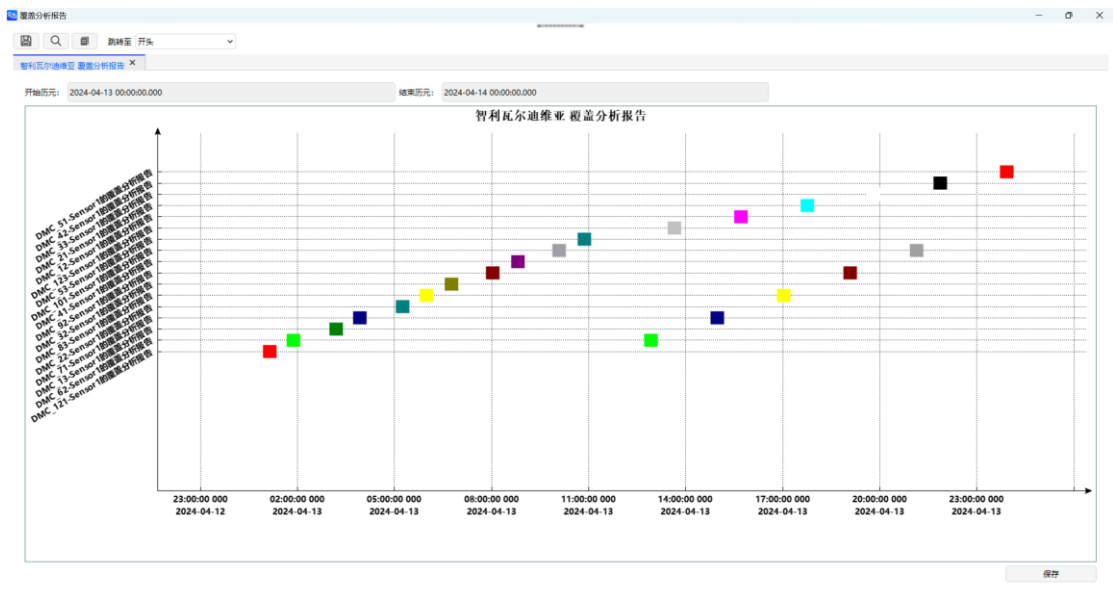
得到的品质因子参数如表所示，从表中可以看出，小型低轨星座对于某特定地点的较难保持长时间连续的检测，难以实现多重覆盖，因此如果想要在全球实现较高的覆盖性能，需要设计巨型低轨星座才能实现。

品质因子类型	总值	最大值	最小值	平均值
覆盖时间	1210.882s	1210.882s	1210.882s	1210.882s
访问时长	1210.882s	55.686s	54.792s	55.040s
重访时间	85189.118s	7340.583s	249.442s	3703.875s
响应时长	1467589.118s	7340.583s	0.000s	2166.320s
多重覆盖		1	0	0.015267176
访问次数	22	22	22	22
覆盖间隔次数	23	23	23	23
可见性约束/ 目标高度范围 约束	45899.932km	696.264km	694.655km	695.453km
光照约束/ 太阳与当地水 平面夹角	200.533	42.473deg	-61.477deg	-7.870deg
时间平均间隔		4332.6392s		

然后选择品质因子为“多重覆盖”，点击“覆盖分析曲线”或其他曲线进行展示，操作如图所示：

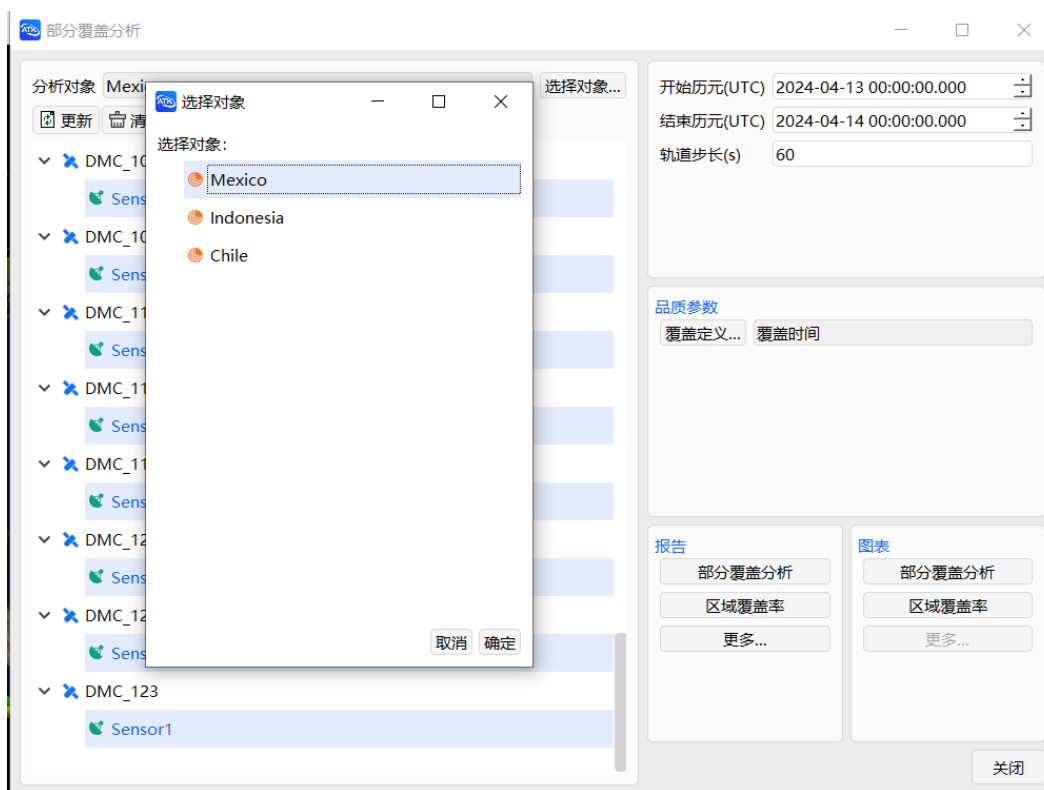


可以得到以下的甘特图，发现对智利瓦尔迪维亚地区的覆盖时间持续较短，也没有多重覆盖的情况，但是该星座对该地区的覆盖在时间的分布上较为均匀，当该地区发生灾害时，能够较短时间内发现。



4.4.2. 区域目标覆盖分析操作与结果

以墨西哥区域覆盖分析为例。在对象视图中右击覆盖定义，选择【区域覆盖】功能，全选所有传感器，仿真时间为 24h，轨道步长为 60s，品质因数为“覆盖时间——每日平均值”，点击【计算】。仿真计算过程约 10 分钟。



仿真计算结束后，可以查看相关报告和图表，如区域覆盖率报告及图表如下：

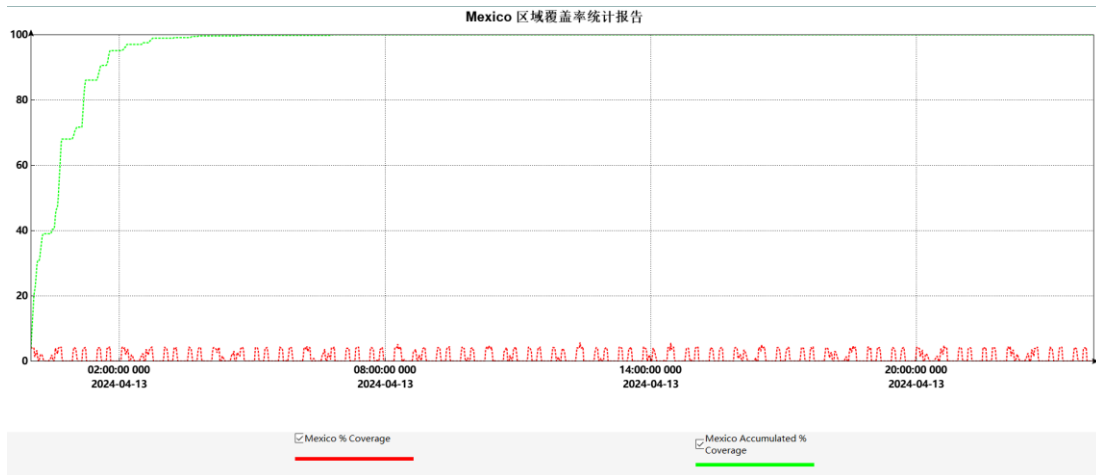
区域覆盖率统计报告
(2024-04-15 14:37:15)

CoverageDefinition-Mexico

Coverage Properties

LatLon Region Coverage
Grid Altitude: 0 (km)
Ground Altitude set from grid altitude reference (On ellipsoid)
Resolution: 1 (deg)
Number of Points: 576

Time (UTCG)	% Coverage	% Accum Coverage
2024-04-13 00:00:00.000	4.17	4.17
2024-04-13 00:01:00.000	3.99	8.16
2024-04-13 00:02:00.000	4.17	12.33
2024-04-13 00:03:00.000	3.99	16.32
2024-04-13 00:04:00.000	3.99	20.31
2024-04-13 00:05:00.000	1.39	21.70
2024-04-13 00:06:00.000	1.74	23.44
2024-04-13 00:07:00.000	2.78	26.22
2024-04-13 00:08:00.000	3.30	29.51
2024-04-13 00:09:00.000	1.22	30.73
2024-04-13 00:10:00.000	0.00	30.73
2024-04-13 00:11:00.000	0.00	30.73



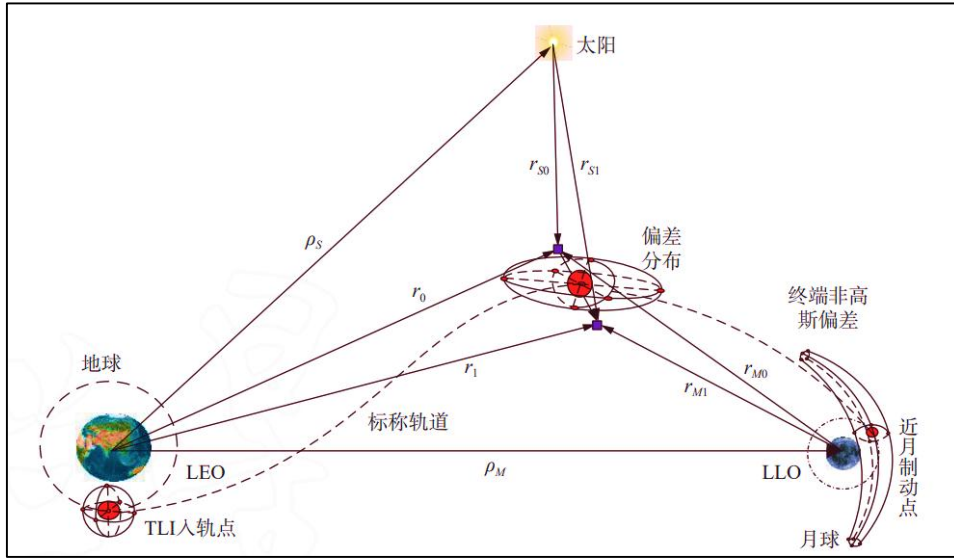
由区域覆盖率报告及图表可以得到每个仿真时刻的实时覆盖率和累计覆盖率，可以看出在约 7h 后该星座对墨西哥区域实现了 100%覆盖。另外，还可以查看部分“部分覆盖弧段统计报告”、“单个访问对象区域覆盖结果统计报告”以及相应图表。

5. ATK 特定问题建模—基于二次开发的轨道偏差演化分析

内容简介

受各种误差影响，航天器入轨参数往往与标称值存在一定偏差，经过一段时

间演化，会出现轨道偏离。为保证轨道方案设计的鲁棒性，需要精确预报航天器的状态偏差，航天器轨道偏差演化分析因此具有重要意义。下图是地月空间轨道偏差演化示意图。工程中为了更精确地获得偏差经非线性传播后的分布情况，一般采用蒙特卡洛方法进行大量随机抽样与仿真。



航天任务工具箱 ATK (Aerospace Tool Kit) 软件本身不具备准确的偏差演化分析功能，但可以通过二次开发模块实现基于蒙特卡洛方法的精确偏差演化。二次开发模块拥有 C++、Matlab、Python 等多种语言接口，可以通过编程形式实现对 ATK 软件的快速操作。本文采用 Matlab 进行二次开发。

本文以地月自由返回轨道作为研究对象，分析存在初始位置速度偏差时，高精度轨道预报模型下的轨道偏差演化情况。初始位置速度来源于前序 ATK 特定问题建模结果（详见 XXX，对应轨道设计的题目，给出链接，点击后直接跳转至相关内容），初值偏差采用高斯型随机分布。本文案例包括 3 个 m 文件，分别是标称轨道仿真、蒙特卡洛打靶仿真和数据分析，完整代码详见文末附录。接下来将结合代码和注解，带大家实现该仿真案例。

5.1 轨道力学与偏差演化基本理论

航天器轨道偏差传播本质是已知航天器初始时刻运动状态的分布情况，计算之后任意时刻运动状态分布的初值问题。本节将介绍基于二次开发的轨道偏差演化分析中与轨道偏差传播问题相关的基本概念、理论和方法。5.1 节介绍偏差传播分析的基本理论，包括偏差统计矩与概率密度函数定义、高斯分布的性质等；

5.2 节描述了航天器轨道偏差演化分析问题; 5.3 节介绍本文中进行轨道偏差传播分析的 Monte Carlo 方法。

5.1.1. 偏差演化分析基本理论

根据大数定律及中心极限定律, 自然界中大多数偏差都服从高斯分布, 因其描述了自然界中的正常状态, 所以又称为正态分布。高斯分布 $\mathbf{x} \sim p_g(\mathbf{x}; \mathbf{m}, \mathbf{P})$ 的概率密度函数 $p(\mathbf{x}, t)$ 定义为

$$p_g(\mathbf{x}; \mathbf{m}, \mathbf{P}) = \frac{1}{\sqrt{(2\pi)^n \det(\mathbf{P})}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1}(\mathbf{x} - \mathbf{m})\right\} \quad (22)$$

其中, n 为不确定性状态量 \mathbf{x} 的维数, \mathbf{m} 为 \mathbf{x} 的均值, \mathbf{P} 为 \mathbf{x} 的协方差矩阵。

对满足高斯分布的偏差, 定义其在某一给定概率值下可能出现的空间为置信空间, 用公式表示为

$$\Omega_N = \left\{ \mathbf{x} \mid (\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1}(\mathbf{x} - \mathbf{m}) \leq N^2 \right\} \quad (23)$$

其中, Ω_N 表示一个 n 维的误差椭球, 其大小依赖于 N 的取值, 又称为 $N - \sigma$ 误差椭球, σ 为误差标准差。根据统计理论, 对一维分布, 高斯分布 3σ 误差椭球包含的概率值是 0.9973, 对三维分布, 3σ 误差椭球包含的概率值是 0.9919。高斯分布的统计特性可完全由其前两阶矩 (即均值和协方差矩阵) 决定。

5.1.2. 轨道偏差演化分析问题描述

要完全确定一个随机变量的分布特征, 须知道其概率密度函数 (或分布函数)。轨道偏差演化分析问题可描述为: 给定航天器初始状态 \mathbf{x}_0 的分布情况, 即概率密度函数 $p(\mathbf{x}, t)$ 。然而, 6 维度的轨道状态偏差演化分析问题很难直接求解。因此很多对轨道偏差演化分析的研究都基于偏差的高斯分布假设和动力系统的线性化假设对问题进行简化, 从而方便求解。

对服从高斯分布的偏差, 其概率密度函数可由其前两阶矩完全确定, 且高斯分布经线性化或线性化的动力学方程传播后仍然为高斯分布。因此在线性假设下, 仅需对其均值和协方差进行预报就可以得到高斯分布偏差任意时刻的概率密度函数。将动力学方程线性化, 可得高斯分布偏差均值和协方差矩阵的传播公式为

$$\begin{aligned} \mathbf{m}(t) &= \Phi(t, t_0) \mathbf{m}_0 \\ \mathbf{P}(t) &= \Phi(t, t_0) \mathbf{P}_0 \Phi(t, t_0)^T \end{aligned} \quad (24)$$

式中， $\Phi(t, t_0)$ 为状态量 $\mathbf{x}(t)$ 的一阶状态转移矩阵，该矩阵可沿参考轨迹 $\bar{\mathbf{x}}(t) = \phi(t; \bar{\mathbf{x}}_0, t_0)$ 积分如下微分方程获得。

$$\dot{\Phi} = \mathbf{A}\Phi, \quad \mathbf{A} = \left. \frac{\partial \mathbf{f}(t, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\bar{\mathbf{x}}}, \quad \Phi(t_0, t_0) = \mathbf{I} \quad (25)$$

现实生活中很多偏差因素都可以用高斯分布描述，因而对初始偏差作高斯分布假设是相对合理的。然而，高斯分布通过非线性系统后，将变成非高斯分布。对非高斯分布偏差，虽然均值和协方差矩阵仍然包含其大部分分布信息，但前两阶矩已不能完整表征其分布特性。轨道偏差的动力系统就包含非线性。一般假设初始误差服从高斯分布，其分布情况可以用 3σ 误差椭球来表示，该误差椭球在非线性轨道动力学中随着时间推移不短放大、旋转、变形，最后变为非高斯分布，该非高斯偏差一般沿航天器标称轨道弧线弯曲。

5.1.3. 轨道偏差演化分析基本方法

偏差演化分析方法可大致分为 Monte Carlo 打靶仿真、线性方法和非线性方法。工程上为了较为精确地获得偏差经非线性传播后的分布情况，一般采用 Monte Carlo 方法进行大量随机抽样与仿真。通过大量随机采样与统计分析，可以计算得到系统输出的均值、方差、乐观值、悲观值、分布函数和概率密度函数等。给定航天器初始状态 $\mathbf{x}(t_0)$ 的概率密度函数 $p(\mathbf{x}_0, t_0)$ ，根据 $p(\mathbf{x}_0, t_0)$ 随机产生 M 个初始样本值 $\mathbf{x}^k(t_0)$, ($k=1, 2, \dots, M$)，将这些样本值预报到任意 t 时刻，获得 t 时刻的样本值 $\mathbf{x}^k(t) = \phi(t; \mathbf{x}_0^k, t_0)$ ，进一步对终端样本进行统计分析，则可以得到终端状态偏差的分布规律。例如，终端样本的均值及协方差矩阵可表示为

$$\begin{aligned} m^i(t) &= \frac{1}{M} \sum_{k=1}^M \phi^i(t; \mathbf{x}_0^k, t_0) \\ \mathbf{P}^{ij}(t) &= \frac{1}{M-1} \sum_{k=1}^M \left\{ \left[\phi^i(t; \mathbf{x}_0^k, t_0) - m^i(t) \right] \left[\phi^j(t; \mathbf{x}_0^k, t_0) - m^j(t) \right] \right\} \end{aligned} \quad (26)$$

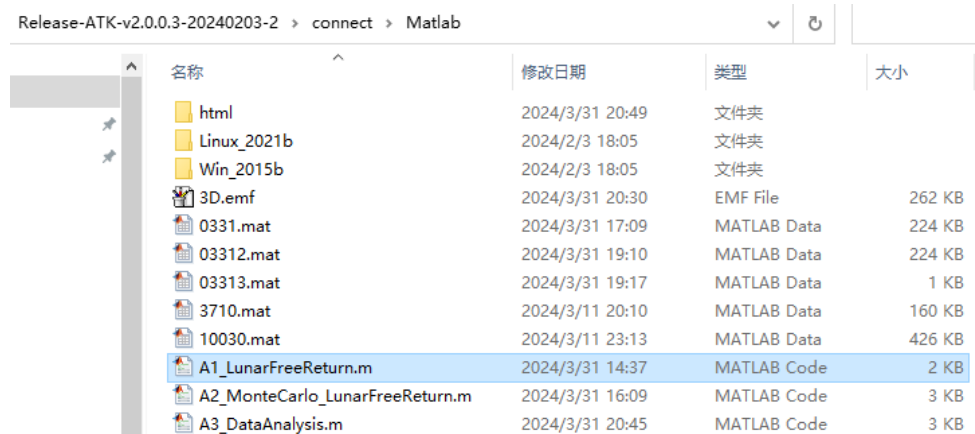
理论上，随着样本点数不断增大 $M \rightarrow \infty$, MC 方法获得的结果可不断逼近终端偏差的真实分布情况。

5.2 标称轨道仿真

5.2.1. 交互配置检查与文件目录设置

首先，确认 ATK 软件与 MATLAB 间交互配置是否正确（文章一：ATK 的安装与配置；ATK 二次开发配置）。

打开 ATK 软件，关闭弹出的“欢迎使用 ATK”窗口，保持 ATK 界面为无场景状态。之后打开 MATLAB，工作目录在 ATK 安装文件夹内，2.0 版本 ATK 为：Release-ATK-v2.0.0.3-20240203-2\connect\Matlab，2.1 版本 ATK 为：Release-ATK-v2.1.0.0-20240329-3\IntegratingWithATK\connect\Matlab\Win_2015b。在 MATLAB 中新建 m 文件，命名为 A1_LunarFreeReturn.m，并保存，保存路径同上。路径及代码文件名称如图所示。



名称	修改日期	类型	大小
html	2024/3/31 20:49	文件夹	
Linux_2021b	2024/2/3 18:05	文件夹	
Win_2015b	2024/2/3 18:05	文件夹	
3D.emf	2024/3/31 20:30	EMF File	262 KB
0331.mat	2024/3/31 17:09	MATLAB Data	224 KB
03312.mat	2024/3/31 19:10	MATLAB Data	224 KB
03313.mat	2024/3/31 19:17	MATLAB Data	1 KB
3710.mat	2024/3/11 20:10	MATLAB Data	160 KB
10030.mat	2024/3/11 23:13	MATLAB Data	426 KB
A1_LunarFreeReturn.m	2024/3/31 14:37	MATLAB Code	2 KB
A2_MonteCarlo_LunarFreeReturn.m	2024/3/31 16:09	MATLAB Code	3 KB
A3_DataAnalysis.m	2024/3/31 20:45	MATLAB Code	3 KB

5.2.2. 标称轨道生成代码实现

地月自由返回轨道的标称轨道生成代码如下，共分为 2 部分，第 1 部分实现与 ATK 建立连接、场景初始化功能，第 2 部分实现标称轨道参数设置及仿真功能。

```
%% 初始化

clear

clc

%与 ATK 建立连接

conID = atkOpen();

%新建场景
```

```
atkConnect(conID, 'New', '/ Scenario LunarFreeReturn');
```

```
%设置场景历元时刻
```

```
atkConnect(conID, 'SetAnalysisTimePeriod', '* "4 Apr 2033 10:05:23.00" "10 Apr 2033 07:01:00.00"');
```

```
%重置场景开始时间
```

```
atkConnect(conID, 'Animate', '* Reset');
```

```
%% 标称轨道
```

```
%新建卫星
```

```
atkConnect(conID, 'New', '/ Satellite sat0');
```

```
%设置轨道坐标系、坐标类型、预报器类型为 HPOP、计算步长、历元时刻、初始位置速度
```

```
atkConnect(conID, 'SetState', ['*/Satellite/sat0 Cartesian HPOP "4 Apr 2033 10:05:00.00" ' ...  
' "10 Apr 2033 07:01:00.00" 60 j2000 "4 Apr 2033 10:05:00.00" ' ...  
'3496183.80951811 -5158739.1800055 -2106180.2471557 7588.16412302495 6884.317409025 -3872.7905851142']);
```

```
%不考虑大气阻力摄动
```

```
atkConnect(conID, 'HPOP', '*/Satellite/sat0 Drag off');
```

```
%考虑太阳光压摄动，设置相关参数
```

```
atkConnect(conID, 'HPOP', '*/Satellite/sat0 Force SRP on Model Spherical 1 20.0');
```

```
%考虑非球形引力摄动，设置相关参数
```

```
atkConnect(conID, 'HPOP', ['*/Satellite/sat0 Force Gravity ' ...  
'"C:\Users\A314-hcy\Desktop\Release-ATK-v2.0.0.3-20240203-2\AstroData\Earth\JGM3.grv' 6 6]);
```

```
%考虑三体引力摄动，设置相关参数
```

```
atkConnect(conID, 'HPOP', '*/Satellite/sat0 Force ThirdBodyGravity true true 0 EGM2008 0 0 false 0  
JGM3 1 2');
```

```
%设置积分算法
```

```
atkConnect(conID, 'HPOP', '*/Satellite/sat0 Integrator IntegMethod RK8');
```

```
%开始仿真
```

```
atkConnect(conID, 'Animate', '* Start');
```

```
%记录标称轨道终端时刻位置速度，放置在数组 position0 中
```

```
position0=str2num(atkConnect(conID, 'Position', '*/Satellite/sat0 "10 Apr 2033 08:01:38.04"');
```

接下来将分块介绍各部分的实现功能。

5.2.3. 场景初始化代码详解

```
%与 ATK 建立连接
conID = atkOpen();

%新建场景
atkConnect(conID, 'New', '/ Scenario LunarFreeReturn');

%设置场景历元时刻
atkConnect(conID, 'SetAnalysisTimePeriod', '* "4 Apr 2033 10:05:00.00" "10 Apr 2033 07:01:00.00"');

%重置场景开始时间
atkConnect(conID, 'Animate', '* Reset');
```

以上代码实现以下功能：与 ATK 建立连接；新建场景并命名为 LunarFreeReturn；设置场景历元起始时刻 4 Apr 2033 10:05:00.00，终止时刻 10 Apr 2033 07:01:00.00。终端时刻接近地月返回轨道的近地点时刻。

5.2.4. 标称轨道卫星参数设置代码详解

```
%新建卫星
atkConnect(conID, 'New', '/ Satellite sat0');

%设置轨道坐标系、坐标类型、预报器类型为 HPOP、计算步长、历元时刻、初始位置速度
atkConnect(conID, 'SetState', ['*/Satellite/sat0 Cartesian HPOP "4 Apr 2033 10:05:00.00" ' ...
    ' "10 Apr 2033 07:01:00.00" 60 j2000 "4 Apr 2033 10:05:00.00" ' ...
    '3496183.80951811 -5158739.1800055 -2106180.2471557 7588.16412302495 6884.317409025 -3872.7905851142']);
```

以上代码实现以下功能：在上述新建的场景中添加卫星，命名为‘sat0’；设置参考坐标系为地心 J2000 坐标系（“J2000”）；设置坐标类型为直角坐标（“Cartesian”）；设置轨道预报器类型为高精度轨道预报（“HPOP”）；设置仿真步长为 60 sec；根据前序轨道设计结果（链接），设置地月转移轨道初始历元为 2033-4-4 10:05:00，分别设置初始位置和初始速度：

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^T = \begin{bmatrix} 3496183.809518110 \\ -5158739.180005480 \\ -2106180.247155680 \end{bmatrix}^T \quad (\text{m}) \quad (27)$$

$$\begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix}^T = \begin{bmatrix} 7588.164123024952 \\ 6884.317409024999 \\ -3872.790585114183 \end{bmatrix}^T \quad (\text{m/s}) \quad (28)$$

`%不考虑大气阻力摄动`

```
atkConnect(conID, 'HPOP', '*Satellite/sat0 Drag off');
```

`%考虑太阳光压摄动，设置相关参数`

```
atkConnect(conID, 'HPOP', '*Satellite/sat0 Force SRP on Model Spherical 1 20.0');
```

`%考虑非球形引力摄动，设置相关参数`

```
atkConnect(conID, 'HPOP', ['*Satellite/sat0 Force Gravity ' ...
```

```
''C:\Users\A314-hcy\Desktop\Release-ATK-v2.0.0.3-20240203-2\AstroData\Earth\JGM3.grv' 6 6]);
```

`%考虑三体引力摄动，设置相关参数`

```
atkConnect(conID, 'HPOP', '*Satellite/sat0 Force ThirdBodyGravity true true 0 EGM2008 0 0 false 0
```

```
JGM3 1 2');
```

`%设置积分算法`

```
atkConnect(conID, 'HPOP', '*Satellite/sat0 Integrator IntegMethod RK8');
```

以上代码实现高精度轨道的摄动项设置。分别对高精度轨道预报器的太阳光压摄动、地球非球形引力摄动、三体引力摄动、数值算法进行设置。其中地球非球形引力摄动模型为“JGM3”，阶数、次数均为 6；三体引力摄动选择太阳、月球，月球引力场模型类型选择点质量模型，阶数、次数均为 0；不考虑大气阻力摄动和太阳光压摄动；积分算法选择“RK8”。

