

2014A

嫦娥三号软着陆轨道设计与控制策略

摘要

针对问题一，建立了嫦娥三号着陆准备轨道模型，根据开普勒定律和引力势能公式及机械能守恒，借助预定着陆点的位置反推法求出嫦娥三号在近月点和远月点的速度分别为 1692.46m/s 和 1614.15m/s 。又根据着陆过程的判断，可知水平方向上距离的变化主要发生在主减速段，借助问题二的建模求出水平路程，转换成纬度差，而竖直距离忽略，得到近月点的经度 19.51W ，纬度 31.4803N 。再通过月球的球对称性，进一步推导出远月点的经度 160.49E ，纬度 31.4803S 。

针对问题二，建立了嫦娥三号软着陆轨道一般性优化模型，在建立合适的坐标系后，给出嫦娥三号的一般受力分析模型，求出嫦娥三号的运动方程后，建立了单目标优化模型。对于不同阶段，修改响应的初始状态、终止状态和约束条件，以及调整目标函数，在一般性优化模型的基础上建立更精细化完善的模型。

对于主减速阶段，建立了主减速轨道优化模型。对于夹角，使用待定系数的三次方程拟合，并使用共轭梯度的搜索法迭代求解，将时间段离散化，估计出最优参数和最优时间，最终得到主发动机的推力方向、运行轨迹、运行速度和高度的变化曲线，最终计算得出消耗燃料 1058.67kg 。

对于快速调整段，建立了快速调整轨道优化模型。通过类似方法，最终计算得出消耗燃料 43.63kg 。

对于粗避障阶段，还需要额外的评价函数选取着陆点。根据包含平坦度和油耗在内的多项指标，我们首先使用聚类算法，筛除陨石坑所在区域。在此基础上，使用二次聚类方法，形成候选着陆点集。接下来，又建立了候选点综合评价及最优决策模型，使用熵权-TOPSIS 候选点排序，综合评估获得最佳着陆点为 $(-34, -21)$ 。接下来通过类似上述阶段方法，最终计算得出消耗燃料 69.53kg 。

对于精避障阶段，建立了精避障阶段优化模型。相比粗避障阶段，由于地图范围较小忽视燃料消耗的差距，通过类似方法，获得最佳着陆点为 $(6.5, 11.5)$ ，最终计算得出消耗燃料 12.55kg 。

对于缓速下降阶段，建立了缓速下降阶段的优化模型。通过类似方法，最终计算得出消耗燃料 24.31kg 。

针对问题三，对于模型的误差和敏感性分析，主要考虑主发动机最大推力的误差影响以及主发动机推力方向角近似控制多项式系数误差的影响，得出主发动机最大推力与推力方向角多项式系数在小范围误差下对轨迹影响较小，但当误差增大到一定程度时会显著改变末端速度和高度，因此需控制其精度以保证着陆安全。

关键词：开普勒定律，四阶龙格库塔，聚类算法，共轭梯度，误差和敏感性分析

一、问题重述

1.1 问题背景

嫦娥三号于 2013 年 12 月 2 日 1 时 30 分成功发射，12 月 6 日抵达月球轨道，计划在 12 月 14 号在月球表面实施软着陆。嫦娥三号如何实现软着陆以及能否成功成为外界关注焦点。

12 月 10 日晚，嫦娥三号已经成功降轨进入预定的月面着陆准备轨道，期间，将稳定飞行姿态，再次确认，为软着陆做最后准备。

嫦娥三号着陆地点选在较为平坦的虹湾区。但由于月球地形的不确定性，最终“落月”地点的选择仍存在一定难度。在整个“落月”过程中，“动力下降”被业内形容为最惊心动魄的环节，嫦娥三号完全依靠自主导航控制，无法人工干预。在距月面 100 米处时，会进行短暂的悬停，扫描月面地形，避开障碍物，寻找着陆点。

1.2 问题描述

1. **问题一**：通过开普勒三定律等物理定律计算出嫦娥三号相应速度的大小与方向，然后根据着陆轨道设计，在已知预定着陆点的情况下得到着陆准备轨道近月点和远月点的位置。
2. **问题二**：在满足嫦娥三号软着陆过程 6 个阶段的要求下，尽量减少软着陆过程的燃料消耗，提出最优控制策略。
3. **问题三**：由于在模型建立过程中不可避免地引入一定的假设与参数设定，同时受限于数值精度，对于设计出的着陆轨道和控制策略做相应的误差分析和敏感性分析。

二、模型假设

- 假设 1：由于月球扁率极小，将月球近似成球体，以月球的平均半径计算。
- 假设 2：忽略月球自转影响。
- 假设 3：忽略月球引力非球项、日月引力摄动等影响因素。
- 假设 4：嫦娥三号软着陆轨道平面穿过月球自转轴，自南向北着陆。

三、符号说明

| 符号 | 含义 | 单位 |
|----------|-------------|---------------------------------------|
| G | 万有引力常量 | $\text{N}\cdot\text{m}^2/\text{kg}^2$ |
| r | 嫦娥三号到月心的距离 | m |
| v | 嫦娥三号的绕行速度 | m/s |
| M | 月球的质量 | kg |
| R | 月球的平均半径 | km |
| m | 嫦娥三号的质量 | kg |
| I_{sp} | 推进剂比冲 | m/s |
| F | 推力 | N |
| Ψ | 制动发动机的推力方向角 | N |
| g_e | 地表重力加速度常数 | m/s^2 |

四、问题一模型的建立与求解

针对问题一，首先通过开普勒定律和能量守恒定理进行联立可得嫦娥三号在近月点和远月点的速度大小和方向。对于着陆准备轨道近月点和远月点的位置，由题目条件不能正向得到，且具体分析较为复杂，因此具体模型将在第二问进行说明，最终通过软着陆模型可计算得出近月点与预着陆位置的纬度差，再以预定着陆点位置反推近月点与远月点的位置。

4.1 问题一模型的构建

4.1.1 嫦娥三号着陆准备轨道模型构建

对于嫦娥三号位于的着陆准备轨道，飞船此时可近似为只受中心天体的引力作用，无外力干扰，因此可运用开普勒定律对其运动过程进行描述，并结合能量守恒定理求解嫦娥三号在近月点和远月点的速度和方向。

如图 1 所示，已知近月点 A 距离月面的高度为 $h_A = 15\text{km}$ ，远月点 B 高度为 $h_B = 100\text{km}$ ，月球平均半径为 $R = 1737.013\text{m}$ 。

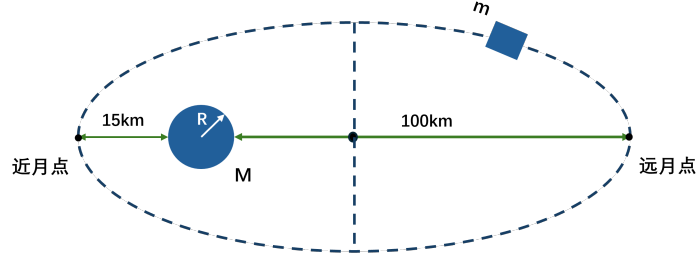


图 1 嫦娥三号轨道示意图

由开普勒第一定律可知，嫦娥三号的运行轨迹为椭圆形，月球位于椭圆的一个焦点上。由第二定理可知，天体在轨道上运动时，其位矢扫过的面积在相同时间内是相等的，即：

$$\frac{1}{2}r_A v_A = \frac{1}{2}r_B v_B \quad (1)$$

其中 $r_A = h_A + R = 1752013\text{m}$ 为近月点到月心的距离， $r_B = h_B + R = 1837013\text{m}$ 为远月点到月心的距离。

然后根据引力势能公式和机械能守恒可知：

$$\frac{1}{2}mv_A^2 - G\frac{Mm}{r_A} = \frac{1}{2}mv_B^2 - G\frac{Mm}{r_B} \quad (2)$$

其中 $M = 77.3477 \times 10^{22}\text{kg}$ 为月球质量， $m = 2.4\text{t}$ 为嫦娥二号质量，联立两者可得：

$$\begin{cases} v_A = \sqrt{2GM\frac{r_B}{r_A} \frac{1}{r_A + r_B}} \\ v_B = \sqrt{2GM\frac{r_A}{r_B} \frac{1}{r_A + r_B}} \end{cases} \quad (3)$$

带入参数得：

$$v_A = 1692.46\text{m/s} \quad (4)$$

$$v_B = 1614.15\text{m/s} \quad (5)$$

其中近月点与远月点的速度方向均为轨道切线方向。

4.1.2 近月点与远月点的确定

近月点和远月点的位置可有其在月球表面投影点的经纬度，以及该点的海拔来描述。因我们已知嫦娥三号的着陆点为 (19.51W, 44.12N)，若再知道飞船着陆轨迹，则可通过着陆点与着陆轨迹逆向推出近月点位置，再根据椭圆的性质推出远月点的位置。但是由于飞船着陆控制策略和轨迹尚未确定，无法直接求出近月点与远月点的精确位置，因此我们将模型进行一定理想状态的简化，得出近月点与远月点的近似位置，再利用问题二中的优化模型和控制策略，对近月点与远月点的位置进行修正。

在实际情况下，飞船会近似沿着经线着陆以减少水平方向的推力做功和能源消耗，因此我们可认为软着陆过程仅改变纬度。

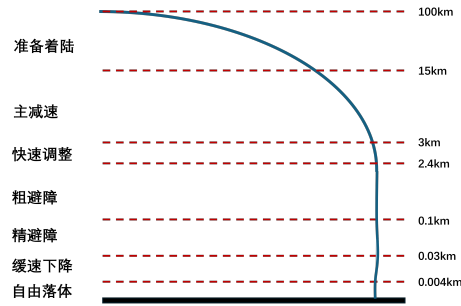


图2 软着陆过程示意图

分析嫦娥三号的6个着陆阶段可知，在主减速阶段，探测器在离月球表面15km处以一段平滑曲线下降至月面3km，始末位置相对于月面的水平路程较大。而在快速调整阶段、粗避障、精避障等阶段，嫦娥三号基本位于着落点上方，水平移动幅度较小，可以忽略。在缓慢下降和自由落体阶段，嫦娥三号基本保持数值下降，不考虑水平路程。

