

参赛队名称	天问		
参赛队单位	数学与统计学院		
参赛队员	马程远		
联系人	姓名：马程远	邮箱：1552650018@qq.com	电话：18770500186

中南大学第二届先进飞行器 虚拟仿真大赛



日期：2021 年 10 月 28 日

目录

1	设计理念的概述	3
2	作品设计的介绍	4
2.1	火星环境分析	4
2.1.1	火星重力场模型	4
2.1.2	火星大气密度模型	5
2.2	飞行器着陆轨迹建模	7
2.2.1	大气进入段轨迹动力学建模	8
2.2.2	伞降减速段轨迹动力学建模	10
2.2.3	发动机制动段动力学建模	11
2.2.4	EDL 过程仿真验证	12
3	轨迹设计	13
3.1	初始点位置估计	13
3.2	着陆点估计	14
3.3	时间最优控制方案设计	15
3.3.1	遗传算法求解轨迹	15
3.4	燃料最优控制策略求解	16
3.4.1	策略求解	17
3.4.2	十进制蚁群优化算法求解	17
3.5	误差控制分析	19
4	作品设计的创新点	20
5	论文/报告的引用和发表	21
6	附件 (视频或 3D 模型等)	22
6.1	程序附录	22
6.2	视频等资料附录	29

1 设计理念的概述

自古以来人类不断探索宇宙的奥妙，为解开宇宙和生命起源付出了一代代的努力。随着航天技术的不断进步，人类从探月工程迈向火星计划，不远的将来人类一定有能力走出太阳系，在更加浩瀚的宇宙中自由穿行。

深空探测具有非常重要的科学研究价值和现实意义，可以为地球资源枯竭提供有效解决途径，为揭开宇宙和生命起源提供科学依据，同时对发现新物质和科学研究具有不可替代的作用。火星探测是近年来深空探测领域的热点，而无论是当前的着陆探测火星阶段还是未来的采样返回和载人探测火星阶段，都需要火星探测器能够安全地着陆在预先选择的具有较高科学价值的预选点处，这就迫切地需要未来火星探测器的着陆系统具有较高的着陆精度。

由此，本作品以天问一号探测器着陆火星为背景，对火星着陆飞行器的着陆轨迹设计进行了分析研究。

首先建立了常用的坐标系和运动变量及转换关系，介绍了火星探测着陆器和降落伞的几何构型，在火星大气模型及相应的不确定性模型的基础上，对着陆巡视器飞行的力学环境进行了建模，包括火星引力、着陆器和降落伞在不同阶段所受到的不同的气动力模型、控制力模型等。

本作品对飞行器的轨迹优化考虑状态约束与路径约束，采用火星重力场模型推算给出大气层进入位置为时间指标的最优进入轨迹，并分析了进入角约束对进入轨迹的影响；飞行着陆器进入火星大气后制导受火星环境影响较大，本作品考虑火星大气不确定性，建立火星大气模型，通过火星卫星气动下降以及降落伞模拟仿真分析了制导律的稳定性和精度，对环境不确定性展开了分析，评估了开伞过程中峰值力和力矩对伞舱组合体动力学性能的影响。接着通过火星 EDL 过程的环境模型、动力学模型及制导控制方法，进行了火星 EDL 全过程一体化仿真，可以对进入、下降与着陆各阶段进行仿真验证。

对于着陆器飞行器轨迹的设计，本作品首先对飞行器分离轨道的初始点和着陆点进行了估计，然后采用控制着陆时间最优策略与燃料最优策略分别对飞行器轨迹进行设计。在时间最优控制策略上，本作品对之前所建立的着陆动力学模型使用了四阶 Runge-Kutta 法估计了控制变量范围。进一步依赖此结果使用遗传算法，通过 1000 次迭代变异概率为 0.01 的含 50 个个体的种群，解得发动机制动部分最优时间为 14.28 秒，而全过程用时在五分钟左右。在燃料最优控制策略上，通过十进制蚁群算法得到动力下降阶段燃料消耗为 $\Delta m = 168.33kg$ 。

最后，本作品进行了让控制方案最接近新闻报道的探索性研究。对误差来源、产生原因做了追溯，并基于此完成了对本作品模型的敏感性分析，探讨了令控制方案满足进一步要求的方法。

2 作品设计的介绍

2.1 火星环境分析

火星是太阳系中的第四颗行星。由于火星与地球的某些物理特性相似，人类对火星探测产生了浓厚兴趣。在长期的深空探测活动中，人们积累了大量关于火星的观测数据。这为本作品建立火星模型提供了帮助。



图 1: 火星环境示意图

2.1.1 火星重力场模型

火星的质量为 $6.578 \times 10^{23}\text{kg}$ ，是地球质量的大约 10%。[3] 火星的质量分布不均匀，由此导致重力场的异常。对于轨道飞行器而言，这种异常会造成扰动。

行星重力场的描述模型通常用球谐函数表示，*Journal of Geophysics Research* 上曾经发表过阶数为 12×12 的火星重力场模型。

为了减少计算量，在最初的分析中，本作品先假定火星质量分布较为均匀，故使用万有引力定律求解火星重力场。

由万有引力定律

$$F = \frac{GMm}{r^2} \quad (1)$$

得到引力关于飞行器与火星距离的函数

$$F(h) = \frac{6.67 \times 10^{-11}\text{N} \cdot \text{m}^2/\text{kg}^2 \times 6.578 \times 10^{23}\text{kg} \times m}{(h + R)^2} \quad (2)$$

通过这个函数，就能够对飞行着陆器 EDL 过程中各高度所受引力大小进行表示。

2.1.2 火星大气密度模型

火星大气密度是火星 EDL 过程的重要环境参数，是火星着陆系统设计与着陆过程仿真的主要大气参数，尤其在进入段，大气密度会直接影响气动力。

火星大气模型包括两类：火星大气数据库和火星大气简化模型。火星大气数据库根据火星大气环流模型和深空探测任务数据构建而成，精确性较高，但计算成本较大。火星大气简化模型是基于火星大气数据库的数据拟合后得到的指数模型，使用简单，计算难度小，适合实时仿真。[5]

由于后续要求解的优化问题动态性强，故本作品使用火星大气简化模型进行计算。

日本大学的 Ushijima 在 2010 年基于 Mars - GRAM 拟合的模型较好，其表达式为

$$T = \begin{cases} 241.0 - 0.999(h/1000), & h < 7000\text{m} \\ 249.5 - 2.22(h/1000), & h \geq 7000\text{m} \end{cases} \quad (3)$$

$$P = 700e^{-0.09(h/1000)}, \rho = P/188.95110711075T \quad (4)$$

这一模型对于两万米以下的大气情况拟合较好，但对于本作品问题的研究范围是不够的。通过 MCD 数据库可以获取天问一号着陆当天的大气数据。

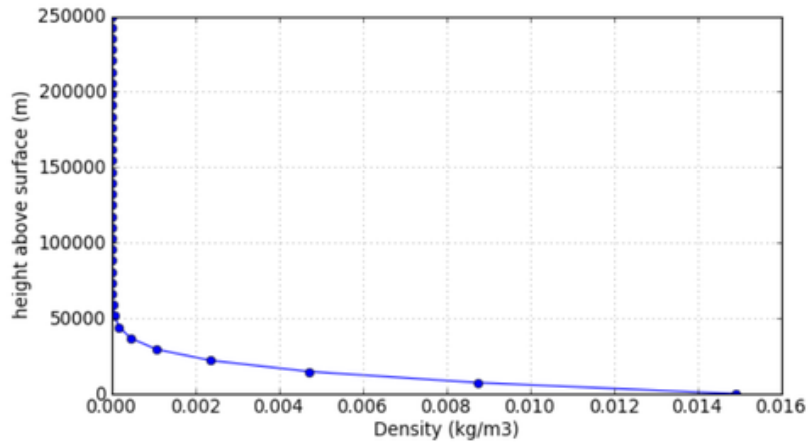


图 2: 大气密度数据

利用 Python 进行多项式拟合，就能得到大气密度的拟合。

其中，三次拟合效果较优。其拟合式为

$$\rho = -1.58370399 \times 10^{-16} h^3 + 2.08948544 \times 10^{-11} h^2 - 9.38428798 \times 10^{-7} h + 1.45762615 \times 10^{-2} \quad (5)$$