`%开始仿真`

```
atkConnect(conID, 'Animate', '* Start');
```

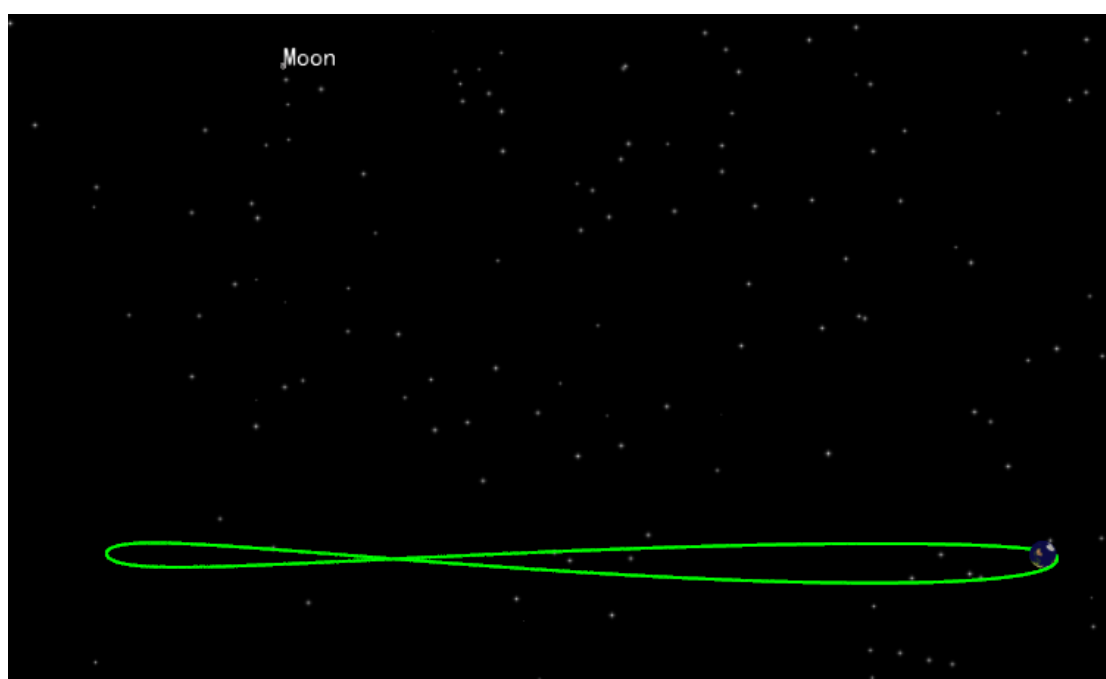
```
%记录标称轨道终端时刻位置速度，放置在数组 position0 中
```

```
position0=str2num(atkConnect(conID, 'Position',' */Satellite/sat0 "10 Apr 2033 07:01:00.00"));
```

上述代码实现以下功能：开始仿真，记录卫星在终端时刻的位置和速度，并保存在数组中，数组命名为 `position0`。

5.2.5. 标称轨道仿真结果查看

按照上述操作步骤完成卫星各参数设置后，点击 MATLAB 中的“运行”按钮，可以看到 ATK 界面中出现了对应场景（场景里面包含卫星），双击卫星，查看各项参数可以发现均与上述代码设置保持一致。读者可以在 ATK 三维视图中查看上述标称轨道在地心 J2000 坐标系中的飞行轨迹，轨迹如图所示。



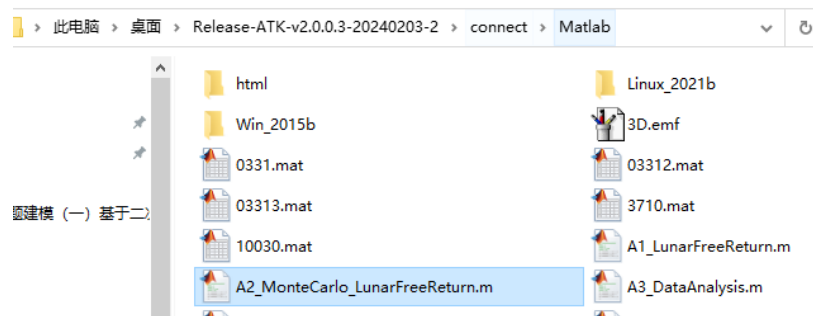
5.3 蒙特卡洛打靶仿真过程

5.3.1. 蒙特卡洛仿真前期准备工作

开始蒙特卡洛仿真之前，先请读者确认一下 ATK 的界面，此时应该仍是标称轨道的场景，我们需要关闭这个场景，点击 ATK 界面“开始”栏的“×”，弹出“是否保存当前想定”窗口，点击“是”，如图所示。

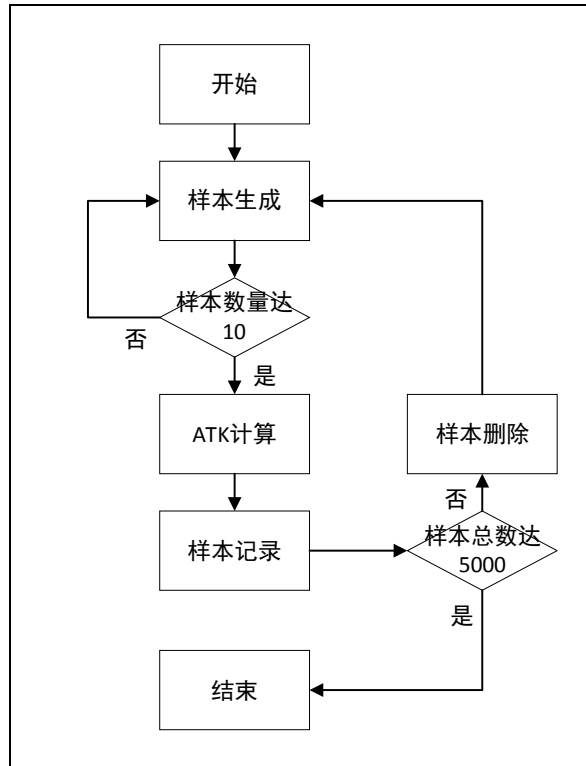


将该 xml 文件保存在自己喜欢的位置，方便随时查看标称轨道的仿真结果。操作完之后，ATK 变为无场景状态，才可以进行下一步的代码原理解析及编写。然后打开 MATLAB，工作目录在 ATK 安装文件夹内，2.0 版本 ATK 为：Release-ATK-v2.0.0.3-20240203-2\connect\Matlab，2.1 版本 ATK 为：Release-ATK-v2.1.0.0-20240329-3\IntegratingWithATK\connect\Matlab\Win_2015b。在 MATLAB 中新建 m 文件，命名为 A2_MonteCarlo_LunarFreeReturn.m，并保存，保存路径同上。如图所示。



5.3.2. 偏差演化的蒙特卡洛方法流程

得到地月自由返回轨道的标称数据之后，接下来针对轨道初始偏差演化进行蒙特卡洛仿真。本文仅考虑航天器导航误差，误差均值及标准差均在地心 J2000 坐标系中描述。蒙特卡洛仿真样本数为 5000。下面先介绍代码实现思路：首先，生成带有随机偏差的样本，以 10 个样本为一组，在 ATK 里进行高精度轨道外推，并记录卫星终端时刻的位置速度数据；然后，为充分利用 ATK 算力，且及时释放内存，删除这 10 个样本点，并重新生成 10 个样本，进行高精度轨道外推，直至样本总数达 5000，仿真结束。思路流程图如图所示。完整代码也一并附上。



%% 初始化

%与 ATK 建立连接

```
conID = atkOpen();
```

%新建场景

```
atkConnect(conID, 'New', '/ Scenario LunarFreeReturn');
```

%设置场景历元时刻

```
atkConnect(conID, 'SetAnalysisTimePeriod', '* "4 Apr 2033 10:05:00.00" "10 Apr 2033 07:01:00.00"');
```

%重置场景开始时间

```
atkConnect(conID, 'Animate', '* Reset');
```

%给定标称轨道的初始位置速度

```
Cartesian=[3496183.80951811,-5158739.1800055,-2106180.2471557,7588.16412302495,6884.317409025,-3872.7905851142];
```

%% 偏差轨道

%用来记录终端时刻偏差的位置速度，共 5000 个样本，每个样本 3 个位置数据、3 个速度数据

```
position_dev=zeros(5000,6);
```

```
for i=0:1:499
```

```

%10 个样本为 1 组

for j=0:1:9

    %转为字符串，适应 atkConnect 命令

    satellite=[' / Satellite sat',num2str(i),num2str(j)];

    %新建卫星

    atkConnect(conID,'New',satellite);

    %位置随机偏差的标准差为 1000m

    location=randn([1,3])*1000+Cartesian(1:3);

    %速度随机偏差的标准差为 0.2m/s

    velocity=randn([1,3])*0.2+Cartesian(4:6);

    %转为字符串，适应 atkConnect 命令

    x=[' ',num2str(location(1))];

    y=[' ',num2str(location(2))];

    z=[' ',num2str(location(3))];

    dx=[' ',num2str(velocity(1))];

    dy=[' ',num2str(velocity(2))];

    dz=[' ',num2str(velocity(3))];

    %卫星初始位置速度发生偏差，创建样本点

    state=['*/Satellite/sat',num2str(i),num2str(j),' Cartesian  HPOP  "4 Apr 2033 10:05:00.00"  "10 Apr 2033
07:01:00.00"  60  j2000  "4 Apr 2033 10:05:00.00",x,y,z,dx,dy,dz];

    atkConnect(conID, 'SetState',state);

    %不考虑大气阻力摄动

    sate_Drag=[' */Satellite/sat',num2str(i),num2str(j),' Drag  off'];

    atkConnect(conID, 'HPOP', sate_Drag);

    %设置太阳光压摄动

    sate_SRP=['*/Satellite/sat',num2str(i),num2str(j),' Force  SRP  on Model Spherical 1 20.0'];

    atkConnect(conID, 'HPOP', sate_SRP);

    %设置地球非球形摄动

```

```

    sata_Grav=['*/Satellite/sat',num2str(i),num2str(j),' Force Gravity "C:\Users\A314-hcy\Desktop\Release-ATK-
v2.0.0.3-20240203-2\AstroData\Earth\JGM3.grv' 6 6'];

    atkConnect(conID, 'HPOP', sata_Grav);

    %设置三体引力摄动

    sata_ThirdBody=['*/Satellite/sat',num2str(i),num2str(j),' Force ThirdBodyGravity true true 0 EGM2008 0
0 false 0 JGM3 1 2'];

    atkConnect(conID, 'HPOP', sata_ThirdBody);

    %设置积分算法

    sata_Inte=['*/Satellite/sat',num2str(i),num2str(j),' Integrator IntegMethod RK8'];

    atkConnect(conID, 'HPOP',sata_Inte);

end

%10 个样本为一组开始仿真

atkConnect(conID, 'Animate', '* Start');

%依次记录 10 个样本的数据，记录完成后为释放内存立即删除该样本点

for j=0:9

    %记录位置速度

    sata_position=[' */Satellite/sat',num2str(i),num2str(j),' "10 Apr 2033 07:01:00.00"'];

    position_dev(10*i+j+1,:)=str2num(atkConnect(conID, 'Position',sata_position));

    %删除样本点

    sata_unload=[' / */Satellite/sat',num2str(i),num2str(j)];

    atkConnect(conID, 'Unload',sata_unload);

end

%每次开始一组样本仿真之前，重置场景时间

atkConnect(conID, 'Animate', '* Reset');

end

```

接下来分块介绍代码功能。

5.3.3. 场景初始化代码详解

```

    %与 ATK 建立连接

conID = atkOpen();

%新建场景

atkConnect(conID, 'New', '/ Scenario LunarFreeReturn');

%设置场景历元时刻

atkConnect(conID, 'SetAnalysisTimePeriod', '* "4 Apr 2033 10:05:00.00" "10 Apr 2033 07:01:00.00"');

%重置场景开始时间

atkConnect(conID, 'Animate', '* Reset');

%给定标称轨道的初始位置速度

Cartesian=[3496183.80951811,-5158739.1800055,-2106180.2471557,7588.16412302495,6884.317409025,-
3872.7905851142];

```

以上代码实现仿真场景的初始化。依次实现与 ATK 建立连接、新建场景并命名为“LunarFreeReturn”、设置场景历元、重置场景开始时间、设置标称轨道的初始位置速度。场景历元和标称轨道位置速度均由前序轨道设计给出。

5.3.4. 样本生成与分组仿真代码详解

```

for j=0:1:9

    %转为字符串，适应 atkConnect 命令

    satellite=['/ Satellite sat',num2str(i),num2str(j)];

    %新建卫星

    atkConnect(conID,'New',satellite);

    %位置随机偏差的标准差为 1000m

    location=randn([1,3])*1000+Cartesian(1:3);

    %速度随机偏差的标准差为 0.2m/s

    velocity=randn([1,3])*0.2+Cartesian(4:6);

    %转为字符串，适应 atkConnect 命令

```

```

x=[' ',num2str(location(1))];

y=[' ',num2str(location(2))];

z=[' ',num2str(location(3))];

dx=[' ',num2str(velocity(1))];

dy=[' ',num2str(velocity(2))];

dz=[' ',num2str(velocity(3))];

%卫星初始位置速度发生偏差，创建样本点

state=['*/Satellite/sat',num2str(i),num2str(j),' Cartesian HPOP "4 Apr 2033 10:05:00.00" "10 Apr 2033
07:01:00.00" 60 j2000 "4 Apr 2033 10:05:00.00"',x,y,z,dx,dy,dz];

atkConnect(conID, 'SetState',state);

%不考虑大气阻力摄动

sate_Drag=['*/Satellite/sat',num2str(i),num2str(j),' Drag off'];

atkConnect(conID, 'HPOP', sate_Drag);

%设置太阳光压摄动

sate_SRP=['*/Satellite/sat',num2str(i),num2str(j),' Force SRP on Model Spherical 1 20.0'];

atkConnect(conID, 'HPOP', sate_SRP);

%设置地球非球形摄动

sata_Grav=['*/Satellite/sat',num2str(i),num2str(j),' Force Gravity "C:\Users\A314-hcy\Desktop\Release-ATK-
v2.0.0.3-20240203-2\AstroData\Earth\JGM3.grv" 6 6'];

atkConnect(conID, 'HPOP', sata_Grav);

%设置三体引力摄动

sata_ThirdBody=['*/Satellite/sat',num2str(i),num2str(j),' Force ThirdBodyGravity true true 0 EGM2008 0
0 false 0 JGM3 1 2'];

atkConnect(conID, 'HPOP', sata_ThirdBody);

%设置积分算法

sata_Inte=['*/Satellite/sat',num2str(i),num2str(j),' Integrator IntegMethod RK8'];

atkConnect(conID, 'HPOP',sata_Inte);

end

```

```
%10 个样本为一组仿真
```

```
atkConnect(conID, 'Animate', '* Start');
```

以上代码实现以下功能：新建卫星；设置加入初始误差后的卫星轨道参数，其中导航误差假设为零均值高斯分布，位置标准差： $\sigma_{\delta r} = [1000, 1000, 1000](m)$ ，速度标准差： $\sigma_{\delta v} = [0.2, 0.2, 0.2](m/s)$ 。为充分利用 ATK 算力，及时释放内存，以 10 个样本点为一组同时进行高精度轨道外推。此外读者也可以尝试 20、30 等样本分组形式。

```
%依次记录 10 个样本的数据，记录完成后为释放内存立即删除该样本点
```

```
for j=0:9
```

```
    %记录位置速度
```

```
    sata_position=[' */Satellite/sat', num2str(i), num2str(j), ' "10 Apr 2033 07:01:00.00"'];
```

```
    position_dev(10*i+j+1,:)=str2num(atkConnect(conID, 'Position', sata_position));
```

```
    %删除样本点
```

```
    sata_unload=[' */Satellite/sat', num2str(i), num2str(j)];
```

```
    atkConnect(conID, 'Unload', sata_unload);
```

```
end
```

```
%每次开始一组样本仿真之前，重置场景时间
```

```
atkConnect(conID, 'Animate', '* Reset');
```

```
end
```

以上代码实现以下功能：依次记录样本的终端偏差数据，包括位置和速度偏差，保存在数组“position1”中；删除样本点；重置场景时间。

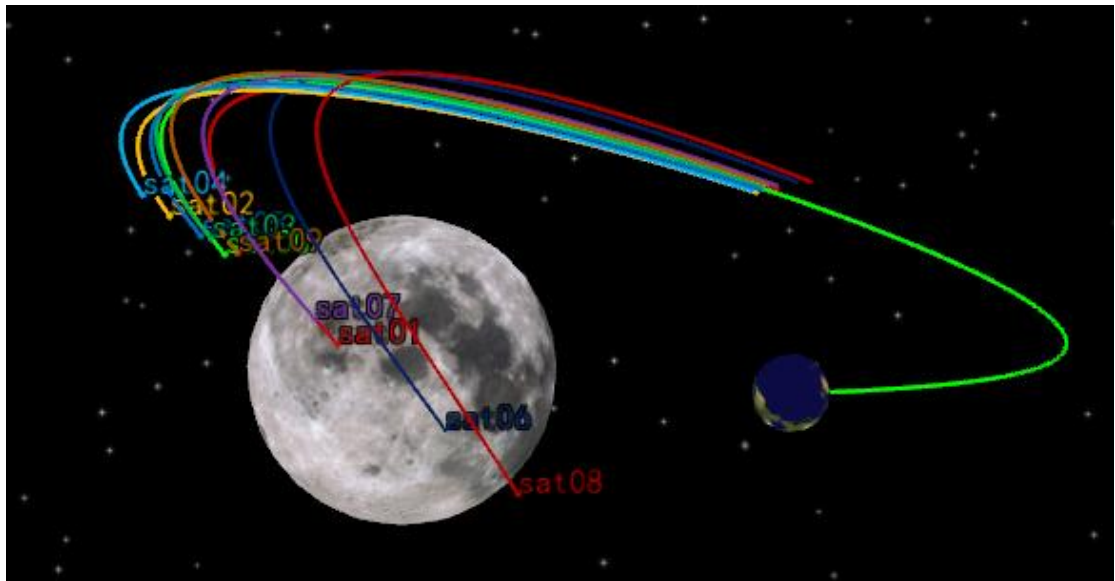
5.3.5. 仿真过程数据查看

按照上述代码示例完成编写后，点击 MATLAB 的“运行”按钮，开始批量生成样本进行蒙特卡洛仿真。（用不同的标识提醒作者，区别于正常的操作）读者也可以在 MATLAB 代码“删除样本点”处打上断点，以观察每组样本点所属卫星

的运行轨迹，如图所示，

```
55     for j=0:9
56         %记录位置速度
57         sata_position=[' */Satellite/sat',num2str(i),num2str(j),' "10 Apr 2033 10:00:00.00"'];
58         position_dev(10*i+j+1,:)=str2num(atkConnect(conID, 'Position',sata_position));
59         %删除样本点
60         sata_unload=[' */Satellite/sat',num2str(i),num2str(j)];
61         atkConnect(conID, 'Unload',sata_unload);
62     end
```

运行至断点后，可在 ATK 三维视图中查看该组样本点对应的蒙特卡洛仿真过程，卫星的飞行轨迹在不同初始导航误差的情况下呈现出明显的差异，第一组样本点地心 J2000 坐标系下的飞行轨迹在 ATK 三维视图中如下所示。



5.4 偏差结果分析

为方便分析偏差的演化情况，将自由返回轨道划分为三个阶段：出发近地点时刻至出发后两天；出发后两天至近月点时刻；近月点时刻至返回近地点时刻。保存每个阶段结束时刻的轨道偏差数据，作为下一阶段的输入。通过对轨道外推进行分段，可以减少重复运行，提高效率。

三个阶段的偏差结果数据，如图所示，标称轨道各阶段的数据同理可获得。

名称	值
position1	[-1.2890e+08,2.852...
position2	[-1.6265e+08,3.446...
position3	[-1.7095e+07,2.684...
position_dev	5000x6 double
position_dev2	5000x6 double
position_dev3	5000x6 double

将上述标称轨道仿真、蒙特卡洛打靶仿真全部运行完毕后，就不再需要与 ATK 连接，不过需要确保 MATLAB 的工作区有前述标称轨道数据和偏差数据数组。

接下来开始进行偏差结果分析。新建 m 文件，文件路径同以上两个代码文件，工作目录为 IntegratingWithATK\connect\Matlab\Win_2015b，命名为 DataAnalysis.m。

```

load('0604-standard.mat');

load('0604-error.mat');

dev1=zeros(5000,6);

dev2=zeros(5000,6);

dev3=zeros(5000,6);

pos_dev1=zeros(5000,6);

pos_dev2=zeros(5000,6);

pos_dev3=zeros(5000,6);

for i=1:5000

    dev1(i,:)=(position_dev(i,:)-position1)*1e-3;%%偏差轨道与标称轨道相对位置速度偏差

    dev2(i,:)=(position_dev2(i,:)-position2)*1e-3;

    dev3(i,:)=(position_dev3(i,:)-position3)*1e-3;

    pos_dev1(i,:)=position_dev(i,:)*1e-3;%%偏差轨道绝对位置速度偏差

    pos_dev2(i,:)=position_dev2(i,:)*1e-3;%%偏差轨道绝对位置速度偏差

    pos_dev3(i,:)=position_dev3(i,:)*1e-3;%%偏差轨道绝对位置速度偏差

```



```

end

dev_p=mean(dev3(:,1:3));

dev_v=mean(dev3(:,4:6));

std=std(dev3);

figure;% 绘制两天后误差三维散点图

scatter3(dev1(:,1), dev1(:,2), dev1(:,3), '.');

hold on;

scatter3(mean(dev1(:,1)), mean(dev1(:,2)), mean(dev1(:,3)), 'filled');

plot3sigma3D([mean(dev1(:,1));mean(dev1(:,2));mean(dev1(:,3))],cov(dev1(:,1:3)),15);

lgd=legend('偏差分布','偏差均值','3σ(标准差椭圆)');

lgd.Location='northwest';

title(lgd,' 偏差结果');

title('终端位置偏差分布 (单位 : 千米) ');

xlabel('X 轴');

ylabel('Y 轴');

zlabel('Z 轴');

figure;% 绘制两天后绝对位置三维散点图

scatter3(pos_dev1(:,1), pos_dev1(:,2), pos_dev1(:,3), '.');

title('终端位置分布 (单位 : 千米) ');

xlabel('X 轴');

ylabel('Y 轴');

zlabel('Z 轴');

figure;% 绘制近月点误差三维散点图

```

```

scatter3(dev2(:,1), dev2(:,2), dev2(:,3), '.');

hold on;

scatter3(mean(dev2(:,1)), mean(dev2(:,2)), mean(dev2(:,3)), 'filled');

plot3sigma3D([mean(dev2(:,1));mean(dev2(:,2));mean(dev2(:,3))],cov(dev2(:,1:3)),15);

lgd=legend('偏差分布','偏差均值','3σ(标准差椭圆)');

lgd.Location='northwest';

title(lgd,' 偏差结果);

title('终端位置偏差分布 (单位 : 千米) ');

xlabel('X 轴');

ylabel('Y 轴');

zlabel('Z 轴');

figure;% 绘制近月点绝对位置三维散点图

scatter3(pos_dev2(:,1), pos_dev2(:,2), pos_dev2(:,3), '.');

hold on

r_M=1737.10;

[x,y,z] = sphere;

X = -161709.6466355966+x*r_M;

Y = 343085.1082100035+y*r_M;

Z = 109073.0619716160+z*r_M;

mesh(X,Y,Z);

lgd=legend('终端轨迹分布','月球');

title('终端位置分布 (单位 : 千米) ');

xlabel('X 轴');

ylabel('Y 轴');

zlabel('Z 轴');

```

```

figure;% 绘制返回近地点误差三维散点图

scatter3(dev3(:,1), dev3(:,2), dev3(:,3), '.');

hold on;

scatter3(mean(dev3(:,1)), mean(dev3(:,2)), mean(dev3(:,3)), 'filled');

plot3sigma3D([mean(dev3(:,1));mean(dev3(:,2));mean(dev3(:,3))],cov(dev3(:,1:3)),15);

lgd=legend('偏差分布','偏差均值','3σ(标准差椭圆)');

lgd.Location='northwest';

title(lgd,' 偏差结果);

title('终端位置偏差分布 (单位 : 千米) ');

xlabel('X 轴');

ylabel('Y 轴');

zlabel('Z 轴');

figure;% 绘制返回近地点绝对位置三维散点图

scatter3(pos_dev3(:,1), pos_dev3(:,2), pos_dev3(:,3), '.');

hold on

r_E=6378.137;

[x,y,z] = sphere;

X = x*r_E;

Y = y*r_E;

Z = z*r_E;

mesh(X,Y,Z);

text(6378.137,6378.137,6378.137,'地球');

lgd=legend('终端轨迹分布','地球');

title('终端位置分布 (单位 : 千米) ');

xlabel('X 轴');

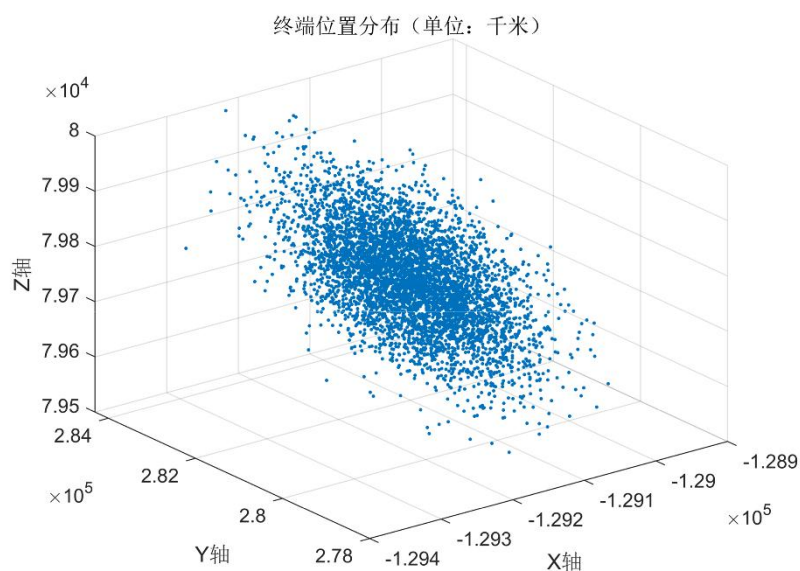
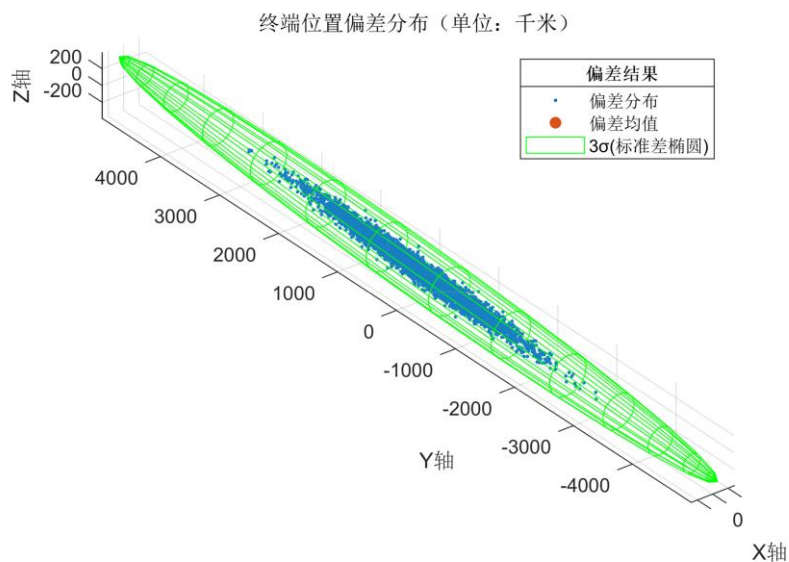
ylabel('Y 轴');

zlabel('Z 轴');

```

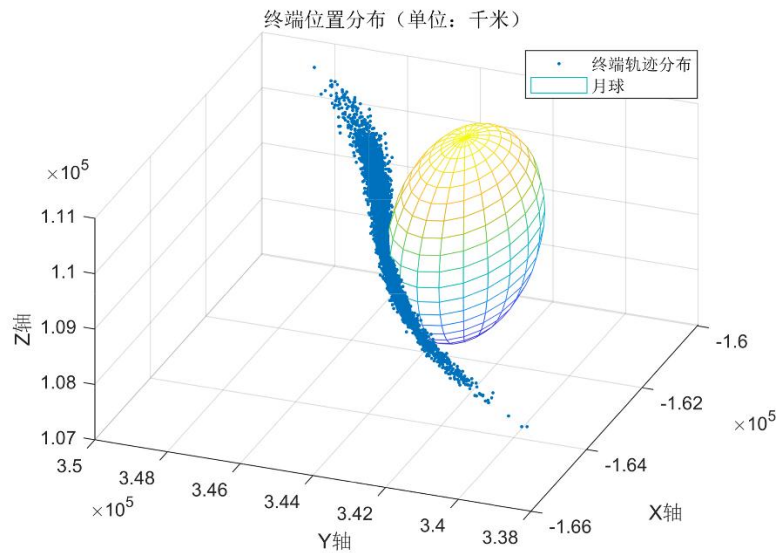
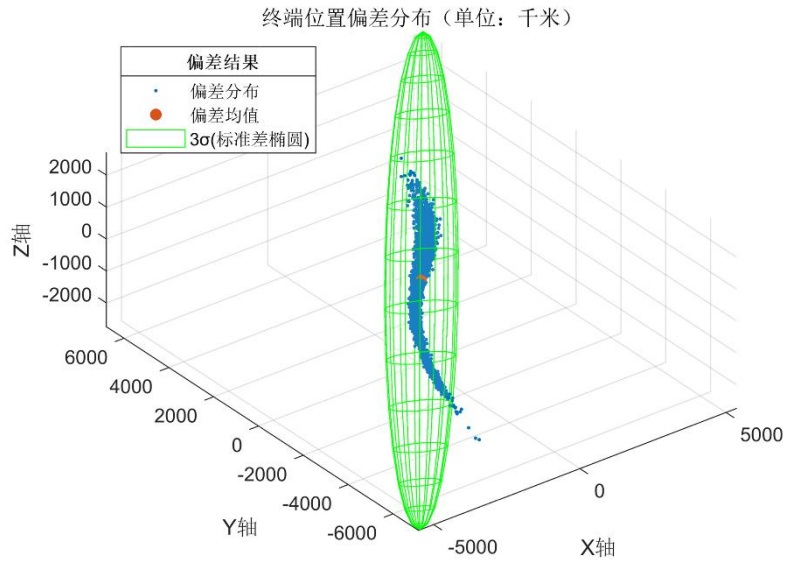
上述代码实现以下功能：计算终端位置、速度误差均值和协方差矩阵；绘制终端误差的三维分布图。