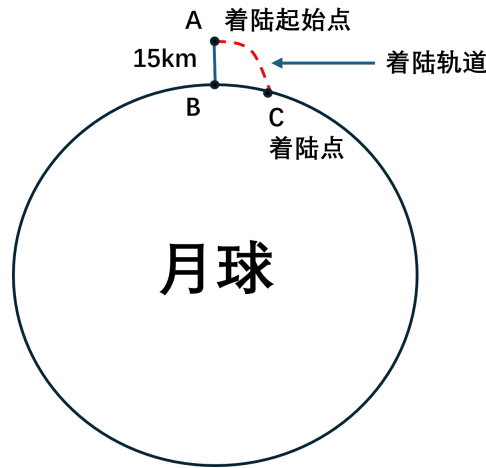


图3 软着陆轨迹示意图

对此，在飞船软着陆过程中，我们可将主减速阶段的水平路程看作整个软着陆过程的水平路程，再由于月球半径为1737.10km远大于飞船在主减速阶段及其之后降落阶段离地高度15km，故可忽略飞船的竖直方向路程(即 $AB \approx BC$)。最后以公式

$$\Delta Angle = \frac{BC}{2\pi R} \quad (6)$$

得出纬度的变化量。得到近月点的位置为(19.51W, 31.4803N)，由月球的球对称性，远月点纬度南北对调，经度东西对调，角度互补。得出远月点的位置为(160.49E, 31.4803S)。根据题意可知，近月点和远月点的高度分别为15km和100km。

综上，着陆准备轨道的近月点、远月点的近似位置，嫦娥三号在该处的速度大小及方向如下表所示：

表 1 近月点、远月点位置与速度信息

| 近/远月点 | 经度 | 纬度 | 高度 | 速度大小 | 速度方向 |
|-------|---------|----------|-------|------------|-----------|
| 近月点 | 19.51W | 31.4803N | 15km | 1692.46m/s | 沿准备轨道切线方向 |
| 远月点 | 160.49E | 31.4803S | 100km | 1614.15m/s | 沿准备轨道切线方向 |

五、问题二模型的建立与求解

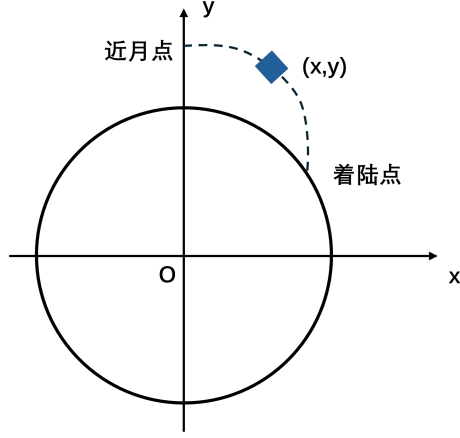
针对问题二，嫦娥三号软着陆过程的六个不同阶段以及状态要求，首先考虑建立嫦娥三号软着陆轨道一般性优化模型，再根据不同阶段的不同要求修改初始状态和改变约束条件，进行细分和详细地求解。

5.1 嫦娥三号软着陆轨道一般性优化模型

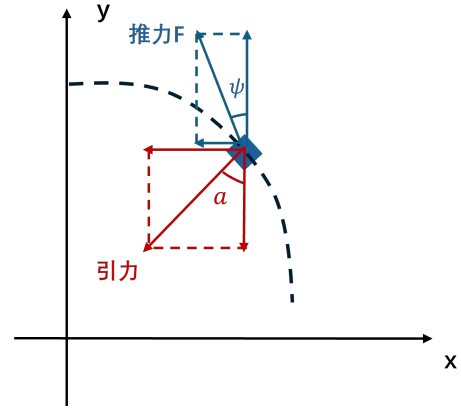
嫦娥三号从近月点处开始进入软着陆轨道，且软着陆轨道与着陆准备轨道在一个平面内，由此可知软着陆轨道在过月心和着陆点的平面上。也正因如此，可知飞船在软着陆轨道运动时，飞船坐标的经度不会发生变化，只需要考虑纬度的变化即可，所以为了简化模型和计算，可以直接以月心为原点，以月心指向近月点为坐标 y 轴，垂直方向为坐标 x 轴建立直角坐标系（如下图（a）所示），通过计算近月点与预着陆位置的纬度差，再以预定着陆点位置反推近月点与远月点的位置。

事实上，满足这个条件的平面是不唯一的。不失一般性，这里将软着陆轨道平面约束在月球的极轴（南北极）所在平面，即将极轴和着陆点所在平面作为软着陆轨道所在平面。

在该坐标系下，将嫦娥三号软着陆过程分为 6 个阶段，分别是主减速阶段、快速调整阶段、粗避障阶段、精细避障阶段、缓速下降阶段和自由落体阶段。这 6 个阶段的受力状态和变化情况基本一致，所以，在这里先给出嫦娥三号的一般受力分析模型，之后分别讨论每个阶段的最优控制策略问题。



(a) 软着陆轨道平面坐标系示意图



(b) 嫦娥三号着陆过程受力分析图

根据该坐标系，可得出嫦娥三号的运动方程为：

$$\frac{d^2x}{dt^2} = \frac{GM}{x^2 + y^2} \sin \alpha(t) + \frac{F_x(t)}{m(t)} \quad (7)$$

$$\frac{d^2y}{dt^2} = \frac{GM}{x^2 + y^2} \cos \alpha(t) + \frac{F_y(t)}{m(t)} \quad (8)$$

$$\alpha(t) = \arctan \frac{x(t)}{y(t)} \quad (9)$$

并且由比冲的定义（即单位质量的推进剂产生的推力）可以求得：

$$\frac{dm}{dt} = -\frac{\sqrt{v_x^2(t) + v_y^2(t)}}{I_{sp}} \quad (10)$$

接着记从着陆准备轨道开始的软着陆过程的初始时刻为 0，并以此开始计时，截至六个阶段中第 i 个阶段所花费的总时间为 t_i ，根据文献 [1,2] 等可先将最少燃油消耗作为目标函数，考虑到各阶段状态的约束条件，则问题归结为下面的优化模型：

$$\min z = \int_0^{t_1} \frac{\sqrt{v_x^2(t) + v_y^2(t)}}{I_{sp}} dt$$

$$\begin{cases}
\frac{d^2x}{dt^2} = \frac{GM}{x^2 + y^2} \sin \alpha(t) + \frac{F_x(t)}{m(t)} \\
\frac{d^2y}{dt^2} = \frac{GM}{x^2 + y^2} \cos \alpha(t) + \frac{F_y(t)}{m(t)} \\
\alpha(t) = \arctan \frac{x(t)}{y(t)} \\
\frac{dm}{dt} = - \frac{\sqrt{v_x^2(t) + v_y^2(t)}}{I_{sp}} \\
\text{s.t. } \begin{cases}
\text{at } t_i : \\
m(t_i) = m_i, \quad x(t_i) = x_i, \\
y(t_i) = y_i, \quad \frac{dx}{dt}(t_i) = v_{x_i}, \\
\frac{dy}{dt}(t_i) = v_{y_i} \\
\text{at } t_{i+1} : \\
m(t_{i+1}) = m_{i+1}, \quad x(t_{i+1}) = x_{i+1}, \\
y(t_{i+1}) = y_{i+1}, \quad \frac{dx}{dt}(t_{i+1}) = v_{x_{i+1}}, \\
\frac{dy}{dt}(t_{i+1}) = v_{y_{i+1}}
\end{cases}
\end{cases}$$

其中模型的初始条件为: $(x(t_i), y(t_i), v_x(t_i), v_y(t_i))$, 终止条件为: $(x(t_{i+1}), y(t_{i+1}), v_x(t_{i+1}), v_y(t_{i+1}))$ 。并且对于不同阶段的最优控制问题, 主要就是相应的初始状态和终止状态约束条件的区别, 并对目标函数进行调整, 其基本模型是类似的。

5.2 主减速轨道优化模型

主减速段的位置区间是距离月面 15km 到 3km, 初状态为近月点, 末状态为进入快速调整阶段的临界, 其距离月面 3km, 由此可确定软着陆模型的初始 0 时刻时的条件限定为:

$$(x(0), y(0)) = (0, 17524013m)$$

$$(v_x(0), v_y(0)) = (1692.46m/s, 0)$$

$$m(0) = 2400kg$$

主减速阶段的末端状态为嫦娥三号位于预定落月点上空 3000m 处, 距离月心 1737013 - 2640 + 3000 = 1737373m, 并且速度要求为 57m/s, 故末状态 t_1 时刻的条件限定为:

$$\sqrt{x^2(t_1) + y^2(t_1)} = 1737373m$$

$$\sqrt{v_x^2(t_1) + v_y^2(t_1)} = 57m/s$$

并且由于主减速阶段的目的主要是减速，因此主发动机采用最大推力，即 $F = 7500N$ 。关键的难点在于确定推力的方向，由于在减速过程中推力的方向往往存在非线性的方向变化，为了简化模型，根据文献 [2], 不妨令一个待定系数的三次方程对夹角进行拟合：

$$\Psi(t) = a_0 t^3 + a_1 t^2 + a_2 t + a_3 \quad (11)$$

其中 $\Psi(t)$ 表示推力与 y 轴正方向的夹角，该多项式具有比较好的光滑性，这样就可以将关于控制函数 $\Psi(t)$ 的优化问题转化为确定控制参数向量 $p = (a_0, a_1, a_2, a_3)$ 的问题。对于目标函数的选取，由于主减速阶段总距离较长、历时久，可能的油耗较高，因此仍以最少油耗量为目标函数。综上所述，主减速阶段的最优控制模型为：

$$\begin{aligned} \min z = & \int_0^{t_1} \frac{\sqrt{v_x^2(t) + v_y^2(t)}}{I_{sp}} \\ \text{s.t. } & \begin{cases} \frac{d^2 x}{dt^2} = \frac{GM}{x^2 + y^2} \sin \alpha(t) + \frac{F_x(t)}{m(t)} \\ \frac{d^2 y}{dt^2} = \frac{GM}{x^2 + y^2} \cos \alpha(t) + \frac{F_y(t)}{m(t)} \\ \alpha(t) = \arctan \frac{x(t)}{y(t)} \\ \frac{dm}{dt} = -\frac{\sqrt{v_x^2(t) + v_y^2(t)}}{I_{sp}} \\ \Psi(t) = a_0 t^3 + a_1 t^2 + a_2 t + a_3 \\ m(0) = 2400, \quad x(0) = 0, \quad y(0) = 1752013 \\ \sqrt{x^2(t_1) + y^2(t_1)} = 1737373 \\ \sqrt{v_x^2(t) + v_y^2(t)} = 57, \quad F = 7500 \end{cases} \end{aligned}$$

