

月面应急直接上升交会轨迹快速规划方法

班焕恒^{1,2}, 周聪^{1,2}, 闫晓东^{1,2,*}

(1. 西北工业大学 航天学院, 西安 710072; 2. 陕西省空天飞行器设计重点实验室, 西安 710068)

摘 要: 当月面探测器出现某些突发状况而需要返回环月轨道器或地球时, 上升器需具备自主规划出合适的应急上升交会轨迹的能力。以上升交会时间最短为目标函数, 基于序列二阶锥规划方法, 建立直接上升交会轨迹优化模型, 并构建求解直接上升交会轨迹优化问题的凸优化算法。为提高计算效率, 对内点法进行定制化改进, 主要改进有: 线性方程组求解过程定制化改进; 内点法热启动; 二阶锥规划子问题求解精度动态调整。仿真结果表明: 所提月面应急上升交会轨迹优化方法可实现上升器的快速上升交会。与采用通用内点法求解器的传统序列二阶锥规划方法相比, 在求解精度不变的情况下, 所提的序列二阶锥规划问题求解加速方法可达到 9.5 倍左右的加速比。综合运用所提方法, 有望实现月面上升器的自主在线轨迹规划。

关 键 词: 月面应急上升交会; 序列二阶锥规划方法; 内点法定制化; 内点法热启动; 在线轨迹规划

中图分类号: V448.2

文献标志码: A

文章编号: 1001-5965(2025)04-1352-15

随着科学技术的发展, 人类对于太空的探索愈加频繁, 实施月球探测工程具有经济、科技和国家发展战略等方面的重大意义^[1]。由于月面工作环境较为恶劣, 探测器在进行采样或探测活动时可能会出现某些突发状况而需要应急返回的情况, 此时, 上升器需要自主规划出合适的应急上升交会轨迹。

对于上升器自主上升交会的研究可分为上升段的研究及交会段的研究。对于上升段轨迹优化的研究, 通常采用的是传统优化方法, 即间接法和直接法。为减少月面上升的燃料消耗量, 文献 [2-4] 基于极大/极小值推导了月面上升段燃料最优或次优的制导律, 但基于极大/极小值原理的间接法对于初始猜想较为敏感, 通过间接法求解轨迹优化问题的局限性较大。相较于间接法, 直接法在优化问题的转化及求解上更具通用性。在基于直接法的上升段轨迹优化中, 研究人员分别基于序列二次规划方法^[5-6]和序列凸优化方法^[7-9]展开了大量的研究。相比于序列二次规划方法, 基于凸优化的方法由于

其计算的快速性而受到广泛的关注, 但受限于箭载或嵌入式计算机的性能, 序列凸优化问题的求解效率仍制约着该方法实时在线应用。

交会段的轨迹优化通常采用智能优化方法及传统优化方法。智能优化方法主要应用于脉冲交会问题的优化^[10-12], 对于应急返回情况下的连续推力上升交会适用性较差。传统优化方法主要包括间接法和直接法, 基于间接法的交会段轨迹优化^[13-14]对于初值的依赖性较大, 文献 [15-16] 对于不同的初值计算方法进行了深入的研究, 但相应的方法使得轨迹优化问题的复杂度与计算耗时均有所增加。在基于直接法的交会段轨迹优化中, 研究人员应用了序列二次规划方法^[17]、非线性规划法^[18]及序列凸优化方法^[19-23]对交会段的轨迹优化展开了深入的研究, 但相关的研究主要聚焦于优化模型的建立, 未对优化问题的求解过程进行研究, 因而相应的优化问题求解效率较低。

对于月面应急上升交会的轨迹优化问题, 如果

收稿日期: 2023-04-04; 录用日期: 2023-07-11; 网络出版时间: 2023-07-31 09:25
网络出版地址: link.cnki.net/urlid/11.2625.V.20230728.1648.002
* 通信作者. E-mail: yan804@nwpu.edu.cn

引用格式: 班焕恒, 周聪, 闫晓东. 月面应急直接上升交会轨迹快速规划方法 [J]. 北京航空航天大学学报, 2025, 51 (4): 1352-1366.
BAN H H, ZHOU C, YAN X D. Rapid planning method for lunar direct ascent and rendezvous trajectory in emergency cases [J]. Journal of Beijing University of Aeronautics and Astronautics, 2025, 51 (4): 1352-1366 (in Chinese).

仍将上升段与交会段分开研究, 则由于未能考虑上升段和交会段的相互影响, 上升交会耗时长, 在应急返回情况下不适用。目前对于月面上升交会联合轨迹优化的研究较少, 文献 [24] 提出了基于高斯伪谱法与序列二次规划的两阶段联合轨迹优化方法, 但相应的方法求解耗时极长, 难以实现在线应用。

在基于序列凸优化的上升与交会轨迹优化方法中, 目前的研究主要聚焦于优化模型的建立与离散方法的选择上, 而在求解相应的优化问题时, 采用的多为通用求解器, 如 SeDuMi^[25]、ECOS^[26] 等。尽管成熟的通用求解器极大方便了凸优化方法研究, 但在应用于工程实践时存在较多问题: 一方面通用求解器针对通用问题开发, 其算法复杂且未对具体问题进行了特定优化, 导致其计算速度通常较慢, 难以实现在线应用; 另一方面, 复杂的代码结构使其难以转化成为嵌入式或箭载计算机使用的代码。

为提高凸优化算法的计算速度, 研究人员对内点法的代码自动生成定制化方法^[27-28] 展开了深入的研究, 但该方法仅对小规模的问题加速效果较好。而且, 受限于嵌入式计算机的储存空间限制, 基于代码自动生成的定制化方法在嵌入式平台上的应用会受到较多限制。随着 GPU 计算加速方法的发展, 利用 CPU-GPU 异构平台对内点法计算过程进行并行加速也受到人们的关注^[29], 但相应的方法仅适用于大规模的优化问题。对于内点法计算过程中的方程组求解问题, 文献 [30] 通过降低方程组求解精度的方法提高计算速度, 但此方法可能会影响内点法最终收敛精度, 因此, 仅对于精度要求较低的问题适用。

除了传统的内点法求解器外, 近年来快速发展的一阶算法为凸优化问题求解提供了更多的选择, 如基于交替方向乘子法 (alternating direction method of multipliers, ADMM) 算法^[31] 的分裂锥求解器 (splitting conic solver, SCS)^[32] 及比例积分投影梯度 (proportional-integral projected gradient, PIPG) 算法^[33] 等。相比于内点法, 一阶算法的优点在于容易实现热启动, 但总迭代次数较多, 当要求的精度较高时, 内点法在计算速度上仍具有明显的优势。而且, 内点法的热启动技术近年来也得到较大突破, 文献 [34] 提出了一种内点法的热启动方法, 该方法将前一个优化问题的最优解与冷启动值的组合作为热启动初值, 以加快内点法计算速度。为减少序列二阶锥规划方法的内点法总迭代次数, 文献 [35] 进一步将热启动初值的计算扩展为前一个优化问题的非最优解与冷启动值的组合。文献 [36] 基于二阶锥规划热启动方法提出了一种火星垂直着陆的模型预

测控制方法, 但该方法在求解初始优化问题时仍需要根据传统内点法计算高精度解。针对大气层内的轨迹优化问题, 文献 [37] 对内点法中 KKT 矩阵进行稀疏化处理, 同时结合热启动方法, 显著加快了序列二阶锥规划方法的求解速度, 但文中引入了额外的轨迹积分方法作为序列二阶锥规划方法终止条件的判断, 而这又增加了相应的计算耗时。

综上所述, 本文基于序列二阶锥规划方法, 提出了一种月面应急上升交会联合轨迹优化方法, 并对内点法进行改进以提高优化问题的求解效率。主要的改进策略有: ①综合应用定制化、并行及动态精度调整的方法提高内点法 KKT 方程组的求解速度; ②采用热启动策略, 加快内点法的收敛速度; ③设计子问题求解精度动态调整策略降低内点法求解的总迭代次数。仿真结果表明, 本文月面应急上升交会轨迹优化方法可实现上升器与轨道器的短时间上升交会。同时, 本文提出的序列二阶锥规划问题求解加速方法相比通用求解器可实现 9.5 倍左右的计算速度提升, 综合运用本文方法, 有望实现上升器的自主在线轨迹规划。

1 月面应急上升交会轨迹规划问题

本文在月心球面坐标系下建立轨迹优化模型, 月心球面坐标系定义如图 1 所示。图中: 月心球面坐标系的原点 O 在月心, R 为月心到飞行器连线 OS 的长度; Ox 轴在交会轨道面内, 指向上升器位置在交会轨道面内投影的方向; Oy 轴在轨道面内, 指向轨道器经过 Ox 轴时的方向; Oz 轴垂直于交会轨道面, 与 Ox 轴和 Oy 轴成右手坐标系, $Oxyz$ 在上升器起飞后不动; θ 为 x 轴按逆时针方向旋转到飞行器在轨道面内的投影所转过的角度; ϕ 为 OS 与轨道面的夹角。定义单位矢量方向 $[e_r, e_t, e_p]^T$ 如下: e_r 沿 OS 的方向; e_p 在平面 zOS 内, 垂直于 e_r , 指向 ϕ 增大的方向; e_t 垂直于 OS , 指向 θ 增大的方向, 且与 e_r 和 e_p 构成直角坐标系。

本文的月面应急上升交会方案如图 2 所示。图中: 在发生应急状况时, 上升器在采样区等待, 当

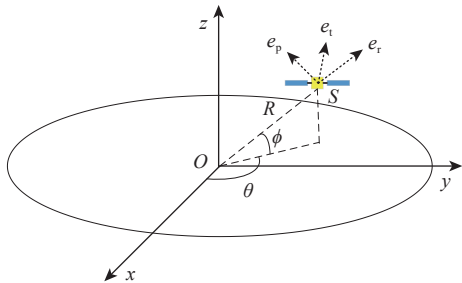


图 1 月心球面坐标系

Fig. 1 Lunar center spherical coordinate system

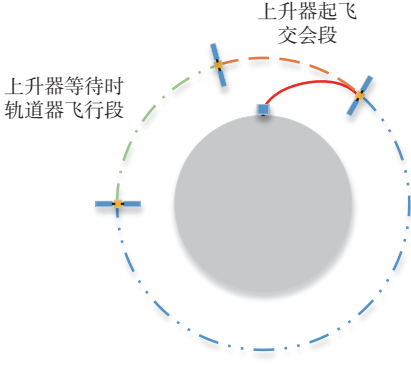


图2 月面应急上升交会示意图

Fig. 2 Lunar ascent and rendezvous process in emergency cases

轨道器飞行到合适的位置时,上升器起飞直接与轨道器交会。

在月心球面坐标系下建立上升器的动力学方程组如下:

$$\begin{cases} \dot{R}(t) = V_r(t) \\ \dot{\theta}(t) = \frac{V_t(t)}{R(t) \cos \phi(t)} \\ \dot{\phi}(t) = \frac{V_p(t)}{R(t)} \\ \dot{V}_r(t) = \frac{V_r^2(t)}{R(t)} + \frac{V_p^2(t)}{R(t)} - \frac{\mu_m}{R^2(t)} + P \frac{u_r(t)}{m(t)} \\ \dot{V}_t(t) = -\frac{V_r(t)V_t(t)}{R(t)} + \frac{V_t(t)V_p(t) \tan \phi(t)}{R(t)} + P \frac{u_t(t)}{m(t)} \\ \dot{V}_p(t) = -\frac{V_r(t)V_p(t)}{R(t)} - \frac{V_t^2(t) \tan \phi(t)}{R(t)} + P \frac{u_p(t)}{m(t)} \\ \dot{m} = -P / (I_{sp} g_0) \end{cases} \quad (1)$$

