

http://fhlx.cbpt.cnki.net fxlbjb@163.com
DOI: 10.13645/j.cnki.f.d.20220929.006

垂直起降运载火箭返回着陆轨迹在线优化研究

姜雪梅¹, 朱博灵², 赵吉松², 王泊乔², 张金明²

(1. 中国飞行试验研究院, 陕西 西安 710089;

2. 南京航空航天大学 航天学院, 江苏 南京 210016)

摘 要: 针对垂直起降运载火箭返回着陆轨迹优化问题开展了研究。首先, 基于多区间 Radau 伪谱法和稀疏非线性规划技术, 采用 C++ 语言编程开发了具有在线优化能力的轨迹优化程序。然后, 针对火箭返回着陆轨迹的落点误差较大问题, 提出了末段轨迹在线更新的应用模式, 以消除轨迹累积误差。最后, 开展了仿真计算。研究表明: 所提方法能够快速求解运载火箭返回着陆轨迹优化问题, 计算效率比 GPOPS-II 软件提高大约 3~12 倍, 并且可以移植到嵌入式系统, 具有在线优化的能力; 对末段轨迹进行在线更新能够有效提高返回轨迹的落点精度。

关 键 词: 运载火箭; 返回着陆; 轨迹优化

中图分类号: V475.1

文献标识码: A 文章编号: 1002-0853(2023)01-0063-08

Online trajectory optimization for descent and landing of vertical takeoff and landing launch rocket

JIANG Xuemei¹, ZHU Boling², ZHAO Jisong², WANG Boqiao², ZHANG Jinming²

(1. Chinese Flight Test Establishment, Xi'an 710089, China;

2. College of Astronautics, NUAA, Nanjing 210016, China)

Abstract: The optimization of descent and landing trajectory of vertical takeoff and landing launch rocket was studied. Firstly, a trajectory optimization method and program with online capability was developed using C++ language based on multi-interval Radau pseudospectral method and sparse non-linear programming (NLP). Then, an application mode of online update of the terminal trajectory was proposed to eliminate the cumulative trajectory error. Finally, a numerical simulation was performed. The results show that the proposed method can quickly generate a fuel-optimal descent and landing trajectory for a launch rocket, and the computational efficiency is about 3~12 times higher than that of the GPOPS-II software. The method can be easily ported to embedded systems and thus has the potential of online optimization. The online optimization of the terminal trajectory can significantly improve the rocket landing accuracy.

Key words: launch rocket; descent and landing; trajectory optimization

0 引言

火箭返回与精准软着陆技术是研制可重复使

用火箭的一项重要支撑技术。近年来, 随着太空探索技术公司的 Falcon-9 系列火箭一级多次成功回收与重复使用, 垂直起降可重复使用火箭返回

收稿日期: 2022-03-22; 修订日期: 2022-09-08; 网络出版时间: 2022-10-08 17:33

基金项目: 国家自然科学基金资助(11602107)

作者简介: 姜雪梅(1990—), 女, 陕西西安人, 工程师, 硕士。

通信作者: 赵吉松(1984—), 男, 安徽寿县人, 副教授, 博士。

着落技术逐渐成为国内外研究的热点^[1-5]。

不同于传统的再入返回飞行器,垂直起降运载火箭子级返回过程中,需要利用火箭的推力实现大范围减速并满足过程约束和定点垂直着陆约束。显然,对火箭返回着落轨迹进行优化,减少再入返回阶段的燃料消耗对提高火箭运载能力和应急能力具有重要意义。目前,不少研究人员基于凸优化方法求解火箭返回着落轨迹优化问题,取得了一系列具有应用潜力的研究成果^[1-4]。凸优化方法的优势是不需要优化初值,收敛较快,并且编程简单,容易在嵌入式系统上运行实现在线优化。凸优化的主要问题是需要对返回着落轨迹进行简化和复杂推导,将其转化为凸优化问题。比如,当考虑气动力时,火箭返回着落轨迹不再是凸优化问题,目前的方法是将火箭返回着落轨迹沿着参考轨迹进行线性化,将其转换为一组凸优化问题,然后通过迭代求解,即序列凸优化(Sequential Convex Programming, SCP)技术。但是,SCP技术一方面需要提供初始轨迹作为线性化的参考轨迹,从而丧失了传统凸优化不需要提供初值的优势;另一方面,多次迭代求解(典型迭代次数为10次左右)降低了凸优化的计算效率,不利于在线优化的快速性要求。随着问题复杂度的增加,比如考虑气动力且气动数据是表格形式,那么SCP技术将更加复杂。