5.3 主减速轨道优化模型的求解与分析

该模型是一个无穷维的优化问题，针对该问题本文结合了文献 [3,4] 等，采用共轭梯度的搜索法进行迭代求解，该算法的流程表如下：

该算法在计算梯度的过程中主要采用四阶 *Runge - Kutta* 法进行求解，将时间段 $[0, 450]$ 内进行离散化，利用哈密顿函数迭代更新最后可以大体估计出需要的时间应在 410s 左右，最终估计出最优参数和最优时间。

表 2 引入约束算子的共轭梯度算法

| 步骤 | 关键操作 | 核心公式 | 说明 |
|----|------------|---|---|
| 1 | 模型简化与控制参数化 | $\theta_1(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0$ | 推力方向角参数化 |
| 2 | 乘子法处理约束 | $J_a = \int_0^{t_f} \frac{T}{v_e} dt + \frac{\sigma}{2} \ \Psi\ ^2 - \nu^T \Psi$ | 终端约束 $\Psi = [r(t_f) - r_f, v(t_f) - v_f]^T$ |
| 3 | 正向积分状态方程 | $\dot{\mathbf{x}} = f(\mathbf{x}, \theta_1(t))$ | Runge-Kutta 法求解 |
| 4 | 反向积分协态方程 | $\dot{\lambda}_r = \lambda_r \frac{2GM_M}{r^3} - \lambda_v \frac{T \sin \theta_1}{mr^2}$ | 协态终端条件 $\lambda(t_f) = \sigma \Psi - \nu$ |
| 5 | 计算梯度 | $g = \int_0^{t_f} \left(\lambda_r \frac{\partial f_r}{\partial p} + \lambda_v \frac{\partial f_v}{\partial p} \right) dt$ | 哈密顿函数对 p 的梯度 |
| 6 | 更新控制参数 | $p^{(k+1)} = p^{(k)} + \alpha_k d^{(k)}, d^{(k)} = \beta_k = -g^{(k)} + \beta_k d^{(k-1)}$ | $\beta_k = \frac{\ g^{(k)}\ ^2}{\ g^{(k-1)}\ ^2}$ |
| 7 | 约束投影 | $\theta_1 \leftarrow \text{clip}(\theta_1, -270^\circ, 180^\circ)$ | 控制量约束处理 |
| 8 | 更新乘子 | $\nu^{(k+1)} = \nu^{(k)} - \sigma \Psi^{(k)}, \sigma \leftarrow \begin{cases} 10\sigma & \text{if } \ \Psi^{(k)}\ > 0.5 \ \Psi^{(k-1)}\ \\ \sigma & \text{otherwise} \end{cases}$ | 自适应罚因子 |
| 9 | 收敛判断 | $\ g^{(k)}\ < \epsilon \text{ and } \ \Psi^{(k)}\ < \epsilon$ | $\epsilon = 10^{-3}$ |

问题的最优控制参数为

$$\mathbf{p}^* = (5.8, -0.000755, 6.0207 \times 10^{-8}, 1.3453 \times 10^{-8}),$$

最优时间为 $t_1 = 415 \text{ s}$ ，即当嫦娥三号的主发动机以最大推力 7500 N 飞行 415 s 后，到达距离预定着陆点上方 3000 m 处。主减速阶段末端速度大小为 57.002 m/s ，速度矢量为 $(39.64, -40.96) \text{ (m/s)}$ ，其末端质量为 1341.33 kg ，即消耗燃料为 1058.67 kg 。

在最优的控制策略下，主发动机的推力方向、运行轨迹、运行速度和高度的变化曲线如图 5 所示。

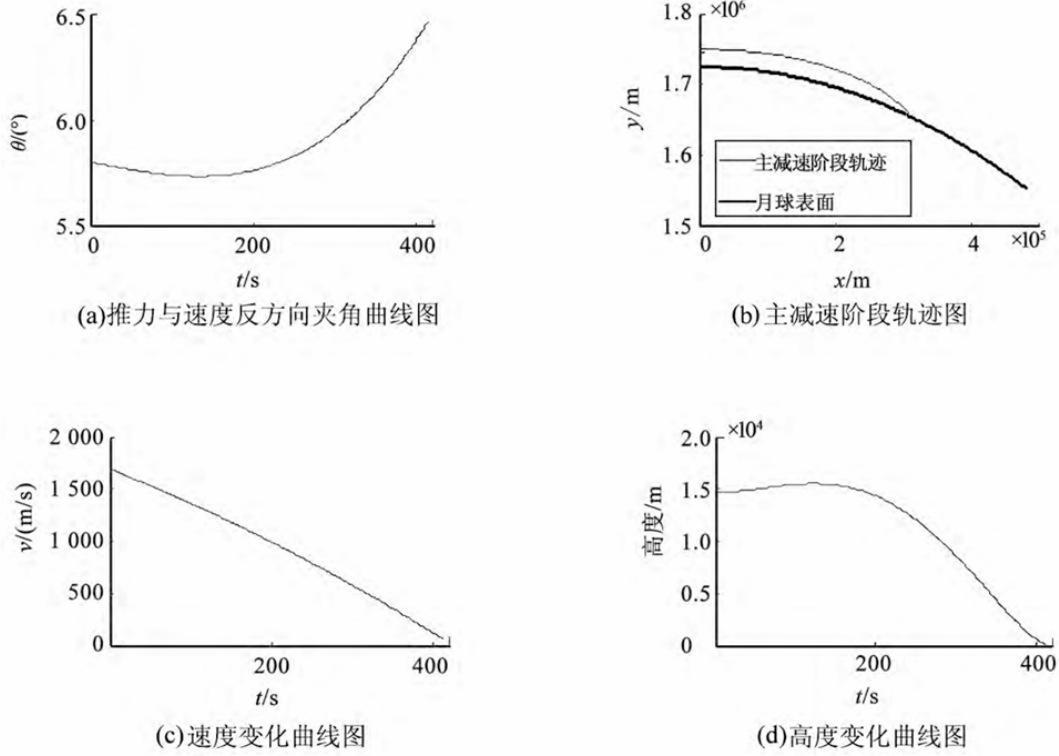


图5 主减速阶段相关变量的变化趋势图

嫦娥三号在主减速阶段共计飞行 415s，其月面投影飞行的纬度数为 12.6397° 。由于嫦娥三号落点为 $(19.51^\circ W, 44.12^\circ N)$ ，假设软着陆轨道为过月球自转轴的平面，且从南至北着陆，则嫦娥三号的近月点的纬度为 $44.12^\circ - 12.6397^\circ = 31.4803^\circ$ ，即嫦娥三号着陆准备轨道的近月点投影的经纬度为 $19.51^\circ W, 31.4803^\circ N$ 。

5.4 快速调整轨道优化模型

快速调整阶段的位置区间为距月面 $3km$ 至 $2.4km$ 处，其初始状态即为主减速阶段的末端状态，末端位置要求悬停在预定落点上空 $2400m$ 处，水平速度调整为 0 ，并将主减速发动机推力方向调整到竖直向上的状态。

$$(x(t_1), y(t_1)) = (380219.57, 1695251.32)$$

$$(v_x(t_1), v_y(t_1)) = (39.64, -40.96)$$

$$m(t_1) = 1341.33kg$$

快速调整阶段末端状态为嫦娥三号位于预定落月点上空 $2400m$ 处，距离月心 $1737013 - 2640 + 2400 = 1736773m$ ，并且速度要求为水平速度达到 $0m/s$ ，由于后续飞船要进行悬停以及自由落体运动，且月球表面重力加速度仅约为 $1.618m/s^2$ ，其值较

小，因此为了简化模型，将末端状态的垂直速度也假设为 0m/s ，故末状态 t_2 时刻的条件限定为：

$$\begin{aligned}\sqrt{x^2(t_2) + y^2(t_2)} &= 1736773\text{m} \\ \sqrt{v_x^2(t_2) + v_y^2(t_2)} &= 0\text{m/s}\end{aligned}$$

对于目标函数的选取，由于快速调整过程的时间短，所以燃料消耗量的优化不是主要问题，因此将目标函数转化为最短调整时间，其余的控制策略保持与主轨道相同。综上所述，快速调整阶段的最优控制模型为：

$$\begin{aligned}\min z &= t_2 - t_1 \\ \text{s.t. } \left\{ \begin{array}{l} \frac{d^2x}{dt^2} = \frac{GM}{x^2 + y^2} \sin \alpha(t) + \frac{F_x(t)}{m(t)} \\ \frac{d^2y}{dt^2} = \frac{GM}{x^2 + y^2} \cos \alpha(t) + \frac{F_y(t)}{m(t)} \\ \alpha(t) = \arctan \frac{x(t)}{y(t)} \\ \frac{dm}{dt} = -\frac{\sqrt{v_x^2(t) + v_y^2(t)}}{I_{sp}} \\ \Psi(t) = a_0 t^3 + a_1 t^2 + a_2 t + a_3 \\ m(t_1) = 1341.33, \quad x(t_1) = 380219.57, \quad y(t_1) = 1695251.32 \\ \sqrt{x^2(t_2) + y^2(t_2)} = 1736773, \quad v_x(t_1) = 39.64, \quad v_y(t_1) = -40.96 \\ \sqrt{v_x^2(t_2) + v_y^2(t_2)} = 0, \quad F = 7500 \end{array} \right.\end{aligned}$$

5.5 快速调整轨道模型的求解与分析

用同样的方法近似求解模型，可以得到：主发动机最佳的推力大小为 4280.7N ，该阶段共计耗时 31s 。其阶段末的状态：速度为 $(0.214, -0.573)\text{(m/s)}$ ，高度为 2409.8m ，剩余质量为 1297.7kg ，消耗燃料 43.63kg 。在该阶段的运行过程中，嫦娥三号的主发动机推力方向、运动轨迹、运行速度和高度的变化规律如图 6 所示