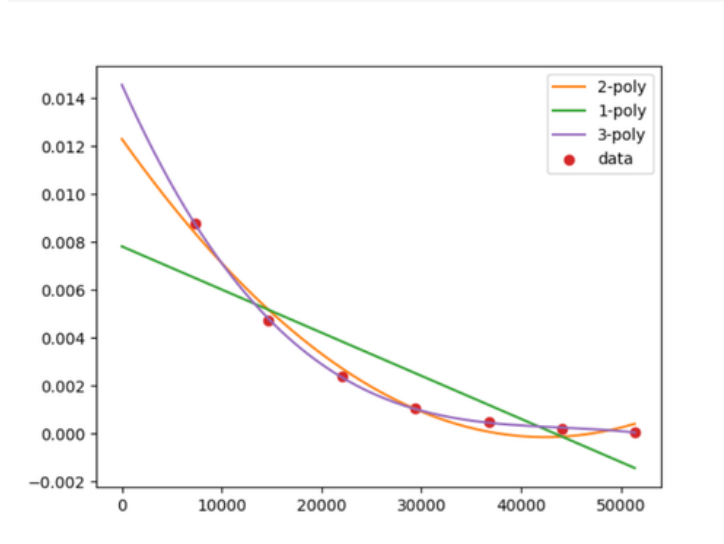


图 3: 大气密度多项式拟合

考虑到火星简化大气模型一般都使用指数模型进行模型，为了形式上的统一，本作品还使用 Excel 获得了指数拟合曲线。

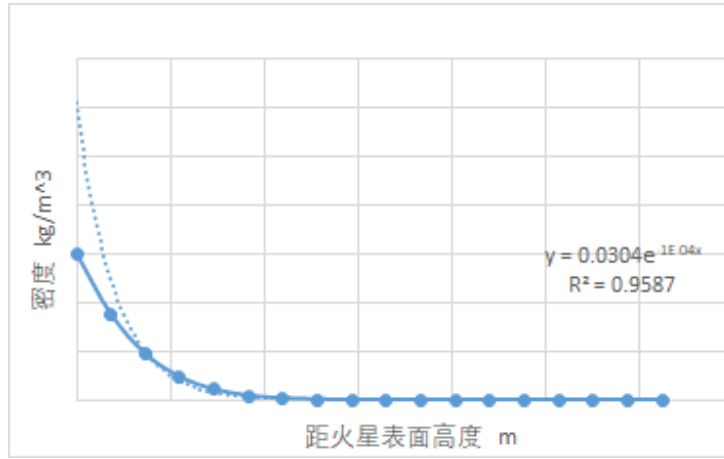


图 4: 大气密度指数拟合

$$\rho = 0.0304e^{-10^{-4}y}, 20000\text{m} \leq h < 125\text{km} \quad (6)$$

拟合结果 R^2 为 0.9587，适合仿真。

综上所述，火星大气密度场模型可以描述为

$$\rho = \begin{cases} 700e^{-0.09(h/1000)} / \{188.95[241.0 - 0.999(h/1000)]\}, h < 7000\text{m} \\ 700e^{-0.09(h/1000)} / \{188.95[249.5 - 2.22(h/1000)]\}, 7000\text{m} \leq h < 20000\text{m} \\ 0.0304e^{-10^{-4}h}, 20000\text{m} \leq h < 125\text{km} \end{cases} \quad (7)$$

2.2 飞行器着陆轨迹建模

建立飞行器着陆火星的轨迹模型需要考虑飞行器的位置、运动状态、受力情况。本作品使用的大量数据来源于天问一号工程叶培建院士的公开讲座内容。

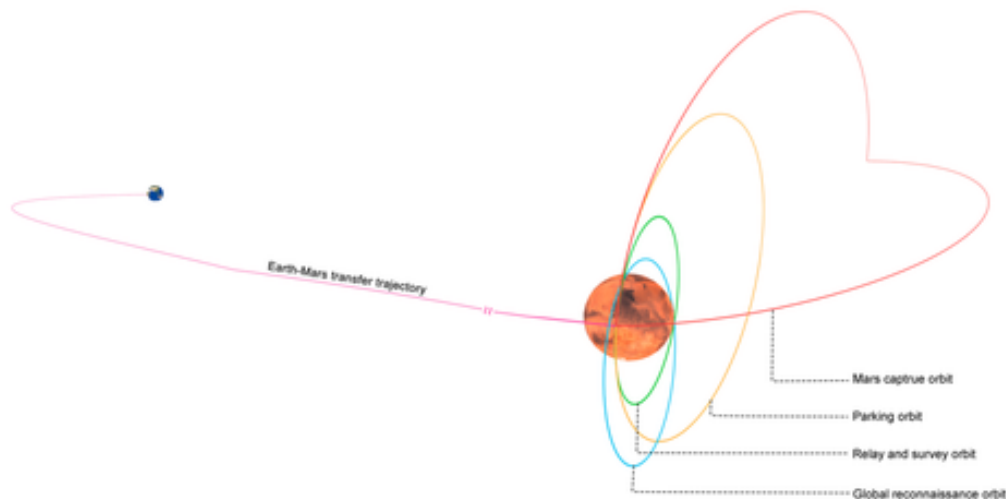


图 5: 飞行器飞行轨道示意图

由于飞行着陆器 EDL 过程是在火星的附近，相距地球、太阳和其他星球甚远，所以可以在求解中忽略除火星外其他天体对着陆器的作用。

天文观测中使用的火心固定坐标系是非惯性的，包含了公转、自转。在 EDL 过程中，由于全过程时间不超过十分钟，所以可以不考虑公转及自转。

进一步地，假定着陆轨道是从近火点出发。为了便于求解，先考虑简化坐标。本作品假定飞行着陆器进入大气位置、着陆位置、火星形心三点均在 EDL 过程的轨迹平面上。即利用上述三点就可以刻画出 EDL 过程平面。

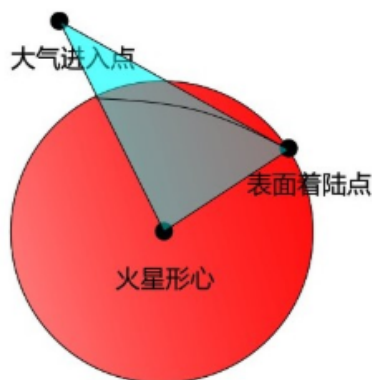


图 6: 共面假设示意图

以飞行器指向火星形心方向作为竖直方向，就可以将运动做正交分解。EDL 过程中各个阶段的切换主要由竖直方向的位置控制，这样的处理极大地方便了建模和求解。

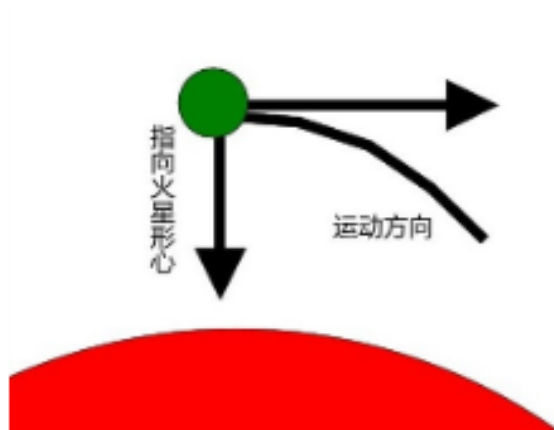


图 7: 坐标示意图

2.2.1 大气进入段轨迹动力学建模

在这一阶段，飞行器受到火星的引力和火星大气对飞行器的阻力。其中，引力方向竖直向下，即简化坐标中的指向火星形心的竖直方向；大气阻力方向与当前运动方向相反。

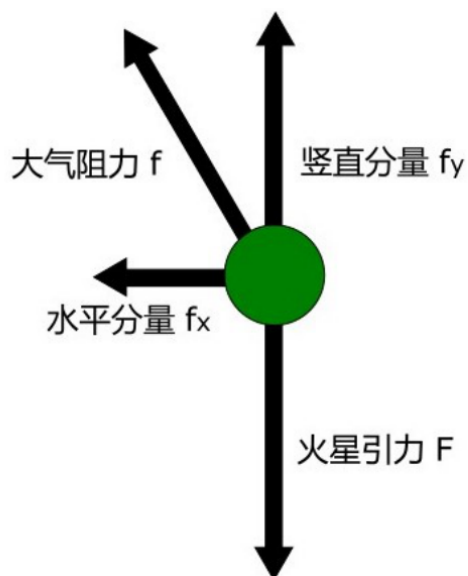


图 8: 进入段受力分析示意图

在相关研究中，大气阻力可用飞行器表面积、速度大小、大气密度以及阻力系数描述。

$$f = \frac{1}{2} \rho v^2 S \cdot C_D \quad (8)$$

于是可知，大气进入段飞行着陆器的运动状态有以下条件

$$\begin{aligned}
f &= \frac{1}{2}\rho(v_x^2 + v_y^2)S \cdot C_D \\
\beta &= \arctan \frac{v_x}{v_y} \\
f_x &= \sin \beta \cdot f \\
f_y &= \cos \beta \cdot f \\
F &= \frac{GMm}{(h+R)^2}
\end{aligned} \tag{9}$$