式中: R 为径向距离; θ 为相角; ϕ 为极角; V_r 、 V_t 和 V_p 为上升器的速度在月心球坐标系下的各分量; μ_m 为月球引力常量; P 为上升器的推力; u_r 、 u_t 和 u_p 为上升器推力方向在月心球坐标系下的各分量; m 为上升器的质量; I_{sp} 为上升器发动机比冲; g_0 为地球引力加速度。

本文假设轨道器在上升器到达交会轨道前,一直处于其停泊轨道上,且轨道器的停泊轨道为圆轨道。在月心球面坐标系下建立轨道器的动力学方程组,在上升器的上升交会阶段,轨道器仅有相角发生改变,即

$$\dot{\theta}_0 = \omega_0 \quad (2)$$

式中: θ_0 为轨道器的相角; ω_0 为轨道器在其停泊轨道上的角速度。

上升器的初始状态约束如下:

$$\begin{cases} R(t_0) = R_0, \theta(t_0) = \theta_0, \phi(t_0) = \phi_0 \\ V_r(t_0) = V_{r0}, V_t(t_0) = V_{t0}, V_p(t_0) = V_{p0} \\ m(t_0) = m_0 \end{cases} \quad (3)$$

式中: R_0 为初始径向距离; θ_0 为初始相角; ϕ_0 为初始

极角; V_{r0} 、 V_{t0} 和 V_{p0} 为上升器的初始速度在月心球坐标系下的各分量; m_0 为上升器初始质量。上升器与轨道器的交会约束可表示如下:

$$\begin{cases} R(t_f) = R_f, \theta(t_f) = \theta_{f0} + \omega_0 t_f, \phi(t_f) = 0 \\ V_r(t_f) = 0, V_t(t_f) = V_f, V_p(t_f) = 0 \\ m(t_f) \geq m_{dry} \end{cases} \quad (4)$$

式中: t_f 为终端交会时间; R_f 为交会轨道的月心距; θ_{f0} 为轨道器初始相角; $V_f = \sqrt{\mu_m / R_f}$ 表示交会速度; m_{dry} 为上升器结构和载荷质量。

为避免上升器撞到月球表面,对月心距 R 作如下约束:

$$R(t) \geq R_m \quad (5)$$

式中: R_m 为月球半径。上升器推力大小约束如下:

$$\sqrt{u_r^2(t) + u_t^2(t) + u_p^2(t)} = 1 \quad (6)$$

为方便求解轨迹优化问题,对式(6)的控制约束作如下松弛:

$$\sqrt{u_r^2(t) + u_t^2(t) + u_p^2(t)} \leq 1 \quad (7)$$

松弛的等价性可参考文献[9]。

由此可得到月面应急上升交会轨迹优化模型的连续轨迹优化问题 (optimal continuous problem, OCP) 如下:

$$\begin{cases} \min J = t_f \\ \text{s.t. 式(1)、式(3)~式(5)、式(7)} \end{cases} \quad (8)$$

2 月面应急上升交会离散优化问题

由于上升器所携带的燃料较少,而主发动机无法二次启动,上升器仅由主发动机无法完成直接上升交会。因此,本文将直接上升交会分为2段,第1段为主发动机工作段,第2段为推力器工作段。

在各飞行阶段,上升器的推力大小不变,因此,质量仅与时间相关,本文以各段飞行时间为变量,即可略去状态量中的质量变量。为表达的简洁,取 $X(t) = [R(t), \theta(t), \phi(t), V_r(t), V_t(t), V_p(t)]^T$ 代表状态量, $U(t) = [u_r(t), u_t(t), u_p(t)]^T$ 代表控制量, $f(X(t), U(t))$ 代表式(1)的动力学方程组的右端项。假设 X_1 和 U_1 为第1段的状态量和控制量, X_2 和 U_2 为第2段的状态量和控制量。

由于 OCP 为无限维的非线性优化问题,为对 OCP 快速求解,需对其进行离散化和凸化处理。

2.1 主发动机工作段

假设主发动机工作段的时间为 t_{1f} , 离散点数为 N_1 , 则离散时间步长 Δt_1 表示如下:

$$\Delta t_1 = t_{1f} / (N_1 - 1) \quad (9)$$

为保证离散精度和优化问题的稀疏性, 本文采用梯形离散法对动力学方程组进行离散化处理。则主发动机工作段的动力学方程组在经过梯形离散后可表示为

$$\begin{aligned} \mathbf{X}_1(k+1) = \mathbf{X}_1(k) + \\ \frac{1}{2} \Delta t_1 (\mathbf{f}(\mathbf{X}_1(k), \mathbf{U}_1(k)) + \mathbf{f}(\mathbf{X}_1(k+1), \mathbf{U}_1(k+1))) \end{aligned} \quad (10)$$

式中: $\mathbf{X}_1(k) = \mathbf{X}_1(t_{1(k)})$; $\mathbf{U}_1(k) = \mathbf{U}_1(t_{1(k)})$ 。为表达的简洁, 将 $\mathbf{f}(\mathbf{X}_1(k), \mathbf{U}_1(k))$ 简写为 $\mathbf{f}_1(k)$ 。

由于 $\mathbf{f}_1(k)\Delta t_1$ 为非线性方程组, 需要对其进行泰勒展开线性化处理, 即

$$\begin{aligned} \mathbf{F}_1(k) = \mathbf{f}_1^0(k)\Delta t_1^0 + \left. \frac{\partial \mathbf{f}_1(k)}{\partial \mathbf{X}_1(k)} \right|_{\mathbf{X}_1^0(k)} (\mathbf{X}_1(k) - \mathbf{X}_1^0(k)) + \\ \left. \frac{\partial \mathbf{f}_1(k)}{\partial \mathbf{U}_1(k)} \right|_{\mathbf{U}_1^0(k)} (\mathbf{U}_1(k) - \mathbf{U}_1^0(k)) + \mathbf{f}_1^0(k)(\Delta t_1 - \Delta t_1^0) \end{aligned} \quad (11)$$

式中: $(\mathbf{X}_1^0, \mathbf{U}_1^0, \Delta t_1^0)$ 为参考轨迹; $\mathbf{f}_1^0(k)$ 为第 k 个参考点处的动力学方程组矢量。在求 $\mathbf{f}_1(k)$ 对 $\mathbf{X}_1(k)$ 的偏导数时, 由于 R 和 ϕ 的变化量较小, 因此, 忽略对该两项的求导, 直接取参考轨迹结果计算即可。

为避免由于初始猜想的不精确性及线性化处理带来的伪不可行现象, 对式 (11) 加入虚拟控制量, 即

$$\begin{cases} \mathbf{F}_1(k) = \mathbf{f}_1^0(k)\Delta t_1^0 + \left. \frac{\partial \mathbf{f}_1(k)}{\partial \mathbf{X}_1(k)} \right|_{\mathbf{X}_1^0(k)} (\mathbf{X}_1(k) - \mathbf{X}_1^0(k)) + \\ \left. \frac{\partial \mathbf{f}_1(k)}{\partial \mathbf{U}_1(k)} \right|_{\mathbf{U}_1^0(k)} (\mathbf{U}_1(k) - \mathbf{U}_1^0(k)) + \\ \mathbf{f}_1^0(k)(\Delta t_1 - \Delta t_1^0) + \mathbf{a}_{R1}(k) \end{cases} \quad (12)$$

在加入虚拟控制量 $\mathbf{a}_{R1}(k)$ 后, 应在目标函数中加入对该项的罚函数, 以避免 $\mathbf{a}_{R1}(k)$ 对结果的影响, 即

$$\begin{cases} \min (N_1 - 1)\Delta t_1 + \gamma_{ar1} \sum_{k=1}^{N_1} \kappa_{ar1k} \\ \text{s.t. } \mathbf{X}_1(k+1) - \mathbf{X}_1(k) = \frac{1}{2}(\mathbf{F}_1(k) + \mathbf{F}_1(k+1)) \\ \|\mathbf{a}_{R1}(k)\|_2 \leq \kappa_{ar1k} \end{cases} \quad (13)$$

式中: κ_{ar1k} 为虚拟控制量的约束变量; γ_{ar1} 为虚拟控制量的罚函数系数。为避免虚拟控制量的加入引起的优化问题无界的现象, 应对性能指标相关的变量加入信赖域约束, 即

$$\varpi_{1\Delta t_1} \Delta t_1^0 \leq \Delta t_1 \leq \varpi_{u\Delta t_1} \Delta t_1^0 \quad (14)$$

式中: $\varpi_{1\Delta t_1}$ 和 $\varpi_{u\Delta t_1}$ 为时间信赖域约束的系数。

对于姿态角变化率的约束, 本文假设推力方向沿着上升器的纵轴方向, 则姿态角变化率的约束可

转换为对推力方向的约束^[38], 如图3所示。

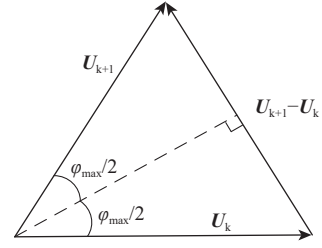


图3 推力方向几何示意图

Fig. 3 Geometry of thrust direction

图3中: φ_{max} 为离散时间步长内的最大姿态变化角。由图3的几何关系可知:

$$\begin{cases} \frac{1}{2} \|U(k+1) - U(k)\|_2 = \sin\left(\frac{\varphi_{max}}{2}\right) \\ \varphi_{max} = \frac{1}{2} \int_{t_k}^{t_{k+1}} \omega_{max} dt \end{cases} \quad (15)$$

式中: ω_{max} 为上升器的最大姿态角变化率。则姿态角变化率可表示如下:

$$\|U(k+1) - U(k)\|_2 \leq 2 \sin\left(\frac{1}{2} \omega_{max} \Delta t^0\right) \quad (16)$$

由于本文是在球坐标系下建立的优化模型, 球坐标系随着上升器位置的变化而变化, 因此, 以式 (16) 约束姿态角会由于坐标系的变化而存在偏差。但由于前后2个离散点之间的时间很短, 即前后2个离散点之间的坐标系偏差很小, 因此, 坐标系变化引起的姿态角约束偏差可以忽略。

为后续表达的简洁, 取 $\mathbf{G}_1(k)$ 表示不等式约束, 即

$$(\mathbf{G}_1(k) \leq \mathbf{0}): \begin{cases} R_1(k) \geq R_m \\ \sqrt{u_{r1}^2(k) + u_{rl}^2(k) + u_{p1}^2(k)} \leq 1 \\ \varpi_{1\Delta t_1} \Delta t_1^0 \leq \Delta t_1 \leq \varpi_{u\Delta t_1} \Delta t_1^0 \\ \|\mathbf{a}_{R1}(k)\|_2 \leq \kappa_{ar1k} \\ \|U_1(k+1) - U_1(k)\|_2 \leq 2 \sin\left(\frac{1}{2} \omega_{1max} \Delta t_1^0\right) \end{cases} \quad (17)$$