出发后两天，观察偏差的演化情况如下所示，



可以看到，在预报 2 天的情况下，偏差演化情况仍基本保持高斯分布特性， 3σ 误差椭球能够很好描述误差的分布情况。

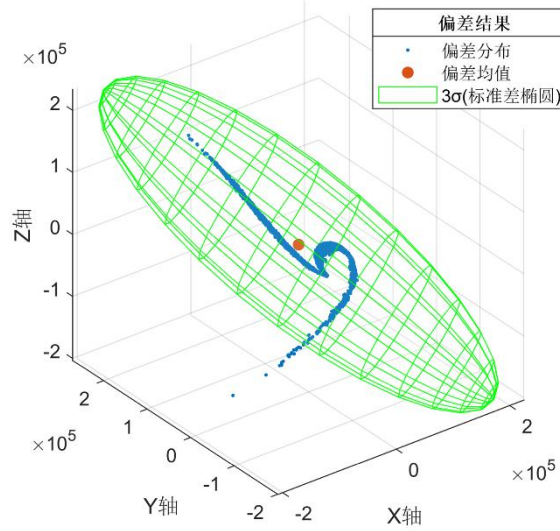
近月点时刻，即“7 Apr 2033 05:48:14.88”，观察偏差的演化情况如下所示，



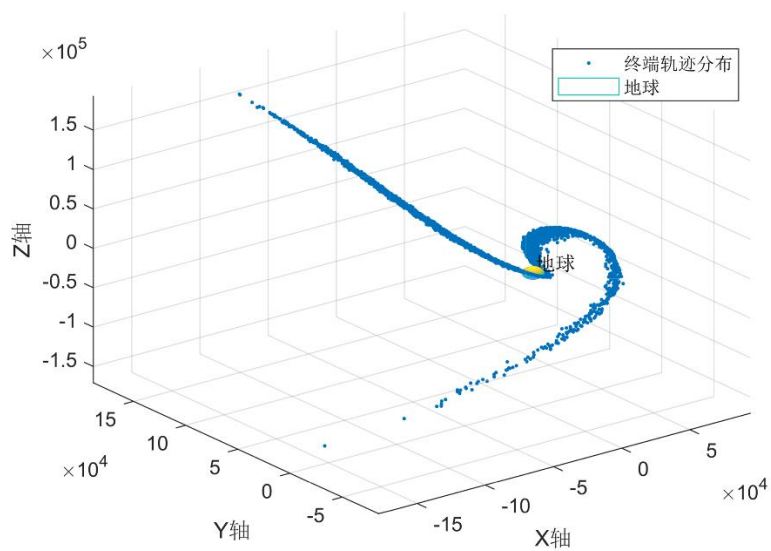
可以看到, 在预报到近月点的情况下, 偏差演化已经初现非高斯分布特性, 发生了变形, 3σ 误差椭球虽能包含绝大多数样本点, 但已不能完整表征其分布特性。

进一步预报到返回近地点时刻, 终端偏差在三维空间的分布以及终端位置的分布情况如下,

终端位置偏差分布 (单位: 千米)



终端位置分布 (单位: 千米)



可以看到, 偏差轨迹沿航天器标称轨道弧线发生弯曲, 初始高斯分布偏差已经演变为非高斯分布, 高斯分布 3σ 误差椭球已不能描述其分布情况。

终端位置偏差的样本均值为:

$$D_p = [9079.1 \quad 35060.0 \quad 15229.0](\text{km}) \quad (29)$$

终端速度偏差的样本均值为:

$$D_v = [-1.7752 \quad 3.9374 \quad 2.3374](\text{km/s}) \quad (30)$$

附录

A1_LunarFreeReturn.m

%% 初始化

clear

clc

%与 ATK 建立连接

conID = atkOpen();

%新建场景

atkConnect(conID, 'New', '/ Scenario LunarFreeReturn');

%设置场景历元时刻

atkConnect(conID, 'SetAnalysisTimePeriod', '* "4 Apr 2033 10:05:00.00" "10 Apr 2033 07:01:00.00"');

%重置场景开始时间

atkConnect(conID, 'Animate', '* Reset');

%% 标称轨道

%新建卫星

atkConnect(conID, 'New', '/ Satellite sat0');

%设置轨道坐标系、坐标类型、预报器类型为 HPOP、计算步长、历元时刻、初始位置速度

atkConnect(conID, 'SetState', ['*/Satellite/sat0 Cartesian HPOP "4 Apr 2033 10:05:00.00" ' ...

' "10 Apr 2033 07:01:00.00" 60 j2000 "4 Apr 2033 10:05:00.00" ' ...

'3496183.80951811 -5158739.1800055 -2106180.2471557 7588.16412302495 6884.317409025 -3872.7905851142'];

%不考虑大气阻力摄动

atkConnect(conID, 'HPOP', ['*/Satellite/sat0 Drag off');

%考虑太阳光压摄动，设置相关参数

atkConnect(conID, 'HPOP', ['*/Satellite/sat0 Force SRP on Model Spherical 1 20.0');

%考虑非球形引力摄动，设置相关参数

atkConnect(conID, 'HPOP', ['*/Satellite/sat0 Force Gravity ' ...

""C:\Users\A314-hcy\Desktop\Release-ATK-v2.0.0.3-20240203-2\AstroData\Earth\JGM3.grv' 6 6]);

%考虑三体引力摄动，设置相关参数

atkConnect(conID, 'HPOP', ['*/Satellite/sat0 Force ThirdBodyGravity true true 0 EGM2008 0 0 false 0

```

JGM3 1 2');

%设置积分算法

atkConnect(conID, 'HPOP', '*Satellite/sat0 Integrator IntegMethod RK8');

%开始仿真

atkConnect(conID, 'Animate', '* Start');

%记录标称轨道终端时刻位置速度，放置在数组 position0 中

position0=str2num(atkConnect(conID, 'Position', '*Satellite/sat0 "10 Apr 2033 07:01:00.00"'));

```

A2_MonteCarlo_LunarFreeReturn.m

%% 初始化

```

%与 ATK 建立连接

conID = atkOpen();

%新建场景

atkConnect(conID, 'New', '/ Scenario LunarFreeReturn');

%设置场景历元时刻

atkConnect(conID, 'SetAnalysisTimePeriod', '* "4 Apr 2033 10:05:00.00" "10 Apr 2033 07:01:00.00"');

%重置场景开始时间

atkConnect(conID, 'Animate', '* Reset');

%给定标称轨道的初始位置速度

Cartesian=[3496183.80951811,-5158739.1800055,-2106180.2471557,7588.16412302495,6884.317409025,-
3872.7905851142];

```

%% 偏差轨道

```

%用来记录终端时刻偏差的位置速度，共 5000 个样本，每个样本 3 个位置数据、3 个速度数据

position_dev=zeros(5000,6);

for i=0:1:499

    %10 个样本为 1 组

    for j=0:1:9

        %转为字符串，适应 atkConnect 命令

        satellite=['/ Satellite sat',num2str(i),num2str(j)];

```



```

%新建卫星

atkConnect(conID,'New',satellite);

%位置随机偏差的标准差为 1000m

location=randn([1,3])*1000+Cartesian(1:3);

%速度随机偏差的标准差为 0.2m/s

velocity=randn([1,3])*0.2+Cartesian(4:6);

%转为字符串，适应 atkConnect 命令

x=[' ',num2str(location(1))];

y=[' ',num2str(location(2))];

z=[' ',num2str(location(3))];

dx=[' ',num2str(velocity(1))];

dy=[' ',num2str(velocity(2))];

dz=[' ',num2str(velocity(3))];

%卫星初始位置速度发生偏差，创建样本点

state=['*/Satellite/sat',num2str(i),num2str(j),' Cartesian HPOP "4 Apr 2033 10:05:00.00" "10 Apr 2033
08:01:38.04" 60 j2000 "4 Apr 2033 10:05:00.00"',x,y,z,dx,dy,dz];

atkConnect(conID, 'SetState',state);

%不考虑大气阻力摄动

sate_Drag=['*/Satellite/sat',num2str(i),num2str(j),' Drag off'];

atkConnect(conID, 'HPOP', sate_Drag);

%设置太阳光压摄动

sate_SRP=['*/Satellite/sat',num2str(i),num2str(j),' Force SRP on Model Spherical 1 20.0'];

atkConnect(conID, 'HPOP', sate_SRP);

%设置地球非球形摄动

sata_Grav=['*/Satellite/sat',num2str(i),num2str(j),' Force Gravity "C:\Users\A314-hcy\Desktop\Release-ATK-
v2.0.0.3-20240203-2\AstroData\Earth\JGM3.grv" 6 6'];

atkConnect(conID, 'HPOP', sata_Grav);

%设置三体引力摄动

```

```

    sata_ThirdBody=[ /*/Satellite/sat',num2str(i),num2str(j),' Force ThirdBodyGravity true true 0 EGM2008 0
0 false 0 JGM3 1 2'];

    atkConnect(conID, 'HPOP', sata_ThirdBody);

    %设置积分算法

    sata_Inte=[ /*/Satellite/sat',num2str(i),num2str(j),' Integrator IntegMethod RK8'];

    atkConnect(conID, 'HPOP',sata_Inte);

end

%10 个样本为一组开始仿真

atkConnect(conID, 'Animate', '* Start');

%依次记录 10 个样本的数据，记录完成后为释放内存立即删除该样本点

for j=0:9

    %记录位置速度

    sata_position=[ /*/Satellite/sat',num2str(i),num2str(j),' "10 Apr 2033 07:01:00.00"'];

    position_dev(10*i+j+1,:)=str2num(atkConnect(conID, 'Position',sata_position));

    %删除样本点

    sata_unload=[ /*/Satellite/sat',num2str(i),num2str(j)];

    atkConnect(conID, 'Unload',sata_unload);

end

%每次开始一组样本仿真之前，重置场景时间

atkConnect(conID, 'Animate', '* Reset');

end

```

```

function [xx,yy,zz] = plot3sigma3D(M0,P0,Nh)

%主轴坐标系

[VectP,ValueP] = eig(P0);

A = 3*sqrt(ValueP(1,1)); B = 3*sqrt(ValueP(2,2)); C = 3*sqrt(ValueP(3,3));

%产生主轴坐标系的 3D 图点

[X,Y,Z] = ellipsoid(0,0,0,A,B,C,Nh);

%将 3D 图点转换到原坐标系

```

```
for ii=1:Nh+1

    for jj=1:Nh+1

        VV = [X(ii,jj);Y(ii,jj);Z(ii,jj)];

        VV = VectP*VV;

        xx(ii,jj) = M0(1) + VV(1);

        yy(ii,jj) = M0(2) + VV(2);

        zz(ii,jj) = M0(3) + VV(3);

    end

end

mesh(xx, yy, zz,'FaceAlpha',0.1,'EdgeColor','green')

axis equal
```

二次开发案例集（下册）

6. 基于 ATK.Connect 模式的轨道快速转移的 C++实现

6.1 案例介绍

本案例实现半径为 6700km 的近地停泊轨道（LEO 轨道）快速转移到半径为 42164.197km 的地球同步轨道（GEO 轨道）的轨道机动规划设计。案例使用基于 Connect 模式的 C++客户端实现。

本案例使用 ATK.Connect 模式提供的 C++客户端程序，使用接口函数进行数据解析与传递，完成案例实现。案例提供客户端及接口函数如下图所示。

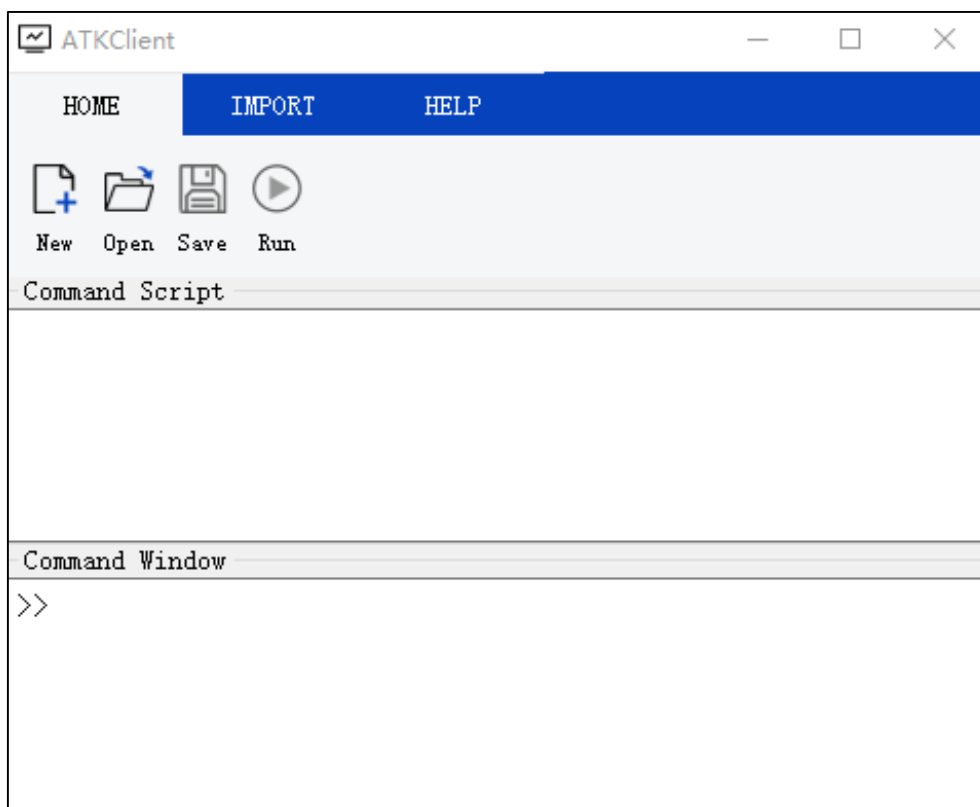


图 6.1-1 客户端界面

```
conID = atkOpen();  
atkConnect(conID, 'New', '/ Scenario FastTransfer');  
atkClose(conID);
```

图 6.1-2 接口函数

通过编辑客户端脚本，使用脚本命令新建想定及属性设置、新建卫星及属性设置和运行机动规划，完成案例构建，如下图所示。

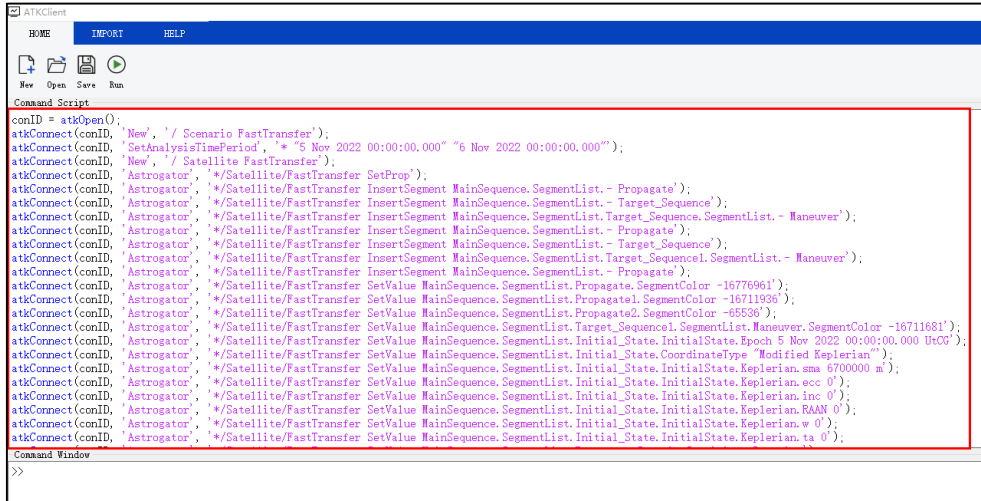


图 6.1-3 命令输入

使用获取报告命令，可查看报告数据，如下图所示：

```

>> atkConnect(conID, 'Report_RM', '* /Satellite/FastTransfer Style "Position" TimePeriod "5 Nov 2022 01:00:00.000" "5 Nov 2022 02:00:00.000"')

```

DateTime	x (m)	y (m)	z (m)
2022-11-05 01:00:03.854310	-3487395.959063	-5702997.687331	37.031653
2022-11-05 01:00:15.535390	-3410097.417406	-5749680.020446	37.029844
2022-11-05 01:00:26.635553	-3336066.688106	-5793069.453548	37.018672
2022-11-05 01:00:38.175524	-3258520.590154	-5837166.749058	36.997375
2022-11-05 01:00:49.883416	-3179253.654062	-5880844.254668	36.965755
2022-11-05 01:01:01.690623	-3098724.688664	-5923801.463142	36.923721
2022-11-05 01:01:13.704185	-3016196.060701	-5966376.964466	36.870564
2022-11-05 01:01:25.721903	-2933057.034431	-6007816.221633	36.806976
2022-11-05 01:01:37.749525	-2849282.934988	-6048129.080897	36.732993
2022-11-05 01:01:50.005486	-2763352.158938	-6088004.974835	36.647069
2022-11-05 01:02:05.166094	-2656290.709703	-6135640.546475	36.526264
2022-11-05 01:02:21.642629	-2539012.860386	-6185274.506404	36.377120
2022-11-05 01:02:37.455551	-2425592.323277	-6230800.747409	36.216960
2022-11-05 01:02:53.454188	-2310015.607737	-6274744.235187	36.038526
2022-11-05 01:03:08.783838	-2198530.559025	-6314838.442791	35.852700
2022-11-05 01:03:25.321285	-2077489.339259	-6355869.485424	35.636703

图 6.1-4 报告数据显示

运行客户端脚本文件，ATK 响应脚本命令，新建想定和构建卫星轨道如下图所示。

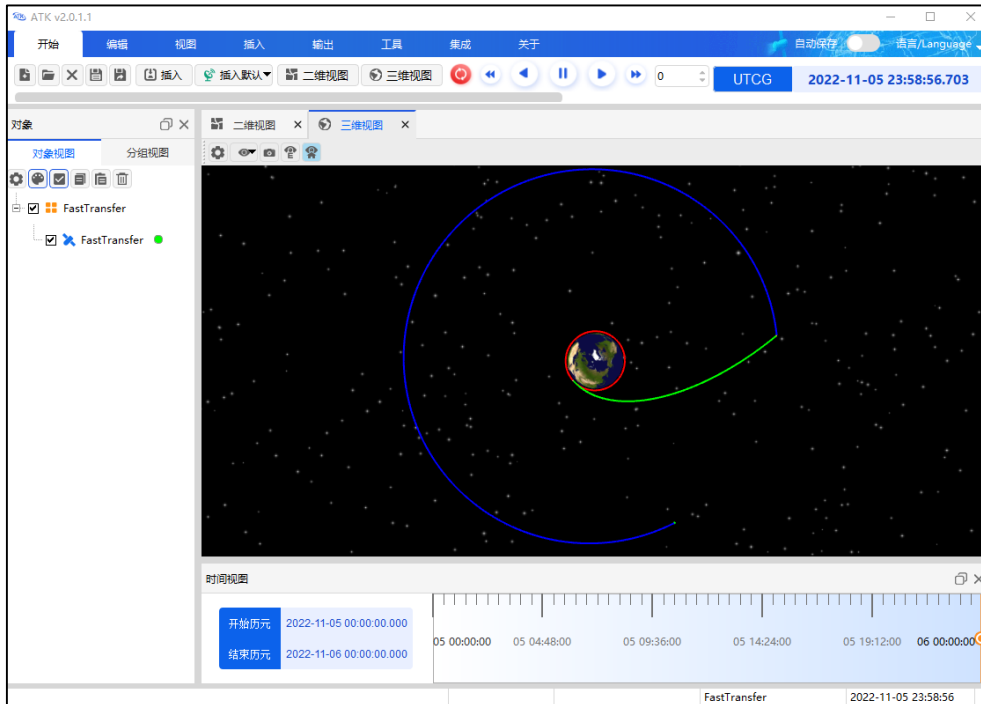


图 6.1-5 快速转移轨道

6.2 案例实现

本案例使用 ATK.Connect 提供的 C++客户端程序，实现 ATK 与 C++客户端之间的数据交互，案例实现流程如下所示。

6.2.1. 启动 ATK 软件，关闭“欢迎使用 ATK”对话框，在“集成”菜单栏中，单击“客户端”按钮，弹出客户端对话框。

6.2.2. 在“HOME”菜单栏中，点击“New”按钮新建脚本文件，在“Command Script”对话框输入脚本命令（代码可以直接拷贝到对话框，注释去掉）。脚本命令代码说明如下：

```
//脚本命令代码结构功能流程说明
// (1) ATK 与 C++客户端连接
// (2) 新建想定并设置属性
// (3) 新建卫星并设置轨道预报器为机动规划
// (4) 机动规划添加段，新添加卫星会有默认初始段
// (5) 初始段属性设置
// (6) 第一个预报段属性设置
// (7) 第一个瞄准段中机动段属性设置
// (8) 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
// (9) 第二个预报段属性设置
```

```

// (10) 第二个瞄准段中机动段属性设置
// (11) 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
// (12) 第三个预报段属性设置
// (13) 机动规划运行
// (14) 查看报告数据
// (15) 想定保存
// (16) ATK 与客户端断开连接

//
// ATK 与 C++客户端连接
//
conID = atkOpen();

//
// 新建想定并设置属性
//
atkConnect(conID, 'New', '/ Scenario FastTransfer');
atkConnect(conID, 'SetAnalysisTimePeriod', '* "5 Nov 2022 00:00:00.000" "6 Nov 2022 00:00:00.000"');

//
// 新建卫星并设置轨道预报器为机动规划
//
atkConnect(conID, 'New', '/ Satellite FastTransfer');
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer SetProp');

//
// 机动规划添加段，新添加卫星会有默认初始段
//
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer InsertSegment MainSequence.SegmentList.- Propagate');
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer InsertSegment MainSequence.SegmentList.- Target_Sequence');

```



```

    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList.-
Maneuver');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Propagate');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Target_Sequence');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.Target_Sequence1.SegmentList.-
Maneuver');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Propagate');

//
// 初始段属性设置
//
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Initial_State.InitialState.Epoch 5
Nov 2022 00:00:00.000 UtCG');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Initial_State.CoordinateType
"Modified Keplerian");
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.sma
6700000 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.ecc
0');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.inc
0');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer

```

```

SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.RAAN
0');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.w 0');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.ta 0');

    //
    // 第一个预报段属性设置
    //
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate.SegmentColor -16776961');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate.StoppingConditions
Duration');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate.StoppingConditions.Duration.TripVa
lue 7200 sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate.StoppingConditions.Duration.Tolera
nce 0.0001 sec');

    //
    // 第一个瞄准段中机动段属性设置
    //
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.Impulsi
veMnvr.ThrustAxes "Satellite VNC(Earth)");
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer

```

```

AddMCSegmentControl
MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver
ImpulsiveMnvr.Cartesian.X');

//
// 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
//
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Target_Sequence.Profiles
Differential_Corrector');
//
//设置属性页中控制变量属性
//
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X Active true');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X Correction 2781.50365947627
m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
//

```

```

//设置属性页中约束条件属性
//
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.Results
"Radius Of Apoapsis");
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Active true');
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Desired 84328394 m');
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Scale 1 m');
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis tolerance 0.1 m');
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Weight 1');

//
// 第二个预报段属性设置
//
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagatel.SegmentColor -
16711936');
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagatel.StoppingConditions

```

```

R_Magnitude');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.TripValue 42164197 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.Tolerance 1e-6 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.RepeatCount 1');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.Condition "Cross Either (Inc. or Dec.)");

//
// 第二个瞄准段中机动段属性设置
//
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.SegmentColor -16711681');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.ImpulsiveMnvr.ThrustAxes "Satellite VNC(Earth)");
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
AddMCSegmentControl
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.ImpulsiveMnvr.Cartesian.X');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
AddMCSegmentControl

```

```

MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver
ImpulsiveMnvr.Cartesian.Z');

//
// 第二个瞄准段添加属性页
//
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Target_Sequence1.Profiles
Differential_Corrector');
//
//设置属性页中控制变量属性
//
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X Active true');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 300 m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X Correction -1581.97670664023
m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue

```

```

MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Active true');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z MaxStep 300 m/sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Correction -2771.82057041661
m/sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Perturbation 0.1 m/sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Scale 1 m/sec');
    //
    //设置属性页中约束条件属性
    //
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.Result
s Eccentricity CosineVFPA');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Active true');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Desired 0');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer

```

```

SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Scale 1');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity tolerance 0.1');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Weight 1');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Active true');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Desired 0 rad');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Scale 1 rad');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA tolerance 0.1 rad');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Weight 1');

//
// 第三个预报段属性设置
//

```



```

    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate2.SegmentColor -65536');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate2.StoppingConditions
Duration');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate2.StoppingConditions.Duration.TripV
alue 86400 sec');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate2.StoppingConditions.Duration.Toler
ance 0.0001 sec');

//
// 机动规划运行
//
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
RunMCS');
    atkConnect(conID, 'Animate', '* Reset');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
ApplyAllProfileChanges');

//
// 查看报告数据
//
    atkConnect(conID, 'Report_RM', '*Satellite/FastTransfer Style
"Position" TimePeriod "5 Nov 2022 01:00:00.000" "5 Nov 2022
02:00:00.000")

//
// 想定保存
//
    atkConnect(conID, 'Save', '/ *');

```

```
//
// ATK 与客户端断开连接
//
atkClose(conID);
```

6.2.3. 点击“Save”按钮，保存 FastTransfer.atks 文件。

6.2.4. 点击“Run”按钮，案例构建完成，可在客户端界面查看报告中数据信息，在 ATK 软件显示想定界面，可查看案例轨迹。

6.3 案例展示

输入展示：

“Command Script”对话框中输入多条命令，如下图：

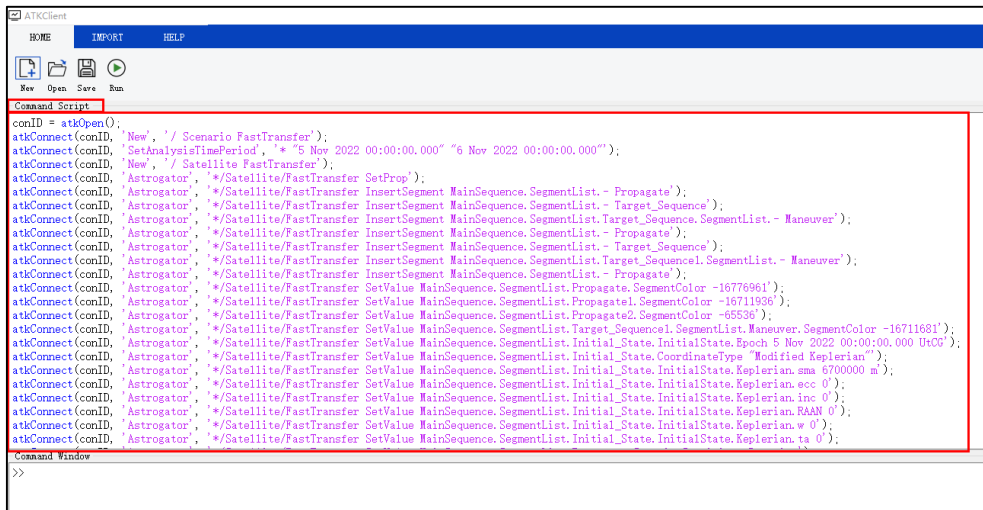


图 6.3-1 “Command Script”对话框中输入多条命令

输出展示：

点击运行后，可在命令行窗口查看报告输出信息，信息如下图：

DateTime	x(m)	y(m)	z(m)
2022-11-05 01:00:03.854310	-3487395.959063	-5702997.687331	37.031653
2022-11-05 01:00:15.535390	-3410097.417406	-5749680.020446	37.029844
2022-11-05 01:00:26.635553	-3336066.688106	-5793069.453548	37.018672
2022-11-05 01:00:38.175524	-3258520.590154	-5837166.749058	36.997375
2022-11-05 01:00:49.883416	-3179253.654062	-5880844.254668	36.965755
2022-11-05 01:01:01.690623	-3098724.688664	-5923801.463142	36.923721
2022-11-05 01:01:13.704185	-3016196.060701	-5966376.964466	36.870564
2022-11-05 01:01:25.721903	-2933057.034431	-6007816.221633	36.806976
2022-11-05 01:01:37.749525	-2849282.934988	-6048129.080897	36.732993
2022-11-05 01:01:50.005486	-2763352.158938	-6088004.974835	36.647069
2022-11-05 01:02:05.166094	-2656290.709703	-6135640.546475	36.526264
2022-11-05 01:02:21.642629	-2539012.860386	-6185274.506404	36.377120
2022-11-05 01:02:37.455551	-2425592.323277	-6230800.747409	36.216960
2022-11-05 01:02:53.454188	-2310015.607737	-6274744.235187	36.038526
2022-11-05 01:03:08.783838	-2198530.559025	-6314838.442791	35.852700
2022-11-05 01:03:25.321285	-2077489.339259	-6355869.485424	35.636703
2022-11-05 01:03:41.504545	-1958305.820615	-6393775.113770	35.410562
2022-11-05 01:03:56.218014	-1849351.412707	-6426298.239638	35.193008