其中, ρ 为大气密度, S 为飞行器表面积, C_D 为阻力系数, β 为运动方向角。进一步地可以得到竖直与水平两个方向上的加速度大小

$$\begin{aligned}
a_x &= \frac{f_x}{m} \\
a_y &= \frac{f_y - mg}{m}
\end{aligned} \tag{10}$$

根据飞行器的运行轨道可以得到进入大气时的初始状态

$$\begin{aligned}
v_{x_0} &= 4694.61\text{m/s}, v_{y_0} = 956\text{m/s} \\
x_0 &= 0\text{m}, y_0 = 125\text{km} \\
a_{x_0} &= a_{y_0} = 0\text{m/s}^2
\end{aligned} \tag{11}$$

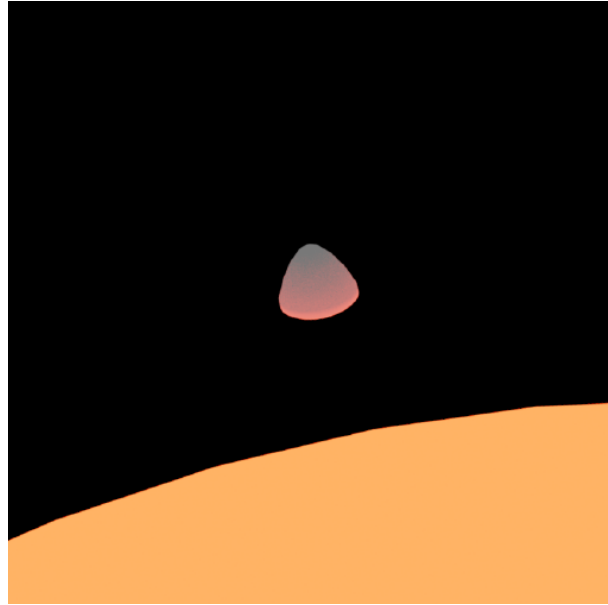


图 9: 大气进入段示意图

从进入火星开始到降落伞开伞完成, 探测器的速度从每秒几千米迅速减小到每秒几百米, 这个过程主要是靠探测器的气动外形减速的。而由于火星大气稀薄, 与在地球着陆相比, 相同的有效载荷质量需要更大的直径外形结构, 体积和质量也会相应的增加;

此外，火星大气成分主要为二氧化碳。在进入过程中，探测器穿过火星大气，会产生大量的摩擦热，这就对探测器外部的防热材料提出了更高的要求。

进入火星的气动减速和热防护外形都是由探测器的弹道系数 β 决定的，研究得到， β 越小，则经气动减速后探测器所能达到的稳降速度越小。而我们有飞行器的阻力系数：

$$C_D = \frac{M}{\beta A} \quad (12)$$

式中， M 为探测器质量， C_D 表示阻力系数， A 为有效面积。为充分发挥探测器的气动减速作用，使稳降速度尽可能减小，即令 β 减小，在探测器质量一定的情况下，就要使阻力系数和有效面积增大。因而，一般采用半锥角较大（头部较钝）的球锥形，一方面阻力系数随着半锥角的增大而增大，另一方面头部钝度越大则气动摩擦热越小。

目前的几种火星飞行探测器的气动外形参数如下图：






气动外形					
直径 /m	3.5	2.65	2.65	2.65	4.6
质量 /kg	990	584	830	600	2800
弹道系数 / ($kg \cdot m^{-2}$)	64	63	94	70	115

图 10: 气动外形参数图

依据天问一号总工程师叶培建院士所做的公开讲座，可以得到天问一号着陆飞行器的一些基本数据。

$$m = 1285kg, S = 12.56m^2, \quad (13)$$

从而根据此数据大致可以得到着陆飞行器的气动外形形状以及阻力系数大小。

这样，天问一号在整个大气进入段的运动状态都是可知的。

2.2.2 伞降减速段轨迹动力学建模

伞降减速阶段的受力情况与大气进入段大体相似。

大气阻力都可以用式 (8) 来确定。不同之处在于，此时还需要考虑降落伞提供的阻力。降落伞的表面积经估算为 $200m^2$ 。阻力系数参考既往火星探测任务：如下表 1：

表 1: 降落伞阻力系数

降落伞类型	“海盗号”降落伞	“火星探路者”降落伞	“火星探测漫游者”降落伞
阻力系数	0.67	0.41	0.43

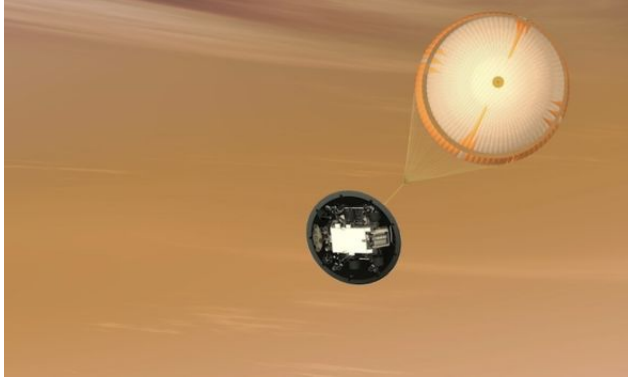


图 11: 伞降控制示意图

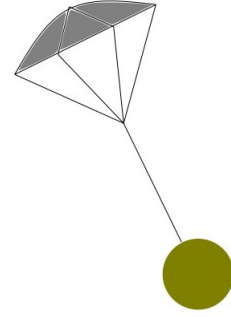


图 12: 伞降减速模型

从而本作品中降落伞阻力系数取 0.43。
而开伞条件为

$$Mach = 2, h > 4km \quad (14)$$

2.2.3 发动机制动段动力学建模

在 EDL 过程中，动力下降过程是最后一个阶段，通过主发动机和姿控发动机的控制，使着陆器安全软着陆。这一阶段主要依靠发动机推力进行减速。当着陆器海拔在 1.5km 时，就进入到发动机制动阶段。

记发动机在水平方向和竖直方向的推力分量分别为 $u_x(t)$ 和 $u_y(t)$ 。着陆器在 t 时刻质量为 $m(t)$ 。可以构建如下运动学方程。

$$\begin{aligned} m(t) \frac{d^2x}{dt^2} &= u_x(t) \\ m(t) \frac{d^2y}{dt^2} &= \frac{GMm(t)}{(R + y(t))^2} + u_y(t) \end{aligned} \quad (15)$$