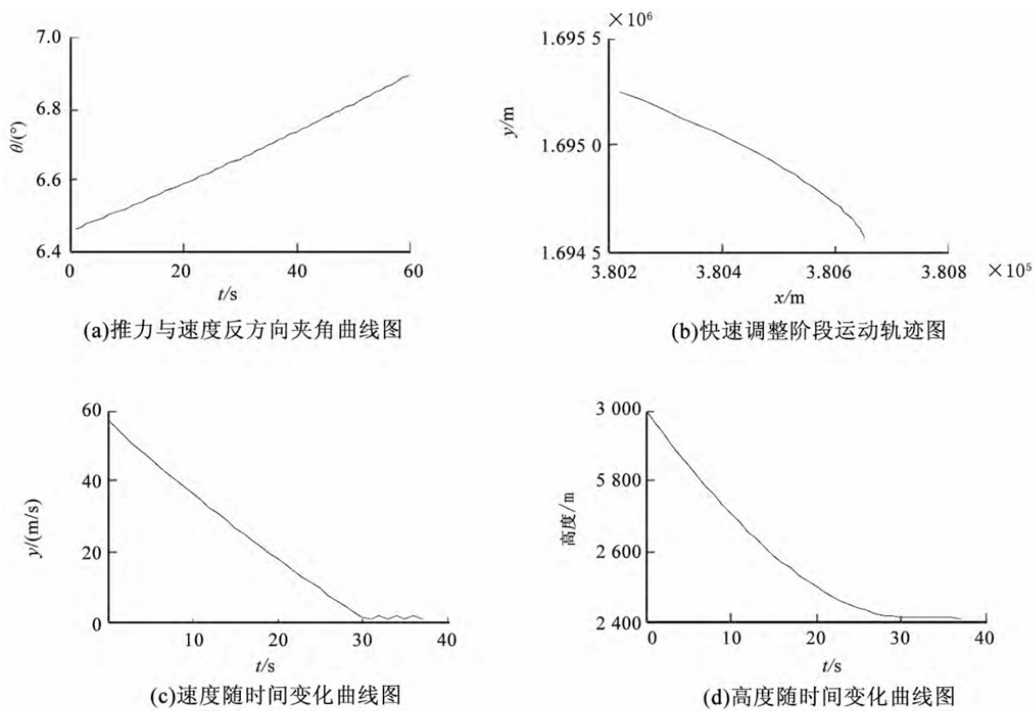


图 6 快速调整阶段各状态变量的变化规律图

快速调整阶段末端的状态:位置为 (380653.334, 1694558.626), 速度为 (0.214, -0.573), 质量为 1297.675kg。

5.6 粗避障阶段优化模型:

嫦娥三号需要在距离月面 2.4km 到 100m 的粗避障段中避开大的陨石坑, 实现在设计着陆点上方 100m 处悬停, 并初步确定落月地点。

5.6.1 评估指标与着陆点的选择

一个理想的着陆点通常满足以下条件:

- 1) 位于陨石坑之外。陨石坑内部通常地形崎岖、凹陷较深, 容易导致着陆器倾斜或损坏, 并且光照条件差。
- 2) 平坦度高, 地面起伏小。有助于保证着陆器的稳定着陆和设备的正常运行。
- 3) 平移距离小, 减少油耗量。

位于陨石坑之外是约束条件, 对于任意一个满足条件的着陆点, 我们需要确定主发动机的控制策略使得油耗量最低, 然后选择一个综合油耗和平坦度的最佳着陆点。

5.6.2 基于聚类算法识别排除陨石坑区域

为确定陨石坑，我们通过 K-means 聚类手肘法确定 k 值，确定 K 为 5。同时通过观察下图数字高程三维视图，可以观察到较大的陨石坑数量为 5，这与程序运行出来的结果相符。

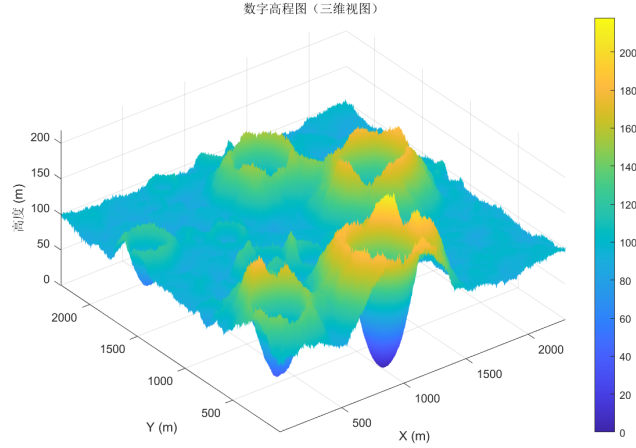


图 7 数字高程三维视图

通过聚类算法，我们得到一系列聚类中心值 127.95, 89.76, 161.94, 101.89, 43.19。

因此，我们将着陆位置完成初步划分，筛除了陨石坑所在区域。相应地，留下了相对安全的候选着陆区域。

在此基础上，我们对该候选区域内的点集进一步处理，利用二次聚类方法对大量候选点进行空间划分，通过计算每个聚类簇的中心点作为代表点，显著降低了候选点数量并保留了空间分布特征。这些代表点构成了最终的候选着陆点集，后续将基于其计算的平坦度和燃油消耗指标，结合熵权法进行综合评价与最优着陆点的选取。

通过前期聚类与阈值剔除，获得非陨石坑区域的候选点集：

$$\mathcal{C} = \{c_i = (x_i, y_i, z_i) \mid i = 1, \dots, N\}.$$

为降低计算复杂度，对 \mathcal{C} 依据空间坐标进行二次聚类，保证代表点均匀覆盖候选区域，设聚类簇数为 $K \ll N$ ，得到簇集合：

$$\mathcal{G} = \{G_k \subset \mathcal{C} \mid k = 1, \dots, K\}.$$

每簇的质心作为代表点：

$$\bar{c}_k = \frac{1}{|G_k|} \sum_{c \in G_k} c, \quad k = 1, \dots, K.$$

代表点集 $\{\bar{c}_k\}$ 既概括空间特征，又便于后续多目标优化，见表???

5.6.3 候选点综合评价及最优决策模型

设有 K 个候选点 \bar{c}_i , 其中 $i = 1..m$ 。对于每个候选点, 我们定义两个原始指标: 平坦度: θ_i , 数值越大表示越平坦, 适合着陆; 耗油量: f_i , 数值越小表示飞行器从当前位置到该点所需的燃油消耗越少。为了统一量纲以便于优化, 将两个指标进行归一化处理, 得到, 得到归一化指标 \tilde{s}_i, \tilde{f}_i , 分别赋予权重 w_s, w_f 通过加权和构建综合目标函数:

$$\max_x \sum_{i=1}^K (w_s \tilde{s}_i + w_f \tilde{f}_i) x_i$$

其中决策变量 $x_i \in \{0, 1\}$ 表示是否选择第 i 个候选点, 且约束

$$\sum_{i=1}^K x_i = 1,$$

保证最终只选择一个最优着陆点。

5.6.4 平坦度及油耗指标的计算及归一化处理

1) 平坦度分数:

设 DEM 高程为 $z(x, y)$, 像素间距为 $\Delta x, \Delta y$ 。用中心差分估算:

$$p = \frac{\partial z}{\partial x} \approx \frac{z(x + \Delta x, y) - z(x - \Delta x, y)}{2\Delta x}$$

$$q = \frac{\partial z}{\partial y} \approx \frac{z(x, y + \Delta y) - z(x, y - \Delta y)}{2\Delta y}$$

标准的坡度 (坡角) 是:

$$\theta(x, y) = \arctan \left(\sqrt{p^2 + q^2} \right)$$

单位: 弧度, θ 越小地形越平。

然后把平坦度分数定义为:

$$\tilde{s}_i = 1 - \frac{\bar{\theta}_i - \min \bar{\theta}}{\max \bar{\theta} - \min \bar{\theta}}$$

2) 油耗分数:

假设飞行器当前位置坐标为 $\mathbf{p}_0 = (x_0, y_0, z_0)$, 候选点 i 的坐标为 $\mathbf{p}_i = (x_i, y_i, z_i)$ 以欧氏距离来估算燃油消耗的代理指标:

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}$$

然后把油耗得分定义为：

$$\bar{f}_i = 1 - \frac{d_i - \min f_i}{\max f_i - \min f_i}$$

熵权-TOPSIS 候选点排序步骤

1. 输入：决策矩阵 $X = [x_{ij}]_{K \times n}$ ，其中 $n = 2$ ；
2. 列向归一化： $p_{ij} = x_{ij} / \sum_{i=1}^K x_{ij}$ ；
3. 计算熵值： $e_j = -k \sum_{i=1}^K p_{ij} \ln p_{ij}$ ， $k = 1 / \ln K$ ；
4. 差异度： $d_j = 1 - e_j$ ；
5. 权重： $w_j = d_j / \sum_{j=1}^n d_j$ ；
6. 向量范数归一化： $r_{ij} = x_{ij} / \sqrt{\sum_{i=1}^K x_{ij}^2}$ ；
7. 加权矩阵： $v_{ij} = w_j r_{ij}$ ；
8. 正负理想解： $v_j^+ = \max_i v_{ij}$ ， $v_j^- = \min_i v_{ij}$ ；
9. 距离： $S_i^+ = \sqrt{\sum_j (v_{ij} - v_j^+)^2}$ ， $S_i^- = \sqrt{\sum_j (v_{ij} - v_j^-)^2}$ ；
10. 贴近度： $C_i = S_i^- / (S_i^+ + S_i^-)$ ；
11. 按 C_i 降序输出排序并选取最优候选点。

经过仿真计算，粗避障阶段的控制误差的标准差约为 $\sigma = 150 \text{ m}$ ，精细避障阶段的控制误差标准差取 $\sigma = 6 \text{ m}$ 。依据式 (3) 可以得到考虑着陆控制误差影响后，着陆区域的综合评估。

从着陆区域综合评估值中选择最大值，即可得到最佳的选择落月点。粗避障阶段的最佳落月点为 $(-34, -21)$ ，即需要在原预定点的基础上向西南方向移动大约 40 m ，如下图所示。