图 6.3-2 报告数据显示

点击运行后，ATK 软件显示想定界面，打开三维视图窗口，拖动时间视图中的时间到结束历元时刻，三维视图展示卫星快速转移轨迹如下图：

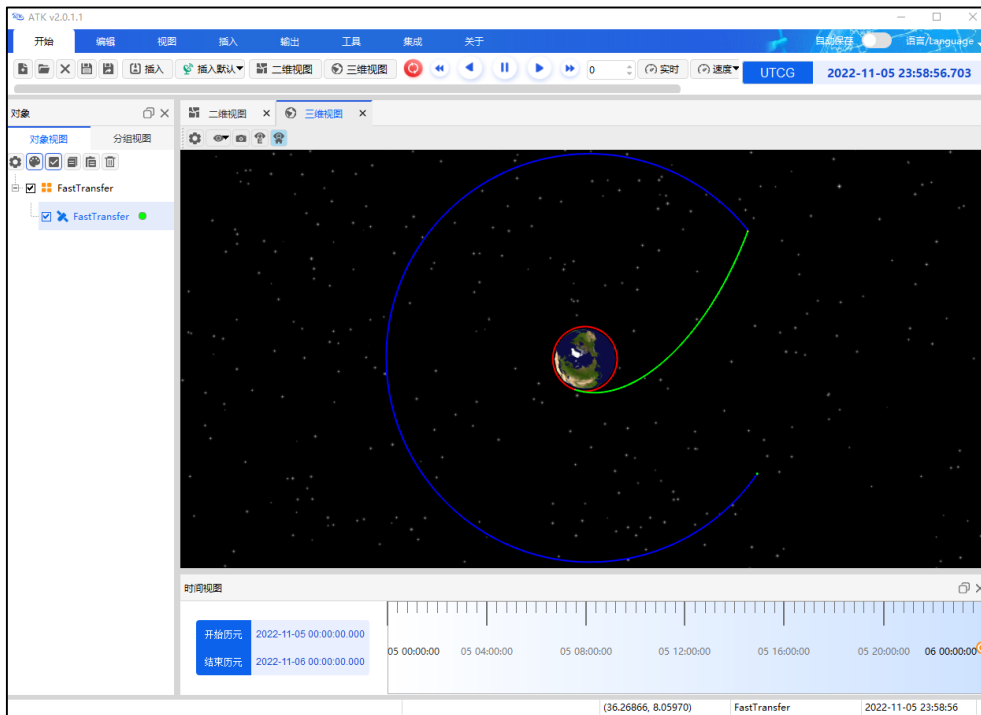


图 6.3-3 卫星快速转移轨迹

7. 基于 ATK.Connect 模式的轨道快速转移的 Matlab 实现

7.1 案例介绍

本案例实现半径为 6700km 的近地停泊轨道（LEO 轨道）快速转移到半径为

42164.197km 的地球同步轨道（GEO 轨道）的轨道机动规划设计。案例使用 Matlab 软件(支持 R2015b 以上版本)基于 Connect 模式的实现。

本案例使用 Matlab R2015b 软件，通过提供的动态库文件和 M 函数文件，使用 M 函数接口完成案例实现。提供的文件和 M 函数接口如下图所示。

Release-ATK-v2.1.0.0-20240329 > IntegratingWithATK > connect > Matlab > Win_2015b

名称	修改日期	类型	大小
atkClose.m	2024/3/29 11:45	M 文件	1 KB
atkConnect.m	2024/3/29 11:45	M 文件	1 KB
ATKConnectorDll.h	2024/3/29 11:45	C/C++ Header	3 KB
ATKConnectorDll64.dll	2024/3/29 11:45	应用程序扩展	46 KB
ATKConnectorDll64.lib	2024/3/29 11:45	Object File Library	3 KB
atkExecuteCommand.m	2024/3/29 11:45	M 文件	1 KB
atkOpen.m	2024/3/29 11:45	M 文件	1 KB
mexATKConnect.cpp	2024/3/29 11:45	C++ Source	4 KB
mexATKConnect.mexw64	2024/3/29 11:45	MEXW64 文件	20 KB

图 7.1-1 提供文件

```

Test.m
1  conID = atkOpen();
2  atkConnect(conID, 'New', '/ Scenario ScenarioTest');
3  atkClose(conID);
    
```

图 7.1-2 M 函数接口

通过编辑 Matlab 脚本文件，使用脚本命令新建想定及属性设置、新建卫星及属性设置和运行机动规划，完成案例构建，如下图所示。

```

FastTransfer.m
1  conID = atkOpen();
2  atkConnect(conID, 'New', '/ Scenario FastTransfer');
3  atkConnect(conID, 'ScenarioAnalysisTimePeriod', '* 5 Nov 2022 00:00:00.000" 6 Nov 2022 00:00:00.000"');
4  atkConnect(conID, 'New', '/ Satellite FastTransfer');
5  atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetProp');
6  atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer InsertSegment MainSequence.SegmentList - Propagate');
7  atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer InsertSegment MainSequence.SegmentList - Target_Sequence');
8  atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList - Maneuver');
9  atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer InsertSegment MainSequence.SegmentList - Propagate');
10 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer InsertSegment MainSequence.SegmentList - Target_Sequence');
11 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer InsertSegment MainSequence.SegmentList - Maneuver');
12 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer InsertSegment MainSequence.SegmentList - Propagate');
13 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Propagate.SegmentColor -16776961');
14 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Propagate1.SegmentColor -16711936');
15 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Propagate2.SegmentColor -65536');
16 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.SegmentColor -16711681');
17 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Epoch 5 Nov 2022 00:00:00.000 UTC');
18 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.CoordinateType 'Modified Keplerian');
19 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Keplerian.sma 6700000 m');
20 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Keplerian.ecc 0');
21 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Keplerian.inc 0');
22 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Keplerian.RAAN 0');
23 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Keplerian.w 0');
24 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Propagate.StoppingConditions.Duration');
25 atkConnect(conID, 'Astrogator', '/Satellite/FastTransfer SetValue MainSequence.SegmentList.Propagate.StoppingConditions.TripValue 7200 sec');
    
```

图 7.1-3 脚本文件

使用获取报告命令，报告数据显示在命令行窗口，如下图所示。

```

142 %
143 % 报告数据查看
144 %
145 % atkConnect(conID, 'Report_RM', '*/Satellite/FastTransfer Style "Position" TimePeriod "5 Nov 2022 01:00:00.000" "5 Nov 2022 02:00:00.000"')
146 %
147 %

```

命令行窗口

```

>> FastTransfer
ans =

```

DateTime	x (m)	y (m)	z (m)
2022-11-05 01:00:03.854310	-3487395.959063	-5702997.687331	37.031653
2022-11-05 01:00:15.535390	-3410097.417406	-5749680.020446	37.029844
2022-11-05 01:00:26.635553	-3336066.688106	-5793069.453548	37.018672
2022-11-05 01:00:38.175524	-3258520.590154	-5837166.749058	36.997375
2022-11-05 01:00:49.883416	-3179253.654062	-5880844.254668	36.965755
2022-11-05 01:01:01.690623	-3098724.688664	-5923801.463142	36.923721
2022-11-05 01:01:13.704185	-3016196.060701	-5966376.964466	36.870564
2022-11-05 01:01:25.721903	-2933057.034431	-6007816.221633	36.806976
2022-11-05 01:01:37.749525	-2849282.934988	-6048129.080897	36.732993
2022-11-05 01:01:50.005486	-2763352.158938	-6088004.974835	36.647069

图 7.1-4 报告数据显示

运行 Matlab 脚本文件，ATK 响应脚本命令，新建想定和构建卫星轨道如下图所示。

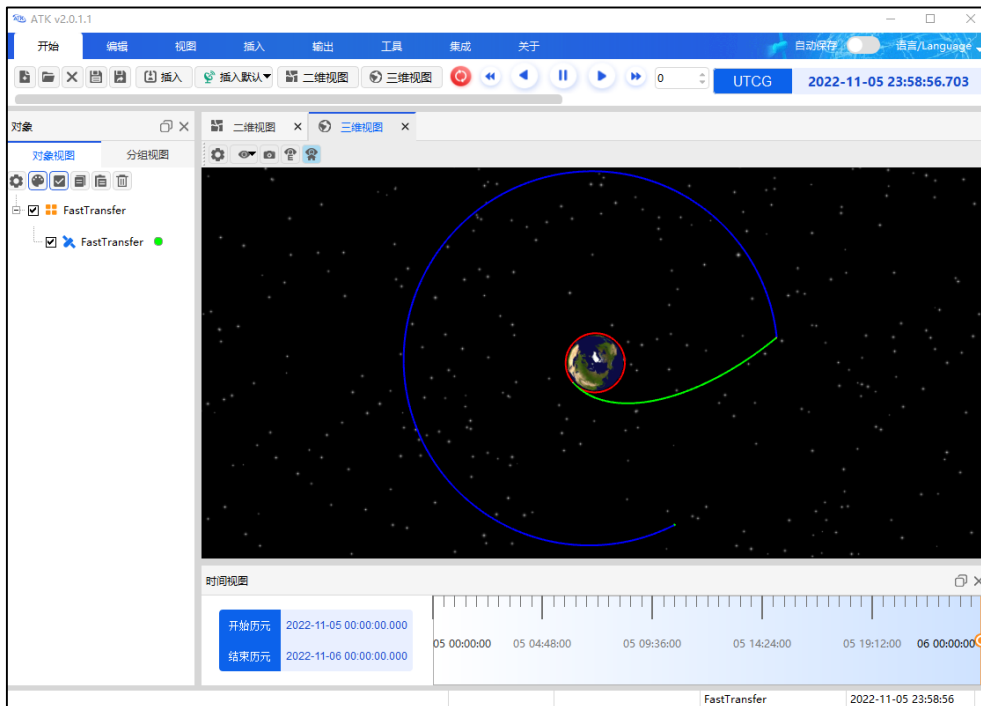


图 7.1-5 快速转移轨道

7.2 案例实现

本案例使用基于 ATK.Connect 模式提供的 M 函数接口来完成 Matlab 软件和 ATK 软件之间的数据交互，案例实现流程如下所示。

7.2.1. 启动 ATK 软件，关闭“欢迎使用 ATK”对话框。

7.2.2. 启动 Matlab R2015b 软件，浏览文件夹并选择提供的动态库文件和 M 函数文件所在的文件夹，然后在菜单栏中，单击“主页”按钮，单击“新建脚本”按钮，新建脚本文件。

7.2.3. 在脚本文件编辑器中输入脚本命令（脚本命令代码可以直接拷贝到脚本文件），脚本命令代码说明如下：

```
//脚本命令代码结构功能流程说明
// (1) Matlab 与 ATK 建立连接
// (2) 新建想定并设置属性
// (3) 新建卫星并设置轨道预报器为机动规划
// (4) 机动规划添加段，新添加卫星会有默认初始段
// (5) 初始段属性设置
// (6) 第一个预报段属性设置
// (7) 第一个瞄准段中机动段属性设置
// (8) 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
// (9) 第二个预报段属性设置
// (10) 第二个瞄准段中机动段属性设置
// (11) 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
// (12) 第三个预报段属性设置
// (13) 机动规划运行
// (14) 查看报告数据
// (15) 想定保存
// (16) Matlab 与 ATK 断开连接

%
% Matlab 与 ATK 建立连接
%
conID = atkOpen();

%
% 新建想定并设置属性
%
atkConnect(conID, 'New', '/ Scenario FastTransfer');
atkConnect(conID, 'SetAnalysisTimePeriod', '* "5 Nov 2022
00:00:00.000" "6 Nov 2022 00:00:00.000"');
```

```

%
% 新建卫星并设置轨道预报器为机动规划
%
atkConnect(conID, 'New', '/ Satellite FastTransfer');
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
SetProp');

%
% 机动规划添加段，新添加卫星会有默认初始段
%
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Propagate');
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Target_Sequence');
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList.-
Maneuver');
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Propagate');
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Target_Sequence');
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.Target_Sequence1.SegmentList.-
Maneuver');
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Propagate');

%
% 初始段属性设置
%
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
SetValue MainSequence.SegmentList.Initial_State.InitialState.Epoch 5
Nov 2022 00:00:00.000 UtCG');
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer

```

```

SetValue MainSequence.SegmentList.Initial_State.CoordinateType
"Modified Keplerian");
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.sma
6700000 m');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.ecc
0');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.inc
0');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.RAAN
0');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.w 0');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.ta 0');

%
% 第一个预报段属性设置
%
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate.SegmentColor -16776961');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate.StoppingConditions
Duration');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue

```



```

MainSequence.SegmentList.Propagate.StoppingConditions.Duration.TripValue 7200 sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate.StoppingConditions.Duration.Tolerance 0.0001 sec');

%
% 第一个瞄准段中机动段属性设置
%
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.ImpulsiveMnvr.ThrustAxes "Satellite VNC(Earth)");
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
AddMCSegmentControl
MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver
ImpulsiveMnvr.Cartesian.X');

%
% 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
%
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue MainSequence.SegmentList.Target_Sequence.Profiles
Differential_Corrector');
%
%设置属性页中控制变量属性
%
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector
Maneuver ImpulsiveMnvr.Cartesian.X Active true');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Corrector

```

```

tor Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X Correction 2781.50365947627
m/sec');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
    %
    % 设置属性页中约束条件属性
    %
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.Results
"Radius Of Apoapsis");
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Active true');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Desired 84328394 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Scale 1 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue

```

```

MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis tolerance 0.1 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Weight 1');

%
% 第二个预报段属性设置
%
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagatel.SegmentColor -
16711936');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagatel.StoppingConditions
R_Magnitude');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.Tr
ipValue 42164197 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.To
lerance 1e-6 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.Re
peatCount 1');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.Co
ndition "Cross Either (Inc. or Dec.)");

%
% 第二个瞄准段中机动段属性设置

```

```

%
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.Segmen
tColor -16711681');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.Impuls
iveMnvr.ThrustAxes "Satellite VNC(Earth)");
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
AddMCSSegmentControl
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver
ImpulsiveMnvr.Cartesian.X');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
AddMCSSegmentControl
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver
ImpulsiveMnvr.Cartesian.Z');

%
% 第二个瞄准段添加属性页
%
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Target_Sequence1.Profiles
Differential_Corrector');
%
%设置属性页中控制变量属性
%
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X Active true');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 300 m/sec');

```

```

    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X Correction -1581.97670664023
m/sec');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Active true');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z MaxStep 300 m/sec');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Correction -2771.82057041661
m/sec');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Perturbation 0.1 m/sec');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Scale 1 m/sec');
%
```

```

% 设置属性页中约束条件属性
%
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.Result
s Eccentricity CosineVFPA');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Active true');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Desired 0');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Scale 1');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity tolerance 0.1');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Weight 1');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Active true');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Desired 0 rad');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer

```

```

SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Scale 1 rad');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA tolerance 0.1 rad');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Weight 1');

%
% 第三个预报段属性设置
%
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate2.SegmentColor -65536');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate2.StoppingConditions
Duration');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate2.StoppingConditions.Duration.TripV
alue 86400 sec');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate2.StoppingConditions.Duration.Toler
ance 0.0001 sec');

%
% 机动规划运行
%
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
RunMCS');
    atkConnect(conID, 'Animate', '* Reset');

```

```

    atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
ApplyAllProfileChanges');

%
% 查看报告数据
%
    atkConnect(conID, 'Report_RM', '* /Satellite/FastTransfer Style
"Position" TimePeriod "5 Nov 2022 01:00:00.000" "5 Nov 2022
02:00:00.000"')

%
% 想定保存
%
    atkConnect(conID, 'Save', '* / *');

%
% ATK 与客户端断开连接
%
    atkClose(conID);

```

7.2.4. 点击“运行”按钮，保存 FastTransfer.m 文件。

7.2.5. 运行结束后，案例构建完成，在命令行窗口可查看报告输出信息，在 ATK 软件显示想定界面，可查看案例轨迹。

7.3 案例结果

输入展示：

脚本文件编辑器中输入多条命令，如下图：


```

4 conID = atkOpen();
5
6 %
7 % 想定新建并设置属性
8 %
9 atkConnect (conID, 'New', '/ Scenario FastTransfer');
10 atkConnect (conID, 'SetAnalysisTimePeriod', '* * 5 Nov 2022 00:00:00.000 * 6 Nov 2022 00:00:00.000');
11
12 %
13 % 卫星新建与轨道预报设置为机动规划
14 %
15 atkConnect (conID, 'New', '/ Satellite FastTransfer');
16 atkConnect (conID, 'Astrogator', '* / Satellite / FastTransfer SetProp');
17
18 %
19 % 机动规划添加段，新添加卫星会有默认初始段
20 %
21 atkConnect (conID, 'Astrogator', '* / Satellite / FastTransfer InsertSegaent MainSequence.SegmentList.- Propagate');
22 atkConnect (conID, 'Astrogator', '* / Satellite / FastTransfer InsertSegaent MainSequence.SegmentList.- Target_Sequence');
23 atkConnect (conID, 'Astrogator', '* / Satellite / FastTransfer InsertSegaent MainSequence.SegmentList.Target_Sequence.SegmentList.- Maneuver');
24 atkConnect (conID, 'Astrogator', '* / Satellite / FastTransfer InsertSegaent MainSequence.SegmentList.- Propagate');
25 atkConnect (conID, 'Astrogator', '* / Satellite / FastTransfer InsertSegaent MainSequence.SegmentList.- Target_Sequence');
26 atkConnect (conID, 'Astrogator', '* / Satellite / FastTransfer InsertSegaent MainSequence.SegmentList.Target_Sequence1.SegmentList.- Maneuver');
27 atkConnect (conID, 'Astrogator', '* / Satellite / FastTransfer InsertSegaent MainSequence.SegmentList.- Propagate');

```

图 7.3-1 脚本文件中输入多条命令

输出展示:

Matlab 软件点击运行脚本文件后，命令行窗口会显示报告数据信息，如下图:

```

命令行窗口
2022-11-05 01:55:27.845048 -949338.371307 6620911.778130 -121.342553
2022-11-05 01:55:29.845048 -964628.404693 6618679.016788 -121.099729
2022-11-05 01:55:33.845048 -995192.947020 6614107.551307 -120.611313
2022-11-05 01:55:42.584496 -1061897.446866 6603628.472886 -119.531377
2022-11-05 01:55:59.079507 -1187496.542782 6582017.396893 -117.445356
2022-11-05 01:56:19.751594 -1344287.349625 6551561.110319 -114.743540
2022-11-05 01:56:40.847362 -1503500.750055 6516629.433310 -111.886841
2022-11-05 01:57:02.788866 -1668148.408649 6476191.775260 -108.810098
2022-11-05 01:57:24.994823 -1833688.784031 6431030.307218 -105.588011
2022-11-05 01:57:46.974401 -1996353.397514 6382160.988683 -102.292920
2022-11-05 01:58:08.638720 -2155425.458196 6329964.871852 -98.943554
2022-11-05 01:58:29.498607 -2307315.756953 6275958.453341 -95.624926
2022-11-05 01:58:49.789471 -2453777.313875 6219926.669323 -92.310485
2022-11-05 01:58:55.759161 -2496613.905446 6202789.385147 -91.319445
2022-11-05 01:59:01.476851 -2537530.858987 6186098.755081 -90.363554
2022-11-05 01:59:07.105734 -2577704.087385 6169403.552517 -89.416174
2022-11-05 01:59:17.211975 -2649557.266537 6138773.958451 -87.699634
2022-11-05 01:59:32.560034 -2757984.179927 6090657.662730 -85.055180
2022-11-05 01:59:52.848427 -2899975.028059 6024115.336792 -81.492094
2022-11-05 01:59:59.999961 -2949649.862775 5999868.184577 -80.218441
2022-11-05 01:59:59.999961 -2949649.862775 5999868.184577 -80.218441
fx >>

```

图 7.3-2 报告数据显示

Matlab 软件点击运行脚本文件后，ATK 软件显示想定界面，打开三维视图窗口，拖动时间视图中的时间到结束历元时刻，三维视图展示卫星快速转移轨迹如下图:

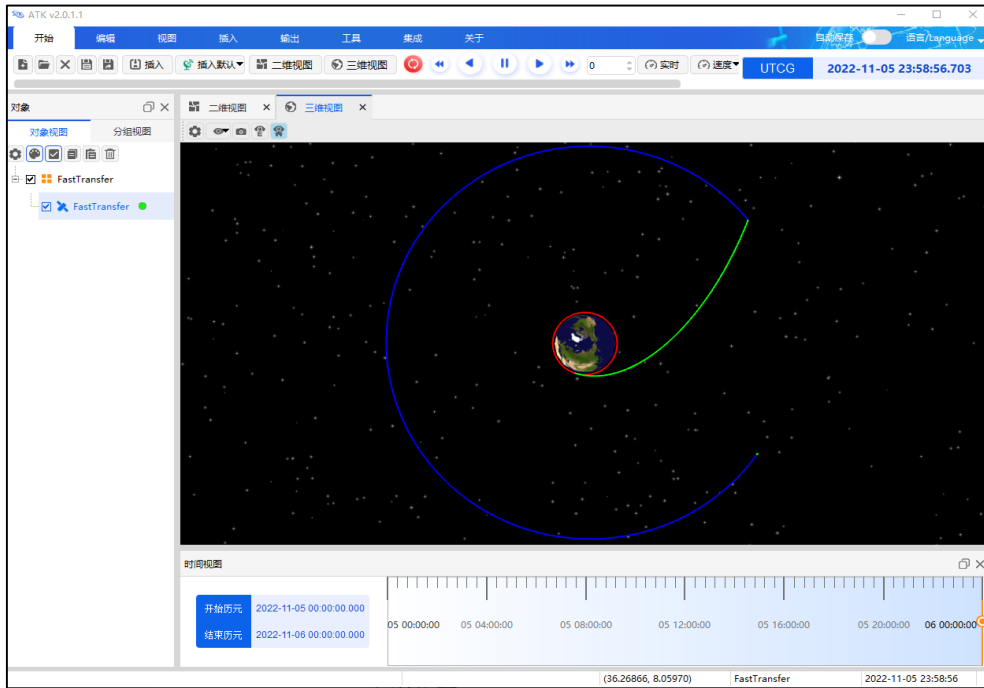


图 7.3-3 卫星快速转移轨迹

8. 基于 ATK.Connect 模式的轨道快速转移的 Java 实现

8.1 案例介绍

本案例实现半径为 6700km 的近地停泊轨道（LEO 轨道）快速转移到半径为 42164.197km 的地球同步轨道（GEO 轨道）的轨道机动规划设计。案例使用基于 Connect 模式的 Java 客户端实现。

本案例通过 ATK.Connect 模式提供的 Java 客户端程序，使用接口函数进行 ATK 软件与 Java 客户端之间的数据解析与传递，完成案例实现。ATK 提供 Java 客户端与接口函数如下图所示。

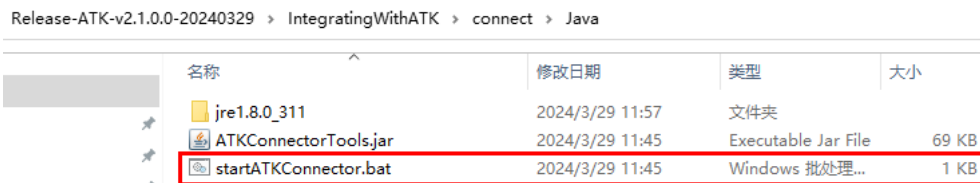


图 8.1-1 客户端界面

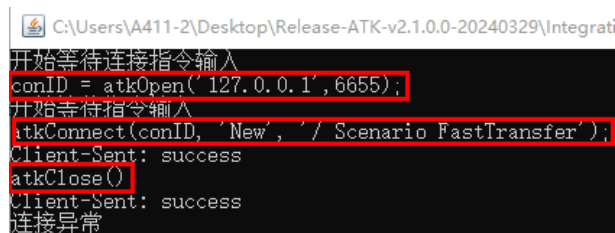


图 8.1-2 接口函数

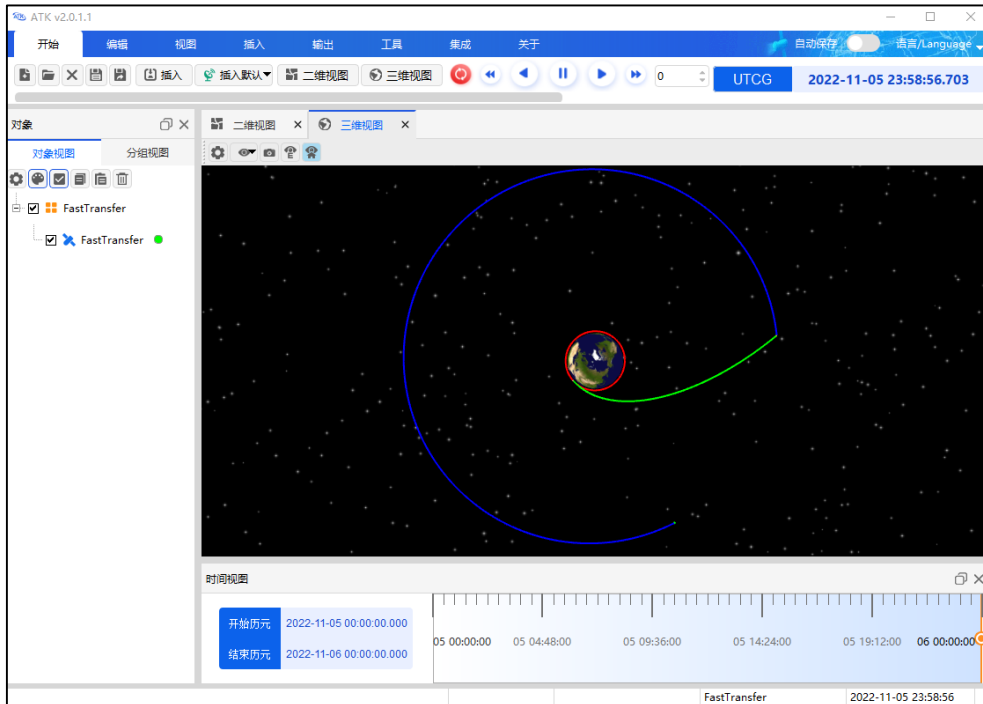


图 8.1-5 快速转移轨道

8.2 案例实现

本案例使用 ATK.Connect 模式提供的 Java 客户端程序，实现 ATK 与 Java 客户端之间的数据交互，案例实现流程如下所示。

8.2.1. 启动 ATK 软件，关闭“欢迎使用 ATK”对话框。

8.2.2. 在 ATK 安装包内打开“Connect”文件夹，打开“Java”文件夹，打开“startATKConnector.bat”对话框。

8.2.3. 在对话框中输入脚本命令（代码可以直接拷贝到对话框）。脚本命令代码说明如下：

```
//脚本命令代码结构功能流程说明
// (1) ATK 与 Java 客户端连接
// (2) 新建想定并设置属性
// (3) 新建卫星并设置轨道预报器为机动规划
// (4) 机动规划添加段，新添加卫星会有默认初始段
// (5) 初始段属性设置
// (6) 第一个预报段属性设置
// (7) 第一个瞄准段中机动段属性设置
// (8) 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
```

```

// (9) 第二个预报段属性设置
// (10) 第二个瞄准段中机动段属性设置
// (11) 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
// (12) 第三个预报段属性设置
// (13) 机动规划运行
// (14) 想定保存
// (15) ATK 与客户端断开连接

//
// ATK 与 Java 客户端连接
//
conID = atkOpen('127.0.0.1',6655);

//
// 新建想定并设置属性
//
atkConnect(conID, 'New', '/ Scenario FastTransfer');
atkConnect(conID, 'SetAnalysisTimePeriod', '* "5 Nov 2022 00:00:00.000" "6 Nov 2022 00:00:00.000"');

//
// 新建卫星并设置轨道预报器为机动规划
//
atkConnect(conID, 'New', '/ Satellite FastTransfer');
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer SetProp');

//
// 机动规划添加段，新添加卫星会有默认初始段
//
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer InsertSegment MainSequence.SegmentList.- Propagate');
atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer InsertSegment MainSequence.SegmentList.- Target_Sequence');

```

```

    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList.-
Maneuver');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Propagate');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Target_Sequence');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.Target_Sequence1.SegmentList.-
Maneuver');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Propagate');

//
// 初始段属性设置
//
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Initial_State.InitialState.Epoch 5
Nov 2022 00:00:00.000 UtCG');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Initial_State.CoordinateType
"Modified Keplerian");
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.sma
6700000 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.ecc
0');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.inc
0');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer

```

```

SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.RAAN
0');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.w 0');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.ta 0');

    //
    // 第一个预报段属性设置
    //
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate.SegmentColor -16776961');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate.StoppingConditions
Duration');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate.StoppingConditions.Duration.TripVa
lue 7200 sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate.StoppingConditions.Duration.Tolera
nce 0.0001 sec');

    //
    // 第一个瞄准段中机动段属性设置
    //
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.Impulsi
veMnvr.ThrustAxes "Satellite VNC(Earth)");
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer

```

```

AddMCSegmentControl
MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver
ImpulsiveMnvr.Cartesian.X');

//
// 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
//
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Target_Sequence.Profiles
Differential_Corrector');
//
// 设置属性页中控制变量属性
//
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X Active true');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X Correction 2781.50365947627
m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
//

```



```

// 设置属性页中约束条件属性
//
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.Results
"Radius Of Apoapsis");
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Active true');
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Desired 84328394 m');
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Scale 1 m');
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis tolerance 0.1 m');
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Weight 1');

//
// 第二个预报段属性设置
//
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagatel.SegmentColor -
16711936');
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagatel.StoppingConditions

```

```

R_Magnitude');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.TripValue 42164197 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.Tolerance 1e-6 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.RepeatCount 1');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.Condition "Cross Either (Inc. or Dec.)");

//
// 第二个瞄准段中机动段属性设置
//
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.SegmentColor -16711681');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.ImpulsiveMnvr.ThrustAxes "Satellite VNC(Earth)");
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
AddMCSegmentControl
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.ImpulsiveMnvr.Cartesian.X');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
AddMCSegmentControl

```

```

MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver
ImpulsiveMnvr.Cartesian.Z');

//
// 第二个瞄准段添加属性页
//
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Target_Sequence1.Profiles
Differential_Corrector');
//
// 设置属性页中控制变量属性
//
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X Active true');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 300 m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X Correction -1581.97670664023
m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue

```

```

MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Active true');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z MaxStep 300 m/sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Correction -2771.82057041661
m/sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Perturbation 0.1 m/sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Scale 1 m/sec');
    //
    // 设置属性页中约束条件属性
    //
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.Result
s Eccentricity CosineVFPA');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Active true');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Desired 0');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer

```

```

SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Scale 1');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity tolerance 0.1');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Weight 1');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Active true');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Desired 0 rad');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Scale 1 rad');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA tolerance 0.1 rad');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Weight 1');

//
// 第三个预报段属性设置
//

```

```

    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate2.SegmentColor -65536');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate2.StoppingConditions
Duration');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate2.StoppingConditions.Duration.TripV
alue 86400 sec');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate2.StoppingConditions.Duration.Toler
ance 0.0001 sec');

//
// 机动规划运行
//
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
RunMCS');
    atkConnect(conID, 'Animate', '* Reset');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
ApplyAllProfileChanges');

//
// 查看报告数据
//
    atkConnect(conID, 'Report_RM', '*Satellite/FastTransfer Style
"Position" TimePeriod "5 Nov 2022 01:00:00.000" "5 Nov 2022
02:00:00.000")

//
// 想定保存
//
    atkConnect(conID, 'Save', '/ *');

```

```
//
// ATK 与客户端断开连接
//
atkClose(conID);
```

8.2.4. 脚本命令输入完成后，案例构建完成，在 ATK 软件显示想定界面，可查看案例轨迹。

8.3 案例展示

输入展示：

Java 客户端对话框中输入命令，如下图：

```
C:\Users\A411-2\Desktop\Release-ATK-v2.1.0.0-20240329\IntegratingWithATK\connect\Java\jre1.8.0_311\bin\java.exe
开始等待接收指令输入
conID = atkOpen( '127.0.0.1', 6655);
开始等待指令输入
atkConnect(conID, 'New', '/ Scenario FastTransfer');
Client-Sent: success
atkConnect(conID, 'SetAnalysisTimePeriod', '* 5 Nov 2022 00:00:00.000' "6 Nov 2022 00:00:00.000");
Client-Sent: success
atkConnect(conID, 'New', '/ Satellite FastTransfer');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer SetProp');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer InsertSegment MainSequence.SegmentList. - Propagate');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer InsertSegment MainSequence.SegmentList. - Target_Sequence');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList. - Maneuver');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer InsertSegment MainSequence.SegmentList. - Propagate');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer InsertSegment MainSequence.SegmentList. - Target_Sequence');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList. - Maneuver');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Epoch 5 Nov 2022 00:00:00.000 UtCG');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.CoordinateType "Modified Keplerian");
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Keplerian.sma 6700000 m');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Keplerian.ecc 0');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Keplerian.inc 0');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Keplerian.RAAN 0');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Keplerian.w 0');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer SetValue MainSequence.SegmentList.Initial_State.InitialState.Keplerian.ta 0');
Client-Sent: success
atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer SetValue MainSequence.SegmentList.Propagate.SegmentColor -16776961');
Client-Sent: success
```

图 8.3-1 Java 客户端对话框中输入多条命令

输出展示：

点击运行后，可在命令行窗口查看报告输出信息，信息如下图：

```
E:\DongMin\dm\二次开发相关\客户端\ATKConnectorTools\Release\jre1.8.0_311\bin\java.exe
atkConnect(conID, 'Report_RU', '*/Satellite/Satellite Style "Position" TimePeriod "5 Nov 2022 01:00:00.000" "5 Nov 2022 02:00:00.000"');
Client-Sent: success
onReceived:gStrCommand:
onReceived:result:
-----
DateTime          x (m)          y (m)          z (m)
-----
2022-11-05 01:00:00.000000 -3398718.583005 -5035706.045801 -2739873.454560
2022-11-05 01:01:00.000000 -2991584.506249 -5232279.960354 -2845380.914184
2022-11-05 01:02:00.000000 -2569955.615060 -5403501.443558 -2937059.953459
2022-11-05 01:03:00.000000 -2135878.806280 -5548548.934132 -3014469.501130
2022-11-05 01:04:00.000000 -1691459.856142 -5666728.112339 -3077238.056879
2022-11-05 01:05:00.000000 -1238853.181560 -5757474.989515 -3125065.336362
2022-11-05 01:06:00.000000 -780251.456951 -5820358.414780 -3157723.597073
2022-11-05 01:07:00.000000 -317875.073057 -5855081.996626 -3175058.655718
2022-11-05 01:08:00.000000 146038.552747 -5861485.397695 -3176990.575269
2022-11-05 01:09:00.000000 609245.755015 -5839544.968059 -3163513.991348
2022-11-05 01:10:00.000000 1069507.406132 -5789373.715571 -3134698.064836
2022-11-05 01:11:00.000000 1524599.644650 -5711220.632946 -3090686.069184
2022-11-05 01:12:00.000000 1972324.507060 -5605469.394258 -3031694.627621
2022-11-05 01:13:00.000000 2410520.426842 -5472636.417010 -2958012.608481
2022-11-05 01:14:00.000000 2837072.545762 -5313368.289569 -2869999.683033
2022-11-05 01:15:00.000000 3249922.775992 -5128438.591505 -2768084.558502
2022-11-05 01:16:00.000000 3647079.579704 -4918744.151726 -2652762.906602
2022-11-05 01:17:00.000000 4026627.464655 -4685300.763098 -2524594.996627
2022-11-05 01:18:00.000000 4386736.175855 -4429238.326936 -2384203.019913
2022-11-05 01:19:00.000000 4725669.500857 -4151795.402716 -2232268.090680
2022-11-05 01:20:00.000000 5041793.580806 -3854313.205573 -2066526.938646
2022-11-05 01:21:00.000000 5333584.682877 -3538229.147945 -1896768.337405
2022-11-05 01:22:00.000000 5599636.469252 -3205069.980628 -1714829.301026
2022-11-05 01:23:00.000000 5838666.774496 -2856444.497390 -1524591.046997
```

图 8.3-2 报告数据显示

点击运行后，ATK 软件显示想定界面，打开三维视图窗口，拖动时间视图中的时间到结束历元时刻，三维视图展示卫星快速转移轨迹如下图：

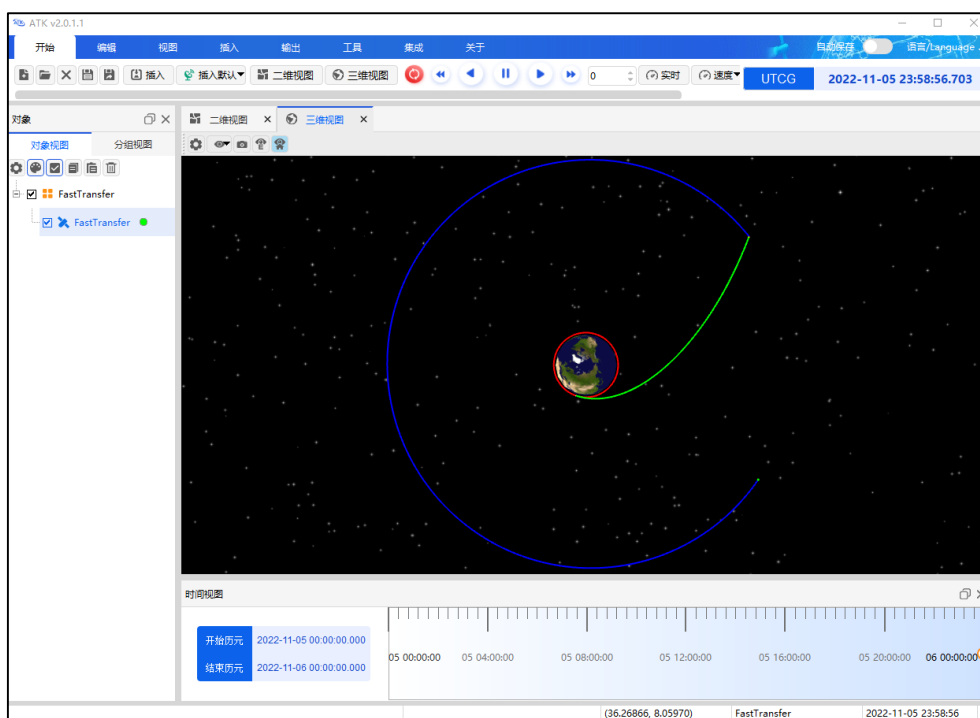


图 8.3-3 卫星快速转移轨迹

9. 基于 ATK.Connent 模式的轨道快速转移的 Python 实现

9.1 案例介绍

本案例实现半径为 6700km 的近地停泊轨道（LEO 轨道）快速转移到半径为 42164.197km 的地球同步轨道（GEO 轨道）的轨道机动规划设计。案例基于 Connect 模式使用 Python 脚本实现。

本案例依赖以下文件（均包含在 ATK 安装包中）：

9.1.1. python-3.11.0-embed-amd64，嵌入式 Python 解释器，实现不安装 Python，也能解释运行 Python 脚本。

9.1.2. ATKConnectModule.py，ATK 数据通信接口文件，提供与 ATK 软件的数据通信模块，在 Python 案例脚本中导入该模块后，就能直接调用里面的数据通信接口。主要包含以下接口：

接口名称	接口功能	返回值	参数
atkOpen	建立 ATK 连接	连接 ID(int)	ATK 设备 IP(str), ATK 端口(int)
atkConnect	发送命令请求	请求响应 (str)	连接 ID(int), 命令类型(str), 命令参数(str)

atkClose	关闭 ATK 连接	无	连接 ID(int)
----------	-----------	---	------------

9.1.3. `_ATKConnectModule.cp311-win_amd64.pyd`, ATK 数据通信动态库, 供 ATK 数据通信接口调用, 包含数据通信接口的具体实现。

9.1.4. `ATKConnectTest.py`, Python 案例脚本, 包含轨道快速转移案例的具体实现过程。该 Python 案例脚本通过嵌入式 Python 解释器解释运行。

名称	修改日期	类型	大小
python-3.11.0-embed-amd64	2024/3/20 星期...	文件夹	
_ATKConnectModule.cp311-win_amd...	2024/3/18 星期...	PYD 文件	98 KB
ATKConnectModule.py	2024/3/18 星期...	PY 文件	5 KB
ATKConnectTest.py	2024/3/22 星期...	PY 文件	24 KB

图 9.1-1 ATK.Connect.Python 提供的文件

名称	修改日期	类型	大小
python.exe	2024/3/18 星期...	应用程序	100 KB
_python311.pth	2024/3/18 星期...	_PTH 文件	1 KB
python311.dll	2024/3/18 星期...	应用程序扩展	5,624 KB
python311.zip	2024/3/18 星期...	WinRAR ZIP 压缩	4,168 KB

图 9.1-2 嵌入式 Python 解释器文件

9.2 案例实现

本案例使用 Python 脚本实现, 通过在案例脚本中导入 ATK 数据通信模块, 调用其中的 `atkOpen` 和 `atkClose` 接口, 完成与 ATK 软件的连接和关闭。调用 `atkConnect` 接口, 完成与 ATK 软件的数据通信, 实现场景卫星的新建与设置, 动画的仿真运行, 数据报告的返回打印, 以及想定文件的保存。最后通过嵌入式 Python 解释器解释运行该 Python 案例脚本, 完成案例实现。具体流程如下所示。

9.2.1. 编辑 Python 案例脚本, 流程图与源码如下:

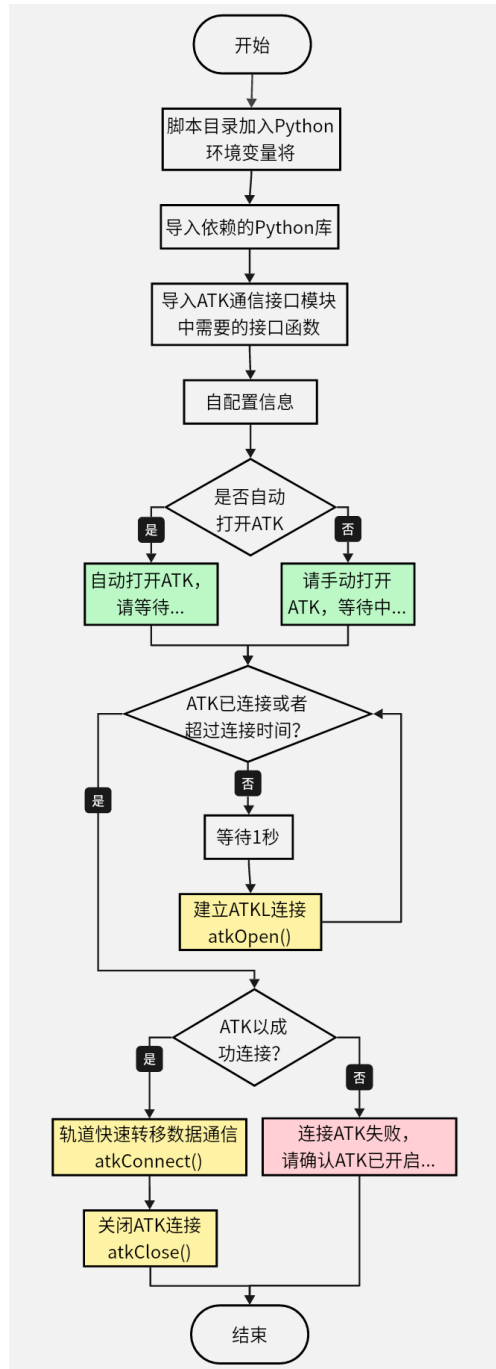


图 9.2-1 Python 案例流程图

#1, 使用内嵌式 python 解释器时, 需要将当前路径加入 Python 系统环境路径

```

import sys
import os
script_dir = os.path.dirname(os.path.abspath(__file__))
sys.path.insert(0, script_dir)

```

#2, 导入依赖的 Python 库

```

import subprocess

```

```

import time

#3, 导入 ATKConnect 接口模块中需要的接口函数
from ATKConnectModule import atkOpen
from ATKConnectModule import atkConnect
from ATKConnectModule import atkClose

#4, 自配置信息
ATKIP          = '127.0.0.1'
ATKPort        = 6655
ATKAutoOpen    = True
ATKPath        = r'E:\ATK 软件\Release-ATK-v2.0.0.3-20240203-
2\ATK.exe'
WaitTime       = 10

#5, 轨道快速转移实现函数
def TestFastTransfer():
    #5.1, 想定新建并设置属性
    atkConnect(conID, 'New', '/ Scenario FastTransfer');
    atkConnect(conID, 'SetAnalysisTimePeriod', '* "5 Nov 2022
00:00:00.000" "6 Nov 2022 00:00:00.000"');
    #5.2, 卫星新建与轨道预报设置为机动规划
    atkConnect(conID, 'New', '/ Satellite FastTransfer');
    atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
SetProp');
    #5.3, 机动规划添加段, 新添加卫星会有默认初始段
    atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
InsertSegment MainSequence.SegmentList. - Propagate');
    atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
InsertSegment MainSequence.SegmentList. - Target_Sequence');
    atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.Target_Sequence.SegmentList. -
Maneuver');
    atkConnect(conID, 'Astrogator', '* /Satellite/FastTransfer
InsertSegment MainSequence.SegmentList. - Propagate');

```

```

    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Target_Sequence');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.Target_Sequence1.SegmentList.-
Maneuver');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
InsertSegment MainSequence.SegmentList.- Propagate');
#5.4, 初始段属性设置
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Initial_State.InitialState.Epoch 5
Nov 2022 00:00:00.000 UtCG');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Initial_State.CoordinateType
"Modified Keplerian");
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.sma
6700000 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.ecc
0');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.inc
0');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.RAAN
0');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Initial_State.InitialState.Keplerian.w 0');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue

```

```

MainSequence.SegmentList.Initial_State.InitialState.Keplerian.ta 0');
#5.5, 第一个预报段属性设置
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate.SegmentColor -16776961');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate.StoppingConditions
Duration');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate.StoppingConditions.Duration.TripVa
lue 7200 sec');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagate.StoppingConditions.Duration.Tolera
nce 0.0001 sec');
#5.6, 第一个瞄准段中机动段属性设置
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.Impulsi
veMnvr.ThrustAxes "Satellite VNC(Earth)");
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
AddMCSegmentControl
MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver
ImpulsiveMnvr.Cartesian.X');
#5.7, 第一个瞄准段添加属性页
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Target_Sequence.Profiles
Differential_Corrector');
#5.8, 设置属性页中控制变量属性
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver ImpulsiveMnvr.Cartesian.X Active true');
atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSControlValue

```

```
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc  
tor Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 100 m/sec');
```

```
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer  
SetMCSControlValue
```

```
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc  
tor Maneuver ImpulsiveMnvr.Cartesian.X Correction 2781.50365947627  
m/sec');
```

```
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer  
SetMCSControlValue
```

```
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc  
tor Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
```

```
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer  
SetMCSControlValue
```

```
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc  
tor Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
```

#5.9, 设置属性页中约束条件属性

```
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer  
SetValue
```

```
MainSequence.SegmentList.Target_Sequence.SegmentList.Maneuver.Results  
"Radius Of Apoapsis");
```

```
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer  
SetMCSConstraintValue
```

```
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc  
tor Maneuver Radius_Of_Apoapsis Active true');
```

```
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer  
SetMCSConstraintValue
```

```
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc  
tor Maneuver Radius_Of_Apoapsis Desired 84328394 m');
```

```
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer  
SetMCSConstraintValue
```

```
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc  
tor Maneuver Radius_Of_Apoapsis Scale 1 m');
```

```
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer  
SetMCSConstraintValue
```

```
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
```

```

tor Maneuver Radius_Of_Apoapsis tolerance 0.1 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence.Profiles.Differential_Correc
tor Maneuver Radius_Of_Apoapsis Weight 1');
    #5.10, 第二个预报段属性设置
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagatel.SegmentColor -
16711936');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagatel.StoppingConditions
R_Magnitude');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.Tr
ipValue 42164197 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.To
lerance 1e-6 m');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.Re
peatCount 1');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Propagatel.StoppingConditions.R_Magnitude.Co
ndition "Cross Either (Inc.or Dec.)");
    #5.11, 第二个瞄准段中机动段属性设置
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.Segmen
tColor -16711681');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue

```

```
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.ImpulsiveMnvr.ThrustAxes "Satellite VNC(Earth)");
```

```
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer  
AddMCSSegmentControl
```

```
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver  
ImpulsiveMnvr.Cartesian.X');
```

```
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer  
AddMCSSegmentControl
```

```
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver  
ImpulsiveMnvr.Cartesian.Z');
```

#5.12, 第二个瞄准段添加属性页

```
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer  
SetValue MainSequence.SegmentList.Target_Sequence1.Profiles  
Differential_Corrector');
```

#5.13, 设置属性页中控制变量属性

```
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer  
SetMCSControlValue
```

```
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre  
ctor Maneuver ImpulsiveMnvr.Cartesian.X Active true');
```

```
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer  
SetMCSControlValue
```

```
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre  
ctor Maneuver ImpulsiveMnvr.Cartesian.X MaxStep 300 m/sec');
```

```
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer  
SetMCSControlValue
```

```
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre  
ctor Maneuver ImpulsiveMnvr.Cartesian.X Correction -1581.97670664023  
m/sec');
```

```
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer  
SetMCSControlValue
```

```
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre  
ctor Maneuver ImpulsiveMnvr.Cartesian.X Perturbation 0.1 m/sec');
```

```
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer  
SetMCSControlValue
```

```
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
```



```

ctor Maneuver ImpulsiveMnvr.Cartesian.X Scale 1 m/sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Active true');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z MaxStep 300 m/sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Correction -2771.82057041661
m/sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Perturbation 0.1 m/sec');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSControlValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver ImpulsiveMnvr.Cartesian.Z Scale 1 m/sec');
    #5.14, 设置属性页中约束条件属性
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetValue
MainSequence.SegmentList.Target_Sequence1.SegmentList.Maneuver.Result
s Eccentricity CosineVFPA');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Active true');
    atkConnect(conID, 'Astrogator', '*/Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Desired 0');

```

```

    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Scale 1');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity tolerance 0.1');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver Eccentricity Weight 1');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Active true');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Desired 0 rad');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Scale 1 rad');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA tolerance 0.1 rad');
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetMCSConstraintValue
MainSequence.SegmentList.Target_Sequence1.Profiles.Differential_Corre
ctor Maneuver CosineVFPA Weight 1');
#5.15, 第三个预报段属性设置
    atkConnect(conID, 'Astrogator', '*Satellite/FastTransfer
SetValue MainSequence.SegmentList.Propagate2.SegmentColor -65536');

```

```

    atkConnect(conID, 'Astrogator', '* / Satellite / FastTransfer
SetValue MainSequence.SegmentList.Propagate2.StoppingConditions
Duration');
    atkConnect(conID, 'Astrogator', '* / Satellite / FastTransfer
SetValue
MainSequence.SegmentList.Propagate2.StoppingConditions.Duration.TripV
alue 86400 sec');
    atkConnect(conID, 'Astrogator', '* / Satellite / FastTransfer
SetValue
MainSequence.SegmentList.Propagate2.StoppingConditions.Duration.Toler
ance 0.0001 sec');

```

#5.16, 机动规划运行

```

    atkConnect(conID, 'Astrogator', '* / Satellite / FastTransfer
RunMCS');
    atkConnect(conID, 'Animate', '* Reset');
    atkConnect(conID, 'Astrogator', '* / Satellite / FastTransfer
ApplyAllProfileChanges');

```

#5.17, 查看报告数据

```

    print(atkConnect(conID, 'Report_RM', '* / Satellite / FastTransfer
Style "Position" TimePeriod "5 Nov 2022 01:00:00.000" "5 Nov 2022
02:00:00.000"))

```

#5.18, 想定保存

```

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

```

#6, 判断是否自动打开 ATK.exe

```

if ATKAutoOpen:
    subprocess.Popen([ATKPath])
    print("自动打开 ATK, 请等待...")
else:
    print("请手动打开 ATK, 等待" + str(WaitTime) + "秒")

```

```

conID = -1

```

```

count = 0

```

#7, 循环检测 ATK 连接状态

```

while 0 != conID and count < WaitTime:

```

```

#等待 1 秒
time.sleep(1)
count += 1;
print(count)
#8, 建立 ATK 连接
conID = atkOpen(ATKIP, ATKPort)

if 0 == conID:
    print("connect atk success")
    #9, 调用轨道快速转移实现函数
    TestFastTransfer();
    #10, 关闭 ATK 连接
    atkClose(conID);
else:
    print("connect atk failed, please confirm atk is open status")

```

9.2.2. 打开 cmd 命令行，切换到 Python 案例脚本目录。

切换磁盘：E:

切换路径：cd E:\ATK 软件\Release-ATK-v2.0.0.3-20240203-2\connect\Python

9.2.3. 调用嵌入式 Python 解释器解释运行 Python 案例脚本。

运行命令：.\python-3.11.0-embed-amd64\python ATKConnectTest.py

```

C:\Users\Administrator>E:
E:\>cd E:\ATK软件\Release-ATK-v2.0.0.3-20240203-2\connect\Python
E:\ATK软件\Release-ATK-v2.0.0.3-20240203-2\connect\Python>.\python-3.11.0-embed-amd64\python ATKConnectTest.py
自动打开ATK, 请等待...
1
2
connect atk success

```

图 9.2-2 cmd 命令行

9.3 案例结果

成功运行 Python 案例脚本 ATKConnectTest.py，ATK 软件会自动打开，自动新建场景和卫星，设置卫星属性实现轨道快速转移。

本案例获取数据报告的方式是通过 atkConnect 接口直接返回，Python 将在 cmd 命令框中输出打印，对应 Python 案例源码步骤 5.17，效果如下图所示

示。

```
管理员: C:\Windows\system32\cmd.exe
D:\Git\ATKConnectPython\Python>. \python-3.11.0-embed-amd64\python ATKConnectTest.py
自动打开ATK, 请等待...
connect atk success
-----
DateTime                x(m)                y(m)                z(m)
-----
2022-11-05 01:00:03.854310 -3487395.959063     -5702997.687331      37.031653
2022-11-05 01:00:15.535390 -3410097.417406     -5749680.020446      37.029844
2022-11-05 01:00:26.635553 -3336066.688106     -5793069.453548      37.018672
2022-11-05 01:00:38.175524 -3258520.590154     -5837166.749058      36.997375
2022-11-05 01:00:49.883416 -3179253.654062     -5880844.254668      36.965755
2022-11-05 01:01:01.690623 -3098724.688664     -5923801.463142      36.923721
2022-11-05 01:01:13.704185 -3016196.060701     -5966376.964466      36.870564
2022-11-05 01:01:25.721903 -2933057.034431     -6007816.221633      36.806976
2022-11-05 01:01:37.749525 -2849282.934988     -6048129.080897      36.732993
2022-11-05 01:01:50.005486 -2763352.158938     -6088004.974835      36.647069
2022-11-05 01:02:05.166094 -2666290.709703     -6135640.546475      36.526264
2022-11-05 01:02:21.642629 -2539012.860386     -6185274.506404      36.377120
2022-11-05 01:02:37.455551 -2425592.323277     -6230800.747409      36.216960
2022-11-05 01:02:53.454188 -2310015.607737     -6274744.235187      36.038526
2022-11-05 01:03:08.783838 -2198530.559025     -6314838.442791      35.852700
2022-11-05 01:03:25.321285 -2077489.339259     -6355869.485424      35.636703
2022-11-05 01:03:41.504545 -1958305.820615     -6393775.113770      35.410562
2022-11-05 01:03:56.218014 -1849351.412707     -6426298.239638      35.193008
```

图 9.3-1cmd 命令框中输出打印的数据报告

点击 ATK “开始” 按钮运行动画，拉动下方时间轴查看二维视图下的轨迹效果，如下图所示。

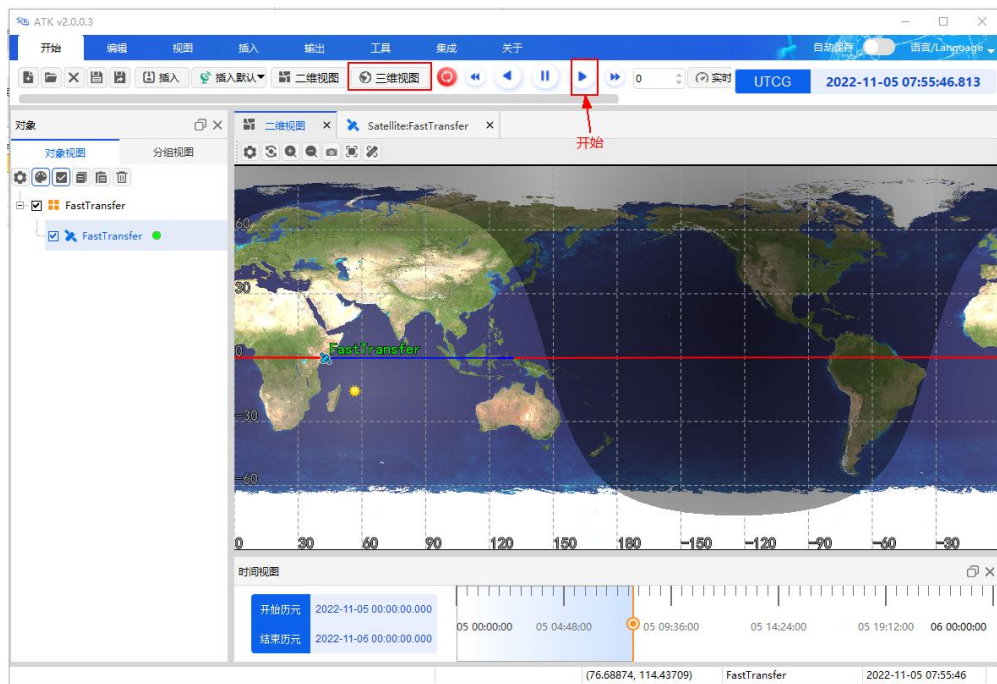


图 9.3-2 ATK 二维视图下的轨道快速转移效果图

点击 ATK “三维视图” 按钮切换至三维视图，查看三维效果，如下图所示。

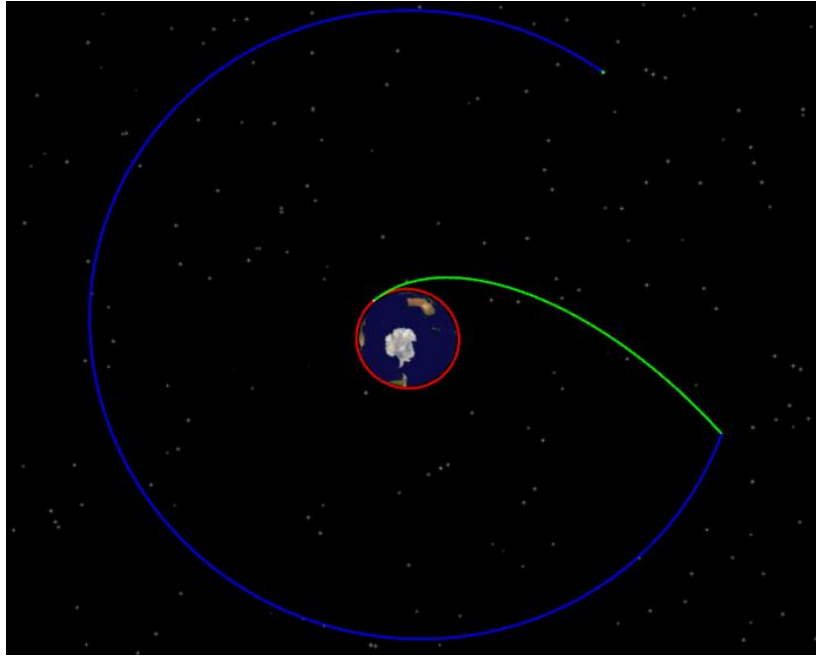


图 9.3-3 ATK 三维视图下的轨道快速转移效果图

成功运行 Python 案例脚本 ATKConnectTest.py 后，还会生成轨道快速转移想定文件“FastTransfer.xml”，存放在 ATK 根目录中，如下图所示。