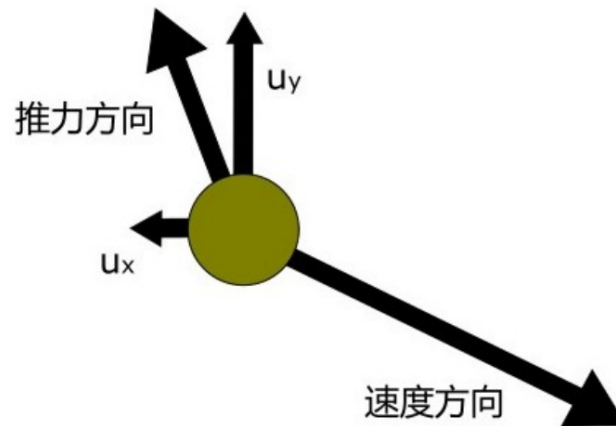


图 13: 发动机制动段示意图

另外，由发动机比冲的定义可以知道

$$\frac{dm}{dt} = \frac{\sqrt{u_x(t)^2 + u_y(t)^2}}{v(t)} \quad (16)$$

式中右端分子部分值为 7500N。

2.2.4 EDL 过程仿真验证

求解优化问题之前，有必要对已建立的 EDL 过程模型进行编程验证。

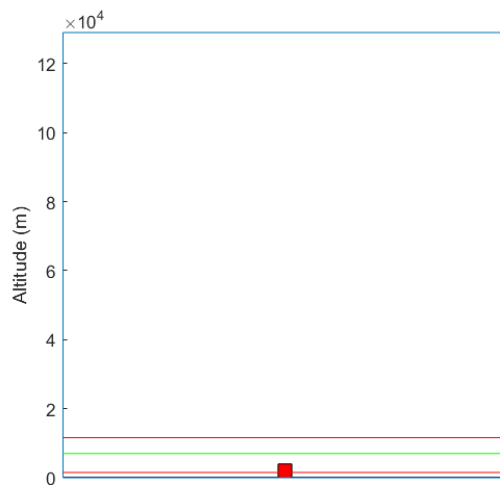


图 14: 仿真动画示意图

仿真使用 Matlab 完成，将火星飞行器降落过程竖直方向以二维动画方式形成。仿真过程中使用竖直方向位移作为位置参数，并将水平方向位移作为结果输出。

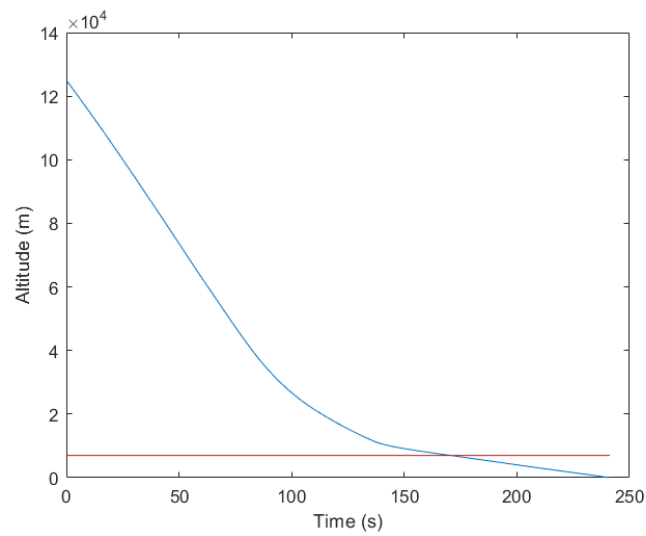


图 15: 海拔变化

3 轨迹设计

3.1 初始点位置估计

如图 16 所示的火星表面无旋转的固连坐标系作为参考坐标系。该坐标系的原点位于目标着陆点，三轴方向分别为横向、侧向及高度方向。

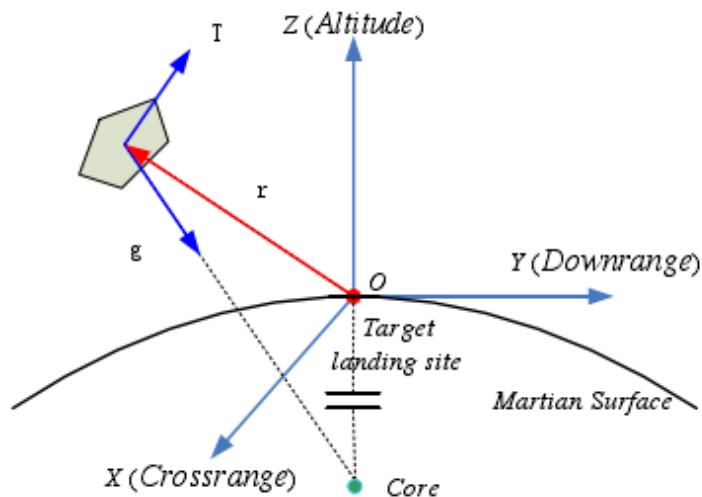


图 16: 参考坐标系示意图

由于天问一号着陆器与环绕器分离点位置坐标并未公布。还不能够求解优化问题。根据前期仿真验证结果，可以得到着陆器在水平方向上的位移。

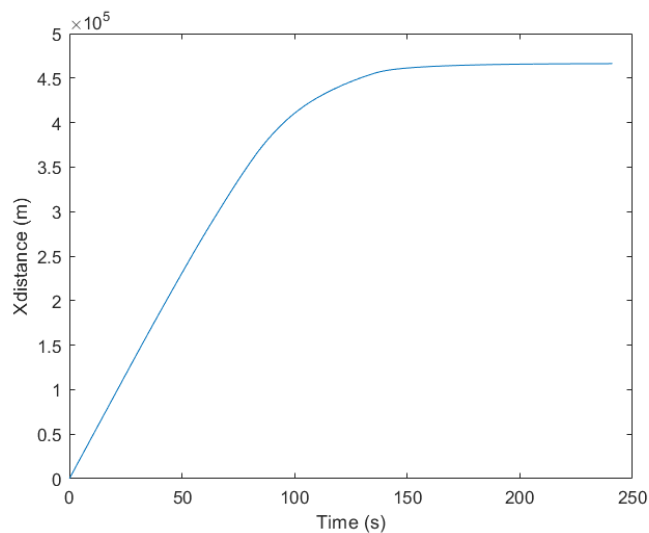


图 17: 水平位移

火星飞行器的着陆准备轨道与火星赤道平面倾角为 86.9° ，环绕方向与火星自转方向一致。所以知道分离点在着陆点西南方，且可以推知分离点经度近似于着陆点。

在探测任务前期规划中，轨道平面倾角被设定为 $90 \pm 5^\circ$ 。不妨假定轨道平面垂直于赤道平面。于是着陆器在水平方向位移就反映了着陆器 EDL 过程中的纬度变化。

由火星半径可以知道，火星上纬度差为 1° 的两点距离为

$$\frac{\pi R}{180} = 59166.66\text{m} \quad (17)$$

根据累积位移可以求得纬度差为 28.7° 。于是分离点经纬度坐标为 $(3.6^\circ S, 109.9^\circ E)$ 。

3.2 着陆点估计

火星表面遍布斜坡、高山、岩石、巨大的火山坑、沟壑等障碍物，地形非常复杂。因而在着陆时，需要精确合理的选择着陆位置，这就要求探测器具有实时对着陆地区障碍地形和着陆点进行检测、评估及轨迹规划、制导的能力。此外，由于火星表面温度分布不均，大气运动十分剧烈，因而常常有尘暴现象出现，其平均风速可达 50m/s 。这就对减速装置及着陆稳定性的设计有较高要求。

对于火星飞行器着陆点的估计，本作品认为需要从这几方面考虑：

第一，着陆区的地形要平坦，这就决定了着陆点很可能位于火星的北半球，在火星北半球有着较大面积的平原，北半球的整体海拔高度相对南半球都要低一些；

第二，着陆区需要有科学考察的意义，从而火星上有水、海洋的地方需要被优先考虑，美国宇航局发射了 4 代火星车，全部都冲着火星上可能有生命的地方，比如好奇号降落在盖尔撞击坑，坑中有一座 5000 多米海拔的夏普山，这座山被认为曾经被海洋覆盖到半山腰的位置，有可能存在生命；勇气号和机遇号降落在古塞夫撞击坑和子午线平原，子午线平原上有灰色的结晶赤铁矿，在地球上，这种矿物形成于温泉或者静水湖泊中，子午线平原上的赤铁矿厚度可达到数百米之巨。古塞夫撞击坑被认为是古老的湖床，还有火山喷发的痕迹，这说明这里是一处火山湖，研究价值较高。

结合火星地图 (如图 18) 以及天问一号的着陆报道，最终我们选定飞行器在乌托邦平原着陆，经纬度坐标为 $(25.1^\circ N, 109.9^\circ E)$ 。

NASA 中火星上功能强大的 HiRISE 摄像机的首席研究员 Alfred McEwen 表示：该地点“大多平坦而光滑，但有火山口，风成的风脊和一些巨石。”该地区被一些科学家解释为被泥石流覆盖，因此可能存在古老的深层地下水，这可能是一个与流浪者一起研究的有趣地点。乌托邦平原高度低，这意味着进入太空飞船将有更多的时间和气氛减速并安全地下降到地面，而且这里是太阳系中最大的撞击盆地，也具备探索价值。[8]