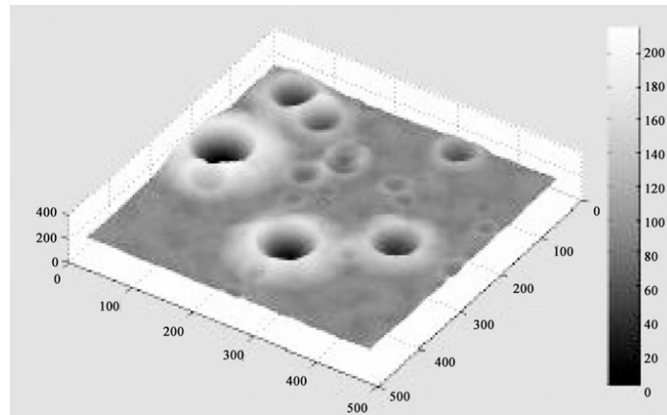


图 8 粗避障阶段最佳落点的位置图

粗避障阶段的起始状态为速度 $(0, 0)$ ，高度 2400 m ；终止状态为垂直速度 0 ，高

度 100m，要求水平位移 40m。为了简化，假设在粗避障阶段的运行轨迹为直线，即月球引力和主减速发动机推力的合力方向在粗避障阶段保持不变。受力情况如下图所示：

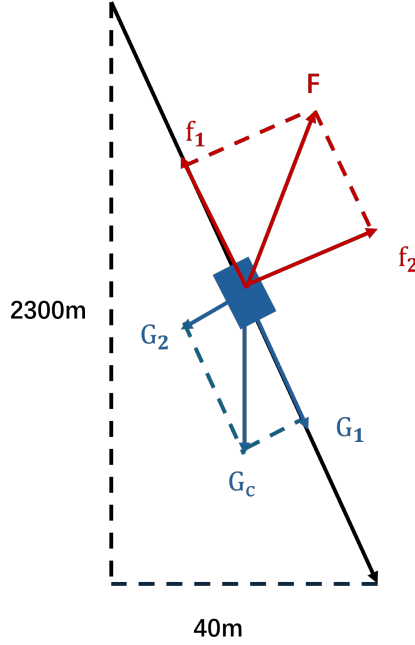


图 9 粗避障阶段的受力分析示意图

在粗避障阶段，令 $G_c = mg_M$ ， $G_1 = G_c \cos \theta$ ， $G_2 = G_c \sin \theta$ ，其中 $\theta = \arctan \frac{4}{230}$ 。若要使粗避障阶段的运行轨迹为直线，即 $f_2 = G_2$ ，现在只需要优化 f_1 随时间变化的曲线即可。

从实际出发，该曲线一般应为抛物线，不妨假设 f_1 随时间变化为三次多项式：

$$f_1(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 \quad (12)$$

则嫦娥三号在运行轨迹上的合力为 $f_1 - G_1$ ，其推力的大小为：

$$F = \sqrt{f_1^2 + f_2^2} \quad (13)$$

以 f_1 方向为 y 轴，以 f_2 方向为 x 轴建立坐标系，则问题的最优控制模型为：

$$\min z = \int_0^{t_f} F(t) dt$$

$$\text{s.t.} \begin{cases} \frac{d^2 y}{dt^2} = -g_M \cos \theta + \frac{f_1}{m}, \\ \frac{dm}{dt} = -\frac{F(t)}{v_e}, \\ F(t) = \sqrt{f_1^2 + f_2^2}, \\ g_M \sin \theta = \frac{f_2}{m}, \\ m(t_2) = 1297.675, y(t_2) = 2300, \frac{dy}{dt}(t_2) = 0, \\ y(t_3) = 0, \frac{dy}{dt}(t_3) = 0. \end{cases}$$

用类似上面的方法求解该模型，则可以得到主发动机推力变化曲线的系数为

$$\mathbf{b}^* = (1104.515, -0.0047, 0.0041, 0.0038),$$

总耗时 99.7 s，剩余质量 1228.15 kg，末端速度 $4.4914 \times 10^{-6} \text{ m s}^{-1}$ 。推力大小和方向随时间变化、高度变化的曲线和粗避障阶段运行轨迹变化如图 10 所示。

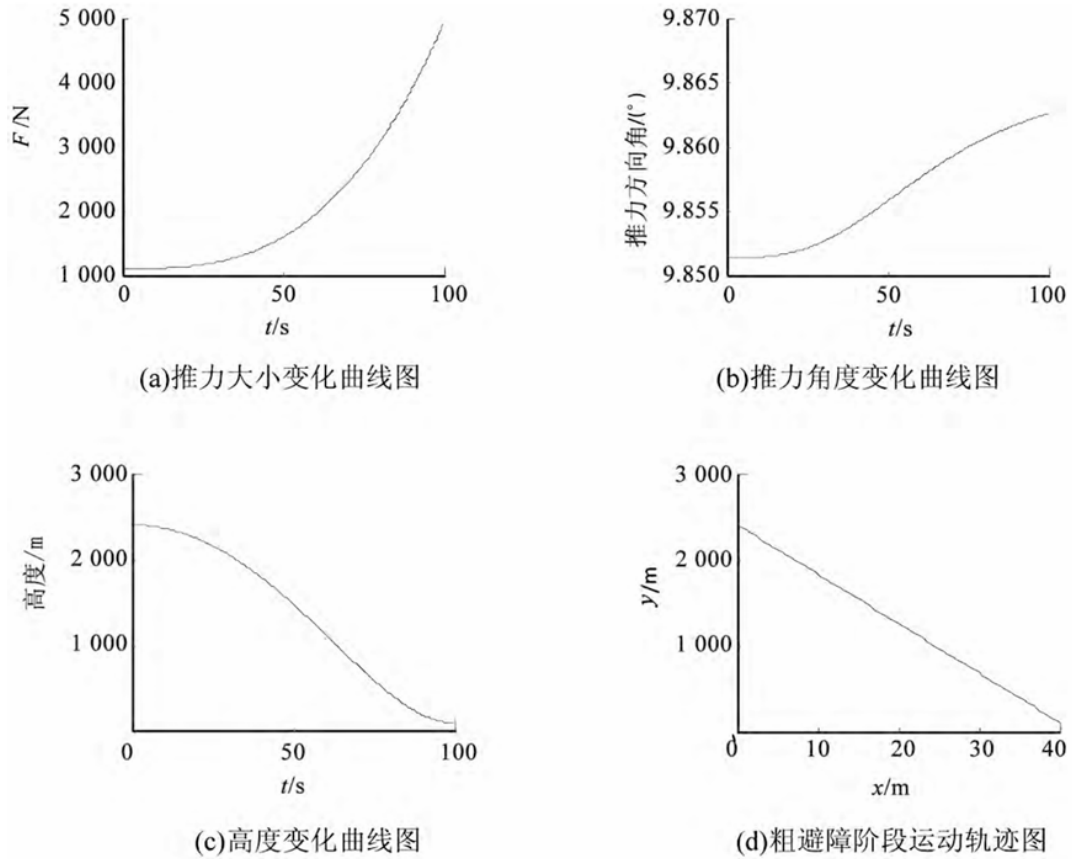


图 10 粗避障阶段的受力分析示意图

由图 10 可以看出，主发动机的推力开始较小，使嫦娥三号能够逐步加速，之后推力逐步增大，以实现在高度 100 m 处悬停。其推力方向角 θ 定义为推力矢量与竖直向上方向的夹角, 可由预定落点的方向确定。

5.7 精避障阶段优化模型：

5.7.1 评估指标与着陆点的选择

在精避障阶段，由于地图范围较小，飞行器的水平平移距离可忽略不计，相应的燃料消耗对着陆决策影响极小。因此，在着陆点的选择中无需考虑水平平移长度，仅需以地形平坦度作为唯一评价指标，选取平坦度最优的着陆点作为目标着陆点。

设候选着陆点集合为

$$P = \{p_1, p_2, \dots, p_K\}$$

其中每个着陆点 p_i 的平坦度指标为 $f_i \in [0, 1]$ ，数值越大表示地形越平坦。则精避障阶段的着陆点选择模型可表示为：

$$p^* = \arg \max_{p_i \in P} f_i$$

其中：

- f_i ：着陆点 p_i 的平坦度评价价值；
- p^* ：最终选定的最佳着陆点。

采用聚类算法对平坦度较高的区域进行分析，并直接选取全局平坦度最高的区域的聚类质心作为最终的最佳着陆点，确定最佳的落月点为 (6.5,11.5)，即需要在粗避障的基础上再向东北方向移动 12.3，如下图所示：

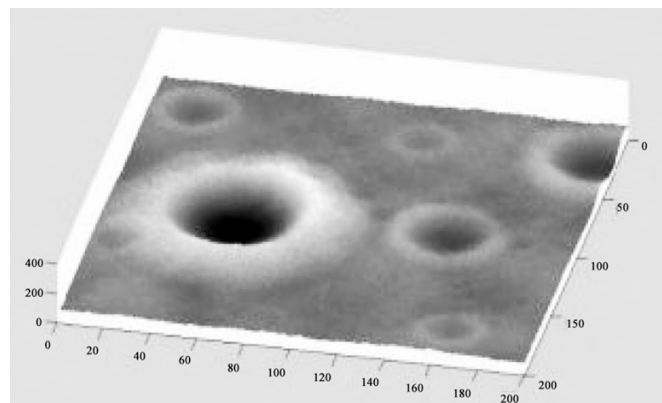


图 11 精避障阶段最佳落点位置图

精避障阶段的控制策略与粗避障阶段略有不同，由于其区间是距离月面 100m 到 30m，该阶段距离较短，因此相比之下速度和调整方案对飞船的行驶时间和油耗影响较小，因此为了简化模型，该阶段可以分两步完成，首先水平平移，然后再垂直下降。

5.7.2 水平平移过程优化模型

首先考虑水平平移过程，依据前文，则相应的优化模型为：

$$\begin{aligned} \min z &= \int_0^{t_f} F(t) dt \\ \text{s.t.} \quad &\begin{cases} \frac{d^2x}{dt^2} = \frac{f_1}{m}, \\ \frac{dm}{dt} = -\frac{F(t)}{v_e}, \\ F(t) = \sqrt{f_1^2 + f_2^2}, \\ g_M = \frac{f_2}{m}, \\ m(t_3) = 1228.15, \quad x(t_3) = 0, \quad \frac{dx}{dt}(t_3) = 0, \\ x(t'_3) = 12.3, \quad \frac{dx}{dt}(t'_3) = 0 \end{cases} \end{aligned}$$