式中: ω_{1max} 为主发动机工作段的最大姿态角变化率。

2.2 推力器工作段

假设推力器工作段的时间为 t_{2f} , 离散点数为 N_2 , 则离散时间步长 Δt_2 表示如下:

$$\Delta t_2 = t_{2f} / (N_2 - 1) \quad (18)$$

与主发动机工作段类似, 动力学方程组右端项 $\mathbf{F}_2$ 如下:

$$\begin{aligned} \mathbf{F}_2(k) = \mathbf{f}_2^0(k)\Delta t_2^0 + \left. \frac{\partial \mathbf{f}_2(k)}{\partial \mathbf{X}_2(k)} \right|_{\mathbf{X}_2^0(k)} (\mathbf{X}_2(k) - \mathbf{X}_2^0(k)) + \\ \left. \frac{\partial \mathbf{f}_2(k)}{\partial \mathbf{U}_2(k)} \right|_{\mathbf{U}_2^0(k)} (\mathbf{U}_2(k) - \mathbf{U}_2^0(k)) + \\ \mathbf{f}_2^0(k)(\Delta t_2 - \Delta t_2^0) + \mathbf{a}_{R2}(k) \end{aligned} \quad (19)$$

式中： $\boldsymbol{a}_{R2}(k)$ 表示推力器工作段的虚拟控制量。不等式约束 $\boldsymbol{G}_2$ 如下：

$$(\boldsymbol{G}_2(k) \leq \mathbf{0}) : \begin{cases} R_2(k) \geq R_m \\ \sqrt{u_{r2}^2(k) + u_{t2}^2(k) + u_{p2}^2(k)} \leq 1 \\ \varpi_{l\Delta t_2} \Delta t_2^0 \leq \Delta t_2 \leq \varpi_{u\Delta t_2} \Delta t_2^0 \\ \|\boldsymbol{a}_{R2}(k)\|_2 \leq \kappa_{ar2k} \\ \|U_2(k+1) - U_2(k)\|_2 \leq 2 \sin\left(\frac{1}{2} \omega_{2\max} \Delta t_2^0\right) \end{cases} \quad (20)$$

式中： $\varpi_{l\Delta t_2}$ 和 $\varpi_{u\Delta t_2}$ 为时间信赖域的系数， κ_{ar2k} 为推力器工作段的虚拟控制量约束变量， $\omega_{2\max}$ 为推力器工作段的最大姿态角变化率。

由于直接上升交会方案并非立即起飞，因而本文在终端相角约束中加入了上升器在月面等待的时间 t_w ，以及松弛项 s_θ ，即：

$$\theta_{o0} + \omega_o((N_1 - 1)\Delta t_1 + (N_2 - 1)\Delta t_2) + \omega_o t_w - \theta_2(N_2) - 2\pi = s_\theta \quad (21)$$

式中：相角约束减去 2π 是因为轨道器与上升器的最终相角偏差应为 2π 。

2.3 离散优化模型

由 2.1 和 2.2 节可得月面应急上升交会的离散轨迹优化问题 (optimal discretized problem, ODP) 如下：

$$\begin{cases} \min t_w + (N_1 - 1)\Delta t_1 + (N_2 - 1)\Delta t_2 + \gamma_\theta s_\theta + \\ \gamma_{ar1} \sum_{k=1}^{N_1} \kappa_{ar1k} + \gamma_{ar2} \sum_{k=1}^{N_2} \kappa_{ar2k} \\ \text{s.t. } \boldsymbol{X}_1(1) = \boldsymbol{X}_0 \\ \boldsymbol{U}_1(1) = [1, 0, 0]^T \\ \boldsymbol{X}_1(k+1) - \boldsymbol{X}_1(k) = \frac{1}{2}(\boldsymbol{F}_1(k) + \boldsymbol{F}_1(k+1)) \\ \boldsymbol{G}_1(k) \leq \mathbf{0} \\ \boldsymbol{X}_2(1) = \boldsymbol{X}_1(N_1) \\ \boldsymbol{U}_2(1) = \boldsymbol{U}_1(N_1) \\ \boldsymbol{X}_2(k+1) - \boldsymbol{X}_2(k) = \frac{1}{2}(\boldsymbol{F}_2(k) + \boldsymbol{F}_2(k+1)) \\ \boldsymbol{G}_2(k) \leq \mathbf{0} \\ R_2(N_2) = R_f, \phi_2(N_2) = 0, \\ V_{r2}(N_2) = 0, V_{t2}(N_2) = V_f, V_{p2}(N_2) = 0 \\ \theta_{o0} + \omega_o t_w + \omega_o((N_1 - 1)\Delta t_1 + (N_2 - 1)\Delta t_2) - \\ \theta_2(N_2) - 2\pi = s_\theta \\ s_\theta \geq 0, t_w \geq 0 \\ m_{e1}(N_1 - 1)\Delta t_1 + m_{e2}(N_2 - 1)\Delta t_2 \leq m_0 - m_{dry} \end{cases} \quad (22)$$

式中：上升器需垂直起飞，因此， $\boldsymbol{U}_1(1) = [1, 0, 0]^T$ ；令 $s_\theta \geq 0$ 是为了避免性能指标取最小值时出现 s_θ 取得负无穷的情况； γ_θ 为 s_θ 的罚函数系数； $m_{e1} = P_1/I_{sp1}g_0$ ，

$m_{e2} = P_2/I_{sp2}g_0$ ， I_{sp1} 和 I_{sp2} 分别为主发动机的比冲和推力器的比冲。

为求解的稳定性，需对上升器的动力学方程组中的变量作归一化处理。径向距离的归一化参数 R_n 为目标轨道高度，相角 θ 和极角 ϕ 的数值在 10 以内，无需归一化； V_r 的归一化参数 $V_{rn} = 400$ ； V_t 的归一化参数 V_{tn} 取为 V_f ； V_p 的归一化参数 $V_{pn} = 30$ 。为表达的简洁，后文仍取原变量的符号作为归一化后的变量。

本文在优化结果达到以下条件时终止序列二阶锥规划的迭代：

$$\begin{cases} |\boldsymbol{X}_1^j - \boldsymbol{X}_1^{j+1}|_{\max} \leq \delta_{\text{SOCP}}, |\boldsymbol{U}_1^j - \boldsymbol{U}_1^{j+1}|_{\max} \leq \delta_{\text{SOCP}} \\ |\boldsymbol{X}_2^j - \boldsymbol{X}_2^{j+1}|_{\max} \leq \delta_{\text{SOCP}}, |\boldsymbol{U}_2^j - \boldsymbol{U}_2^{j+1}|_{\max} \leq \delta_{\text{SOCP}} \end{cases} \quad (23)$$

式中： δ_{SOCP} 为设定的值； $\boldsymbol{X}_1^j$ 和 $\boldsymbol{X}_1^{j+1}$ 及 $\boldsymbol{U}_1^j$ 和 $\boldsymbol{U}_1^{j+1}$ 分别为第 j 次和第 $j+1$ 次序列优化得到的主发动机工作段的状态量和控制量； $\boldsymbol{X}_2^j$ 和 $\boldsymbol{X}_2^{j+1}$ 及 $\boldsymbol{U}_2^j$ 和 $\boldsymbol{U}_2^{j+1}$ 表示第 j 次和第 $j+1$ 次序列优化得到的推力器工作段的状态量和控制量。

由于动力学方程组的线性化需要初始参考轨迹，本文初始猜想取值方法如下：

$$\begin{cases} R_1^0 = R_m/R_n \\ V_{r1}^0 = 1 \\ R_2^0 = 1 \\ V_{t2}^0 = 1 \end{cases} \quad (24)$$

$[\boldsymbol{X}_1, \boldsymbol{U}_1]$ 和 $[\boldsymbol{X}_2, \boldsymbol{U}_2]$ 中其余变量初始猜想均为 0， t_{1f} 的初始猜想为 150， t_{2f} 的初始猜想值根据式(25)计算：

$$t_{2f} = (m_0 - m_{dry} - m_{e1}t_{1f})/m_{e2} \quad (25)$$

3 内点法定制化改进求解设计

首先对 ECOS 中的内点法作简要说明，之后再对内点法的计算过程进行加速，以加快二阶锥规划问题的求解速度。

3.1 二阶锥规划问题的内点法求解方法

3.1.1 标准二阶锥规划问题

标准的二阶锥规划问题如下：

$$\begin{cases} \min \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t. } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ \boldsymbol{G}\boldsymbol{x} + \boldsymbol{s} = \boldsymbol{h} \\ \boldsymbol{s} \in \mathcal{K} \end{cases} \quad (26)$$

式中： $\boldsymbol{x}$ 为原变量； $\boldsymbol{s}$ 为松弛变量； $\boldsymbol{c} \in \mathbf{R}^n$ ， $\boldsymbol{G} \in \mathbf{R}^{m \times n}$ ， $\boldsymbol{h} \in \mathbf{R}^m$ ， $\boldsymbol{A} \in \mathbf{R}^{p \times n}$ ， $\boldsymbol{b} \in \mathbf{R}^p$ 为待解问题的矩阵或向量； $\mathcal{K}$ 为约束锥。

式(26)的对偶问题为

$$\begin{cases} \max & -\mathbf{b}^T \mathbf{y} - \mathbf{h}^T \mathbf{z} \\ \text{s.t.} & \mathbf{G}^T \mathbf{z} + \mathbf{A}^T \mathbf{y} + \mathbf{c} = \mathbf{0} \\ & \mathbf{z} \in \mathcal{K} \end{cases} \quad (27)$$

式中: $\mathbf{y}$ 和 $\mathbf{z}$ 为对偶问题的变量。

3.1.2 内点法流程

求解式 (26) 所示的二阶锥规划问题通常可采用原对偶内点法, 原对偶内点法即根据 KKT 条件的修改形式迭代计算最优解的方法。对于式 (26) 所描述的问题, 其 KKT 条件为

$$\begin{cases} \mathbf{c} + \mathbf{A}^T \mathbf{y} + \mathbf{G}^T \mathbf{z} = \mathbf{0} \\ \mathbf{A} \mathbf{x} - \mathbf{b} = \mathbf{0} \\ \mathbf{G} \mathbf{x} + \mathbf{s} - \mathbf{h} = \mathbf{0} \\ \mathbf{s}^T \mathbf{z} = 0 \\ \mathbf{s} \in \mathcal{K} \quad \mathbf{z} \in \mathcal{K} \end{cases} \quad (28)$$