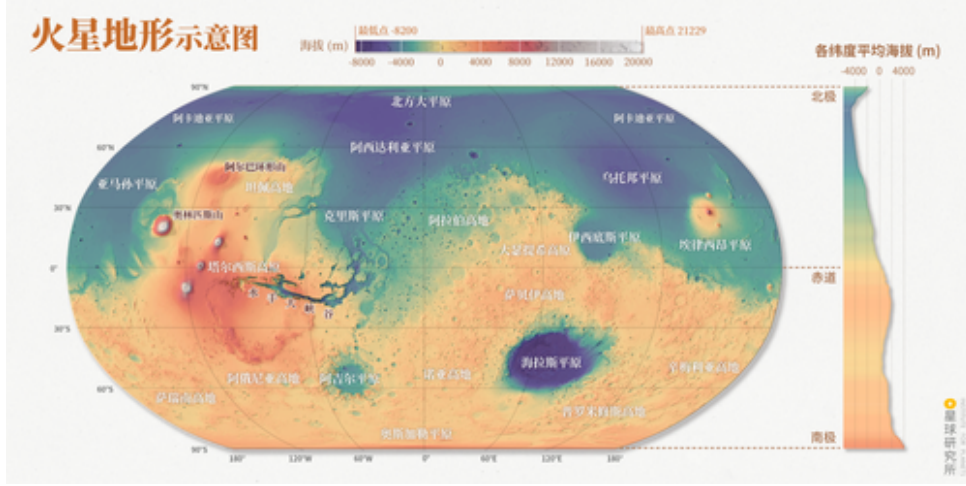


图 18: 火星地形示意图

3.3 时间最优控制方案设计

3.3.1 遗传算法求解轨迹

注意到，EDL 过程中可以通过预设方案进行控制的部分只有发动机制动段。所以只需求解发动机制动段的最优方案。

不妨假定发动机始终以最大推力运作，在制动过程中着陆器仅受到发动机推力和火星引力。于是有

$$\begin{cases} m(t) \frac{d^2 x}{dt^2} = u_x(t) \\ m(t) \frac{d^2 y}{dt^2} = \frac{GMm(t)}{(R + y(t))^2} + u_y(t) \\ \frac{dm}{dt} = \frac{7500}{v_e} \end{cases} \quad (18)$$

而约束条件有

$$\begin{cases} \sqrt{u_x^2 + u_y^2} = 7500 \\ m(0) = 1285 \\ x(0) = 0, y(0) = 1192.56 \\ \frac{dx}{dt} \Big|_{t=0} = 2.2 \\ \frac{dy}{dt} \Big|_{t=0} = 95.1 \\ y(t_f) = -100 \\ \sqrt{\left(\frac{dx}{dt} \Big|_{t=t_f}\right)^2 + \left(\frac{dy}{dt} \Big|_{t=t_f}\right)^2} = 3.6 \end{cases} \quad (19)$$

在本策略中，目标泛函为 $J = t_f$ 。显然，这是一个 Mayer 型泛函。[12] 状态变量为着陆器的位置参数、运动状态等，控制变量是水平和竖直两个方向上的推力分量。

如果记着陆器方向角为 θ ，那么可以得到控制变量的更精确形式

$$\begin{cases} u = a_1 t^3 + a_2 t^2 + a_3 t + a_4 \\ \theta = b_1 t^3 + b_2 t^2 + b_3 t + b_4 \end{cases} \quad (20)$$

可以将 u_x 及 u_y 表成 $u \cos \theta$ 及 $u \sin \theta$ 。根据该阶段所处的状态，大体估计出该阶段所需要的时间小于 90 秒。因此，将时间段在 $[0, 90]$ 内进行离散化，采用四阶 Runge-Kutta 方法进行参数范围的估计，见表 2。

参数	下界	上界
a_1	-10	10
a_2	-10	10
a_3	-100	100
a_4	1500	10000
b_1	-10	10
b_2	-10	10
b_3	-10	10
b_4	-100	100

表 2: 参数估计

在此基础上，利用遗传算法求解时间最优的控制方案。优化变量个数为 8，每代种群数量为 50，迭代 1000 次，变异概率为 0.01。解得最优时间为 14.28 秒，参数结果见表 3。

变量
$u = -7.1t^3 + 90.3t^2 + 89.4t + 5134$
$\theta = 9.4t^3 - 3.6t^2 + 2.9t - 24.4$

表 3: 优化变量

3.4 燃料最优控制策略求解

动力下降阶段，发动机主要任务是抵消着陆器的初始动能和势能。在实际制动中，为了实现精准降落的目的，发动机的推力方向会随时间变化，不同时刻的推力大小也不同，但仍处于发动机推力的变化范围中。[13] 除此之外，随着燃料的消耗，着陆器本身的质量也会随之降低。

由以上特点，我们建立如下的优化模型：

$$\begin{aligned} \min \quad & J(u, \theta) = \int_0^{t_f} \frac{u}{v_e} dt \\ \text{subject to} \quad & (18 - 19) \end{aligned} \quad (21)$$

3.4.1 策略求解

显然，目标泛函是一个 Bolza 型泛函 [12]。由于控制变量 $\theta(t)$ 和 $u(t)$ 的搜索空间是一个泛函空间，无法直接应用优化算法，因此首先我们将这个轨迹优化问题转化为参数优化问题。即用多项式来表示控制变量，如 (20)。

这样轨迹优化问题就转化为对多项式系数的优化。但是，这些参数并没有明确的物理意义，确定初值及搜索空间比较困难。因此我们采用离散化的方法对优化问题进行改进。

首先将火星着陆器动力下降轨迹离散化分割成 n 个小段，每段的节点设定一个推力大小和方向角，那么可以将 $n+1$ 个节点的推力大小和方向角作为待优化的参数。每个节点的时刻可以由下式得到：

$$t_i = t_0 + i(t_f - t_0)/n \quad (22)$$

这样，就使得每个节点的推力大小和方向角都有一个对应的节点时刻。那么利用这 $n+1$ 个节点的推力大小和方向角及对应时刻，对式 (20) 进行拟合，可以求得多项式的各项系数，进而得到整个动力下降轨迹的推力大小和方向角的曲线。

3.4.2 十进制蚁群优化算法求解

对燃料的优化问题本质上是一个二元函数连续优化的问题，上述数学模型可以简化为：

$$\begin{aligned} \min \quad & J(u, \theta) \\ \text{subject to} \quad & u \in [u_0, u_1], \theta \in [\theta_0, \theta_1] \end{aligned} \quad (23)$$

先通过变换将优化参数的变化范围转化为 $[0,1]$ 。设转换后的优化参数 x 要求精确到小数点后 d 位，则该优化参数值可以用 $1+d$ 个十进制数字来构成

这里构造 $1 + d \times 10$ 个城市如图 19 所示其结构为 $1 + d$ 列第一列代表个位上的数这一列只有一个城市代表个位上只能选的数字 0。其余 d 列从左至右分别代表小数点后十分位、百分位、千分位等每列包括 10 个城市分别代表每个数位上可选的 0 到 9 这 10 个十进制数字。

蚂蚁在这样的城市群上选择路径时根据一定的路径转移规则从左向右走每列上必须且只能选择一个城市。这样蚂蚁从左到右完成一个循环以后走过的路径包含有 $1 + d$

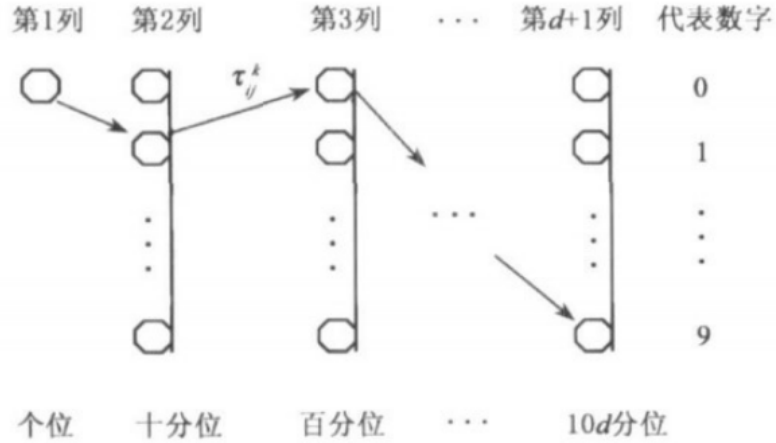


图 19: 城市群构造

个城市它们所代表的数字可以构成一个在 $[0, 1]$ 上有 d 位有效数字的小数。经过路径解码后，成为优化参数的一个可行解。

所有蚂蚁完成一次循环后要根据信息素更新规则更新相关路径上的信息素量。使蚂蚁经过多次循环之后，逐渐向最优路径聚集，也就是优化参数逐渐向最优解收敛。

用参数化方法的相关参数设置为：设将轨迹离散化为 9 段，那么待优化参数共 20 个参数即 10 个推力方向角和 10 个推力大小。在用蚁群算法进行优化过程中需要确定这 20 个优化参数的搜索范围。对于 10 个方向角，由经验可知，推力方向与水平方向夹角不会超过 180 度，否则着陆器将会被加速，燃料消耗将更多。