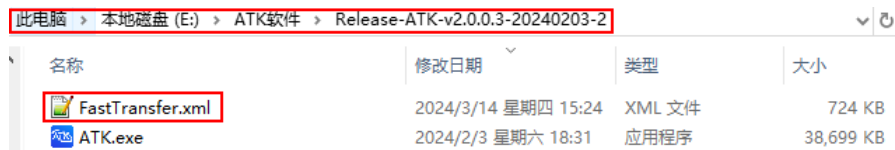


图 9.3-4 生成的轨道快速转移想定文件

10. 基于 ATK.Component 模式的轨道快速转移的 C++实现

10.1 案例介绍

本案例实现半径为 6700km 的近地停泊轨道（LEO 轨道）快速转移到半径为 42164.197km 的地球同步轨道（GEO 轨道）的轨道机动规划设计。案例使用 C++ 语言基于组件模式的 ATK.Component 动态库编码实现。

本案例使用 VS 工程，通过提供的头文件及库文件，使用 C++ 语言进行代码编译，完成案例实现，并保存想定文件。提供的文件及路径如下图所示。

Release-ATK-v2.1.0.0-20240329 > IntegratingWithATK > component

名称	修改日期	类型	大小
Include	2024/3/29 11:57	文件夹	
Lib	2024/3/29 11:57	文件夹	
Python	2024/3/29 11:57	文件夹	
IAtkObj.h	2024/3/29 11:45	C/C++ Header	11 KB
IAtkObject.dll	2024/3/29 11:45	应用程序扩展	9,670 KB
IAtkObjectD.dll	2024/3/29 11:45	应用程序扩展	18,256 KB
IAtkObjectH.h	2024/3/29 16:15	C/C++ Header	16 KB
MinSolver.dll	2024/3/29 11:45	应用程序扩展	135 KB
MinSolverD.dll	2024/3/29 11:45	应用程序扩展	207 KB

图 10.1-1 ATK.Component 提供的文件

Release-ATK-v2.1.0.0-20240329

名称	修改日期	类型
AstroData	2024/3/29 11:57	文件夹
AtkInput	2024/3/29 11:57	文件夹
bearer	2024/3/29 11:57	文件夹

图 10.1-2 ATK 提供的文件

本案例会生成想定文件“FastTransfer.xml”，报告文件“J2000 位置速度.txt”，控制变量累计校正量文件“CorrectionsData.txt”，生成文件均在工程目录 Output 文件夹中，如下图所示。

名称	修改日期	类型	大小
J2000位置速度.txt	2024/3/29 16:29	文本文档	1 KB
CorrectionsData.txt	2024/3/29 16:29	文本文档	1 KB
FastTransfer.xml	2024/3/29 16:29	XML 文档	720 KB

图 10.1-3 案例生成文件

使用 ATK 软件打开想定文件，快速转移轨道如下图所示。

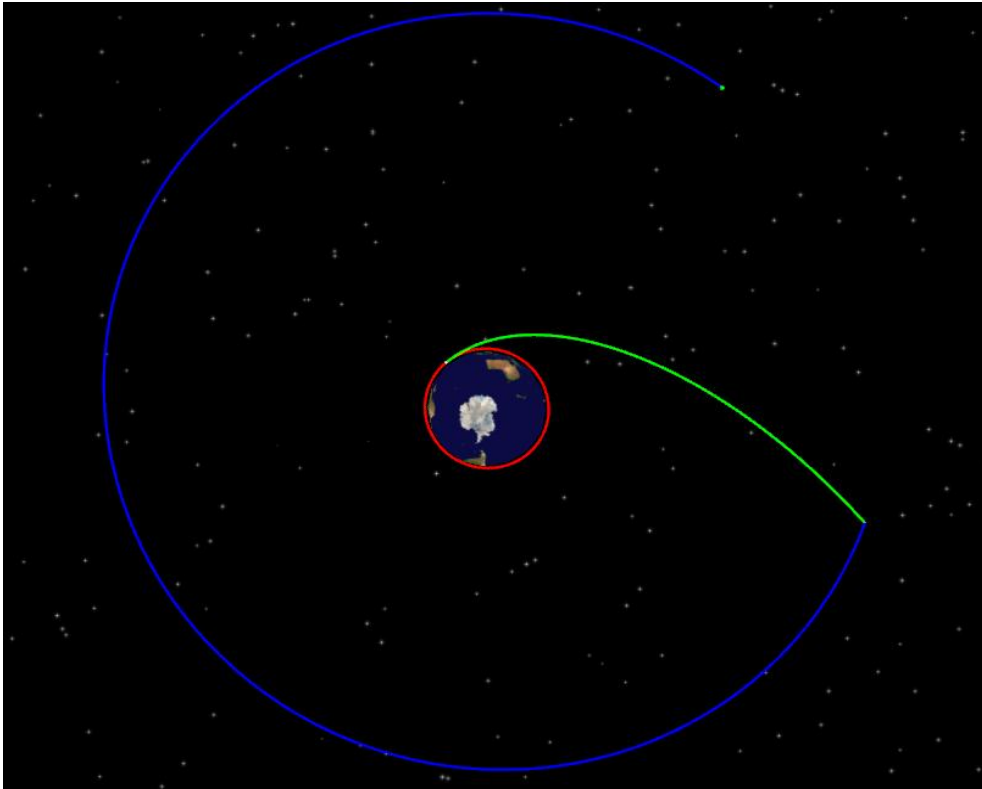


图 10.1-4 快速转移轨道

10.2 案例实现

本案例使用 Visual Studio 2015 开发实现，(ATK.Component 动态库可支持 Visual Studio 更高版本)，案例实现流程如下所示。

10.2.1. 启动 Visual Studio 2015，在“文件”菜单栏，单击“新建”然后单击“项目”，弹出新建项目对话框。

10.2.2. 在项目类型模板里选择 Visual C++项目，然后选择空项目，输入项目名称“Test”，点击“确定”按钮完成项目创建。

10.2.3. 将所需的文件拷贝到工程目录下（具体操作可查看文件末尾附录）；

10.2.4. 添加 FastTransfer.cpp 文件，包含头文件“IAtkObjectH.h”，添加 main() 函数，编写案例代码（代码可以直接拷贝编译运行）；
FastTransfer.cpp 文件代码说明如下：

```
// 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, "Satellit1");
    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 (eVASegmentTypeProp
    agate, "Propagate", "-");
    IVAMCSTargetSequence* pIVAMCSTargetSequence =
    (IVAMCSTargetSequence*)pIVAMCSSegmentCollection->Insert (eVASegmentTyp
    eTargetSequence, "TargetSequence", "-");
    IVAMCSManeuver* pIVAMCSManeuver =
    (IVAMCSManeuver*)pIVAMCSTargetSequence->GetSegments ()->Insert (eVASegm
    entTypeManeuver, "Maneuver", "-");
    IVAMCSPropagate* pIVAMCSPropagate1 =
    (IVAMCSPropagate*)pIVAMCSSegmentCollection->Insert (eVASegmentTypeProp
    agate, "Propagate", "-");
    IVAMCSTargetSequence* pIVAMCSTargetSequence1 =
    (IVAMCSTargetSequence*)pIVAMCSSegmentCollection->Insert (eVASegmentTyp
    eTargetSequence, "TargetSequence1", "-");
    IVAMCSManeuver* pIVAMCSManeuver1 =
    (IVAMCSManeuver*)pIVAMCSTargetSequence1->GetSegments ()->Insert (eVASeg
    mentTypeManeuver, "Maneuver", "-");
    IVAMCSPropagate* pIVAMCSPropagate2 =
    (IVAMCSPropagate*)pIVAMCSSegmentCollection->Insert (eVASegmentTypeProp
    agate, "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
//
IVStoppingConditionElement* pIVStoppingConditionElement =
pIVAMCSPropagate->StoppingConditions->Add("Duration");
IVStoppingCondition* pIVStoppingCondition =
(IVStoppingCondition*)pIVStoppingConditionElement->Properties;
pIVStoppingCondition->SetTrip(7200);
pIVStoppingCondition->SetTolerance(0.0001);

//
//第一个瞄准段中机动段属性设置
//
pIVAMCSManeuver->SetManeuverType(eVAManeuverTypeImpulsive);
IVManeuverImpulsive* pIVManeuverImpulsive =
(IVManeuverImpulsive*)pIVAMCSManeuver->Maneuver;
IVAAttitudeControlImpulsiveThrustVector*
pIVAAttitudeControlImpulsiveThrustVector =
(IVAAttitudeControlImpulsiveThrustVector*)pIVAManeuverImpulsive->AttitudeControl;
pIVAAttitudeControlImpulsiveThrustVector->SetThrustAxesName("Satellite VNC(Earth)");
pIVAMCSManeuver->EnableControlParameter(eVAControlManeuverImpulsiveCartesianX);
pIVAMCSManeuver->Results->Add("Radius_Of_Apoapsis");

//
//第一个瞄准段添加属性页
//
IVAProfileDifferentialCorrector1*
pIVAProfileDifferentialCorrector1 =
(IVAProfileDifferentialCorrector1*)pIVAMCSTargetSequence->GetProfiles()->Add("Differential Corrector");
IVADCCControl* pIVADCCControl =
pIVAProfileDifferentialCorrector1->ControlParameters->GetControlByPaths("Maneuver", "ImpulseX");
IVADCCResult* pIVADCCResult =
pIVAProfileDifferentialCorrector1->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 =
pIVAMCSPropagatel->StoppingConditions->Add("RMagnitude");
IVASToppingCondition* pIVASToppingCondition1 =
(IVASToppingCondition*)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 =
(IVAManeuverImpulsive*)pIVAMCSManeuver1->Maneuver;
IVAAttitudeControlImpulsiveThrustVector*pIVAAttitudeControlImpuls
iveThrustVector1 =
(IVAAttitudeControlImpulsiveThrustVector*)pIVAManeuverImpulsive1->Att
itudeControl;
pIVAAttitudeControlImpulsiveThrustVector1->SetThrustAxesName("Sat
ellite VNC(Earth)");
pIVAMCSManeuver1->EnableControlParameter(eVAControlManeuverImpuls
iveCartesianX);
pIVAMCSManeuver1->EnableControlParameter(eVAControlManeuverImpuls
iveCartesianZ);
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->GetProfiles()->Add("Differential Corrector");
//
//属性页中控制变量属性设置
//
    IVADCControl* pIVADCControl1 =
pIVAProfileDifferentialCorrector11->ControlParameters->Item(0);
    pIVADCControl1->SetEnable(true);
    pIVADCControl1->SetMaxStep(300);
    pIVADCControl1->SetCorrection(-1581.97670664023);
    pIVADCControl1->SetPerturbation(0.1);
    pIVADCControl1->SetScalingValue(1);
    IVADCControl* pIVADCControl2 =
pIVAProfileDifferentialCorrector11->ControlParameters->Item(1);
    pIVADCControl2->SetEnable(true);
    pIVADCControl2->SetMaxStep(300);
    pIVADCControl2->SetCorrection(-2771.82057041661);
    pIVADCControl2->SetPerturbation(0.1);
    pIVADCControl2->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");
}

```

10.2.5. 点击“本地 windows 调试器”运行程序，使用保存想定接口，会生成想定文件，在项目目录 Output 文件夹中可查看生成文件。

10.3 案例结果

结果展示：

生成文件均在工程目录 Output 文件夹中，如下图：

名称	修改日期	类型	大小
J2000位置速度.txt	2024/3/29 16:29	文本文档	1 KB
CorrectionsData.txt	2024/3/29 16:29	文本文档	1 KB
FastTransfer.xml	2024/3/29 16:29	XML 文档	720 KB

图 10.3-1 案例生成文件

调用输出数据报告接口，会生成报告文件（接口返回值为数据报告文件位置），如下图：

```
报告文件位置： C:\Users\A411-2\Desktop\临时\Test\Test\Output\ J2000位置速度.txt
6.7e+06 0 0 0 7713.14 0
std::string strReportFilePath = pRoot->OutputDataReport(pISatellite,
```

图 10.3-2 生成数据报告文件位置

可输出卫星位置速度变化过程，如下图：

```
6.7e+06 0 0 0 7713.14 0
6.69979e+06 53100.9 -0.0023229 -61.2196 7712.9 -0.000671403
6.69909e+06 110616 -0.00996945 -127.528 7712.09 -0.00137523
6.69798e+06 164450 -0.0218037 -189.594 7710.82 -0.00201144
6.69594e+06 232904 -0.0431385 -268.514 7708.48 -0.00278819
6.69346e+06 295812 -0.0686991 -341.041 7705.62 -0.00346958
6.68898e+06 383923 -0.1136 -442.624 7700.46 -0.00437087
6.68343e+06 470513 -0.167465 -542.454 7694.08 -0.00519554
6.67722e+06 551643 -0.226106 -635.991 7686.92 -0.00591299
6.66839e+06 649551 -0.306616 -748.871 7676.77 -0.00670784
6.65825e+06 746261 -0.395735 -860.371 7665.09 -0.00741755
6.64721e+06 838759 -0.489071 -967.016 7652.39 -0.00802755
6.63433e+06 935084 -0.593853 -1078.07 7637.56 -0.00859321
```

图 10.3-3 卫星位置速度数据输出

保存想定后，本案例生成 FastTransfer.xml 想定文件；启动 ATK 软件，打开生成的想定文件 FastTransfer.xml，三维展示卫星快速转移轨迹如下图：

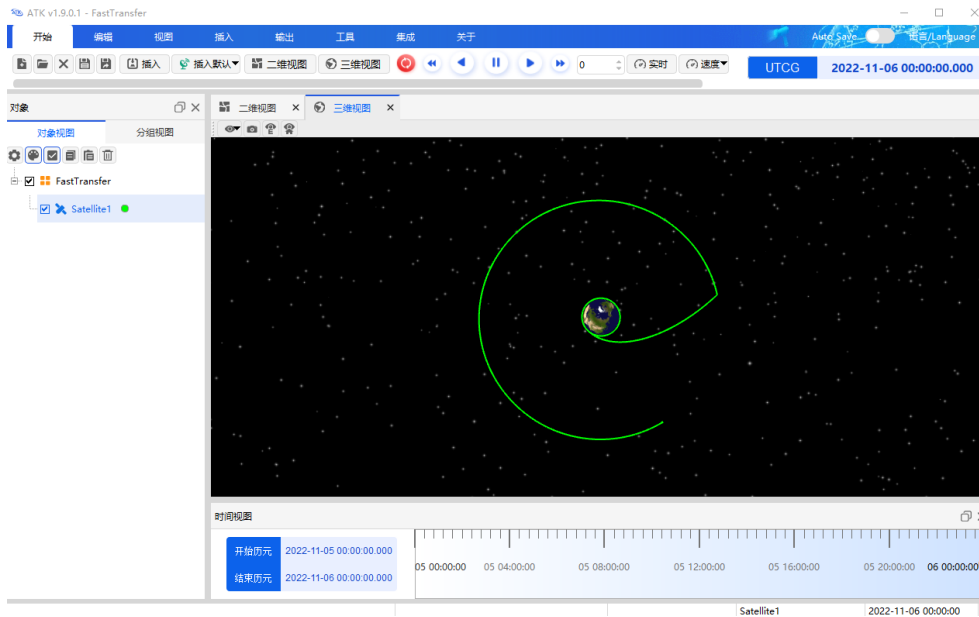


图 10.3-4 卫星快速转移轨迹

10.4 附录

以 VisualStudio2015 为例，（库文件可支持 vs 更高版本），使用动态库进行项目规划。已新建工程 Test，并新建 cpp 文件 FastTransfer.cpp。

10.4.1. 文件配置

在 ATK 安装包目录中，点开 Component 文件夹。

Release-ATK-v2.1.0.0-20240329 > IntegratingWithATK > component

名称	修改日期	类型	大小
Include	2024/3/29 11:57	文件夹	
Lib	2024/3/29 11:57	文件夹	
Python	2024/3/29 11:57	文件夹	
IAtkObj.h	2024/3/29 11:45	C/C++ Header	11 KB
IAtkObject.dll	2024/3/29 11:45	应用程序扩展	9,670 KB
IAtkObjectD.dll	2024/3/29 11:45	应用程序扩展	18,256 KB
IAtkObjectH.h	2024/3/29 16:15	C/C++ Header	16 KB
MinSolver.dll	2024/3/29 11:45	应用程序扩展	135 KB
MinSolverD.dll	2024/3/29 11:45	应用程序扩展	207 KB

图 10.4-1 ATK.Component 文件夹

将所有文件添加至工程 Test 根目录中。

Test > Test

名称	修改日期	类型	大小
Include	2024/3/29 15:44	文件夹	
Lib	2024/3/29 15:44	文件夹	
IAtkObj.h	2024/3/27 16:55	C/C++ Header	11 KB
IAtkObject.dll	2024/3/29 11:32	应用程序扩展	9,670 KB
IAtkObjectD.dll	2024/3/29 11:27	应用程序扩展	18,256 KB
IAtkObjectH.h	2024/3/29 16:17	C/C++ Header	17 KB
MinSolver.dll	2024/3/13 15:51	应用程序扩展	135 KB
MinSolverD.dll	2024/3/20 16:01	应用程序扩展	207 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 10. 4-2 文件添加

将需要的动态库与配置文件添加至此文件夹。（其中 AstroData 文件夹在 ATK 安装目录下）。

AstroData	2024/3/29 15:45	文件夹	
Include	2024/3/29 15:44	文件夹	
Lib	2024/3/29 15:44	文件夹	
IAtkObj.h	2024/3/27 16:55	C/C++ Header	11 KB
IAtkObject.dll	2024/3/29 11:32	应用程序扩展	9,670 KB
IAtkObjectD.dll	2024/3/29 11:27	应用程序扩展	18,256 KB
IAtkObjectH.h	2024/3/29 16:17	C/C++ Header	17 KB
MinSolver.dll	2024/3/13 15:51	应用程序扩展	135 KB
MinSolverD.dll	2024/3/20 16:01	应用程序扩展	207 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 10. 4-3 动态库与配置文件添加

10.4.2. Debug 属性配置

打开项目右键->属性。

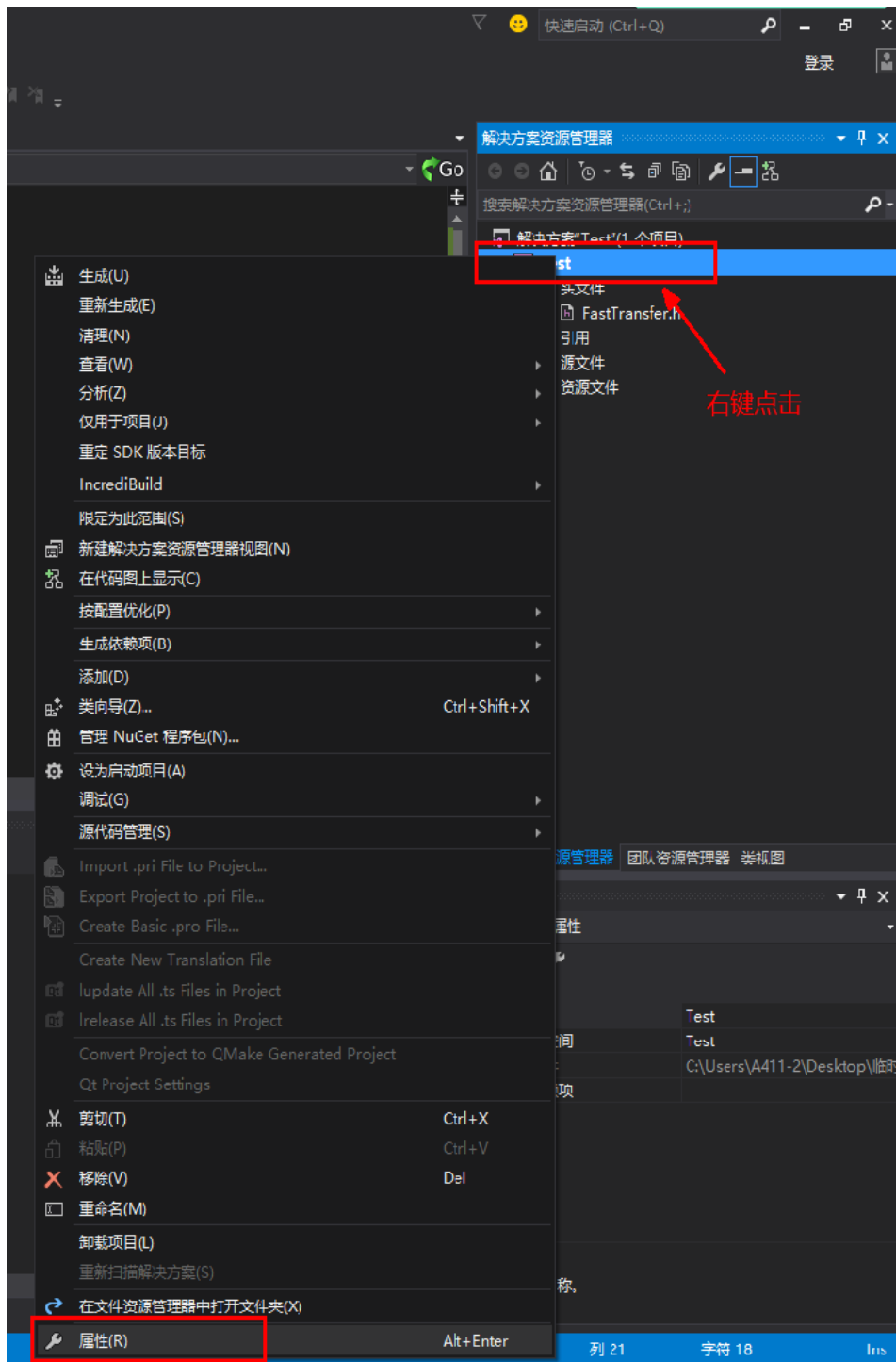


图 10.4-4 属性设置步骤

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。

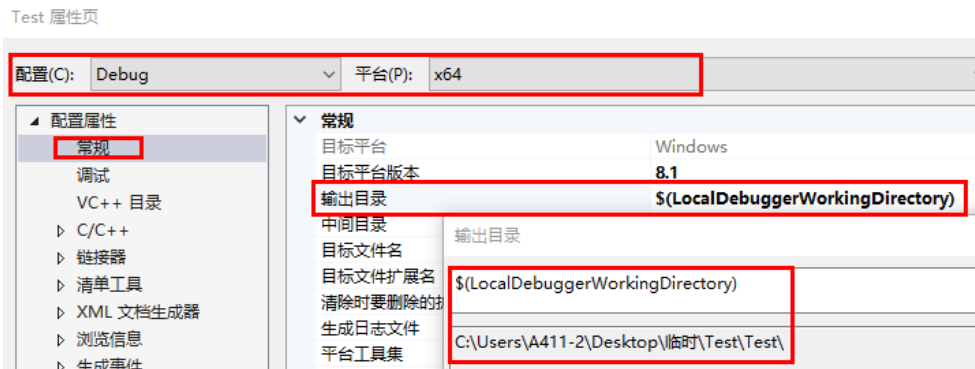


图 10.4-5 修改输出目录

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 10.4-6 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。