内点法的详细计算流程可参考文献 [39] 和文献 [40], 本文仅对关键步骤作简要说明。

1) 中心路径

为便于对 KKT 条件进行求解, 需要对其进行松弛, 在对 KKT 条件进行松弛后可得到中心路径 (29), 对 KKT 条件的求解等价于对中心路径的求解, 中心路径即为满足式 (29) 的 $(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{z}, \tau, \kappa)$ 的集合。

$$\begin{cases} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{s} \\ \kappa \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{A}^T & \mathbf{G}^T & \mathbf{c} \\ -\mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{b} \\ -\mathbf{G} & \mathbf{0} & \mathbf{0} & \mathbf{h} \\ -\mathbf{c}^T & -\mathbf{b}^T & -\mathbf{h}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \\ \tau \end{bmatrix} + \mu \begin{bmatrix} \mathbf{q}_x \\ \mathbf{q}_y \\ \mathbf{q}_z \\ \mathbf{q}_\tau \end{bmatrix} \\ (\mathbf{W}^{-T} \mathbf{s}) \circ (\mathbf{W} \mathbf{z}) = \mu \mathbf{e} \quad \kappa \tau = \mu \\ \mathbf{s} \in \mathcal{K} \quad \mathbf{z} \in \mathcal{K} \quad \kappa \geq 0 \quad \tau \geq 0 \end{cases} \quad (29)$$

式中: $\mu = (\mathbf{s}^T \mathbf{z} + \kappa \tau) / (D + 1)$, 当 $\mu = 0$ 时, 式 (29) 等价于 KKT 条件; D 为锥的阶数, 即锥 $\mathcal{K}$ 由 D 个标准锥组成; τ 和 κ 为额外引入的变量; $\mathbf{q}_i$ 的定义如下:

$$(\mathbf{q}_x, \mathbf{q}_y, \mathbf{q}_z, \mathbf{q}_\tau) = \frac{D+1}{\mathbf{s}^T \mathbf{z} + \tau \kappa} (\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z, \mathbf{r}_\tau) \quad (30)$$

$$\begin{cases} \mathbf{r}_x = -\mathbf{A}^T \mathbf{y} - \mathbf{G}^T \mathbf{z} - \mathbf{c} \tau \\ \mathbf{r}_y = \mathbf{A} \mathbf{x} - \mathbf{b} \tau \\ \mathbf{r}_\tau = \kappa + \mathbf{c}^T \mathbf{x} + \mathbf{b}^T \mathbf{y} + \mathbf{h}^T \mathbf{z} \\ \mathbf{r}_z = \mathbf{s} + \mathbf{G} \mathbf{x} - \mathbf{h} \tau \end{cases} \quad (31)$$

式中: 运算符号 $\circ$ 为欧式约当算子^[39]。

$\mathbf{e} = [\mathbf{e}_1^T, \mathbf{e}_2^T, \dots, \mathbf{e}_D^T]^T$, 对于线性锥, $\mathbf{e}_i = \mathbf{1}$; 对于 m 维二阶锥, $\mathbf{e}_i = (1, \mathbf{0}_{m-1})$ 。 $\mathbf{W}$ 为内斯特罗夫-托德 (Nesterov-Todd, NT) 缩放矩阵, 其定义见文献 [39]。

2) 更新方向与更新步长

内点法在求解二阶锥规划问题时, 首先计算变量初值 $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{s}_0, \mathbf{z}_0, \tau_0, \kappa_0)$, 再根据变量初值计算更新方向 $(\Delta \mathbf{x}, \Delta \mathbf{y}, \Delta \mathbf{s}, \Delta \mathbf{z}, \Delta \tau, \Delta \kappa)$ 与更新步长 α 。在得到更新方向与更新步长后, 再对变量进行更新:

$$\begin{aligned} (\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{z}, \tau, \kappa) &= (\mathbf{x}_0, \mathbf{y}_0, \mathbf{s}_0, \mathbf{z}_0, \tau_0, \kappa_0) + \\ &\alpha (\Delta \mathbf{x}, \Delta \mathbf{y}, \Delta \mathbf{s}, \Delta \mathbf{z}, \Delta \tau, \Delta \kappa) \end{aligned} \quad (32)$$

在完成对变量的更新后, 根据新的变量计算更新方向和更新步长, 不断循环此过程, 直至变量收敛至满足 KKT 条件的解。

在 ECOS 中, 更新方向的计算包括仿射方向、中心化方向、矫正方向 3 个部分。在计算仿射方向时, 令 $\mu = 0$, 其目的在于让变量尽可能地满足 KKT 条件。计算中心化方向的目的在于让变量更接近中心路径, 以便在下一次迭代中可以取得更大的更新步长。计算矫正方向是为了补偿在计算仿射方向时, 由于对式 (29) 泰勒展开线性化所引入的误差。中心化方向和矫正方向的计算可进行合并, 该合并方向称为组合方向。

仿射方向和组合方向的计算可转化为式 (33) 和 (34) 的线性方程组的计算:

$$\mathbf{K} [\Delta \mathbf{x}_1, \Delta \mathbf{y}_1, \Delta \mathbf{z}_1]^T = [-\mathbf{c}, \mathbf{b}, \mathbf{h}]^T \quad (33)$$

$$\mathbf{K} [\Delta \mathbf{x}_2, \Delta \mathbf{y}_2, \Delta \mathbf{z}_2]^T = [\mathbf{d}_x, \mathbf{d}_y, \mathbf{d}_z]^T \quad (34)$$

式中:

$$\mathbf{K} = \begin{bmatrix} \mathbf{0} & \mathbf{A}^T & \mathbf{G}^T \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{G} & \mathbf{0} & -\mathbf{W}^2 \end{bmatrix} \quad (35)$$

式 (33) 与 (34) 称为 KKT 方程组, 其系数矩阵称为 KKT 矩阵。在计算仿射方向和组合方向时, 式 (34) 的右端项会有差别, 详情可参考文献 [26]。

3) 内点法迭代停止条件

当变量迭代至最优解时, 内点法即停止迭代, 一般的, 最优条件的设定如下:

$$\zeta = \max \left\{ \frac{\|\mathbf{r}_x\|_2}{r_{xs} \tau}, \frac{\|\mathbf{r}_y\|_2}{r_{ys} \tau}, \frac{\|\mathbf{r}_z\|_2}{r_{zs} \tau} \right\} \quad (36)$$

式中:

$$r_{xs} = \max \{1, \|\mathbf{c}\|_2 + \|\mathbf{y}\|_2 + \|\mathbf{z}\|_2\} \quad (37)$$

$$r_{ys} = \max \{1, \|\mathbf{b}\|_2 + \|\mathbf{x}\|_2\} \quad (38)$$

$$r_{zs} = \max \{1, \|\mathbf{h}\|_2 + \|\mathbf{x}\|_2 + \|\mathbf{s}\|_2\} \quad (39)$$

其次, 定义对偶间隙与相对对偶间隙分别如式 (40) 和 (41) 所示:

$$\delta_{\text{gap}} = \mathbf{s}^T \mathbf{z} \quad (40)$$

$$\delta_{\text{relgap}} = \mathbf{s}^T \mathbf{z} / \nu \quad (41)$$

式中: $\nu = \max \{-\mathbf{c}^T \mathbf{x}, -\mathbf{b}^T \mathbf{y} - \mathbf{h}^T \mathbf{z}\}$ 。

当 $\nu > 0$ 时, 如果 $\zeta < o_{\text{feastol}}$, 而且 $\delta_{\text{abs}} < o_{\text{abstol}}$ 或 $\delta_{\text{relgap}} < o_{\text{reltol}}$, 即认为求得最优解。 o_{feastol} 为等式约束残差精度, o_{abstol} 为绝对对偶残差精度, o_{reltol} 为相对

对偶残差精度。

4) KKT 矩阵的变换

在求解式 (33) 和式 (34) 所示的 KKT 方程组时,一般采用直接法进行求解,即先对 KKT 矩阵进行分解,再根据分解得到的三角矩阵计算变量的值。然而,由于式 (35) 的 KKT 矩阵不是正定的,无法对其直接进行快速矩阵分解,因此,需对 KKT 矩阵做如下转换:

$$\tilde{\mathbf{K}} = \mathbf{K} + \Delta\mathbf{K} = \begin{bmatrix} \delta\mathbf{I}_n & \mathbf{A}^T & \mathbf{G}^T \\ \mathbf{A} & -\delta\mathbf{I}_p & \mathbf{0} \\ \mathbf{G} & \mathbf{0} & -\mathbf{W}^2 \end{bmatrix} \quad (42)$$

式中:

$$\Delta\mathbf{K} = \begin{bmatrix} \delta\mathbf{I}_n & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\delta\mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (43)$$

式中: $\delta\mathbf{I}_n$ 和 $\delta\mathbf{I}_p$ 分别为 n 阶和 p 阶对角微元单位矩阵, $\delta = 7 \times 10^{-8}$ 。

5) 内点法流程

内点法计算流程如下:

算法 内点法算法。

输入: 二阶锥规划问题的 $\mathbf{c}, \mathbf{G}, \mathbf{h}, \mathbf{A}, \mathbf{b}$ 。

输出: $(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{z}, \tau, \kappa)$ 。

1. 输入二阶锥规划问题的 $\mathbf{c}, \mathbf{G}, \mathbf{h}, \mathbf{A}, \mathbf{b}$;
2. 计算初值 $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{s}_0, \mathbf{z}_0, \tau_0, \kappa_0)$;
3. 计算残差, 式 (30);
4. 判断循环条件, 当达到最优或问题无解时, 退出循环;
5. 更新 KKT 矩阵中的 $\mathbf{W}$ 矩阵;
6. KKT 矩阵分解;
7. 仿射方向和仿射步长的计算;
8. 中心化参数计算;
9. 更新方向与更新步长的计算;
10. 更新变量 $(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{z}, \tau, \kappa)$, 回到步骤 3。

3.2 线性方程组定制化求解加速方法

在内点法的单次迭代中, 仿射方向和组合方向均需计算式 (33) 和 (34) 的 KKT 方程组, 但由于式 (33) 的 KKT 方程组右端项不发生改变, 因此, 在计算组合方向时, 无需对式 (33) 进行重复计算。因此, 每一次内点法的迭代需要计算 3 次 KKT 方程组。而 KKT 方程组求解是内点法中最耗时的部分, 因此, 本文主要对 KKT 方程组的求解进行加速。

3.2.1 定制化矩阵排序

在式 (42) 中, $\tilde{\mathbf{K}}$ 矩阵为拟正定矩阵, 为了减少对 $\tilde{\mathbf{K}}$ 矩阵进行分解时产生的填充元数量, 需要对 $\tilde{\mathbf{K}}$ 矩阵进行重新排序。即

$$\tilde{\mathbf{K}}_P = \mathbf{P}\tilde{\mathbf{K}}\mathbf{P}^T = \mathbf{L}\mathbf{D}\mathbf{L}^T \quad (44)$$

式中: $\mathbf{P}$ 矩阵为排序矩阵; $\mathbf{L}$ 为矩阵分解后得到的下三角矩阵; $\mathbf{D}$ 矩阵为矩阵分解后得到的对角矩阵。

为综合排序耗时与排序效果, ECOS 采用近似最小度排序算法^[41]对 $\tilde{\mathbf{K}}$ 矩阵进行重新排序。但近似最小度排序并不一定是填充最少的排序方法, 常用的排序方法还有最小度排序^[42], 局部填充最小排序^[43]等。