除了凸优化方法,还可以采用传统的直接法求解火箭返回着陆轨迹优化问题。直接法通过对控制变量和状态变量进行离散把连续最优控制问题转化为非线性规划(Nonlinear Programming, NLP),然后采用NLP算法^[6-7]求解。直接法中的配点法由于不需要推导最优性必要条件,对初值的敏感性较低,容易收敛,近年来得到广泛研究和应用。多区间Radau伪谱法^[8]是配点法中的一种,它将时间分成多个区间,在每个区间采用低阶插值多项式在LGR(Legendre Gauss Radau)配点上近似状态变量。与在整个时间区间应用高阶插值多项式的传统伪谱法相比,多区间Radau伪谱法离散得到的NLP偏导数矩阵更稀疏,具有更高的计算效率。近年来,多区间Radau伪谱法依托于开源软件GPOPS-II^[8]在航空航天领域得到广泛研究和应用。

基于配点法的轨迹优化方法由于需要调用依赖于操作系统的NLP求解器(SNOPT、IPOPT

等),以前只能在桌面计算机上运行。近年来,嵌入式计算机技术得到了快速发展,市场上出现了大量基于Linux系统的嵌入式开发板,比如树莓派等,使得将基于配点法的轨迹优化方法移植到嵌入式开发板成为可能。比如,Agamawi等^[9]最近基于C++语言开发的CGPOPS软件可以移植到支持Linux的嵌入式系统上运行。

本文基于多区间Radau伪谱配点法,采用C++语言开发具有在线优化潜力的轨迹优化程序,在此基础上开展火箭返回着落轨迹优化研究,提出了在线优化末段轨迹的应用模式,并通过数值仿真验证了所述方法和应用模式的有效性。

1 火箭返回着陆轨迹优化问题

1.1 火箭返回着陆段质心运动方程

本文采用地球固定坐标系,原点位于着陆点,单位矢量 e_1 指向上方, e_2 指向东方, e_3 与其他单位矢量组成右手系。在该坐标系中,描述火箭下降过程中质心运动的微分方程组^[4]为:

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{g} + \frac{\mathbf{T}}{m} - \frac{\mathbf{D}}{m} \\ \dot{m} = -\alpha \|\mathbf{T}\| \end{cases} \quad (1)$$

式中: $\mathbf{r} = (r_1, r_2, r_3)^T \in \mathbf{R}^3$ 为位置矢量; $\mathbf{v} = (v_1, v_2, v_3)^T \in \mathbf{R}^3$ 为速度矢量; m 为质量; $\mathbf{g} = (-g, 0, 0)^T \in \mathbf{R}^3$ 为重力加速度常矢量; $\mathbf{T} = (T_1, T_2, T_3)^T \in \mathbf{R}^3$ 为发动机推力矢量; \mathbf{D} 为气动阻力矢量; $\alpha = 1/(I_{sp}g_0)$ 是与燃油消耗率相关的正常数; I_{sp} 为燃料比冲; g_0 为海平面重力加速度。

本文假设火箭飞行过程升阻比很低,升力可以忽略不计。无量纲阻力矢量 \mathbf{D} 为:

$$\mathbf{D} = \frac{1}{2} \rho S_{ref} C_D \|\mathbf{v}\| \mathbf{v} \quad (2)$$

式中: ρ 为空气密度; S_{ref} 为火箭参考面积; C_D 为阻力系数。因为火箭的动力下降阶段通常处于亚声速飞行状态,本文认为阻力系数为常数。

1.2 初始条件

运载火箭返回着陆问题的初始条件如下:

$$\begin{cases} \mathbf{r}(t_0) = \mathbf{r}_0 \\ \mathbf{v}(t_0) = \mathbf{v}_0 \\ m(t_0) = m_0 \end{cases} \quad (3)$$