综上，本算例中算法参数应设置为：

num-ant	num-clc	K	Q	ρ	τ_0
1000	50	5	0.5	0.2	5

表 4: 蚁群参数

按照前述方法用十进制蚁群算法对火星软着陆轨迹进行优化。由于十进制蚁群算法是基于概率转移的仿生算法，优化结果具有一定的随机性，所以这里进行了 10 次仿真，结果全部收敛，证明在本方案中，十进制蚁群算法具有良好的收敛性。

由十进制蚁群算法得到的优化参数见表 5，由于参数过多，故不一一列出。

u_1	u_2	\cdots	u_{10}	θ_1	θ_2	\cdots	θ_{10}
3280	4493	\cdots	2105	32.86	45.72	\cdots	88.08

表 5: 蚁群参数

将以上 20 个优化参数投入到参数化方法中的多项式中，我们可以得到发动机推力

大小和方向角随时间变化曲线和参数。见图 20 及表 6。

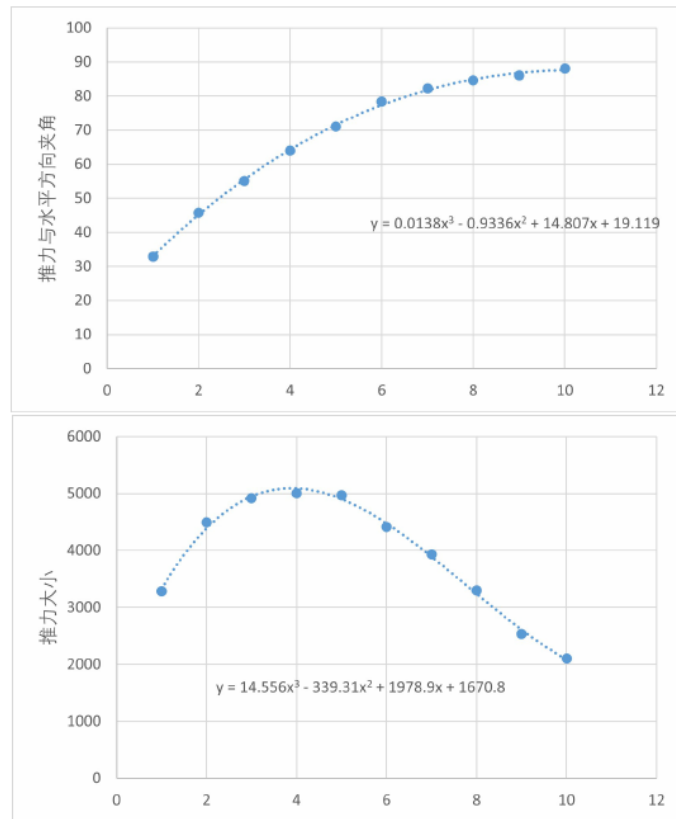


图 20: 推力方向角及大小变化

变量
$u = 14.556t^3 - 339.31t^2 + 197.9t + 5134$
$\theta = 0.0138t^3 - 0.9336t^2 + 14.807t + 1670.8$

表 6: 优化变量

将上述结果代入动力学模型中,我们最终可以得到,动力下降阶段燃料消耗为 $\Delta m = 168.33\text{kg}$ 。

3.5 误差控制分析

飞行器在实际着陆的控制过程中,往往客观存在一定的控制误差,譬如着陆准备轨道参数(近月点位置和速度)的误差、发动机推力(大小和方向)的控制误差、模型的简化假设和近似求解误差等。诸如这些误差势必会对实际的轨道设计和控制结果造成或多或少的影响,从工程应用的角度需要做相应的敏感性分析。而且涉及到的相关参数,如坐标系的选取、变量的选择、参数的选取、约束条件、函数的简化和月面的观测等都存

在一定的偏差，那么对轨道设计和控制策略的影响程度如何，也需要就某些情况做相应的敏感度分析。

这里仅就主减速发动机推力变化的敏感性进行分析。

主发动机最大推力的误差影响对于主减速阶段，主发动机要求以 7500N 最大推力运行，但事实上发动机的推力和实际推力会有一些误差。

主发动机推力方向角近似控制多项式系数误差的影响主发动机推力方向角的控制函数是用三次多项式来实现的，其系数是根据各阶段的状态要求通过数值方法搜索得到的，势必也存在一定的误差。

飞行器的运行状态对方向角的控制多项式系数的改变是非常敏感的，尤其是对运行轨迹和高度的影响较大，但对速度无影响。因此，在实际中需要提高该参数的传输和控制的精度，才能保证实际需求。

对于其他参数的敏感性分析，方法是类似的，这里不再赘述。

4 作品设计的创新点

本作品提出的解决方案对飞行器轨迹设计是较为适恰的。经过前期大量地对数据进行收集和处理，我们得到了解决本问题所必需的各项数据。在动力学建模阶段，我们通过对问题适当的简化并基于物理学分析，较为准确地建立了飞行着陆器 EDL 过程动力学模型。

求解时间最优控制策略方案时，由于已知的路径点过少，本作品选择了智能优化算法求解。这类算法不能保证所得结果是全局最优的。事实也是如此，天问一号 EDL 过程中与地球无通信，其状态无从获知。该问题的解即便最优，也未必就符合实际情况。回到方法的讨论上，在本问题中遗传算法的收敛性较好，在 1000 次迭代中就得到了结果。

在考虑燃料最优控制策略方案的求解时，本作品采用了对整个动力下降过程进行一定程度上的离散化，再利用十进制蚁群算法对 20 个优化参数基于概率转移进行求解，在进行多次仿真后，结果全部收敛，说明在本方案中，十进制蚁群算法具有良好的收敛性。由此，我们可以考虑将十进制蚁群算法推广到其他优化模型中，解决了蚁群算法很难对连续目标函数进行优化的问题，且该方法具有良好的收敛性，值得在其他问题中尝试、应用。

在本次求解过程中，我们发现燃料最优模型结果对轨迹离散化处理程度有一定的要求，仅将动力下降过程离散为 10 个过程，很难使模型具有较高的精度，因此会出现一定的误差。但如果要提高精度就会带来难以预测的计算量。这表明本作品的算法仍然有改进的空间。