对于定制化的问题, $\tilde{\mathbf{K}}$ 矩阵的非零元素位置不发生改变, 即排序矩阵可提前计算得到。由文献 [44] 可知, 对于任意的排序矩阵 $\mathbf{P}$, $\mathbf{P}\tilde{\mathbf{K}}\mathbf{P}^T$ 矩阵的分解是稳定的。因此, 在选择排序方法时, 无需考虑排序耗时的影响, 仅需根据排序效果选取 $\mathbf{L}$ 矩阵非零元素个数最少的排序方法即可。

以本文的轨迹规划问题为例, 通过取不同的离散点数得到不同维数的 KKT 矩阵, 对比近似最小度排序算法以及局部填充最小排序算法的排序效果如表 1 所示, 表 1 中离散点数 ($N_1 + N_2$) 表示式 (22) 中的离散点数 N_1 和 N_2 之和, 后文与此相同。可看到, 采用局部填充最小排序算法, $\mathbf{L}$ 矩阵的非零元素个数更少。

表 1 不同排序方法的矩阵非零元素个数对比
Table 1 Number of non-zero elements of matrix for different ordering methods

离散点数(N_1+N_2)	非零元素个数	
	近似最小度算法	局部填充最小算法
11+21	8 194	7 209
21+41	17 059	14 416
31+61	26 977	21 709
41+81	34 435	28 916

3.2.2 定制化矩阵分解

为加快矩阵分解的速度, 一般采用稀疏矩阵 Cholesky 分解法^[45]对 $\tilde{\mathbf{K}}_P$ 矩阵进行分解。对于定制化的问题, $\tilde{\mathbf{K}}_P$ 矩阵的非零元素个数及位置不随着优化问题的求解而改变, 因此, 可对矩阵的分解过程进行定制化。

本文将 $\tilde{\mathbf{K}}_P$ 矩阵分解所需的定制化信息, 如 $\mathbf{L}$ 矩阵的非零元素位置信息等提前计算得出, 内置计算结果。本文方法相较于采用代码自动生成的方法而言, 需要额外的读取非零元素位置等数据的时间, 但代码量小, 所需的储存空间也较小。

3.2.3 数据储存定制化

通用求解器在求解相应的优化问题时, 一般采用动态内存储存数据。对于动态申请的储存空间, 储存空间的申请与释放均需要消耗一部分时间, 且此时数据储存于内存中, 读取与写入的速度较慢。

然而, 对于定制化的问题, $\tilde{\mathbf{K}}_P$ 矩阵的非零元素

位置不改变, 排序矩阵不改变, 数据所需的储存空间是确定的。因此, 本文采用静态内存储存数据, 对于静态内存而言, 储存空间申请与释放的耗时少, 且此时数据储存于缓存区, 读取与写入的速度快。

3.2.4 仿射方向的并行计算

在仿射方向时, 需要计算式 (33) 和 (34) 的 2 个线性方程组。为加快仿射方向的计算速度, 可将式 (33) 和 (34) 并行计算, 如图 4 所示。

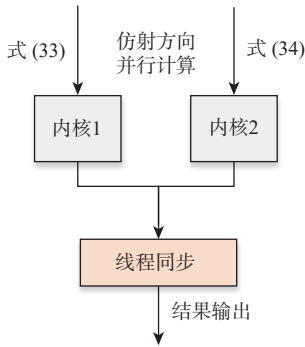


图 4 仿射方向并行计算流程

Fig. 4 Parallel calculation process of affine direction

3.2.5 动态调整 KKT 方程组求解精度

取 $\tilde{K}\xi = r$ 进行分析, 在采用 $\tilde{K}$ 矩阵对 KKT 方程组进行求解时, 由于引入了微元矩阵 ΔK , 求解结果会存在偏差。在对求解精度要求较高时, 一般采用迭代法提高 KKT 方程组的求解精度, 具体流程如下:

算法 KKT 方程组求解算法。

输入: $\tilde{K}$ 、 r 、 K 。

输出: ξ 。

1. $\tilde{K}\xi = r$;
2. $\tilde{K}\Delta\xi = r - K\xi$;
3. $\xi = \xi + \Delta\xi$;
4. 若 ξ 精度满足要求, 或精度提高缓慢, 则跳出循环; 否则回到步骤 2。

由 KKT 方程组求解算法可知, 求解 KKT 方程组需迭代多次。然而本文研究发现, 在内点法计算初期, 即对偶间隙与相对对偶间隙较大时, 微元 δ 对组合方向与更新步长的计算结果影响微弱。

因此, 可以对 KKT 方程组的求解精度进行动态调整, 在对偶间隙 δ_{gap} 与相对对偶间隙 δ_{relgap} 较大时, 不修正 KKT 方程组的精度; 在 δ_{gap} 与 δ_{relgap} 较小时, 再加入对 KKT 方程组求解的迭代修正, 其改进算法如下所示, 其中: ε_{abs} 和 ε_{rel} 为设定的 KKT 方程组修正阈值。

算法 改进 KKT 方程组求解算法。

输入: $\tilde{K}$ 、 r 、 K 、 ε_{abs} 、 ε_{rel} 。

输出: ξ 。

1. $\tilde{K}\xi = r$;
2. 当 $\delta_{\text{gap}} > \varepsilon_{\text{abs}}$ 且 $\delta_{\text{relgap}} > \varepsilon_{\text{rel}}$ 时, 跳出循环; 否则转到 3;
3. $\tilde{K}\Delta\xi = r - K\xi$;
4. $\xi = \xi + \Delta\xi$;
5. 若 ξ 精度满足要求, 或精度提高缓慢, 则跳出循环; 否则回到步骤 3。

3.3 热启动方法

内点法在求解优化问题时需给优化变量赋初值, 常用的初值计算方法有冷启动法、传统热启动法以及最近提出的序列二阶锥规划改进热启动法 (后称改进热启动法)。

3.3.1 冷启动法

内点法的冷启动一般如下选择冷启动参数:

$$(x, s, y, z, \kappa, \tau) = (0, e, 0, e, 1, 1)$$

(45)

式中: $e = [e_1^T, e_2^T, \dots, e_b^T]^T$ 为约束锥的单位向量。

3.3.2 传统热启动法

在传统热启动方法中, 内点法初值计算方法如下:

- 1) 变量 x 和 s

对变量 x 的初值计算等价于求解以下优化问题:

$$\begin{cases} \min \frac{1}{2} \|\bar{s}\|^2 \\ \text{s.t. } \bar{s} = h - Gx \\ Ax = b \end{cases}$$

(46)

变量 s 的初值可由式 (46) 中的 $\bar{s}$ 计算得到, 即

$$s = \begin{cases} \bar{s} & \bar{s} \in \mathcal{K} \\ [\bar{s} + (1 + \alpha_p)e] \in \mathcal{K} & \bar{s} \notin \mathcal{K} \end{cases}$$

(47)

- 2) 对偶变量 y 和 z

对偶变量 y 的初值取以下优化问题的结果:

$$\begin{cases} \min \frac{1}{2} \|\bar{z}\|^2 \\ \text{s.t. } A^T y + G^T \bar{z} + c = 0 \end{cases}$$

(48)

对偶变量 z 的初值可通过式 (48) 中的 $\bar{z}$ 计算得到, 即

$$z = \begin{cases} \bar{z} & \bar{z} \in \mathcal{K} \\ [\bar{z} + (1 + \alpha_p)e] \in \mathcal{K} & \bar{z} \notin \mathcal{K} \end{cases}$$

(49)

- 3) 变量 κ 和 τ

变量 κ 和 τ 的初值均取 1。

3.3.3 改进热启动法

在求解序列二阶锥规划问题时, 冷启动内点法及传统热启动内点法未能充分利用前一次序列优化的结果。因此, 文献 [35] 提出了一种序列二阶锥

规划改进热启动方法,该方法取热启动参数如下:

$$\begin{cases} \boldsymbol{x} = \omega \boldsymbol{x}' \\ \boldsymbol{y} = \omega \boldsymbol{y}' \\ \boldsymbol{s} = \omega \boldsymbol{s}' + (1 - \omega) \boldsymbol{e} \\ \boldsymbol{z} = \omega \boldsymbol{z}' + (1 - \omega) \boldsymbol{e} \\ \kappa = \boldsymbol{s}^T \boldsymbol{z} / D \\ \tau = 1 \end{cases} \quad (50)$$

式中: $(\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{s}', \boldsymbol{z}')$ 为序列二阶锥规划中上一次优化的结果; ω 取值接近 1。

热启动初值较为理想时,初始 $(\boldsymbol{r}_x, \boldsymbol{r}_y, \boldsymbol{r}_z, \boldsymbol{r}_\tau, \mu)$ 的值较小,此时,内点法收敛到指定精度所需的迭代次数较少,相关的证明可参考文献 [35]。

3.4 二阶锥规划子问题求解精度调整策略

在序列二阶锥规划方法中,各二阶锥规划子问题的精度可分为凸化精度与求解精度两部分,凸化精度受前后 2 次优化量的接近程度、线性化方法、离散方法与离散点数等的影响^[46];求解精度即内点法求解二阶锥规划子问题的精度。

传统的序列二阶锥规划方法在求解各二阶锥规划子问题时,均是让内点法在收敛到同一较高精度时才停止迭代。然而,在序列二阶锥规划初期的计算中,由于初始猜想的不精确性,在经过泰勒展开线性化等凸化方法处理后,二阶锥规划子问题的凸化精度较低,此时将求解精度设置得过高并无意义。

因此,可对各二阶锥规划子问题的求解精度进行动态调整,在二阶锥规划子问题的凸化精度较低时,通过降低内点法的求解精度提高求解速度;在后续求解二阶锥规划子问题时,再逐步提高内点法的求解精度,使得序列二阶锥规划问题最终收敛到高精度的解。

假设内点法初始猜想 $(\boldsymbol{x}_0, \boldsymbol{s}_0, \boldsymbol{y}_0, \boldsymbol{z}_0, \kappa_0, \tau_0)$ 的等式约束残差为 ζ_0 , 对偶间隙为 $\delta_{\text{gap}0}$, 相对对偶间隙为 $\delta_{\text{relgap}0}$, 则求解精度可如下设置:

$$\varpi = \min \left\{ \frac{\zeta_0}{M}, \frac{\delta_{\text{gap}0}}{M}, \frac{\delta_{\text{relgap}0}}{M} \right\} \quad (51)$$

$$o_{\text{feastol}} = o_{\text{abstol}} = o_{\text{reltol}} = \max \{ \varpi, o_{\text{max}} \} \quad (52)$$

式中: M 为设定的精度提高倍数; o_{max} 为内点法最高优化精度。

由式 (51) 和式 (52) 可知,如果 $\varpi > o_{\text{max}}$, 内点法求解结果的精度达到初始猜想的精度的 $1/M$ 时退出循环;如果 $\varpi < o_{\text{max}}$, 内点法求解结果的精度达到最高精度时退出循环。

为避免序列二阶锥规划方法在初始几次的迭代中,因为内点法优化精度 ϖ 太低而导致前后 2 个

二阶锥规划子问题变化量较小,从而出现序列二阶锥规划过早终止迭代的情况。首先,精度提高倍数 M 的值不应太小;其次,在序列二阶锥规划方法迭代的终止条件中还应加上内点法优化精度的判断,即

$$\varpi < o_{\min} \quad (53)$$

式中: $o_{\min}$ 为内点法最低优化终止精度。

3.5 内点法定制化改进的序列二阶锥规划流程

综合 3.1 节~3.4 节的 KKT 方程组求解加速方法、改进热启动方法及二阶锥规划子问题求解精度调整策略,可得到内点法定制化改进的序列二阶锥规划流程图如图 5 所示。