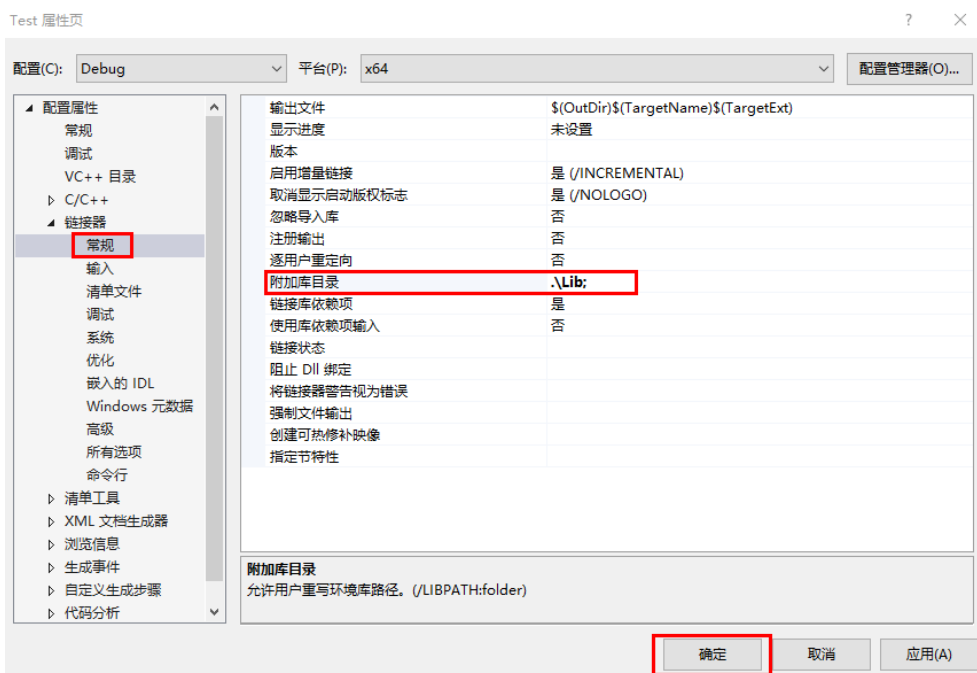


图 10.4-7 附加库设置

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。

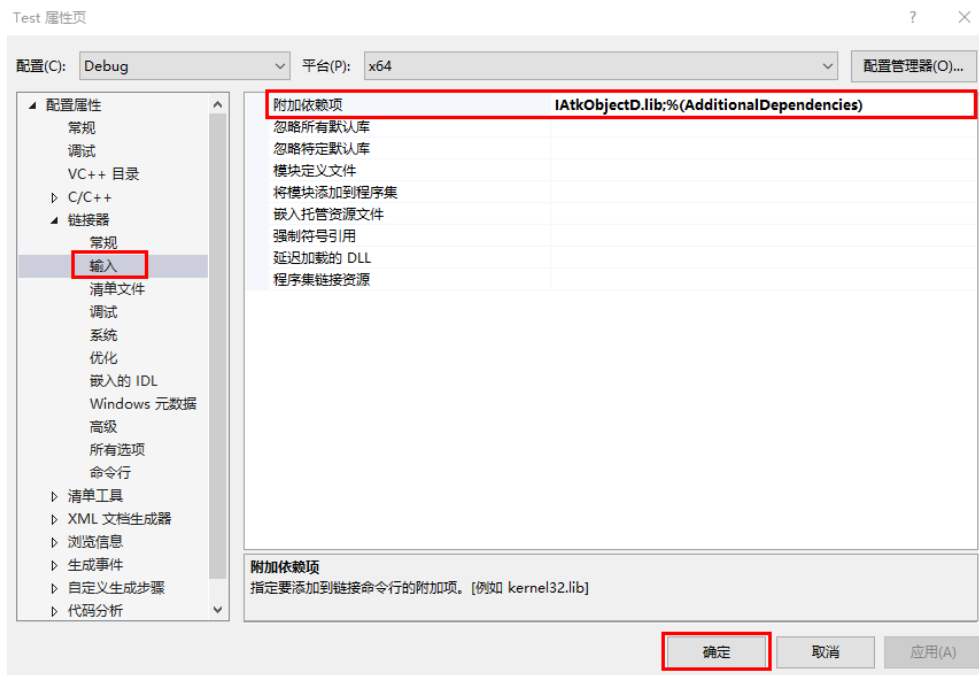


图 10.4-8 附加库设置

10.4.3. Release 属性配置

打开项目右键->属性。

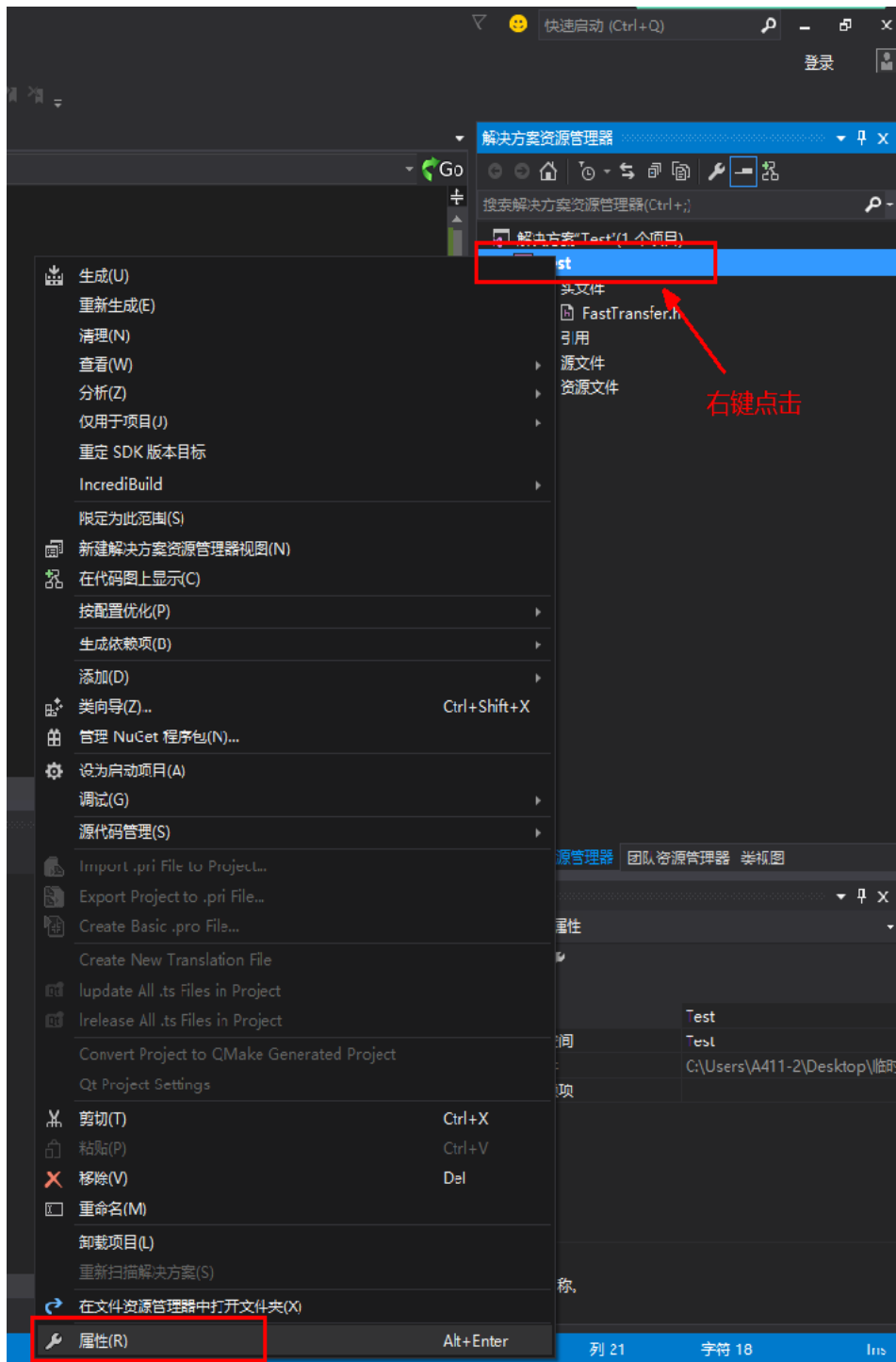


图 10.4-9 属性设置步骤

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。

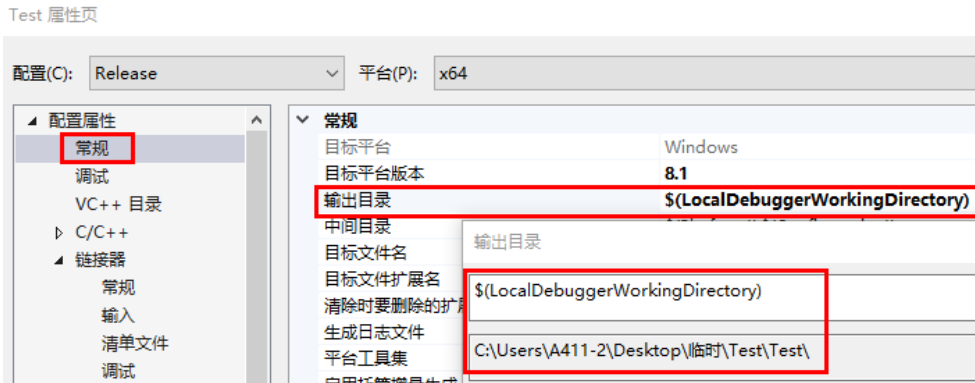


图 10.4-10 修改输出目录

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 10.4-11 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。

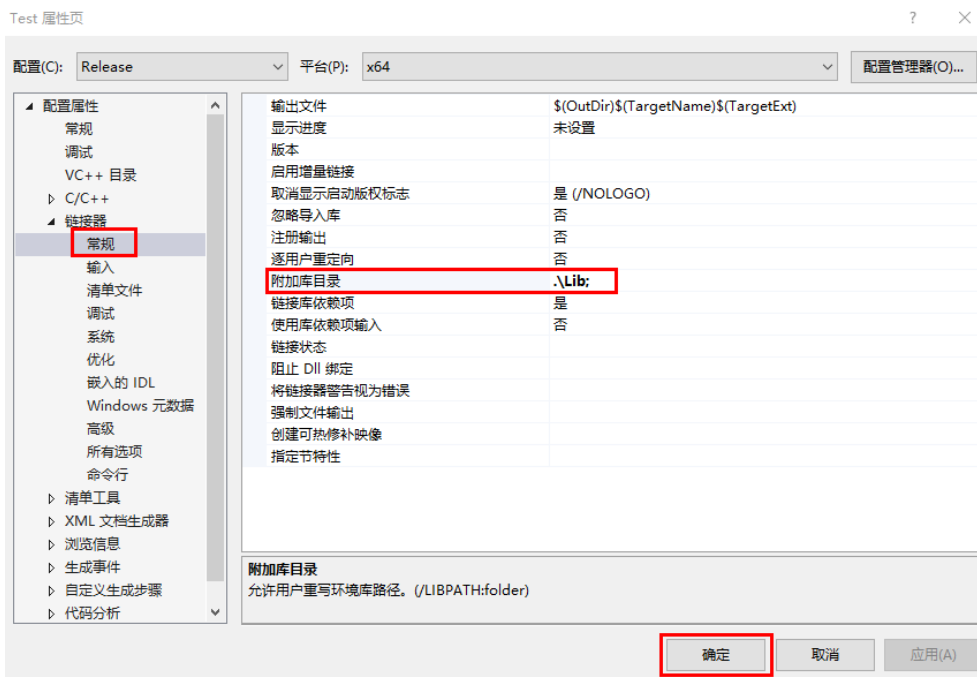


图 10.4-12 附加库设置

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。

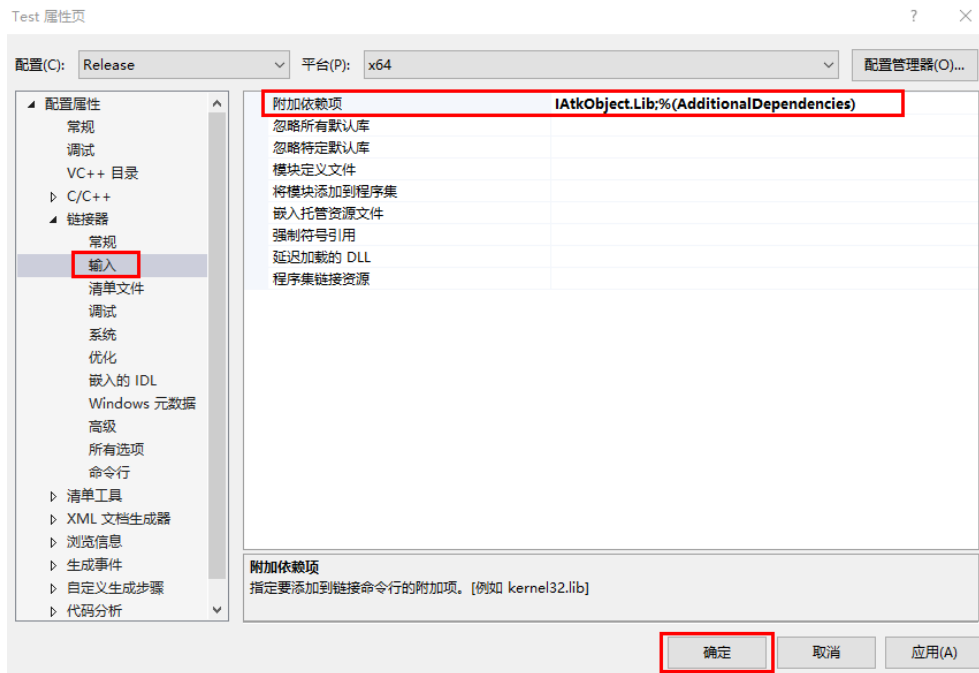


图 10.4-13 附加库设置

11. 基于 ATK.Component 模式的轨道快速转移的 Java 实现

11.1 案例介绍

本案例实现半径为 6700km 的近地停泊轨道（LEO 轨道）快速转移到半径为 42164.197km 的地球同步轨道（GEO 轨道）的轨道机动规划设计。案例基于 Component 模式使用 Java 脚本实现。

11.1.1. 本案例依赖 Java 环境

JDK (Java Development Kit), 官网下载地址:

www.oracle.com/java/technologies/download, 选择相应版本下载, 按默认流程安装后, 配置以下 Java 路径到系统环境变量:

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;

注意, 添加到 Path 的路径建议上移到第一行, 不然可能会被电脑上其他软件里的 Java 环境影响。

11.1.2. 本案例依赖以下文件 (均包含在 ATK 安装包中):

文件 1. com/atk/component, ATKComponent 模式接口类目录, 提供 Component 模式下所有公用类对象与函数的接口封装。已编译, 可直接调用执行。

文件 2. IAtkObjectD.dll, ATKComponent 模式动态库, 供 Java 案例脚本加载导入, 包含 Component 模式下所有公用类对象与函数的具体实现。

文件 3. MinSolverD.dll, ATKComponent 模式依赖动态库, 供 Java 案例脚本加载导入, ATK 运行计算依赖该文件。

文件 4. ATKComponentJavaTest.java, Java 案例脚本, 包含相关动态库导入, 以及轨道快速转移案例的具体实现过程。该脚本通过 Java 先编译再执行。



名称	修改日期	类型	大小
1 com	2024/3/28 星期...	文件夹	
4 ATKComponentJavaTest.java	2024/3/29 星期...	JAVA 文件	30 KB
2 IAtkObjectD.dll	2024/3/28 星期...	应用程序扩展	19,080 KB
3 MinSolverD.dll	2024/3/1 星期五 ...	应用程序扩展	207 KB

图 11.1-1 ATK.Component.Java 提供的文件



名称	修改日期	类型	大小
ATKComponentJavaModule.class	2024/3/28 星期...	CLASS 文件	29 KB
ATKComponentJavaModuleJNI.class	2024/3/28 星期...	CLASS 文件	113 KB
EAccessConstraints.class	2024/3/28 星期...	CLASS 文件	5 KB

图 11.1-2 ATKComponent 模式接口类目录

11.2 案例实现

本案例使用 Java 脚本实现, 通过在案例脚本的编译与执行命令中, 添加接口类目录的相对路径, 使案例脚本能直接调用 ATKComponent 模式下的所有类接口与函数接口, 从而完成场景与卫星类对象的新建与设置, 动画的仿真运行, 数据报告与想定文件的输出保存。具体流程如下所示。

11.2.1. 编辑 Java 案例脚本, 流程图与源码如下:

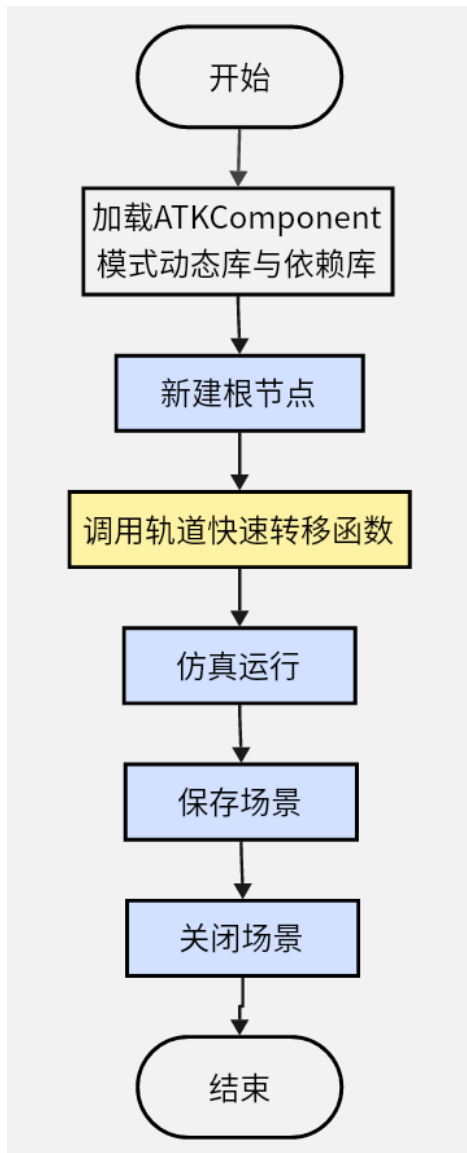


图 11.2-1 Java 案例流程图

```

//ATKComponentJavaTest.java
public class ATKComponentJavaTest{
  //1, 加载 ATKComponent 模式动态库
  static {
    try {
      System.loadLibrary("MinSolverD");
      System.loadLibrary("IAtkObjectD");
    } catch (UnsatisfiedLinkError e) {
      System.err.println("load dll failed\n" + e);
      System.exit(1);
    }
  }
}
  
```

```

public static void main(String []argv) {
    //2, 新建根结点
    IAtkObjectRoot pIAtkObjectRoot = new IAtkObjectRoot();
    //3, 调用轨道快速转移函数
    TestFastTransfer(pIAtkObjectRoot);
    //4, 仿真运行
    pIAtkObjectRoot.getM_Animation().PlayForward();
    //5, 保存场景
    pIAtkObjectRoot.SaveScenario();
    //6, 关闭场景
    pIAtkObjectRoot.CloseScenario();
}

//3, 轨道快速转移函数
public static void TestFastTransfer(IAtkObjectRoot
pIAtkObjectRoot) {
    //3.1, 场景新建与属性设置
    IAtkObject pIAtkObjectIScenario =
    pIAtkObjectRoot.GetChildren().New(EATKObjectType.eScenario, "FastTrans
fer");
    IScenario pIScenario =
    ATKComponentJavaModule.IAtkObject2IScenario(pIAtkObjectIScenario);
    pIScenario.SetTimePeriod("5 Nov 2022 00:00:00.000", "6 Nov 2022
00:00:00.000");
    //3.2, 卫星新建与轨道预报设置为机动规划
    IAtkObject pIAtkObjectSatellite =
    pIScenario.GetChildren().New(EATKObjectType.eSatellite, "Satellite1");
    ISatellite pISatellite =
    ATKComponentJavaModule.IAtkObject2ISatellite(pIAtkObjectSatellite);
    pISatellite.SetPropagatorType(EVePropagatorType.ePropagatorAstrog
ator);
    IVADriverMCS pIVADriverMCS =
    ATKComponentJavaModule.IVePropagator2IVADriverMCS(pISatellite.GetProp
agator());
}

```

```

    IVAMCSSegmentCollection pIVAMCSSegmentCollection =
pIVADriverMCS.GetMainSequence();
    //3.3, 机动规划添加段, 新添加卫星会有默认初始段
    if (EVASegmentType.eVASegmentTypeInitialState !=
pIVAMCSSegmentCollection.Item(0).getType()){
        return;
    }
    IVAMCSInitialState pIVAMCSInitialState =
ATKComponentJavaModule.IVAMCSSegment2IVAMCSInitialState(pIVAMCSSegmen
tCollection.Item(0));
    IVAMCSPropagate pIVAMCSPropagate =
ATKComponentJavaModule.IVAMCSSegment2IVAMCSPropagate(pIVAMCSSegmentCo
llection.Insert(EVASegmentType.eVASegmentTypePropagate, "Propagate",
"-"));
    IVAMCSTargetSequence pIVAMCSTargetSequence =
ATKComponentJavaModule.IVAMCSSegment2IVAMCSTargetSequence(pIVAMCSSegm
entCollection.Insert(EVASegmentType.eVASegmentTypeTargetSequence,
"TargetSequence", "-"));
    IVAMCSManeuver pIVAMCSManeuver =
ATKComponentJavaModule.IVAMCSSegment2IVAMCSManeuver(pIVAMCSTargetSequ
ence.GetSegments().Insert(EVASegmentType.eVASegmentTypeManeuver,
"Maneuver", "-"));
    IVAMCSPropagate pIVAMCSPropagate1 =
ATKComponentJavaModule.IVAMCSSegment2IVAMCSPropagate(pIVAMCSSegmentCo
llection.Insert(EVASegmentType.eVASegmentTypePropagate, "Propagate",
"-"));
    IVAMCSTargetSequence pIVAMCSTargetSequence1 =
ATKComponentJavaModule.IVAMCSSegment2IVAMCSTargetSequence(pIVAMCSSegm
entCollection.Insert(EVASegmentType.eVASegmentTypeTargetSequence,
"TargetSequence1", "-"));
    IVAMCSManeuver pIVAMCSManeuver1 =
ATKComponentJavaModule.IVAMCSSegment2IVAMCSManeuver(pIVAMCSTargetSequ
ence1.GetSegments().Insert(EVASegmentType.eVASegmentTypeManeuver,
"Maneuver", "-"));
    IVAMCSPropagate pIVAMCSPropagate2 =

```

```

ATKComponentJavaModule. IVAMCSSegment2IVAMCSPropagate (pIVAMCSSegmentCo
llection. Insert (EVASegmentType. eVASegmentTypePropagate, "Propagate",
"-"));
    //3.4, 初始段属性设置
    pIVAMCSInitialState.SetOrbitEpoch("5 Nov 2022 00:00:00.000");
    pIVAMCSInitialState.SetElementType (EVAElementType. eVAElementTypeK
eplerian);
    IVAElementKeplerian pIVAElementKeplerian =
ATKComponentJavaModule. IVAElement2IVAElementKeplerian (pIVAMCSInitialS
tate. GetElement ());
    pIVAElementKeplerian. SetSemiMajorAxis (6700);
    pIVAElementKeplerian. SetEccentricity (0);
    pIVAElementKeplerian. SetInclination (0);
    pIVAElementKeplerian. SetRAAN (0);
    pIVAElementKeplerian. SetArgOfPeriapsis (0);
    pIVAElementKeplerian. SetTrueAnomaly (0);
    //3.5, 第一个预报段属性设置
    IVAStoppingConditionElement pIVAStoppingConditionElement =
pIVAMCSPropagate. GetStoppingConditions (). Add ("Duration");
    IVAStoppingCondition pIVAStoppingCondition =
ATKComponentJavaModule. IVAStoppingConditionComponent2IVAStoppingCondi
tion (pIVAStoppingConditionElement. GetProperties ());
    pIVAStoppingCondition. SetTrip (7200);
    pIVAStoppingCondition. SetTolerance (0.0001);
    //3.6, 第一个瞄准段中机动段属性设置
    pIVAMCSManeuver. SetManeuverType (EVAManeuverType. eVAManeuverTypeIm
pulsive);
    IVAManeuverImpulsive pIVAManeuverImpulsive =
ATKComponentJavaModule. IVAManeuver2IVAManeuverImpulsive (pIVAMCSManeuver.
GetManeuver ());
    IVAAttitudeControlImpulsiveThrustVector
pIVAAttitudeControlImpulsiveThrustVector =
ATKComponentJavaModule. IVAAttitudeControl2IVAAttitudeControlImpulsive
ThrustVector (pIVAManeuverImpulsive. GetAttitudeControl ());
    pIVAAttitudeControlImpulsiveThrustVector. SetThrustAxesName ("Satel

```

```

lite VNC(Earth)");
    pIVAMCSManeuver.EnableControlParameter(EVAControlManeuver.eVAControlManeuverImpulsiveCartesianX);
    pIVAMCSManeuver.getResults().Add("Radius_Of_Apoapsis");
    //3.7, 第一个瞄准段添加属性页
    IVAProfileDifferentialCorrector1
pIVAProfileDifferentialCorrector1 =
ATKComponentJavaModule.IVAProfile2IVAProfileDifferentialCorrector1(pIVAMCSTargetSequence.GetProfiles().Add("Differential Corrector"));
    IVADCCControl pIVADCCControl =
pIVAProfileDifferentialCorrector1.GetControlParameters().GetControlByPaths("Maneuver", "ImpulseX");
    IVADCCResult pIVADCCResult =
pIVAProfileDifferentialCorrector1.GetResults().GetResultByPaths("Maneuver", "RadiusOfApoapsis");
    //3.8, 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);
    //3.9, 属性页中约束条件属性设置
    pIVADCCResult.SetEnable(true);
    pIVADCCResult.SetDesiredValue(84328394);
    pIVADCCResult.SetScalingValue(1);
    pIVADCCResult.SetTolerance(0.1);
    pIVADCCResult.SetWeight(1);
    //3.10, 第二个预报段属性设置
    IVAStoppingConditionElement pIVAStoppingConditionElement1 =
pIVAMCSPropagate1.GetStoppingConditions().Add("RMagnitude");
    IVAStoppingCondition pIVAStoppingCondition1 =
ATKComponentJavaModule.IVAStoppingConditionComponent2IVAStoppingCondition(pIVAStoppingConditionElement1.GetProperties());
    pIVAStoppingCondition1.SetTrip(42164197);
    pIVAStoppingCondition1.SetTolerance(1e-6);

```

```

    pIVAStoppingCondition1.SetRepeatCount(1);
    pIVAStoppingCondition1.SetCriterion(EVACriterion.eVACriterionCrossEither);
    //3.11, 第二个瞄准段中机动段属性设置
    pIVAMCSManeuver1.SetManeuverType(EVAManeuverType.eVAManeuverTypeImpulsive);
    IVAManeuverImpulsive pIVAManeuverImpulsive1 =
    ATKComponentJavaModule.IVAManeuver2IVAManeuverImpulsive(pIVAMCSManeuver1.GetManeuver());
    IVAAttitudeControlImpulsiveThrustVector
    pIVAAttitudeControlImpulsiveThrustVector1 =
    ATKComponentJavaModule.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");
    //3.12, 第二个瞄准段添加属性页
    IVAProfileDifferentialCorrector1
    pIVAProfileDifferentialCorrector11 =
    ATKComponentJavaModule.IVAProfile2IVAProfileDifferentialCorrector1(pIVAMCSManeuver1.GetProfiles().Add("Differential Corrector"));
    //3.13, 属性页中控制变量属性设置
    IVADCCControl pIVADCCControl11 =
    pIVAProfileDifferentialCorrector11.GetControlParameters().Item(0);
    pIVADCCControl11.SetEnable(true);
    pIVADCCControl11.SetMaxStep(300);
    pIVADCCControl11.SetCorrection(-1581.97670664023);
    pIVADCCControl11.SetPerturbation(0.1);
    pIVADCCControl11.SetScalingValue(1);
    IVADCCControl pIVADCCControl12 =

```

```

pIVAProfileDifferentialCorrector11.GetControlParameters().Item(1);
    pIVADCControl2.SetEnable(true);
    pIVADCControl2.SetMaxStep(300);
    pIVADCControl2.SetCorrection(-2771.82057041661);
    pIVADCControl2.SetPerturbation(0.1);
    pIVADCControl2.SetScalingValue(1);
//3.14, 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);
//3.15, 第三个预报段属性设置
    IVAStoppingConditionElement pIVAStoppingConditionElement2 =
pIVAMCSPropagate2.GetStoppingConditions().Add("Duration");
    IVAStoppingCondition pIVAStoppingCondition2 =
ATKComponentJavaModule.IVAStoppingConditionComponent2IVAStoppingCondi
tion(pIVAStoppingConditionElement2.GetProperties());
    pIVAStoppingCondition2.SetTrip(86400);
    pIVAStoppingCondition2.SetTolerance(0.0001);
//3.16, 机动规划运行
    pIVADriverMCS.RunMCS();
    pIVADriverMCS.ApplyAllProfileChanges();
//3.17, 生成数据到文件
    String strReportFilePath =
pIAtkObjectRoot.OutputDataReport(pISatellite, "J2000 位置速度", "5

```

```

Nov 2022 00:00:00.000", "6 Nov 2022 00:00:00.000");
    pIAtkObjectRoot.getM_Animation().OutputPropCortsData(pISatellite)
;
    //3.18, 输出动画数据
    pIAtkObjectRoot.getM_Animation().OutputData();
}
}
}

```

11.2.2. 打开 cmd 命令行，切换到 Java 案例脚本目录。

切换磁盘：E:

切换路径：cd E:\ATK 软件\Release-ATK-v2.1.0.0-20240329\IntegratingWithATK\component\Java

11.2.3. 编译并执行 Java 案例脚本

编译命令：javac -cp ./com/atk/component -encoding utf-8
ATKComponentJavaTest.java

执行命令：java -cp .;./com/atk/component ATKComponentJavaTest
-cp ./com/atk/component：表示添加接口类目录路径到编译执行环境。
-encoding utf-8：表示设置 utf-8 编码，解决中文编译报错问题。

```

D:\>e:
E:\>cd E:\ATK软件\Release-ATK-v2.1.0.0-20240329\IntegratingWithATK\component\Java
E:\ATK软件\Release-ATK-v2.1.0.0-20240329\IntegratingWithATK\component\Java>javac -cp ./com/atk/component -encoding utf-8 ATKComponentJavaTest.java
E:\ATK软件\Release-ATK-v2.1.0.0-20240329\IntegratingWithATK\component\Java>java -cp .;./com/atk/component ATKComponentJavaTest
=====
S N O P T 7.2-8 (Apr 2008)
=====
SNMEMA EXIT 100 -- finished successfully
SNMEMA INFO 104 -- memory requirements estimated

```

图 11.2-2cmd 命令行

11.3 案例结果

成功执行 Java 案例脚本 ATKComponentJavaTest.java，会在 cmd 命令框中输出打印仿真动画数据，对应 Java 案例源码步骤 3.18，效果如下图所示。


```

6.7e+06 0 0 0 7713.14 0
6.69979e+06 53100.9 -0.0023229 -61.2196 7712.9 -0.000671403
6.69908e+06 110788 -0.0100001 -127.727 7712.09 -0.0013773
6.69773e+06 174426 -0.0244807 -201.095 7710.53 -0.0021269
6.69596e+06 232400 -0.0429562 -267.932 7708.5 -0.0027826
6.69331e+06 299154 -0.0702114 -344.894 7705.44 -0.0035049
6.68939e+06 376698 -0.109532 -434.294 7700.93 -0.00429931

```

图 11.3-1 cmd 命令框中输出打印的动画数据

同时生成的还有，卫星数据报告 J2000 位置速度.txt 和卫星机动规划数据修正报告 CorrectionsData.txt，对应案例源码步骤 3.17。以及轨道快速转移想定文件 FastTransfer.xml，对应案例源码步骤 5。以上文件均存放在输出目录 Output，如下图所示。

Release-ATK-v2.1.0.0-20240329 > IntegratingWithATK > component > Java > Output

名称	修改日期	类型	大小
CorrectionsData.txt	2024/4/1 星期一 ...	TXT 文件	1 KB
FastTransfer.xml	2024/4/1 星期一 ...	XML 文件	4,484 KB
J2000位置速度.txt	2024/4/1 星期一 ...	TXT 文件	1 KB

图 11.3-2 生成的想定文件与数据报告

打开 ATK 检查生成的想定文件，选择轨道快速转移想定文件 FastTransfer.xml，点击“开始”按钮运行动画，拉动下方时间轴查看二维视图下的轨迹效果，如下图所示。