5 论文/报告的引用和发表

- [1] 新华社. 五大看点! 这份中国首次自主火星探测任务“观赏指南”请收好.[EB/OL].http://www.xinhuanet.com/2020-07/23/c_1126275770.htm, 2021-08-07
- [2] 新华社. 习近平代表党中央、国务院和中央军委祝贺我国首次火星探测任务天问一号探测器成功着陆火星的贺电.[EB/OL].http://www.xinhuanet.com/2021-05/15/c_1127448736.htm, 2021-08-07
- [3] 欧阳自远, 肖福根. 火星及其环境 [J]. 航天器环境工程, 2012, 29(06): 591-601.
- [4] Christensen E J, Balmino G. Development and analysis of a twelfth degree and order gravity model for Mars [J]. Journal of Geophysics Research, 1979, 84: 7943-7945
- [5] 秦同, 王硕, 高艾, 朱圣英, 崔平远. 一种火星大气密度三维解析模型 [J]. 深空探测学报, 2014, 1(02): 117-122.
- [6] Ushijima M, Hara H, Muramatsu A, et al. The operating characteristics of a ducted rocket in the Mars atmosphere [J]. AIAA paper, 20106999.
- [7] 欧洲航天局.MCD 数据库.[DB].http://www-mars.lmd.jussieu.fr/mcd_python/, 2021-08-07
- [8] 航天爱好者网. 2020 中国火星探测计划 (根据叶院士报告整理).[EB/OL].
<http://www.spaceflightfans.cn/28219.html>, 2021-08-07
- [9] 陈旭, 荣伟, 陈国良. 大气对降落伞开伞动载阻尼作用的分析 [A]. 中国科学技术协会. 提高全民科学素质、建设创新型国家——2006 中国科协年会论文集 (下册) [C]. 中国科学技术协会: 中国科学技术协会学会学术部, 2006: 5.
- [10] 于莹潇, 田佳林. 火星探测器降落伞系统综述 [J]. 航天返回与遥感, 2007(04): 12-16.
- [11] 人民日报. 祝融号火星车顺利发回遥测信号我国首次火星探测任务着陆火星成功.[EB/OL].<http://finance.people.com.cn/n1/2021/0515/c1004-32104220.html>, 2021-08-07
- [12] 雍炯敏. 最优控制教程 [M]. 第 1 版. 北京: 高等教育出版社, 2006.
- [13] 杜剑平, 韩中庚. 嫦娥三号软着陆轨道设计与控制策略的优化模型 [J]. 数学建模及其应用, 2014, 3(04): 39-53.
- [14] 戴娟. 火星着陆器轨迹跟踪控制方法研究 [D]. 北京理工大学, 2016.
- [15] 张瑶. 火星探测器动力下降段软着陆制导研究 [D]. 哈尔滨工业大学, 2018.
- [16] 崔平远, 胡海静, 朱圣英. 火星精确着陆制导问题分析与展望 [J]. 宇航学报, 2014, 35(03): 245-253.
- [17] 任高峰, 朱圣英. 火星着陆任务落点误差快速分析方法 [J]. 哈尔滨工业大学学报, 2012, 44(07): 14-20. [18] 王超, 徐瑞. 基于时间-燃耗优化的火星反推制动制导律设计 [A]. 中国宇航学会深空探测技术专业委员会, 2013: 6.
- [19] Wolf A A, Graves C, Powell R, et al. Systems for pinpoint landing at Mars [J]. 2004.

6 附件 (视频或 3D 模型等)

6.1 程序附录

Listing 1: guidaomini.py

```
1  # -*- coding: UTF-8 -*-
2
3  from matplotlib.patches import Ellipse, Circle
4  import matplotlib.pyplot as plt
5  import numpy as np
6  import util
7
8  # 地球/起点
9  AX = 0.5293788380171227
10 AY = -0.8669958827267433
11
12 # 霍曼转移理论火星位置
13 H_BX = -0.814
14 H_BY = (AY/AX) * H_BX
15 H_aFull, H_bFull, H_centerX, H_centerY, H_angle, radiusA, H_radiusB, H_slope = util.ellipseCoord
    (
16     AX, AY, H_BX, H_BY, 0.9)
17
18 # 快速转移理论火星位置
19 F_BX = -1
20 F_BY = (AY/AX) * F_BX
21 F_aFull, F_bFull, F_centerX, F_centerY, F_angle, radiusA, F_radiusB, F_slope = util.ellipseCoord
    (
22     AX, AY, F_BX, F_BY, 0.85)
23
24
25 # 快速转移理论椭圆长轴终点
26 DX = 0.35
27 DY = 1.52
28
29 EX = 0.08648922946393636
30 EY = 1.559225411955731
31
32
33 fig = plt.figure()
34 ax = fig.add_subplot(111)
35 ax.set_title('TianWen 1 Orbit Simulation - OGWW.COM', fontsize=12, color='k')
36
37 # 椭圆和圆画线, 其中椭圆 heigh为长轴 angle 为逆时针旋转角度
38 ell1 = Ellipse(xy=(H_centerX, H_centerY), width=H_bFull, height=H_aFull, angle=H_angle,
39                fc='white', fill=False, ec='grey', lw=1, linestyle='--', label='Hohmann transfer
    orbit')
40 ell2 = Ellipse(xy=(F_centerX, F_centerY), width=F_bFull, height=F_aFull, angle=F_angle,
```

```

41         fc='white', fill=False, ec='red', lw=1, linestyle='--', label='Fast transfer
           orbit')
42 cir1 = Circle(xy=(0.0, 0.0), radius=radiusA,
43             fc='white', fill=False, ec='#8e8e8e', lw=2, label='Earth')
44 cir2 = Circle(xy=(0.0, 0.0), radius=H_radiusB,
45             fc='white', fill=False, ec='orange', lw=2, label='Mars')
46 F_X = np.linspace(AX, F_BX, 2)
47 F_Y = F_slope * F_X
48 plt.plot(F_X, F_Y, linestyle='--', lw=1,
49         color='red', alpha=0.6, label='Orbit axis')
50
51 ax.add_patch(ell1)
52 ax.add_patch(ell2)
53 ax.add_patch(cir1)
54 ax.add_patch(cir2)
55 ax.plot(0, 0, 'bo')
56 ax.plot(F_centerX, F_centerY, 'ro')
57 plt.axis('scaled')
58 plt.axis('equal')
59 plt.xlim(-2.8, 2.8)
60 plt.ylim(-2.8, 2.8)
61 plt.legend(fontsize="xx-small")
62 ax.plot(AX, AY, 'k*')
63 ax.annotate('Start', xy=(AX, AY), xytext=(AX+0.1, AY-0.2))
64 ax.plot(DX, DY, 'k*')
65 ax.annotate('End', xy=(DX, DY), xytext=(DX+0.1, DY+0.1))
66 ax.set_ylim(-2, 2)
67 ax.grid(True)
68
69
70 plt.show()

```

Listing 2: A1sim.py

```

1 import numpy as np
2 from matplotlib import pyplot as plt
3
4 def gacc(y):
5     return G*mars_mass/(y**2)
6
7
8 def dens(y):
9     if y>7000:
10         temp=249.5-2.22*(y/1000)
11         pressure= 700*np.exp(-0.09*(y/1000))
12     if y<=7000:
13         temp= 241-0.999*(y/1000)
14         pressure= 700*np.exp(-0.09*(y/1000))
15     return pressure/(9188.95110711075*temp)

```

```

16
17
18 mars_mass= 6.39*(10**23)
19 G= 6.67*(10**(-11))
20 mars_radius= 3.39 * (10**9)
21 y0= 12500*(10**3) + mars_radius
22 v0= -975
23 yAcc= 0
24 yVel= v0
25 y=y0
26 m=1285
27 t=0
28
29 ydata= []
30 yVeldata= []
31 yAccdata= []
32 time= []
33 density= []
34 gdata= []
35 dragdata= []
36
37 delta_t= 0.01
38 i=0
39 while t>=0:
40     if y<=mars_radius:
41         break
42     t=i*delta_t
43     time.append(t)
44     if yVel<-460:
45         g=gacc(y)
46         rho=dens(y)
47         A= 12.56
48         Cd= 1.05
49         drag=A/2*Cd*rho*(yVel**2)
50         yAcc=(drag - m*g)/m
51         yVel += yAcc*delta_t
52         y += yVel*delta_t
53     else:
54         if yVel<-95:
55             g=g(y)
56             rho=rho(y)
57             A= 200
58             Cd= 0.43
59             drag=A/2*Cd*rho*(yVel**2)
60             yAcc=(drag - m*g)/m
61             yVel += yAcc*delta_t
62             y += yVel*delta_t
63         else:
64             if yVel<-3.6:
65                 g=g(y)