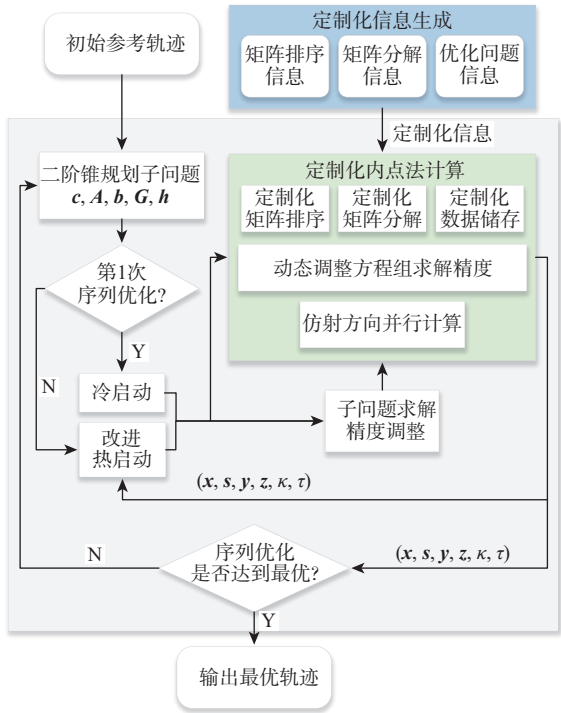


图 5 基于定制化改进内点法的序列二阶锥规划流程

Fig. 5 Successive second-order cone programming based on customized and modified interior point method

如图 5 所示,在实施时,①根据二阶锥规划子问题的结构生成定制化信息(比如矩阵排序的信息,矩阵定制化分解时所需的 L 矩阵非零元素位置信息等);②根据初始参考轨迹生成二阶锥规划子问题的 $\boldsymbol{c}, \boldsymbol{A}, \boldsymbol{b}, \boldsymbol{G}, \boldsymbol{h}$ 矩阵及向量;③由于在第 1 次序列二阶锥规划迭代时无法获得热启动值,因此,第 1 个二阶锥规划子问题需根据冷启动法计算内点法优化变量的初值,在后续的二阶锥规划子问题的求解中,再根据改进热启动法计算内点法优化变量的初值;④在计算得到内点法优化变量的初值后,根据二阶锥规划子问题求解精度调整策略,计算当前优化问题的求解精度;⑤将内点法变量初值及优化精度赋予定制化内点法求解器中进行求解;⑥在定

制化求解器计算时, 求解器根据得到的定制化信息进行快速矩阵排序、矩阵分解及线性方程组并行计算, 在内点法的计算结果达到设定的二阶锥规划子问题求解精度时, 求解器输出求解结果; ⑦根据求解器的求解结果判断是否达到序列二阶锥规划算法的最优条件, 即式 (23) 及式 (53), 当达到最优条件时, 输出最优轨迹, 否则更新二阶锥规划子问题, 再次求解。

值得注意的是, 定制化信息的计算需要二阶锥规划子问题 $\mathbf{A}$ 矩阵和 $\mathbf{G}$ 矩阵的非零元素位置信息,

以及 $\mathbf{c}$ 、 $\mathbf{b}$ 、 $\mathbf{h}$ 向量的维数信息, 但定制化信息的计算仅需在整个过程实施一次, 且计算结果可在主程序运行前预置于定制化求解器中。

4 数值仿真与分析

为验证本文的月面应急上升交会轨迹规划方法及基于定制化改进内点法的序列二阶锥规划方法(后文称改进序列二阶锥规划方法), 用改进序列二阶锥规划方法求解 ODP 问题。数值仿真参数如表 2 所示。

Table 2 Parameters of ascender					
上升器总重 m_{wet}/kg	上升器干重 m_{dry}/kg	主发动机推力 P_1/N	主发动机比冲 I_{sp1}/s	推力器总推力 P_2/N	推力器比冲 I_{sp2}/s
800	370	3 000	313.7	960	290

轨道器的轨道参数为: 半长轴 $a_0 = 1\,937.4\text{ km}$, 偏心率 $e_0 = 0$, 轨道倾角 $i_0 = 43.75^\circ$, 升交点赤经 $\Omega_0 = 232^\circ$; 轨道器初始相角 $\theta_{00} = 104.7^\circ$ 。上升器的初始位置为: $R_0 = 1\,737.4\text{ km}$, $\theta_0 = 0^\circ$, $\phi_0 = 0.744\,9^\circ$; 由于月球自转, 上升器具有初始速度: $V_{r0} = 0\text{ m/s}$, $V_{t0} = 3.382\,5\text{ m/s}$, $V_{p0} = 0.007\,2\text{ m/s}$ 。

4.1 求解结果及分析

以表 2 的仿真参数为例, 应用本文的定制化求解器(命名为 MCOS)及改进序列二阶锥规划方法求解 ODP 问题。此外, 为验证本文方法的正确性, 应用通用求解器 ECOS 及传统序列二阶锥规划方法求解 ODP 问题以进行对比分析。为避免降低优化精度对 ECOS 最终优化结果的影响, ECOS 的优化精度仍设置为其默认的 1×10^{-8} ; 在 MCOS 中, KKT 方程组修正阈值设置为 $\varepsilon_{\text{abs}} = \varepsilon_{\text{rel}} = 1 \times 10^{-3}$, 热启动参数 $\omega = 0.992$, 内点法最低优化终止精度 $\sigma_{\text{min}} = 1 \times 10^{-4}$, 内点法最高优化精度 $\sigma_{\text{max}} = 1 \times 10^{-8}$, 序列二阶锥规划收敛精度 $\delta_{\text{SOCp}} = 1 \times 10^{-3}$ 。

取主发动机工作段离散点数 $N_1 = 11$, 推力器工作段离散点数 $N_2 = 21$ 。仿真结果如图 6~图 8 所示, 图中改进方法表示以定制化求解器 MCOS 及改

进序列二阶锥规划方法求解 ODP 问题的结果, 原方法表示通用求解器 ECOS 及传统序列二阶锥规划方法求解 ODP 问题的结果。

图 6 中, 上升器在月面等待轨道器飞行到相应的位置后再起飞, 月面等待时间为 $4\,897.4\text{ s}$, 上升交会飞行时间为 952.4 s , 即上升器在 $5\,849.8\text{ s}$ 后完成与轨道器的远程交会。在图 7 中, 为实现应急上升交会, 原方法与改进方法均耗尽了燃料。在图 8 中, 由于推力角度变化率的约束, 主发动机工作段

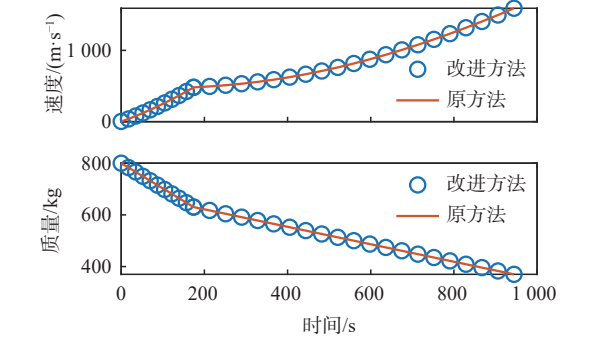


图 7 速度和质量曲线
Fig. 7 Velocity and mass curves

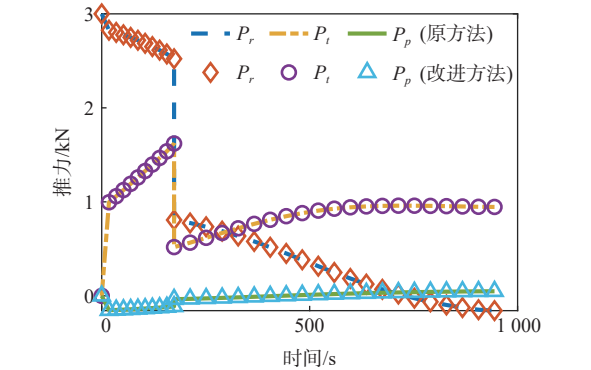


图 8 推力曲线
Fig. 8 Thrust curves

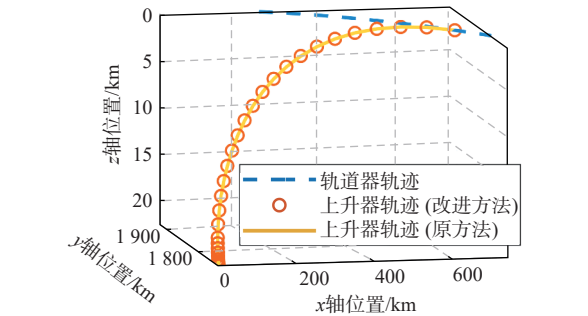


图 6 三维轨迹
Fig. 6 3D trajectories

的第 1 个离散点和第 2 个离散点的变化是上升器从垂直起飞到转弯完成的过程。由图 6~图 8 可知,原方法和改进方法的求解结果一致。

4.2 计算效率对比分析

4.2.1 热启动效果及分析

以求解 ODP 为例,对比各种内点法启动方法的迭代次数,取离散节点数 $N_1 = 21$, $N_2 = 41$,本节的热启动效果分析不涉及二阶锥规划子问题的精度调整,因此,内点法在求解各二阶锥规划子问题时,求解精度均设置为 $\alpha_{\text{feastol}} = \alpha_{\text{abstol}} = \alpha_{\text{reltol}} = 1 \times 10^{-8}$,热启动参数 $\omega = 0.999\ 2$,不同启动方法的迭代次数对比如图 9 所示。

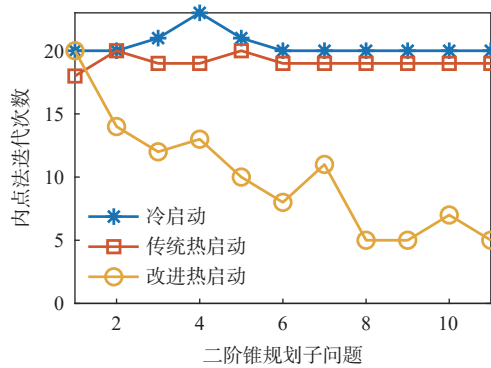


图 9 不同启动方法迭代次数

Fig. 9 Iteration times of different starting methods

由图 9 可知,改进热启动内点法所需的迭代次数相较于冷启动内点法以及传统热启动内点法而言更少,且随着迭代的进行,改进热启动内点法所需的迭代次数呈下降趋势。原因是对于上升交会问题 ODP,不等式约束的 $\mathbf{G}$ 和 $\mathbf{h}$ 及等式约束的 $\mathbf{A}$ 和 $\mathbf{b}$ 随着序列二阶锥规划迭代的进行变化量越来越小,取 δ_A , δ_b , δ_G , δ_h 如下:

$$\delta_A = \|\mathbf{A}^{i+1} - \mathbf{A}^i\|_F \tag{54}$$

$$\delta_b = \|\mathbf{b}^{i+1} - \mathbf{b}^i\|_2 \tag{55}$$

$$\delta_G = \|\mathbf{G}^{i+1} - \mathbf{G}^i\|_F \tag{56}$$

$$\delta_h = \|\mathbf{h}^{i+1} - \mathbf{h}^i\|_2 \tag{57}$$

式中: $\mathbf{A}^i$ 和 $\mathbf{A}^{i+1}$ 为第 i 个和第 $i+1$ 个二阶锥规划子问题的等式约束矩阵; $\mathbf{b}^i$ 和 $\mathbf{b}^{i+1}$ 为第 i 个和第 $i+1$ 个二阶锥规划子问题的等式约束右端项; $\mathbf{G}^i$ 和 $\mathbf{G}^{i+1}$ 为第 i 个和第 $i+1$ 个二阶锥规划子问题的不等式约束矩阵; $\mathbf{h}^i$ 和 $\mathbf{h}^{i+1}$ 为第 i 个和第 $i+1$ 个二阶锥规划子问题不等式约束的右端项。优化问题的变化趋势如图 10 所示。