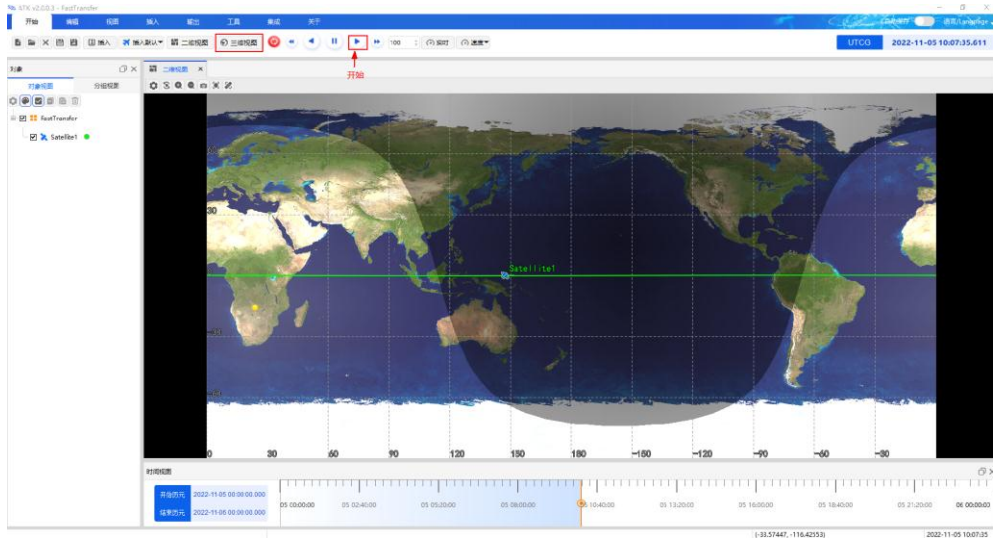


图 11.3-3 ATK 二维视图下的轨道快速转移效果图

点击 ATK “三维视图”按钮切换至三维视图，查看三维效果，如下图所示。

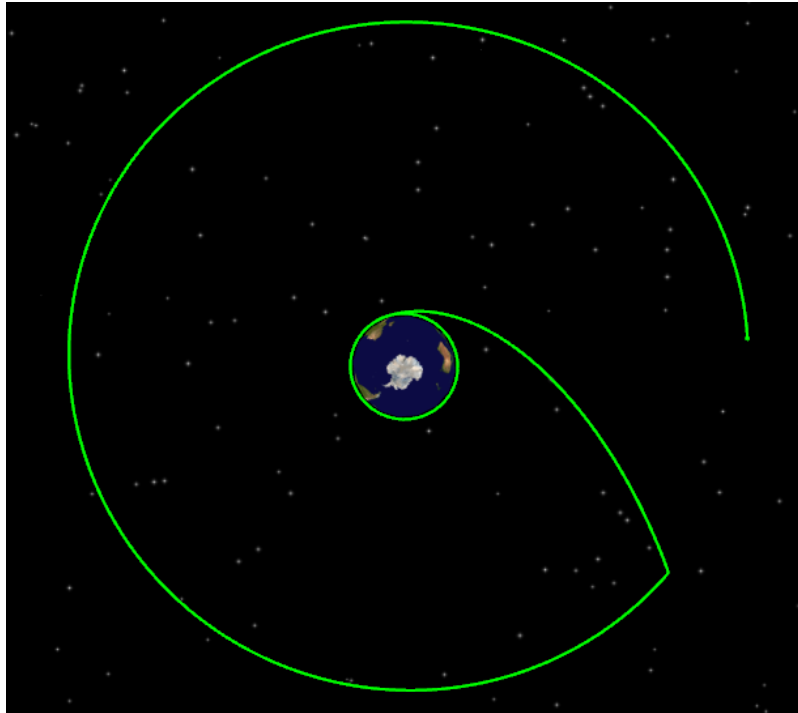


图 11.3-4 ATK 三维视图下的轨道快速转移效果图

12. 基于 ATK.Component 模式的轨道快速转移的 Python 实现

12.1 案例介绍

本案例实现半径为 6700km 的近地停泊轨道（LEO 轨道）快速转移到半径为 42164.197km 的地球同步轨道（GEO 轨道）的轨道机动规划设计。案例基于 Component 模式使用 Python 脚本实现。

本案例依赖以下文件（均包含在 ATK 安装包中）：

文件 1. python-3.11.0-embed-amd64，嵌入式 Python 解释器，实现不安装 Python，也能解释运行 Python 脚本。

文件 2. ATKComponentPythonModule.py，ATKComponent 模式接口文件，提供 Component 模式下所有公用类对象与函数的接口模块，在 Python 案例脚本中导入该模块后，就能调用这些公用的类对象与函数。

文件 3. _ATKComponentPythonModule.cp311-win_amd64.pyd，ATKComponent 模式动态库，供接口模块调用，包含 Component 模式下所有公用类对象与函数的具体实现。

文件 4. MinSolver.dll，ATKComponent 模式依赖动态库。

文件 5. ATKComponentPythonTest.py，Python 案例脚本，包含轨道快速转

移案例的具体实现过程。该 Python 案例脚本通过嵌入式 Python 解释器解释运行。

文件 6. AstroData, ATK 默认配置文件, 包含类对象的默认值, 以及显示计算所需的相关参数。该文件夹由案例脚本自动复制到依赖目录。

名称	修改日期	类型	大小
python-3.11.0-embed-amd64	2024/3/29 星期...	文件夹	
_ATKComponentPythonModule.cp31...	2024/3/25 星期...	PYD 文件	11,542 KB
ATKComponentPythonModule.py	2024/3/25 星期...	PY 文件	318 KB
ATKComponentPythonTest.py	2024/3/29 星期...	PY 文件	25 KB
MinSolver.dll	2023/8/4 星期五 ...	应用程序扩展	135 KB

图 12.1-1 ATK.Component.Python 提供的文件

名称	修改日期	类型	大小
AstroData	2024/3/6 星期三 ...	文件夹	
python.exe	2024/3/18 星期...	应用程序	100 KB
python311.pth	2024/3/18 星期...	.PTH 文件	1 KB
python311.dll	2024/3/18 星期...	应用程序扩展	5,624 KB
python311.zip	2024/3/18 星期...	WinRAR ZIP 压缩	4,168 KB

图 12.1-2 嵌入式 Python 解释器文件

12.2 案例实现

本案例使用 Python 脚本实现, 通过在案例脚本中导入 ATKComponent 模式接口模块 ATKComponentPythonModule, 调用接口模块中公用的类接口与函数接口, 完成场景与卫星类对象的新建与设置, 动画的仿真运行, 数据报告与想定文件的输出保存。最后通过嵌入式 Python 解释器解释运行该 Python 案例脚本, 完成案例实现。具体流程如下所示。

12.2.1. 编辑 Python 案例脚本, 流程图与源码如下:

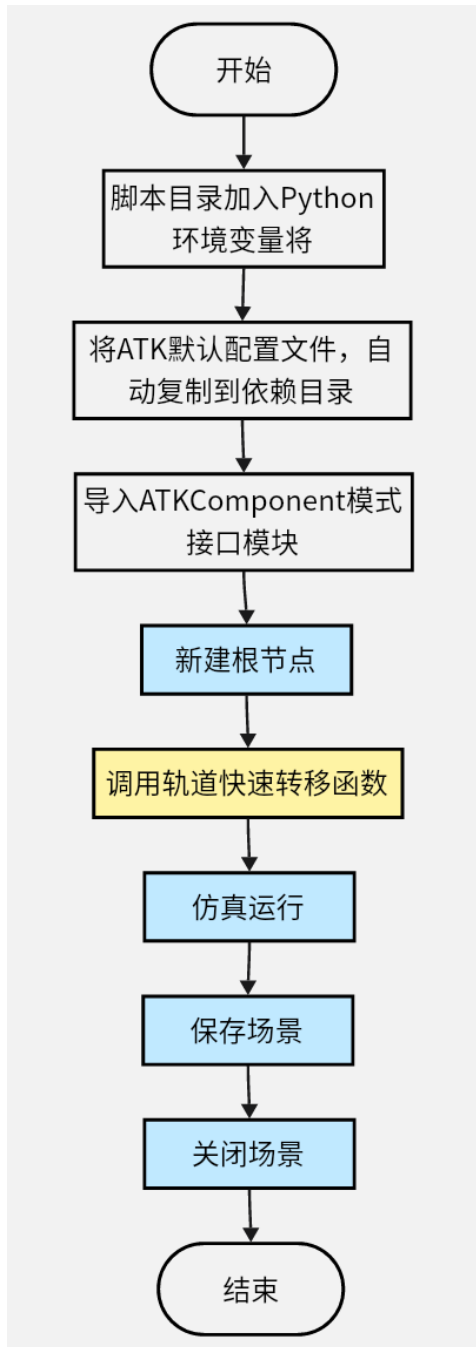


图 12.2-1 Python 案例流程图

#1, 使用内嵌式 python 解释器时, 需要将当前路径加入 Python 系统环境路径

```

import sys
import os
script_dir = os.path.dirname(os.path.abspath(__file__))
sys.path.insert(0, script_dir)
  
```

#2, 将 ATK 默认配置文件, 自动复制到依赖目录

```

import shutil
  
```

```

srcPath = script_dir + "..\..\..\AstroData"
destPath = script_dir + "\python-3.11.0-embed-amd64\AstroData"
if not (os.path.exists(destPath)):
    shutil.copytree(srcPath, destPath)

#3, 导入 ATKComponent 模式接口模块, 并取个简短的别名
import ATKComponentPythonModule as ATKModule

#4, 轨道快速转移实现函数
def TestFastTransfer(pIAtkObjectRoot):
    #4.1, 场景新建与属性设置
    pIAtkObject =
pIAtkObjectRoot.GetChildren().New(ATKModule.eScenario,
"FastTransfer");
    pIScenario = ATKModule.IAtkObject2IScenario(pIAtkObject);
    pIScenario.SetTimePeriod("5 Nov 2022 00:00:00.000", "6 Nov 2022
00:00:00.000");
    #4.2, 卫星新建与轨道预报设置为机动规划
    pIAtkObject = pIScenario.GetChildren().New(ATKModule.eSatellite,
"Satellit1");
    pISatellite = ATKModule.IAtkObject2ISatellite(pIAtkObject);
    pISatellite.SetPropagatorType(ATKModule.ePropagatorAstrogator);
    pIVADriverMCS =
ATKModule.IVePropagator2IVADriverMCS(pISatellite.GetPropagator());
    pIVAMCSSegmentCollection = pIVADriverMCS.GetMainSequence();
    #4.3, 机动规划添加段, 新添加卫星会有默认初始段
    if (ATKModule.eVASegmentTypeInitialState !=
pIVAMCSSegmentCollection.Item(0).Type):
        return;
    pIVAMCSInitialState =
ATKModule.IVAMCSSegment2IVAMCSInitialState(pIVAMCSSegmentCollection.I
tem(0));
    pIVAMCSSegment =
pIVAMCSSegmentCollection.Insert(ATKModule.eVASegmentTypePropagate,
"Propagate", "-");

```

```

    pIVAMCSPropagate =
ATKModule. IVAMCSSegment2IVAMCSPropagate (pIVAMCSSegment);
    pIVAMCSSegment =
pIVAMCSSegmentCollection. Insert (ATKModule. eVASegmentTypeTargetSequenc
e, "TargetSequence", "-");
    pIVAMCSTargetSequence =
ATKModule. IVAMCSSegment2IVAMCSTargetSequence (pIVAMCSSegment);
    pIVAMCSSegment =
pIVAMCSTargetSequence. GetSegments (). Insert (ATKModule. eVASegmentTypeMa
neuver, "Maneuver", "-");
    pIVAMCSManeuver =
ATKModule. IVAMCSSegment2IVAMCSManeuver (pIVAMCSSegment);
    pIVAMCSSegment =
pIVAMCSSegmentCollection. Insert (ATKModule. eVASegmentTypePropagate,
"Propagate", "-");
    pIVAMCSPropagate1 =
ATKModule. IVAMCSSegment2IVAMCSPropagate (pIVAMCSSegment);
    pIVAMCSSegment =
pIVAMCSSegmentCollection. Insert (ATKModule. eVASegmentTypeTargetSequenc
e, "TargetSequence1", "-");
    pIVAMCSTargetSequence1 =
ATKModule. IVAMCSSegment2IVAMCSTargetSequence (pIVAMCSSegment);
    pIVAMCSSegment =
pIVAMCSTargetSequence1. GetSegments (). Insert (ATKModule. eVASegmentTypeM
aneuver, "Maneuver", "-");
    pIVAMCSManeuver1 =
ATKModule. IVAMCSSegment2IVAMCSManeuver (pIVAMCSSegment);
    pIVAMCSSegment =
pIVAMCSSegmentCollection. Insert (ATKModule. eVASegmentTypePropagate,
"Propagate", "-");
    pIVAMCSPropagate2 =
ATKModule. IVAMCSSegment2IVAMCSPropagate (pIVAMCSSegment);
    #4. 4, 初始段属性设置
    pIVAMCSInitialState. SetOrbitEpoch ("5 Nov 2022 00:00:00.000");
    pIVAMCSInitialState. SetElementType (ATKModule. eVAElementTypeKepler

```

```

ian);
    pIVAElementKeplerian =
ATKModule. IVAElement2IVAElementKeplerian (pIVAMCSInitialState. GetElement());
    pIVAElementKeplerian. SetSemiMajorAxis (6700);
    pIVAElementKeplerian. SetEccentricity (0);
    pIVAElementKeplerian. SetInclination (0);
    pIVAElementKeplerian. SetRAAN (0);
    pIVAElementKeplerian. SetArgOfPeriapsis (0);
    pIVAElementKeplerian. SetTrueAnomaly (0);
#4.5, 第一个预报段属性设置
    pIVASToppingConditionElement =
pIVAMCSPropagate. GetStoppingConditions (). Add ("Duration");
    pIVASToppingCondition =
ATKModule. IVASToppingConditionComponent2IVASToppingCondition (pIVASToppingConditionElement. GetProperties ());
    pIVASToppingCondition. SetTrip (7200);
    pIVASToppingCondition. SetTolerance (0.0001);
#4.6, 第一个瞄准段中机动段属性设置
    pIVAMCSManeuver. SetManeuverType (ATKModule. eVAManeuverTypeImpulsive);
    pIVAManeuverImpulsive =
ATKModule. IVAManeuver2IVAManeuverImpulsive (pIVAMCSManeuver. GetManeuver());
    pIVAAAttitudeControlImpulsiveThrustVector =
ATKModule. IVAAttitudeControl2IVAAttitudeControlImpulsiveThrustVector (
pIVAManeuverImpulsive. GetAttitudeControl ());
    pIVAAAttitudeControlImpulsiveThrustVector. SetThrustAxesName ("Satellite VNC (Earth)");
    pIVAMCSManeuver. EnableControlParameter (ATKModule. eVAControlManeuverImpulsiveCartesianX);
    pIVAMCSManeuver. Results. Add ("Radius_Of_Apoapsis");
#4.7, 第一个瞄准段添加属性页
    pIVAProfileDifferentialCorrector1 =
ATKModule. IVAProfile2IVAProfileDifferentialCorrector1 (pIVAMCSTargetSe

```

```

quence.GetProfiles().Add("Differential Corrector"));
    pIVADCControl =
pIVAProfileDifferentialCorrector1.GetControlParameters().GetControlBy
Paths("Maneuver", "ImpulseX");
    pIVADCResult =
pIVAProfileDifferentialCorrector1.GetResults().GetResultByPaths("Mane
uver", "RadiusOfApoapsis");
#4.8, 属性页中控制变量属性设置
pIVADCControl.SetEnable(True);
pIVADCControl.SetMaxStep(100);
pIVADCControl.SetCorrection(2781.50365947627);
pIVADCControl.SetPerturbation(0.1);
pIVADCControl.SetScalingValue(1);
#4.9, 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);
#4.10, 第二个预报段属性设置
pIVASToppingConditionElement1 =
pIVAMCSPropagate1.GetStoppingConditions().Add("RMagnitude");
    pIVASToppingCondition1 =
ATKModule.IVASToppingConditionComponent2IVASToppingCondition(pIVASTop
pingConditionElement1.GetProperties());
    pIVASToppingCondition1.SetTrip(42164.197);
    pIVASToppingCondition1.SetTolerance(1e-6);
    pIVASToppingCondition1.SetRepeatCount(1);
    pIVASToppingCondition1.SetCriterion(ATKModule.eVACriterionCrossEi
ther);
#4.11, 第二个瞄准段中机动段属性设置
pIVAMCSManeuver1.SetManeuverType(ATKModule.eVAManeuverTypeImpulsi
ve);
    pIVAManeuverImpulsive1 =
ATKModule.IVAManeuver2IVAManeuverImpulsive(pIVAMCSManeuver1.GetManeu

```



```

er());
    pIVAAAttitudeControlImpulsiveThrustVector1 =
ATKModule. IVAAttitudeControl2IVAAttitudeControlImpulsiveThrustVector(
pIVAManeuverImpulsive1.GetAttitudeControl());
    pIVAAAttitudeControlImpulsiveThrustVector1.SetThrustAxesName("Satellite VNC(Earth)");
    pIVAMCSManeuver1.EnableControlParameter(ATKModule.eVAControlManeuverImpulsiveCartesianX);
    pIVAMCSManeuver1.EnableControlParameter(ATKModule.eVAControlManeuverImpulsiveCartesianZ);
    pIVAMCSManeuver1.Results.Add("Eccentricity");
    pIVAMCSManeuver1.Results.Add("Cosine_of_Vertical_FPA");
#4. 12, 第二个瞄准段添加属性页
    pIVAProfileDifferentialCorrector11 =
ATKModule. IVAProfile2IVAProfileDifferentialCorrector1(pIVAMCSTargetSequence1.GetProfiles().Add("Differential Corrector"));
#4. 13, 属性页中控制变量属性设置
    pIVADCControl1 =
pIVAProfileDifferentialCorrector11.GetControlParameters().Item(0);
    pIVADCControl1.SetEnable(True);
    pIVADCControl1.SetMaxStep(300);
    pIVADCControl1.SetCorrection(-1581.97670664023);
    pIVADCControl1.SetPerturbation(0.1);
    pIVADCControl1.SetScalingValue(1);
    pIVADCControl2 =
pIVAProfileDifferentialCorrector11.GetControlParameters().Item(1);
    pIVADCControl2.SetEnable(True);
    pIVADCControl2.SetMaxStep(300);
    pIVADCControl2.SetCorrection(-2771.82057041661);
    pIVADCControl2.SetPerturbation(0.1);
    pIVADCControl2.SetScalingValue(1);
#4. 14, 属性页中约束条件属性设置
    pIVADCCResult1 =
pIVAProfileDifferentialCorrector11.GetResults().Item(0);
    pIVADCCResult1.SetEnable(True);

```

```

    pIVADCResult1.SetDesiredValue(0);
    pIVADCResult1.SetScalingValue(1);
    pIVADCResult1.SetTolerance(0.1);
    pIVADCResult1.SetWeight(1);
    pIVADCResult2 =
pIVAProfileDifferentialCorrector11.GetResults().Item(1);
    pIVADCResult2.SetEnable(True);
    pIVADCResult2.SetDesiredValue(0);
    pIVADCResult2.SetScalingValue(1);
    pIVADCResult2.SetTolerance(0.1);
    pIVADCResult2.SetWeight(1);
#4.15, 第三个预报段属性设置
    pIVASToppingConditionElement2 =
pIVAMCSPropagate2.GetStoppingConditions().Add("Duration");
    pIVASToppingCondition2 =
ATKModule.IVASToppingConditionComponent2IVASToppingCondition(pIVASTop
pingConditionElement2.GetProperties());
    pIVASToppingCondition2.SetTrip(86400);
    pIVASToppingCondition2.SetTolerance(0.0001);
#4.16, 机动规划运行
    pIVADriverMCS.RunMCS();
    pIVADriverMCS.ApplyAllProfileChanges();
#4.17, 生成数据到文件
    strReportFilePath = pIAtkObjectRoot.OutputDataReport(pISatellite,
"J2000 位置速度", "5 Nov 2022 00:00:00.000", "6 Nov 2022
00:00:00.000");
    pIAtkObjectRoot.m_Animation.OutputPropCortsData(pISatellite);
#4.18, 输出动画数据
    pIAtkObjectRoot.m_Animation.OutputData();

#5, 新建根节点
pIAtkObjectRoot = ATKModule.IAtkObjectRoot();
#6, 轨道快速转移
TestFastTransfer(pIAtkObjectRoot);
#7, 仿真运行

```

```
pIAtkObjectRoot.m_Animation.PlayForward();
```

#8, 保存场景

```
pIAtkObjectRoot.SaveScenario();
```

#9, 关闭场景

```
pIAtkObjectRoot.CloseScenario();
```

12.2.2. 打开 cmd 命令行，切换到 Python 案例脚本目录。

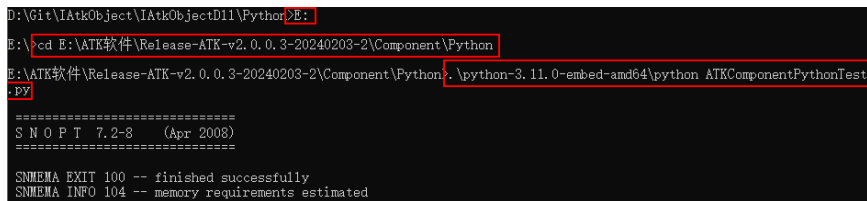
切换磁盘：E:

切换路径：cd E:\ATK 软件\Release-ATK-v2.0.0.3-20240203-2\Component\Python

12.2.3. 调用嵌入式 Python 解释器解释运行 Python 案例脚本。

运行命令：.\python-3.11.0-embed-amd64\python

ATKComponentPythonTest.py

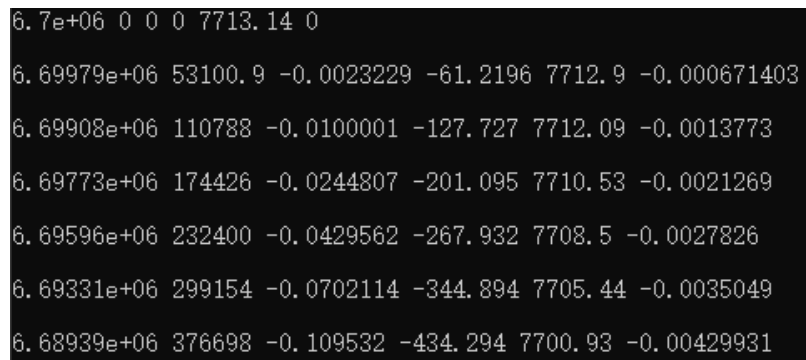


```
D:\Git\IAtkObject\IAtkObjectD11\Python>E:
E:\>cd E:\ATK 软件\Release-ATK-v2.0.0.3-20240203-2\Component\Python
E:\ATK 软件\Release-ATK-v2.0.0.3-20240203-2\Component\Python>.\python-3.11.0-embed-amd64\python ATKComponentPythonTest
.py
=====
SNOPIT 7.2-8 (Apr 2008)
=====
SNMEMA EXIT 100 -- finished successfully
SNMEMA INFO 104 -- memory requirements estimated
```

图 12.2-2 cmd 命令行

12.3 案例结果

成功运行 Python 案例脚本 ATKComponentPythonTest.py，会在 cmd 命令框中输出打印仿真动画数据，对应 Python 案例源码步骤 4.18，效果如下图所示。



```
6.7e+06 0 0 0 7713.14 0
6.69979e+06 53100.9 -0.0023229 -61.2196 7712.9 -0.000671403
6.69908e+06 110788 -0.0100001 -127.727 7712.09 -0.0013773
6.69773e+06 174426 -0.0244807 -201.095 7710.53 -0.0021269
6.69596e+06 232400 -0.0429562 -267.932 7708.5 -0.0027826
6.69331e+06 299154 -0.0702114 -344.894 7705.44 -0.0035049
6.68939e+06 376698 -0.109532 -434.294 7700.93 -0.00429931
```

图 12.3-1 cmd 命令框中输出打印的动画数据

同时生成的还有，卫星数据报告 J2000 位置速度.txt 和卫星机动规划数据修正报告 CorrectionsData.txt，对应案例源码步骤 4.17。以及轨道快速转移想定文件 FastTransfer.xml，对应案例源码步骤 8。以上文件均存放在输出目

录 Output，如下图所示。

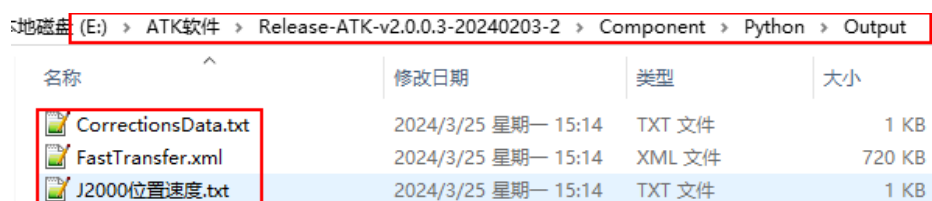


图 12.3-2 生成的想定与数据报告

打开 ATK 检查生成的想定文件，选择轨道快速转移想定文件 FastTransfer.xml，点击“开始”按钮运行动画，拉动下方时间轴查看二维视图下的轨迹效果，如下图所示。

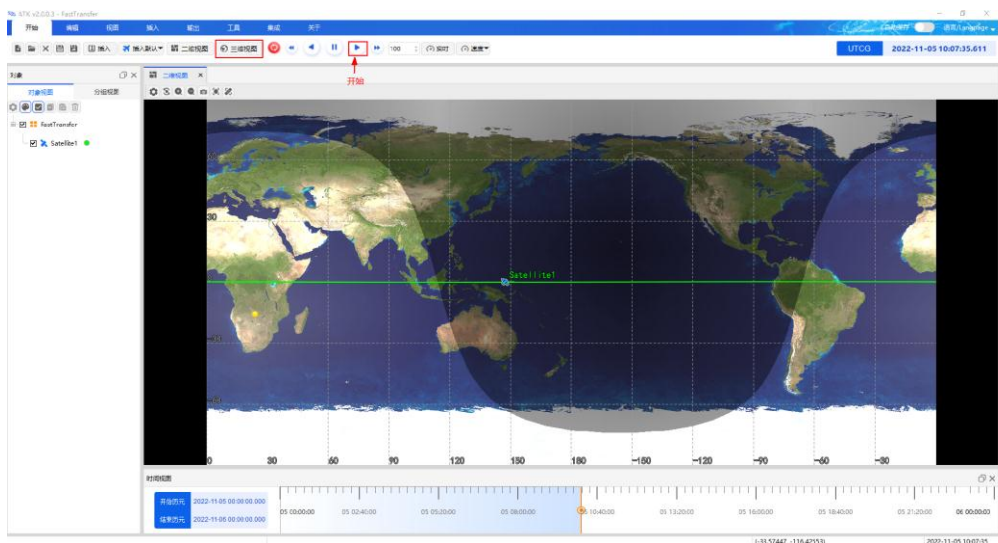


图 12.3-3 ATK 二维视图下的轨道快速转移效果图

点击 ATK “三维视图”按钮切换至三维视图，查看三维效果，如下图所示。

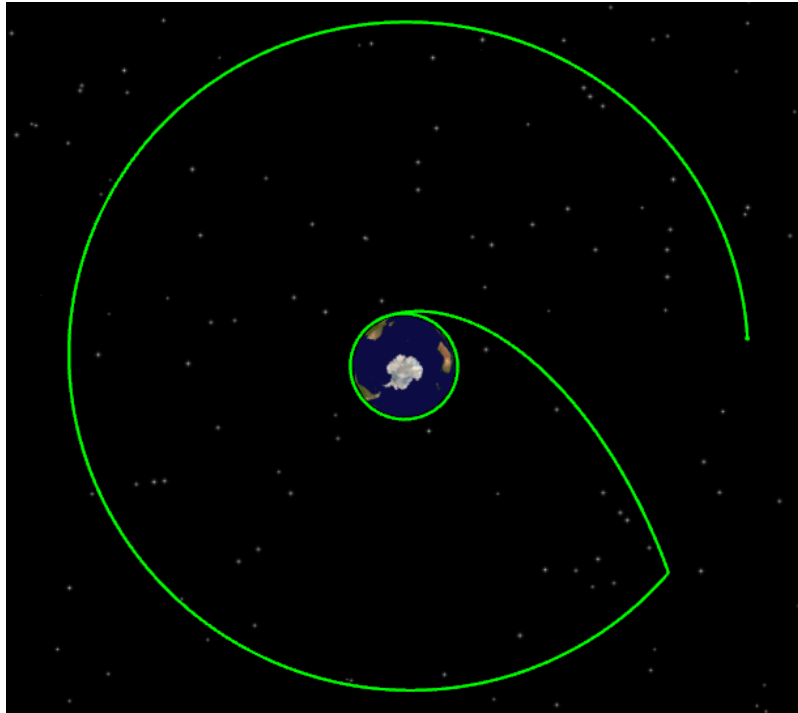


图 12.3-4 ATK 三维视图下的轨道快速转移效果图

参考文献

- [1] 张冉, 殷建丰, 韩潮. 航天器受迫绕飞构型设计与控制[J]. 北京航空航天大学学报, 2017, 第 43 卷(10): 2030-2039
- [2] 张赛, 杨震, 王华, 罗亚中. 航天器受控绕飞任务规划与自主通用软件实现[J]. 载人航天, 2024, 第 30 卷(1): 94-103
- [3] 贺波勇, 李恒年, 周庆瑞, 等. Walker 星座性能修复构型重构的弹性力学解法[J]. 西北工业大学学报, 2021, 39(2): 278-284.
- [4] 李骁. 灾害监测小卫星星座设计与仿真[D]. 中南大学, 2009.