用离散化搜索求解该模型，则可得到主发动机推力的水平分量系数向量为

$$\mathbf{b}^* = (123.4003, -0.2052, 0.0963, -0.0618),$$

总耗时为 18.6 s，剩余质量 1215.6 kg。相应的推力大小和方向随时间变化曲线、水平位移变化曲线如下图所示。

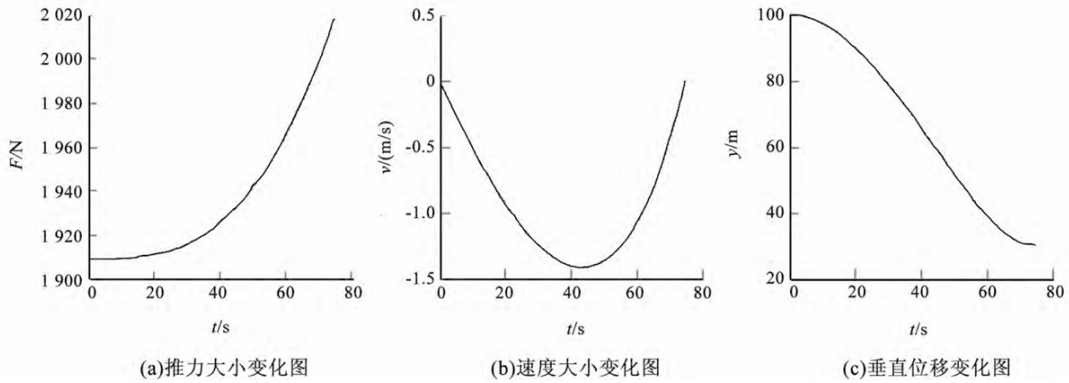


图 12 精避障阶段水平移动过程参数变化图

由图可知，主发动机的推力方向首先是正方向，大约 113s 后变为负方向，这是因为前一段时间推动平移，后一段时间使嫦娥三号制动悬停。

5.7.3 垂直下降过程优化模型

$$\begin{aligned} \min z &= \int_0^{t_f} F(t) dt \\ \text{s.t.} \quad &\begin{cases} \frac{d^2 y}{dt^2} = \frac{F(t)}{m} - g_M, \\ \frac{dm}{dt} = -\frac{F(t)}{v_e}, \\ m(t'_3) = 1215.6, y(t'_3) = 100, \frac{dy}{dt}(0) = 0, \\ y(t_4) = 30, \frac{dy}{dt}(t_4) = 0 \end{cases} \end{aligned}$$

同样用离散化方法求解该模型，则可以得到主发动机推力的系数向量为

$$\mathbf{b}^* = (1908.918, 1.273 \times 10^{-5}, -0.000647, 0.000268),$$

总耗时为 75 s，剩余质量 1166.21 kg。相应的主发动机推力的大小、速度和垂直位移变化曲线如下图所示。

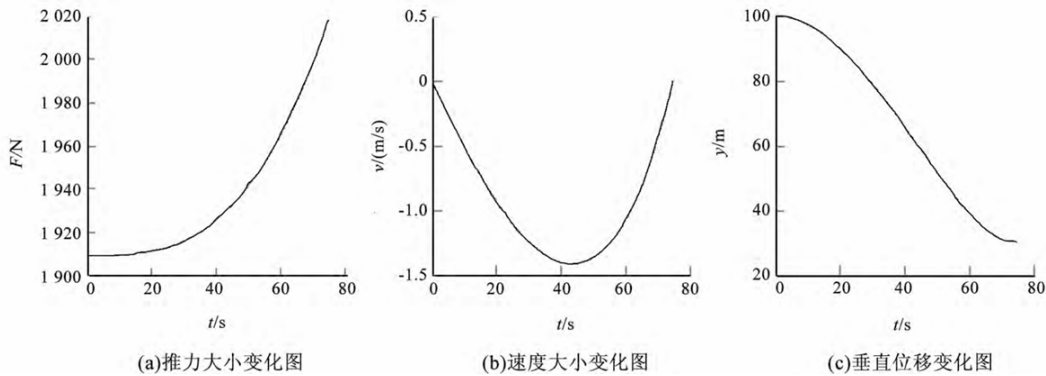


图 13 精避障阶段垂直下降过程的参数变化图

由图可知，主发动机的推力是逐步增大的，从而使得下落速度先加速后减速，最终末端速度为 0，即悬停于距月面 30m 高度。

5.8 缓速下降阶段的优化模型：

起始状态为精避障阶段的末端状态（距月面 30m，速度 0，质量 1166.21 kg），发动机推力方向垂直向上。末端状态要求距离月面 4m，速度为 0。其控制过程和优化模型与精避障模型类似，则有：

$$\begin{aligned} \min z &= \int_0^{t_f} F(t) dt \\ \text{s.t.} \quad &\begin{cases} \frac{d^2 y}{dt^2} = \frac{F(t)}{m} - g_M, \\ \frac{dm}{dt} = -\frac{F(t)}{v_e}, \\ m(0) = 1166.21, y(0) = 30, \frac{dy}{dt}(0) = 0, \\ y(t_f) = 4, \frac{dy}{dt}(t_f) = 0. \end{cases} \end{aligned}$$

用类似如上的方法求解该模型，得到主发动机推力的系数向量为

$$\mathbf{b}^* = (1850.99, -0.00024, -0.00047, -0.000387),$$

总耗时 55.7 s，剩余燃料质量为 1130.9 kg。主发动机的推力大小、速度和垂直位移变化曲线如下图所示。

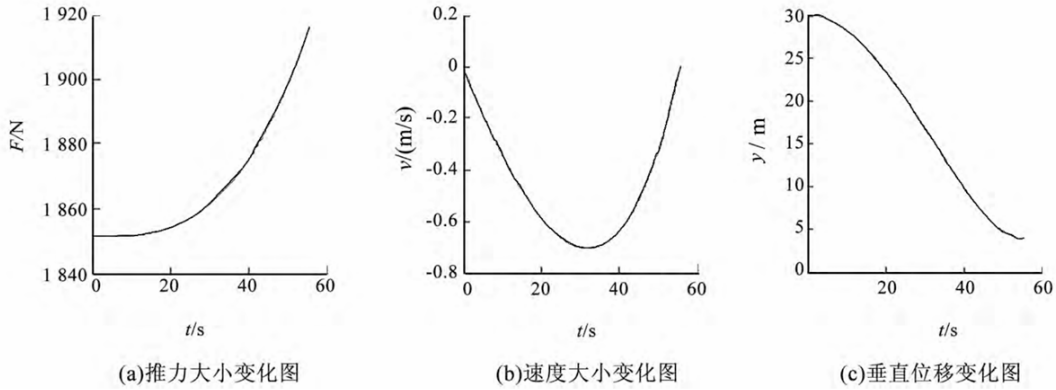


图 14 缓速下降阶段垂直下落过程状态变量的变化图

由图 23 可知，主发动机推力逐步增大，从而使得下落速度先加速后减速，最终速度变为 0，即悬停于距月面 4 m 处，燃料消耗 24.31 kg。

5.9 自由下降阶段的控制模型：

自由下降阶段的起始状态为距月面高度 4 m，主发动机推力为 0，垂直自由落体运动，整个过程耗时大约 1.5686 s，落月速度为 2.55 m s^{-1} 。

六、模型的误差和敏感性分析

嫦娥三号在实际软着陆的控制过程中，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。

6.1 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 的最大推力运行，但事实上发动机的推力会有一些误差，这个误差对运行轨道、速度和高度的影响情况如下图所示。

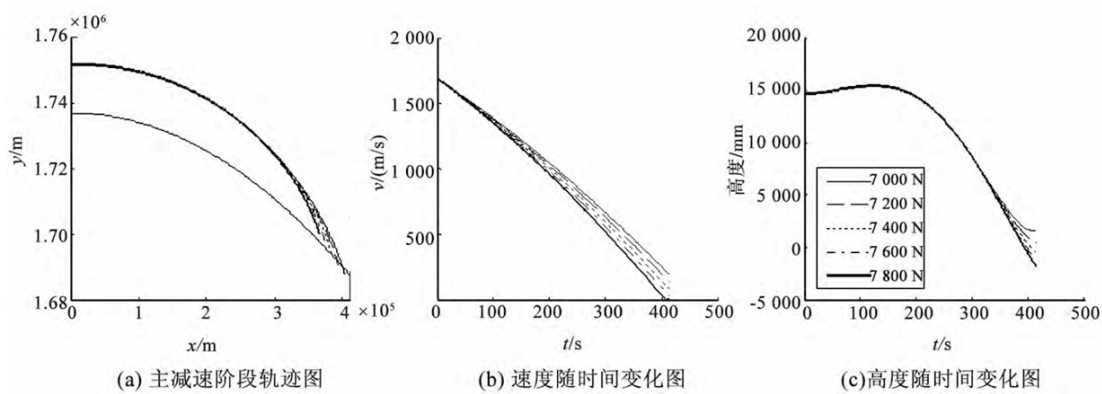


图 15 主发动机最大推力误差的影响分析图

由图可以看出，主发动机的推力越大，使嫦娥三号的速度越早地变为 0；推力越小，使其飞行距离越远。当最大推力误差不太大时，对其运行轨道、速度和高度的影响都不大，只有当推力误差达到500 N 时，会使末端速度差距 200 m s^{-1} 左右，高度差为300 mm 左右。因此，发动机的推力误差对模型的影响是稳定的。

6.2 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，该误差对运行轨道、速度和高度的影响情况如下图所示

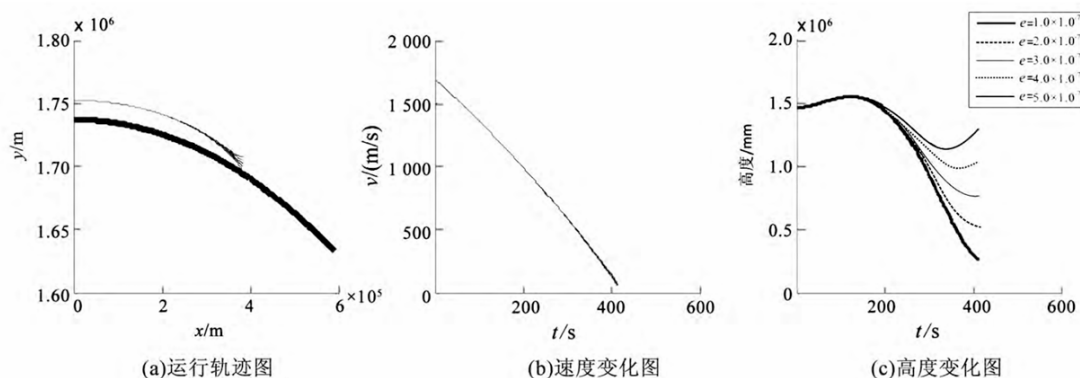


图 16 推力方向角度的控制误差分析图

由图 25 可以看出，嫦娥三号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。

七、模型的评价与推广

7.1 模型的优点

本模型的**优点**在于能够完整刻画嫦娥三号从着陆准备到最终着陆的全过程，阶段划分明确，建模思路清晰，结合了轨道力学、最优控制和多指标决策方法，计算精度高且具有较强的工程可实施性。

7.2 模型的不足

部分阶段的动力学模型仍基于理想化假设（如忽略部分扰动、地形模型分辨率有限），且参数受误差影响的分析仍有简化，需在更高精度环境和更复杂条件下验证。

7.3 模型推广

该方法可适配其他深空探测器的轨道设计与自主着陆任务，并可推广至火星、小行星等不同引力天体的软着陆规划与路径优化中，通过调整引力模型、地形数据与控制约束即可实现移植。

参考文献

- [1] 王大轶, 关轶峰. 月球软着陆制导、导航与控制技术研究 [C]//国防科学技术工业委员会科技与质量司, 中国宇航学会深空探测技术专业委员会. 中国宇航学会深空探测技术专业委员会第二届学术会议论文集. 中国空间技术研究院 502 所; 中国空间技术研究院 502 所, 2005:206-213.
- [2] 孙军伟, 崔平远. 月球软着陆多项式制导控制方法 [J]. 宇航学报, 2007, (05):1171-1174+1218.
- [3] 蔡艳芳. 月球探测器软着陆制导控制方法研究 [D]. 西北工业大学, 2006.
- [4] 唐琼. 月球软着陆轨道快速优化 [J]. 计算机仿真, 2007, (12):24-27.
- [5] 张建辉, 张峰. 月球软着陆轨道优化方法比较研究 [J]. 工程数学学报, 2012, 29(03):355-365.
- [6] 田浩, 白争锋, 赵阳. 月球软着陆器动力学建模研究 [J]. 中国空间科学技术, 2009, 29(06):32-38+44.