由图 10 可知,随着序列二阶锥规划迭代的进行,优化问题的数值变化趋于下降,优化变量趋向

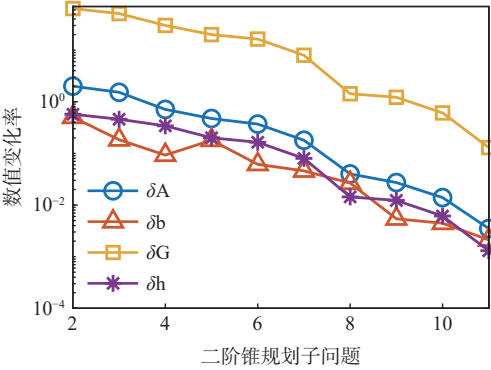


图 10 优化问题变化趋势

Fig. 10 Optimization problem evolution trend

于收敛,即本次序列二阶锥规划的结果与上一次序列二阶锥规划的结果越来越接近,因而热启动内点法所需的迭代次数随着序列二阶锥规划迭代的进行越来越少。

4.2.2 二阶锥规划子问题求解精度调整效果及分析

以 4.2.1 节的仿真参数为例,对比不同的精度提高倍数 M ,对序列二阶锥规划迭代次数、内点法总迭代次数以及平均每次内点法的迭代次数的影响。对比结果如表 3 所示。

表 3 不同 M 值下的计算结果对比

Table 3 Comparison of calculation results under different M values

M	序列二阶锥规划 迭代次数	内点法 总迭代次数
20	15	96
40	15	102
60	15	102
100	10	60
200	10	64
400	10	72
600	12	84
1 000	11	89

由表 3 可知, M 值的大小会对序列二阶锥规划的迭代次数产生影响,在 M 值较小时,二阶锥规划子问题的求解精度较低,各二阶锥规划子问题的线性化参考轨迹的误差较大,因而序列二阶锥规划的迭代次数较多,内点法迭代次数也较多。在 M 值增大时,序列二阶锥规划的迭代次数减少,但 M 值增大到一定程度时,二阶锥规划子问题的求解精度过高,因而内点法的迭代次数也较多。随着 M 值的增加,总的内点法迭代次数呈先下降后上升的趋势。

4.3 综合加速效果与求解精度对比分析

以 4.1 节的仿真为例,通过取不同的离散点数得到不同规模的优化问题,对于不同的优化问题,运行 100 次取平均值计算求解耗时。仿真设置与

4.1 节一致。以台式计算机作为仿真平台, 仿真环境为 Visual Studio 2019, 仿真语言为 C 语言, 台式计算机的 CPU 为 Intel Core i3 10100F(基准频率 3.6 GHz, 最高频率 4.3 GHz, 三级缓存 6 MB), 内存 8 G。不同离散点数的热启动参数 ω 以及精度提高倍数 M 如表 4 所示, 统计结果如图 11 及表 5 和表 6 所示, 离散点数 11+21 表示主发动机工作段离散点数为 11 个, 推力器工作段的离散点数为 21 个, 其余离散点数同理。

表 4 不同离散点数下的参数设置

离散点数(N_1+N_2)	热启动参数 ω	精度提高倍数 M
11+21	0.992 0	200
21+41	0.999 2	100
31+61	0.999 6	100
41+81	0.999 6	100

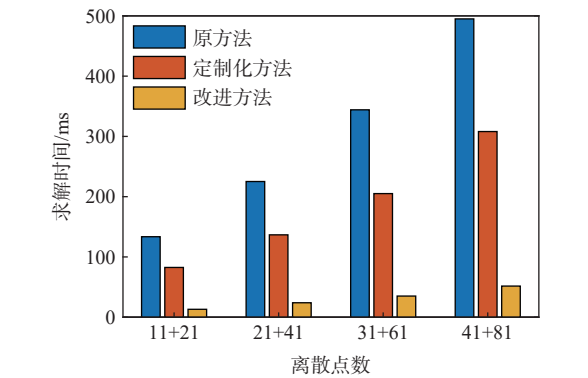


图 11 不同方法的求解耗时统计图

Fig. 11 Statistics of runtime by different methods

表 5 不同方法的求解耗时统计

离散点数(N_1+N_2)	原方法 耗时/ms	定制化方法 耗时/ms	改进方法 耗时/ms
11+21	133.60	82.60	13.15
21+41	225.20	136.70	24.00
31+61	344.10	205.10	35.10
41+81	495.00	308.10	51.60

表 6 不同方法加速比

离散点数(N_1+N_2)	定制化方法 加速比	改进方法 加速比
11+21	1.62	10.19
21+41	1.65	9.39
31+61	1.68	9.81
41+81	1.61	9.58

为进一步体现本文方法的有效性, 除了与原方法对比外, 本文还增加了与文献 [27] 中定制化方法(后文简称定制化方法)的对比, 但由于文献 [27]

中的 Bsocp 求解器不开源, 因此, 本文仅应用文献 [27] 中的定制化方法的原理(即本文的 3.2.1 节、3.2.2 节和 3.2.3 节中的方法)对 ECOS 进行改进。

定义加速比 χ 如下:

$$\chi = \frac{\text{加速前的计算耗时}}{\text{加速后的计算耗时}} \quad (58)$$

由图 11、表 5 和表 6 可知, 相比于传统的序列二阶锥规划方法, 在仅应用定制化方法时可达到 1.6 倍的加速比。与文献 [27] 相比, 在应用本文的定制化方法时未出现加速效果随问题规模增大而下降的情况, 这是因为本文在对求解器进行定制化时未通过代码自动生成的方式实现, 无需生成庞大的可执行代码, 因而不会出现文献 [27] 中所述的由于生成的代码量过于庞大而导致的缓存丢失的现象。

对比原方法与改进方法可知, 本文提出的方法相比于传统的序列二阶锥规划方法可达到 9.5 倍左右的加速比。

为进一步对比各种方法的求解精度, 本文通过对优化问题 ODP 取不同的离散点数, 对比优化结果的精度。在计算优化结果的精度时, 本文根据优化得到的控制量进行积分计算, 积分方法为四阶龙格库塔法, 积分步长为 0.01 s。精度对比结果如表 7 所示。

表 7 不同方法的求解精度

离散点数(N_1+N_2)	方法	位置误差/km	速度误差/ ($\text{m}\cdot\text{s}^{-1}$)
11+21	原方法	16.280 4	33.529 9
	定制化方法 ^[27]	16.280 4	33.529 9
	改进方法	16.155 7	33.268 1
21+41	原方法	7.942 4	16.482 4
	定制化方法 ^[27]	7.942 4	16.482 4
	改进方法	8.118 8	16.841 4
31+61	原方法	5.262 3	10.938 1
	定制化方法 ^[27]	5.262 3	10.938 1
	改进方法	5.312 4	11.029 8
41+81	原方法	4.080 0	8.472 8
	定制化方法 ^[27]	4.080 0	8.472 8
	改进方法	3.933 3	8.172 2

由表 7 可知, 原方法与定制化方法相比, 求解精度完全一样, 这是因为在对求解器进行定制化改进后, 内点法的求解流程没有发生任何改变, 因而求解结果完全相同。但原方法与本文的改进方法求解结果存在偏差, 这是因为改进方法在求解各二阶锥规划子问题时的初值与收敛精度和原方法相比存在偏差, 因而两方法的优化结果也会存在微弱的偏差。由位置误差和速度误差也可看出, 原方法

与改进方法的求解精度差距微弱。

5 结 论

1) 本文基于序列二阶锥规划方法,提出了一种月面应急上升交会联合轨迹优化方法,并且应用内点法定制化方法、改进热启动法、二阶锥规划子问题精度调整的方法,对月面上升交会的序列二阶锥规划算法进行加速。

2) 仿真表明,在普通台式计算机上,综合应用本文的方法,相较于传统序列二阶锥规划方法,本文方法在不降低求解精度的情况下可达到 9.5 倍左右的加速比。

参考文献 (References)

[1] 欧阳自远. 我国月球探测的总体科学目标与发展战略[J]. 地球科学进展, 2004, 19(3): 351-358.
OUYANG Z Y. Scientific objectives of Chinese lunar exploration project and development strategy[J]. Advances in Earth Science, 2004, 19(3): 351-358(in Chinese).

[2] 李鑫, 刘莹莹, 周军. 载人登月舱上升入轨段的制导律设计[J]. 系统工程与电子技术, 2011, 33(11): 2480-2484.
LI X, LIU Y Y, ZHOU J. Design of guidance law for lunar ascent phase of manned lunar module[J]. Systems Engineering and Electronics, 2011, 33(11): 2480-2484(in Chinese).

[3] 马克茂, 陈海朋. 登月舱上升段最优轨迹设计[J]. 中国空间科学技术, 2013, 33(2): 54-60.
MA K M, CHEN H P. Optimal trajectory design for the lunar module in ascent stage[J]. Chinese Space Science and Technology, 2013, 33(2): 54-60(in Chinese).

[4] HULL D G, HARRIS M W. Optimal solutions for quasiplanar ascent over a spherical moon[J]. Journal of Guidance Control Dynamics, 2012, 35(4): 1218-1224.

[5] MA L, SHAO Z J, CHEN W F, et al. Three-dimensional trajectory optimization for lunar ascent using Gauss pseudospectral method[C]// Proceedings of the AIAA Guidance, Navigation, and Control Conference. Reston: AIAA, 2016.

[6] 张磊. 月面上升的轨迹优化与参数分析[J]. 深空探测学报, 2019, 6(4): 391-397.
ZHANG L. Trajectory optimization and parameter analysis for lunar ascent[J]. Journal of Deep Space Exploration, 2019, 6(4): 391-397(in Chinese).

[7] 宋征宇, 王聪, 巩庆海. 运载火箭上升段推力下降故障的自主轨迹规划方法[J]. 中国科学 (信息科学), 2019, 49(11): 1472-1487.
SONG Z Y, WANG C, GONG Q H. Autonomous trajectory planning for launch vehicle under thrust drop failure[J]. Scientia Sinica (Informationis), 2019, 49(11): 1472-1487(in Chinese).

[8] HAO Z M, ZHANG R. Onboard real-time generation of launch vehicle abort orbits[J]. Journal of Guidance Control Dynamics, 2021, 44(8): 1541-1549.