```



```

66         rho=rho(y)
67         A= 12.56
68         Cd= 1.05
69         drag=A/2*Cd*rho*(yVel**2)+5000
70         yAcc=(drag - m*g)/m
71         yVel += yAcc*delta_t
72         y += yVel*delta_t
73     else:
74         yAcc=0
75         yVel=-3.6
76         y += yVel*delta_t
77     time.append(t)
78     ydata.append(y)
79     yVeldata.append(yVel)
80     yAccdata.append(yAcc)
81     gdata.append(g)
82     density.append(rho)
83     dragdata.append(drag)
84     i+=1
85
86 s= 0.001
87 impactForce= 1/2*m*(yVeldata[-1]**2)/s
88 impactStress= impactForce/0.01
89
90 if impactStress <= 276*(10**2):
91     print('着陆成功')
92 else:
93     print('着陆失败')
94     print(time[-1])
95     print(ydata[-1]-mars_radius)
96     print(yVeldata[-1])
97 '''
98 acc=[]
99 v=[]
100 height=[]
101 kl=[]
102 for k in range(0,len(yAccdata),100):
103     kl.append(k)
104     acc.append(yAccdata[k])
105     v.append(yVeldata[k])
106     height.append(ydata[k])
107 plt.plot(kl,acc,"C1",label="acc")
108 plt.show()
109 plt.plot(kl,v,"C2",label="vel")
110 plt.show()
111 plt.plot(kl,height,"C3",label="y")
112 plt.show()
113 '''

```

Listing 3: DACA.m

```

1  %十进制蚁群算法，用于解决连续函数优化问题，求min(f)
2  function fx_best=DACA(NC_max,m,Rho,Q,K,f,num,Range,precision)
3  %主要变量定义和符号说明
4  % f;           %目标函数
5  num=11;       %优化参数的个数
6  %Range;       %优化参数的范围，num行，2列，每行对应一个参数范围
7  precision=5;%每个优化参数的精度，精确到小数点后第几位
8  n=num*precision*10+1;%n表示城市个数，一共产生num个数，每个数有precision位精度，每位精度有10个可能值；第一个城市为起始城市
9  D=zeros(n,n); %D表示完全图的赋权邻接矩阵
10 NC_max=50;    %最大迭代次数
11 m=1000;      %蚂蚁个数wo
12 Rho=0.2;     %信息素蒸发系数
13 Q=0.5;       %信息素增加强度系数，小于1
14 K=5;         %信息素的权重系数
15 R_best=zeros(NC_max,floor(n/10+1)); %各代最佳路线
16 x=0;
17 fx_best=inf;
18 fx_ave=zeros(NC_max,1); %各代路线的平均长度
19
20 %% 第一步：变量初始化
21 %构造D
22 D(1,1)=eps;
23 for i=2:n
24     if i<=11
25         D(1,i)=1;
26     else
27         D(1,i)=inf;
28     end
29     D(i,1)=inf;
30 end
31 for i1=1:num
32     for i2=1:precision
33         for i3=1:10
34             i=(i1-1)*precision*10+(i2-1)*10+i3+1;
35             for j1=1:num
36                 for j2=1:precision
37                     for j3=1:10
38                         j=(j1-1)*precision*10+(j2-1)*10+j3+1;
39                         if i==j
40                             D(i,j)=eps; %i=j时不计算，应该为0，但后面的启发因子要取倒数，用eps（浮点相对精度）表示
41                         else
42                             if i1==j1 && i2==j2-1 || (i1==j1-1 && j2==1 && i2==precision)
43                                 D(i,j)=1;
44                             else D(i,j)=inf;
45                             end
46                         end

```

```

47         end
48     end
49 end
50 end
51 end
52 end
53 Eta=1./D; %Eta为启发因子，这里设为距离的倒数
54 Tau=5*ones(n,n); %Tau为信息素矩阵
55 Tabu=zeros(m,floor(n/10+1)); %存储并记录路径的生成
56 NC=1; %迭代计数器
57
58 while NC<=NC_max %停止条件之一：达到最大迭代次数
59     %% 第二步：将m只蚂蚁放到初始城市上
60     for i=1:m
61         Tabu(i,1)=1;
62     end
63     %% 第三步：m只蚂蚁按概率函数选择下一座城市，完成各自的周游
64     for j=2:n/10+1 %所在城市不计算
65         for i=1:m
66             last_city=Tabu(i,j-1);
67             P=zeros(1,10);%概率分布
68             next_line=ceil((last_city-1)/10)*10+2;%待访问的城市第一个，即待访问的城市编号为：J
69                 到J+9
70             %下面计算待选城市的概率分布
71             p=0;
72             for k=next_line:next_line+9
73                 if D(last_city,k)==1
74                     p=p+Tau(last_city,k);
75                 end
76             end
77             for k=next_line:next_line+9
78                 if D(last_city,k)==1
79                     P(k-next_line+1)=Tau(last_city,k)/p;
80                 end
81             end
82             %按 概率原则 选取下一个城市
83             Pcum=cumsum(P); %cumsum, 求序列的和序列
84             Select=find(Pcum>=rand); %若计算的概率大于原来的就选择这条路线
85             to_visit=next_line+Select(1)-1;
86             Tabu(i,j)=to_visit;
87         end
88     end
89     %% 杀死第一只蚂蚁，取而代之的是之前的迭代中最优的蚂蚁
90     if NC>=2
91         Tabu(1,:)=R_best(NC-1,:);
92     end
93     %% 第四步：记录本次迭代最佳路线
94     data=zeros(m,num); %m只蚂蚁，每只蚂蚁对应num个数。data记录各优化参数
95     fx=zeros(m,1); %保存优化函数值
96     pos=0; %保存本次迭代取到最优的位置

```

```

96     for i=1:m
97         R=Tabu(i,2:end)-floor((Tabu(i,2:end)-2)./10)*10-2;%R记录各个城市对应的数，0~9
98         for j=1:num
99             sum=0.0;
100            for k=1:precision
101                sum=sum/10+R((j-1)*precision+k)/10;
102            end
103            sum=sum*(Range(j,2)-Range(j,1))+Range(j,1);
104            data(i,j)=sum;
105        end
106        fx(i,1)=f(data(i)); %求该优化参数对应的优化函数值
107        if fx(i)<=fx_best
108            x=data(i);
109            fx_best=fx(i);
110            pos=i;
111        end
112    end
113    R_best(NC,:)=Tabu(pos,:); %此轮迭代后的最佳路线
114    fx_ave(NC)=fx(pos);%mean(fx);
115    NC=NC+1
116    %% 第五步：更新信息素
117    Delta_Tau=zeros(n,n); %开始时信息素为n*n的0矩阵
118    for i=1:size(pos,2) %如果每轮迭代保存前若干优的路径，则要用到这一句
119        for j=1:(n/10-1)
120            Delta_Tau(Tabu(pos(i),j),Tabu(pos(i),j+1))=K*Q^(fx(i)-fx_best); %此次循环在路径 (i
121                , j) 上的信息素增量
122            if Tabu(pos(i),j+1)-1>0 && D(Tabu(pos(i),j),Tabu(pos(i),j+1)-1)==1
123                Delta_Tau(Tabu(pos(i),j),Tabu(pos(i),j+1)-1)=K*Q^(fx(i)-fx_best)/4;
124            end
125            if Tabu(pos(i),j+1)+1<n && D(Tabu(pos(i),j),Tabu(pos(i),j+1)+1)==1
126                Delta_Tau(Tabu(pos(i),j),Tabu(pos(i),j+1)+1)=K*Q^(fx(i)-fx_best)/4;
127            end
128        end
129    end
130    Tau=(1-Rho).*Tau+Delta_Tau; %考虑信息素挥发，更新后的信息素
131    %% 第六步：禁忌表清零
132    Tabu=zeros(m,floor(n/10+1));
133    end
134    %% 第七步：输出结果
135    x
136    fx_best
137    figure;
138    plot(fx_ave,'r')
139    hold on; %保持图形
140    plot(R_best,'b');
141    title('平均'); %标题

```

Listing 4: odeRK4.m

```

1  %4阶龙格库塔法解常微分方程组
2  function[Y,t]=odeRK4(f,tspan,Y0,Nt)
3  Nvar=nmuel(Y0);
4  dt=(tspan(2)-tspan(1))/(Nt-1);
5  Y=zeros(Nvar,Nt);
6  Y(:,1)=Y0(:);
7  t=linspace(tspan(1),tspan(2),Nt);
8  for ii=1:Nt-1
9      K1=f(Y(:,ii),t(ii));
10     K2=f(Y(:,ii)+K1*dt/2,t(ii)+dt/2);
11     K3=f(Y(:,ii)+K2*dt/2,t(ii)+dt/2);
12     K4=f(Y(:,ii)+K3*dt,t(ii)+dt);
13     Y(:,ii+1)=Y(:,ii)+dt/6*(K1+2*K2+2*K3+K4);
14 end
15 end

```

6.2 视频等资料附录

见附件文件夹中.avi 等格式附件。