附录 A 问题一

```
1 function problem1
2     %% 常量定义
3     GM = 4.9048695e12;          % 月球引力常数 (m^3/s^2)
4     I_sp = 2940.0;              % 比冲 (m/s)
5     F_max = 7500.0;            % 最大推力 (N)
6     m0 = 2400.0;               % 初始质量 (kg)
7     R_moon = 1737013.0;        % 月球半径 (m)
8
9     % 初始条件
10    x0 = 0; y0 = R_moon + 15000;
11    vx0 = 1692.46; vy0 = 0;
12
13    % 终端条件
14    r_target = R_moon + 3000;
15    v_target = 57;
16
17    %% 初始猜测 (调节这里提高可行性)
18    x_init = [120, 0, 0, 0, pi/6]; % [tf, a0, a1, a2, a3]
19
20    %% 变量上下界
21    lb = [50, -0.1, -0.1, -0.01, 0];
22    ub = [200, 0.1, 0.1, 0.1, pi/2];
23
24    %% 优化选项
25    opts = optimoptions('fmincon', ...
26        'Algorithm', 'interior-point', ...
```

```

27     'MaxIterations', 500, ...
28     'OptimalityTolerance', 1e-6, ...
29     'ConstraintTolerance', 1e-6, ...
30     'StepTolerance', 1e-10, ...
31     'Display', 'iter', ...
32     'EnableFeasibilityMode', true, ...
33     'FiniteDifferenceType', 'forward');
34
35 %% 运行优化
36 problem.objective = @(x) objective(x, x0, y0, vx0, vy0, m0, GM, F_max, I_sp);
37 problem.x0 = x_init;
38 problem.lb = lb;
39 problem.ub = ub;
40 problem.nonlcon = @(x) constraints(x, x0, y0, vx0, vy0, m0, GM, F_max, I_sp, r_target, v_target);
41 problem.options = opts;
42 problem.solver = 'fmincon';
43
44 [x_opt, fval, exitflag, output] = fmincon(problem);
45
46 %% 输出结果
47 if exitflag > 0
48     fprintf('优化成功! \n');
49     fprintf('最优飞行时间 tf = %.4f s\n', x_opt(1));
50     fprintf('最优多项式系数 a0=%.6f, a1=%.6f, a2=%.6f, a3=%.6f rad (%.2f deg)\n', ...
51         x_opt(2), x_opt(3), x_opt(4), x_opt(5), rad2deg(x_opt(5)));
52     fprintf('最小燃料消耗 = %.4f kg\n', fval);
53
54 % 终端约束误差
55 [c, ceq] = constraints(x_opt, x0, y0, vx0, vy0, m0, GM, F_max, I_sp, r_target, v_target);
56 fprintf('终端位置误差: %.6f m\n', ceq(1));
57 fprintf('终端速度误差: %.6f m/s\n', ceq(2));
58 else
59     fprintf('优化未收敛, exitflag = %d\n', exitflag);
60     fprintf('输出信息: %s\n', output.message);
61 end
62 end
63
64 %% 目标函数: 最小化燃料消耗
65 function fuel_used = objective(x, x0, y0, vx0, vy0, m0, GM, F_max, I_sp)
66     tf = x(1);
67     a_coeffs = x(2:end);
68     state0 = [x0; y0; vx0; vy0; m0];
69
70     opts = odeset('RelTol', 1e-9, 'AbsTol', 1e-12);
71     [~, Y] = ode113(@(t,s) dynamics(t,s,a_coeffs,GM,F_max,I_sp), [0 tf], state0, opts);
72
73     mf = Y(end,5);
74     fuel_used = m0 - mf;
75 end
76
77 %% 非线性约束函数: 终端状态约束
78 function [c, ceq] = constraints(x, x0, y0, vx0, vy0, m0, GM, F_max, I_sp, r_target, v_target)
79     tf = x(1);
80     a_coeffs = x(2:end);
81     state0 = [x0; y0; vx0; vy0; m0];
82

```

```

83     opts = odeset('RelTol',1e-9,'AbsTol',1e-12);
84     [~, Y] = ode113(@(t,s) dynamics(t,s,a_coeffs,GM,F_max,I_sp), [0 tf], state0, opts);
85
86     xf = Y(end,1);
87     yf = Y(end,2);
88     vxf = Y(end,3);
89     vyf = Y(end,4);
90
91     r_final = sqrt(xf^2 + yf^2);
92     v_final = sqrt(vxf^2 + vyf^2);
93
94     c = []; % 无不等式约束
95     ceq = [r_final - r_target; v_final - v_target];
96 end
97
98 %% 系统动力学方程
99 function dsdt = dynamics(t, s, a_coeffs, GM, F_max, I_sp)
100     x = s(1); y = s(2); vx = s(3); vy = s(4); m = s(5);
101
102     % 推力方向角  $\psi(t)$ 
103     psi = a_coeffs(1)*t^3 + a_coeffs(2)*t^2 + a_coeffs(3)*t + a_coeffs(4);
104
105     % 位置矢量角度  $(t)$ 
106     alpha = atan2(x, y);
107
108     % 重力加速度
109     r = sqrt(x^2 + y^2);
110     g = GM / r^2;
111     gx = -g * sin(alpha);
112     gy = -g * cos(alpha);
113
114     % 推力分量
115     Fx = F_max * sin(psi);
116     Fy = F_max * cos(psi);
117
118     % 速度大小
119     v = sqrt(vx^2 + vy^2);
120
121     dxdt = vx;
122     dydt = vy;
123     dvxdt = gx + Fx / m;
124     dvydt = gy + Fy / m;
125     dmdt = -v / I_sp;
126
127     dsdt = [dxdt; dydt; dvxdt; dvydt; dmdt];
128 end

```

附录 B 问题二

```

1
2 A=imread("2014年国赛A题附件 (3).tif");
3 image(A);
4 % 读取 tif 文件

```

```

5 [Z, R] = readgeoraster("2014年国赛A题附件 (3).tif"); % Z是高程矩阵, R是空间参考
6 Z = double(Z); % 转成double便于计算
7
8 % 生成 X、Y 网格 (假设每像素分辨率是 1m)
9 [xGrid, yGrid] = meshgrid(1:size(Z,2), 1:size(Z,1));
10
11 % 绘制三维地形
12 figure;
13 surf(xGrid, yGrid, Z, 'EdgeColor', 'none'); % 无网格线
14 colormap('parula');
15 colorbar; % 添加颜色条
16 axis tight; % 紧凑坐标
17 xlabel('X (m)');
18 ylabel('Y (m)');
19 zlabel('高度 (m)');
20 title('数字高程图 (三维视图)');
21 view(-37.5,50); % 调整视角
22 shading interp; % 平滑颜色
23
24 % 将矩阵转成一列向量
25 elevVals = Z(:);
26
27 % 设置聚类数 k (低洼区 + 其他)
28 k = 2;
29
30 % 使用 k-means 聚类
31 [idx, C] = kmeans(elevVals, k);
32
33 % 找到低洼类别 (均值最小的类)
34 [~, lowCluster] = min(C);
35
36 % 将聚类结果重塑为原始图像大小
37 craterMask = reshape(idx == lowCluster, size(Z));
38
39 % 显示原始高程
40 figure;
41 subplot(1,2,1);
42 imagesc(Z);
43 axis image;
44 colorbar;
45 title('原始数字高程图 (m)');
46
47 % 显示陨石坑位置
48 subplot(1,2,2);
49 imagesc(craterMask);
50 axis image;
51 colorbar;
52 title('K-means识别的低洼区 (陨石坑候选)');
53 colormap(subplot(1,2,2), [1 1 1; 0 0 0]); % 白色背景, 黑色表示坑
54
55 % 嫦娥三号粗避障段 - 陨石坑识别与分析
56 % 读取数字高程数据
57 clc; clear; close all;
58
59 %% 1. 数据读取与预处理
60 try

```

```

61 % 方法1: 使用readgeoraster (MATLAB R2019a及以后版本)
62 [Z, R] = readgeoraster("2014年国赛A题附件 (3).tif");
63 Z = double(Z);
64 catch
65 % 方法2: 使用imread作为备选
66 Z = imread("2014年国赛A题附件 (3).tif");
67 Z = double(Z);
68 end
69
70 % 获取数据基本信息
71 [rows, cols] = size(Z);
72 fprintf('数字高程图尺寸: %d × %d 像素\n', rows, cols);
73 fprintf('覆盖范围: %.1f × %.1f 米\n', cols, rows);
74 fprintf('高程范围: %.1f - %.1f 米\n', min(Z(:)), max(Z(:)));
75
76 % 生成坐标网格 (每像素1米分辨率)
77 [xGrid, yGrid] = meshgrid(1:cols, 1:rows);
78
79 %% 2. 三维地形可视化
80 figure('Position', [100, 100, 1200, 400]);
81
82 % 子图1: 三维地形图
83 subplot(1,3,1);
84 surf(xGrid, yGrid, Z, 'EdgeColor', 'none');
85 colormap('parula');
86 colorbar;
87 axis tight;
88 xlabel('X (m)');
89 ylabel('Y (m)');
90 zlabel('高度 (m)');
91 title('数字高程图 (三维视图)');
92 view(-37.5, 50);
93 shading interp;
94
95 % 子图2: 二维高程图
96 subplot(1,3,2);
97 imagesc(Z);
98 axis image;
99 colorbar;
100 title('数字高程图 (俯视图)');
101 xlabel('X (m)');
102 ylabel('Y (m)');
103
104 %% 3. K-means聚类分析
105 % 将高程数据转换为列向量, 去除NaN值
106 elevVals = Z(:);
107 validIdx = ~isnan(elevVals);
108 elevVals = elevVals(validIdx);
109
110 % 设置聚类数
111 k = 5; % 低洼区 (陨石坑) + 正常地形
112
113 % 执行K-means聚类
114 fprintf('\n正在执行K-means聚类分析...\n');
115 [idx, C] = kmeans(elevVals, k);
116