[9] BENEDIKTER B, ZAVOLI A, COLASURDO G, et al. Convex approach to three-dimensional launch vehicle ascent trajectory optimization[J]. Journal of Guidance Control Dynamics, 2021, 44(6):

1116-1131.

[10] 王华, 唐国金. 用遗传算法求解双冲量最优交会问题[J]. 中国空间科学技术, 2003, 23(1): 26-30.
WANG H, TANG G J. Solving optimal rendezvous using two impulses based on genetic algorithms[J]. Chinese Space Science and Technology, 2003, 23(1): 26-30(in Chinese).

[11] 冉茂鹏, 王青. 一种基于 EPSO 的航天器交会轨迹优化方法[J]. 宇航学报, 2013, 34(9): 1195-1201.
RAN M P, WANG Q. Spacecraft rendezvous trajectory optimization method based on EPSO[J]. Journal of Astronautics, 2013, 34(9): 1195-1201(in Chinese).

[12] 李君龙, 李松洲, 周获. 一种多约束条件下的三脉冲交会优化设计方法[J]. 系统工程与电子技术, 2022, 44(8): 2612-2620.
LI J L, LI S Z, ZHOU D. Optimization method for three-impulse rendezvous under multi-constraints[J]. Systems Engineering and Electronics, 2022, 44(8): 2612-2620(in Chinese).

[13] 任远, 崔平远, 栾恩杰. 最优两脉冲行星际轨道转移优化算法[J]. 航空学报, 2007, 28(6): 1307-1311.
REN Y, CUI P Y, LUAN E J. Interplanetary optimum two-impulse transfer trajectories[J]. Acta Aeronautica et Astronautica Sinica, 2007, 28(6): 1307-1311(in Chinese).

[14] YUE X C, YANG Y, GENG Z Y. Indirect optimization for finite-thrust time-optimal orbital maneuver[J]. Journal of Guidance, Control, and Dynamics, 2010, 33(2): 628-634.

[15] 武冠群. 在轨服务航天器交会轨迹优化与近距离安全接近控制研究[D]. 哈尔滨: 哈尔滨工业大学, 2019.
WU G Q. Research on rendezvous trajectory optimization and close-range safe approach control of on-orbit service spacecraft[D]. Harbin: Harbin Institute of Technology, 2019(in Chinese).

[16] 管凯颜. 不同质量航天器在连续推力下快速协同交会探讨[D]. 济南: 山东大学, 2021.
GUAN K Y. Discussion on rapid cooperative rendezvous of spacecraft with different masses under continuous thrust[D]. Jinan: Shandong University, 2021(in Chinese).

[17] 李萌, 龚胜平, 彭坤, 等. 直接优化算法在快速交会组合变轨策略中的应用[J]. 载人航天, 2017, 23(2): 156-162.
LI M, GONG S P, PENG K, et al. Application of direct optimization algorithm in synthetic orbit maneuver strategy of short rendezvous[J]. Manned Spaceflight, 2017, 23(2): 156-162(in Chinese).

[18] 周新耀, 陶勇鹏, 曾亮, 等. 空间轨道交会非线性优化方法研究[J]. 上海航天, 2018, 35(1): 54-58.
ZHOU X Y, TAO Y P, ZENG L, et al. Study on nonlinear optimization of space orbit interception[J]. Aerospace Shanghai (Chinese & English), 2018, 35(1): 54-58(in Chinese).

[19] ORTOLANO N, GELLER D K, AVERY A. Autonomous optimal trajectory planning for orbital rendezvous, satellite inspection, and final approach based on convex optimization[J]. The Journal of the Astronautical Sciences, 2021, 68(2): 444-479.

[20] 靳锴, 罗建军, 郑茂章, 等. 考虑导航误差和摄动影响的椭圆轨道最优交会制导[J]. 控制理论与应用, 2018, 35(10): 1484-1493.
JIN K, LUO J J, ZHENG M Z, et al. Guidance design with navigation errors for relative motion in noncircular perturbed orbits[J]. Control Theory & Applications, 2018, 35(10): 1484-1493(in Chinese).

[21] WANG Z B, GRANT M J. Optimization of minimum-time low-

thrust transfers using convex programming[J]. *Journal of Spacecraft and Rockets*, 2018, 55(3): 586-598.

[22] WANG Z B, GRANT M J. Minimum-fuel low-thrust transfers for spacecraft: a convex approach[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2018, 54(5): 2274-2290.

[23] 王访寒, 周如好, 余薛浩, 等. 基于序列凸优化的上面级远程交会轨迹优化[J]. *上海航天 (中英文)*, 2022, 39(2): 45-52.
WANG F H, ZHOU R H, YU X H, et al. Trajectory optimization of remote rendezvous for upper stage based on sequential convex optimization[J]. *Aerospace Shanghai(Chinese& English)*, 2022, 39(2): 45-52(in Chinese).

[24] 彭祺肇. 考虑应急返回能力的载人登月轨道优化设计及特性分析[D]. 长沙: 国防科学技术大学, 2012.
PENG Q B. Optimal design and characteristic analysis of manned moon orbit considering emergency return capability[D]. Changsha: National University of Defense Technology, 2012(in Chinese).

[25] STURM J F. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones[J]. *Optimization Methods and Software*, 1999, 11(1-4): 625-653.

[26] DOMAHIDI A, CHU E, BOYD S. ECOS: an SOCP solver for embedded systems[C]//*Proceedings of the 2013 European Control Conference*. Piscataway: IEEE Press, 2013: 3071-3076.

[27] DUERI D, ACIKMESE B, SCHARF D P, et al. Customized real-time interior-point methods for onboard powered-descent guidance[J]. *Journal of Guidance Control Dynamics*, 2017, 40(2): 197-212.

[28] MATTINGLEY J, BOYD S. CVXGEN: a code generator for embedded convex optimization[J]. *Optimization and Engineering*, 2012, 13(1): 1-27.

[29] JUNG J H, O'LEARY D P. Implementing an interior point method for linear programs on a CPU-GPU system[J]. *Electronic Transactions on Numerical Analysis*, 2008, 28: 174-189.

[30] WANG J W, ZHANG R, LI H F. Onboard optimization of multi-arc trajectories with constraints on duration of arcs[J]. *Acta Astronautica*, 2022, 192: 434-442.

[31] BOYD S. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. *Foundations and Trends® in Machine Learning*, 2010, 3(1): 1-122.

[32] O'DONOGHUE B, CHU E, PARIKH N, et al. Conic optimization via operator splitting and homogeneous self-dual embedding[J]. *Journal of Optimization Theory and Applications*, 2016, 169(3): 1042-1068.

[33] ELANGO P, KAMATH A, YU Y, et al. A customized first-order solver for real-time powered-descent guidance[C]//*Proceedings of the AIAA Scitech Forum*. Reston: AIAA, 2022: 0951.

[34] SKAJA A, ANDERSEN E D, YE Y Y. Warmstarting the homogeneous and self-dual interior point method for linear and conic quadratic problems[J]. *Mathematical Programming Computation*, 2013, 5(1): 1-25.

[35] CHEN Y S, YANG G W, WANG L, et al. Iteration complexity of an infeasible interior point methods for seconder-order cone programming and its warmstarting[EB/OL]. (2023-01-24)[2025-02-04]. <https://doi.org/10.48550/arXiv.2205.03570>.

[36] ZHANG Z Y, ZHAO Q C, DAI F A. A warm-start strategy in interior point methods for shrinking horizon model predictive control with variable discretization step[J]. *IEEE Transactions on Automatic Control*, 2023, 68(6): 3830-3837.

[37] CHEN Y S, YANG G W, WANG L, et al. A fast algorithm for onboard atmospheric powered descent guidance[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2023, 59(5): 6112-6123.

[38] 郝泽明, 张冉, 王嘉炜, 等. 大气层内固体火箭实时轨迹优化方法[J]. *宇航学报*, 2021, 42(11): 1416-1426.
HAO Z M, ZHANG R, WANG J W, et al. Real-time atmospheric trajectory optimization for solid rockets[J]. *Journal of Astronautics*, 2021, 42(11): 1416-1426(in Chinese).

[39] VANDENBERGHE L. The CVXOPT linear and quadratic cone program solvers[EB/OL]. (2010-03-20)[2023-03-28]. <http://cvxopt.org/documentation/coneprog.pdf>.

[40] DOMAHIDI A. Methods and tools for embedded optimization and control[D]. Zurich: ETH Zurich, 2013.

[41] DAVIS T A, GILBERT J R, LARIMORE S I, et al. A column approximate minimum degree ordering algorithm[J]. *ACM Transactions on Mathematical Software*, 2004, 30(3): 353-376.

[42] GEORGE A, LIU J W H. The evolution of the minimum degree ordering algorithm[J]. *SIAM Review*, 1989, 31(1): 1-19.

[43] DUFF I S, ERISMAN A M, REID J K. Direct methods for sparse matrices[M]. New York: Oxford University Press, 2017.

[44] GILL P E, SAUNDERS M A, SHINNERL J R. On the stability of cholesky factorization for symmetric quasidefinite systems[J]. *SIAM Journal on Matrix Analysis and Applications*, 1996, 17(1): 35-46.

[45] DAVIS T A. Algorithm 849[J]. *ACM Transactions on Mathematical Software*, 2005, 31(4): 587-591.

[46] GAO Y, SHAO Z J, SONG Z Y. Enhanced successive convexification based on error-feedback index and line search filter[J]. *Journal of Guidance Control Dynamics*, 2022, 45(12): 2243-2257.

Rapid planning method for lunar direct ascent and rendezvous trajectory in emergency cases

BAN Huanheng^{1, 2}, ZHOU Cong^{1, 2}, YAN Xiaodong^{1, 2, *}

(1. School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China;
2. Shaanxi Aerospace Flight Vehicle Design Key Laboratory, Xi'an 710068, China)

Abstract: When a lunar detector encounters unexpected conditions and needs a return to the lunar orbiter or Earth, the lunar ascender should be able to plan an appropriate ascent and rendezvous trajectory independently in emergency cases. In this paper, a direct ascent and rendezvous trajectory optimization model was established based on a successive second-order cone programming method, and minimum ascent and rendezvous time was chosen as the objective function. Additionally, a convex optimization algorithm was proposed to solve the ascent and rendezvous trajectory optimization problem. In order to enhance computational efficiency, the interior point method was customized and modified. The main adaptations included: the process of solving linear equations was customized and modified; a warm starting was used in the interior point method; the second-order cone programming subproblem solving accuracy was adjusted dynamically. Simulation results demonstrate that the proposed lunar ascent and rendezvous trajectory optimization method in emergency cases facilitates the rapid ascent and rendezvous process of the lunar ascender. Moreover, when compared to the traditional successive second-order cone programming method employing a general interior point method solver, the proposed acceleration technique achieves a speedup ratio of approximately 9.5 times while maintaining the same level of solving accuracy. Consequently, the method outlined in this paper holds significant potential for enabling autonomous online trajectory planning of lunar ascenders.

Keywords: lunar ascent and rendezvous process in emergency cases; successive second-order cone programming method; customized interior point method; warm starting in interior point method; online trajectory planning