```

```

117 % 显示聚类中心
118 fprintf('聚类中心值:\n');
119 for i = 1:k
120     fprintf('  聚类 %d: %.2f 米\n', i, C(i));
121 end
122
123 % 确定低洼区类别（高程值较小的类）
124 [~, lowCluster] = min(C);
125 [~, highCluster] = max(C);
126
127 fprintf('\n聚类结果分析:\n');
128 fprintf('  低洼区（陨石坑候选）聚类: %d（高程均值: %.2f 米）\n', lowCluster, C(lowCluster));
129 fprintf('  正常地形聚类: %d（高程均值: %.2f 米）\n', highCluster, C(highCluster));
130
131 %% 4. 生成陨石坑掩码
132 % 重建完整的聚类索引矩阵
133 fullIdx = nan(size(Z));
134 fullIdx(validIdx) = idx;
135
136 % 创建陨石坑掩码
137 craterMask = (fullIdx == lowCluster);
138
139 % 计算陨石坑统计信息
140 craterPixels = sum(craterMask(:));
141 craterArea = craterPixels; % 平方米（1像素=1m2）
142 craterRatio = craterPixels / (rows * cols) * 100;
143
144 fprintf('\n\n陨石坑识别统计:\n');
145 fprintf('  陨石坑像素数: %d\n', craterPixels);
146 fprintf('  陨石坑面积: %.0f 平方米\n', craterArea);
147 fprintf('  陨石坑面积占比: %.2f%%\n', craterRatio);
148
149 % 嫦娥三号粗避障段 - 陨石坑识别与分析
150 % 读取数字高程数据
151 clc; clear; close all;
152
153 %% 1. 数据读取与预处理
154 try
155     % 方法1: 使用readgeoraster（MATLAB R2019a及以后版本）
156     [Z, R] = readgeoraster("2014年国赛A题附件（3）.tif");
157     Z = double(Z);
158 catch
159     % 方法2: 使用imread作为备选
160     Z = imread("2014年国赛A题附件（3）.tif");
161     Z = double(Z);
162 end
163
164 % 获取数据基本信息
165 [rows, cols] = size(Z);
166 fprintf('数字高程图尺寸: %d × %d 像素\n', rows, cols);
167 fprintf('覆盖范围: %.1f × %.1f 米\n', cols, rows);
168 fprintf('高程范围: %.1f - %.1f 米\n', min(Z(:)), max(Z(:)));
169
170 % 生成坐标网格（每像素1米分辨率）
171 [xGrid, yGrid] = meshgrid(1:cols, 1:rows);
172

```

```

173 %% 2. 三维地形可视化
174 figure('Position', [100, 100, 1200, 400]);
175
176 % 子图1: 三维地形图
177 subplot(1,3,1);
178 surf(xGrid, yGrid, Z, 'EdgeColor', 'none');
179 colormap('parula');
180 colorbar;
181 axis tight;
182 xlabel('X (m)');
183 ylabel('Y (m)');
184 zlabel('高度 (m)');
185 title('数字高程图 (三维视图)');
186 view(-37.5, 50);
187 shading interp;
188
189 % 子图2: 二维高程图
190 subplot(1,3,2);
191 imagesc(Z);
192 axis image;
193 colorbar;
194 title('数字高程图 (俯视图)');
195 xlabel('X (m)');
196 ylabel('Y (m)');
197
198 %% 3. K-means聚类分析 - 识别陨石坑
199 % 将高程数据转换为列向量, 去除NaN值
200 elevVals = Z(:);
201 validIdx = ~isnan(elevVals);
202 elevVals = elevVals(validIdx);
203
204 % 首先用肘部法则确定最优陨石坑数量
205 k_range = 2:10;
206 sse = zeros(size(k_range));
207
208 fprintf('\n正在分析最优陨石坑聚类数量...\n');
209 for i = 1:length(k_range)
210     [idx_temp, C_temp] = kmeans(elevVals, k_range(i), 'Replicates', 5);
211     % 计算SSE
212     sse_temp = 0;
213     for j = 1:k_range(i)
214         cluster_data = elevVals(idx_temp == j);
215         sse_temp = sse_temp + sum((cluster_data - C_temp(j)).^2);
216     end
217     sse(i) = sse_temp;
218 end
219
220 % 使用肘部法则选择最优k值
221 diff1 = diff(sse);
222 diff2 = diff(diff1);
223 [~, elbow_idx] = max(abs(diff2));
224 k = k_range(elbow_idx + 1);
225
226 fprintf('根据肘部法则, 最优陨石坑聚类数量: %d\n', k);
227
228 % 执行K-means聚类

```



```

229 fprintf('正在执行K-means聚类分析...\n');
230 [idx, C] = kmeans(elevVals, k, 'Replicates', 5);
231
232 % 显示聚类中心 (按高程从低到高排序)
233 [sortedC, sortIdx] = sort(C);
234 fprintf('\n聚类中心值 (按高程排序):\n');
235 for i = 1:k
236     fprintf('  陨石坑 %d: %.2f 米\n', i, sortedC(i));
237 end
238
239 fprintf('\n识别到 %d 个不同深度的陨石坑区域\n', k);
240
241 %% 4. 生成多层陨石坑掩码
242 % 重建完整的聚类索引矩阵
243 fullIdx = nan(size(Z));
244 fullIdx(validIdx) = idx;
245
246 % 重新映射聚类索引以按高程排序
247 remappedIdx = fullIdx;
248 for i = 1:k
249     remappedIdx(fullIdx == sortIdx(i)) = i;
250 end
251
252 % 创建不同深度的陨石坑掩码
253 fprintf('\n各陨石坑区域统计:\n');
254 craterMasks = cell(k, 1);
255 totalCraterArea = 0;
256
257 for i = 1:k
258     craterMasks{i} = (remappedIdx == i);
259     craterPixels = sum(craterMasks{i}(:));
260     craterArea = craterPixels; % 平方米 (1像素=1m²)
261     craterRatio = craterPixels / (rows * cols) * 100;
262     totalCraterArea = totalCraterArea + craterArea;
263
264     fprintf('  陨石坑区域 %d (深度%.1fm): %d像素, %.0f平方米, %.2f%%\n', ...
265         i, sortedC(i), craterPixels, craterArea, craterRatio);
266 end
267
268 fprintf('总陨石坑覆盖面积: %.0f 平方米 (%.2f%%)\n', totalCraterArea, totalCraterArea/(rows*cols)*100);
269
270 %% 5. 可视化聚类结果
271 % 子图3: 多层陨石坑分布
272 subplot(1,3,3);
273 imagesc(remappedIdx);
274 axis image;
275 colorbar;
276 title(sprintf('识别的%d类陨石坑区域', k));
277 xlabel('X (m)');
278 ylabel('Y (m)');
279 colormap(gca, jet(k));
280
281 %% 6. 详细的多层陨石坑可视化
282 figure('Position', [900, 100, 1000, 800]);
283
284 % 显示各层陨石坑

```

```

285 subplot_rows = ceil(sqrt(k+1));
286 subplot_cols = ceil((k+1)/subplot_rows);
287
288 % 原始地形
289 subplot(subplot_rows, subplot_cols, 1);
290 imagesc(Z);
291 axis image;
292 colorbar;
293 title('原始数字高程图');
294
295 % 各层陨石坑
296 for i = 1:k
297     subplot(subplot_rows, subplot_cols, i+1);
298     imagesc(craterMasks{i});
299     axis image;
300     colorbar;
301     title(sprintf('陨石坑区域%d (%.1fm)', i, sortedC(i)));
302     colormap(gca, [1 1 1; 0.8 0.2 0.2]);
303 end
304
305 % 所有陨石坑叠加图
306 if k+1 <= subplot_rows * subplot_cols
307     subplot(subplot_rows, subplot_cols, k+2);
308     imagesc(Z);
309     hold on;
310     colors = jet(k);
311     for i = 1:k
312         contour(craterMasks{i}, [0.5 0.5], 'Color', colors(i,:), 'LineWidth', 2);
313     end
314     axis image;
315     colorbar;
316     title('所有陨石坑叠加图');
317     legend_labels = cell(k, 1);
318     for i = 1:k
319         legend_labels{i} = sprintf('区域%d', i);
320     end
321     legend(legend_labels, 'Location', 'best');
322 end
323
324 %% 7. 输出详细统计报告
325 fprintf('\n=== 嫦娥三号粗避障段分析报告 ===\n');
326 fprintf('分析区域: 2300×2300米\n');
327 fprintf('分辨率: 1米/像素\n');
328 fprintf('高程范围: %.1f - %.1f 米\n', min(Z(:)), max(Z(:)));
329 fprintf('\nK-means聚类参数:\n');
330 fprintf('  陨石坑聚类数量k: %d\n', k);
331 fprintf('  聚类中心值:\n');
332 for i = 1:k
333     fprintf('    区域%d: %.2f 米\n', i, sortedC(i));
334 end
335
336 fprintf('\n陨石坑识别结果:\n');
337 fprintf('  总陨石坑覆盖面积: %.0f 平方米 (%.2f%%)\n', totalCraterArea, totalCraterArea/(rows*cols)*100);
338 fprintf('  识别的陨石坑区域层次: %d 层\n', k);
339
340 craterDensity = totalCraterArea/(rows*cols)*100;

```

```

341 fprintf('\n建议着陆策略:\n');
342 if craterDensity > 30
343     fprintf('    陨石坑密度很高(%.1f%%), 强烈建议重新选择着陆区域\n', craterDensity);
344 elseif craterDensity > 15
345     fprintf('    陨石坑密度较高(%.1f%%), 需要精细避障, 建议降低下降速度\n', craterDensity);
346 else
347     fprintf('    陨石坑密度适中(%.1f%%), 可以继续精细避障\n', craterDensity);
348 end
349
350 fprintf('  建议避开高程在 %.1f - %.1f 米的区域\n', sortedC(1), sortedC(min(2,k)));
351 fprintf('  推荐着陆区域高程: %.1f 米以上\n', sortedC(max(k-1,1)));

```