式中: t_0 为初始时刻; \mathbf{r}_0 、 \mathbf{v}_0 、 m_0 分别为位置向量、速度向量和火箭质量在初始时刻的取值。

1.3 路径约束

运载火箭返回着陆问题的控制变量为发动机的推力矢量,其调节范围受到以下限制:

$$T_{\min} \leq \|T(t)\| \leq T_{\max} \quad (4)$$

式中: T_{\min} 、 T_{\max} 分别为火箭的最小和最大推力。

在火箭返回着陆过程中需要对火箭俯仰角进行约束。本文假设火箭推力沿轴线方向,因而推力方向与竖直方向的夹角需要满足如下约束:

$$\|T(t)\| \cos \theta_{\max} \leq T_1(t) \quad (5)$$

式中: θ_{\max} 为推力倾斜角的上限; $T_1(t)$ 为沿 r_1 方向的推力分量。当火箭靠近着陆点时, θ_{\max} 应该足够小以避免着陆时侧翻。

另一个重要的约束条件是要求运载火箭在下降着陆过程中始终位于锥角逐渐变小的圆锥体内,以实现最终接近垂直着陆,即在下降着陆过程中火箭的下滑倾斜角不能超过 γ_{\max} ,即:

$$\|r(t)\|_2 \cos \gamma_{\max} \leq r_1 \quad (6)$$

式中: γ_{\max} 为允许的最大下滑倾斜角。

式(5)和式(6)中的最大推力倾斜角 θ_{\max} 和最大下滑倾斜角 γ_{\max} 可以随 r_1 分量变化。此外,当火箭靠近着陆点时, θ_{\max} 应该足够小以避免火箭在着陆时发生侧翻。在后续仿真计算时,将采用随高度变化的 θ_{\max} 和 γ_{\max} 函数。

1.4 末端约束

为了精确和安全着陆,还需要对火箭的着陆点位置和着陆速度进行约束。终端约束条件为:

$$\begin{cases} r(t_f) = r_f \\ v(t_f) = v_f \end{cases} \quad (7)$$

式中: t_f 为待优化的火箭着陆时刻; r_f 为要求的火箭着陆位置; v_f 为要求的火箭着陆速度。

1.5 目标函数

火箭下降着陆轨迹优化问题的目标函数通常取着陆过程燃料消耗最少(等效于着陆时刻火箭的剩余质量最大)。因此,目标函数可以写为:

$$\text{maximize } J = m(t_f) \quad (8)$$

2 火箭返回着陆轨迹在线优化方法

2.1 多区间 Radau 伪谱法

本文采用多区间伪谱法^[8]求解垂直起降火箭返回着陆轨迹优化问题。为了便于描述,本节以一般形式的 Bolza 型最优控制问题为例,介绍多区间伪谱法。Bolza 型问题可描述为:求解控制变量 $u(t) \in \mathbf{R}^m$,使得如下目标函数最小化:

$$J = M(x(t_0), t_0, x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t) dt \quad (9)$$

式中: $M: \mathbf{R}^n \times \mathbf{R} \times \mathbf{R}^n \times \mathbf{R} \rightarrow \mathbf{R}$, $L: \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R} \rightarrow \mathbf{R}$, $x \in \mathbf{R}^n$, $u \in \mathbf{R}^m$, $t \in [t_0, t_f] \subseteq \mathbf{R}$ 。

状态方程为:

$$\dot{x} = f(x(t), u(t), t) \quad (t \in [t_0, t_f]) \quad (10)$$

端点条件为:

$$e(x(t_0), t_0, x(t_f), t_f) = 0 \quad (11)$$

路径约束为:

$$c(x(t), u(t), t) \leq 0 \quad (t \in [t_0, t_f]) \quad (12)$$

式中: $f: \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R} \rightarrow \mathbf{R}^n$, $e: \mathbf{R}^n \times \mathbf{R} \times \mathbf{R}^n \times \mathbf{R} \rightarrow \mathbf{R}^e$, $c: \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R} \rightarrow \mathbf{R}^c$ 。

式(9)~(12)所描述的连续最优控制问题称为 Bolza 型最优控制问题。

为了便于应用伪谱法对上述最优控制问题进行离散化,需要将原始最优控制问题的时间区间 $t \in [t_0, t_f]$ 变换至 $\tau \in [-1, 1]$,变换方式如下:

$$t = \frac{t_f - t_0}{2} \tau + \frac{t_f + t_0}{2} \quad (13)$$

将积分变量变化范围 $\tau \in [-1, 1]$ 划分为 K 个网格区间 $I_k = [T_{k-1}, T_K]$ ($k = 1, \dots, K$),其中 $-1 = T_0 < T_1 < \dots < T_K = T_f = 1$ 。记 $x^{(k)}(\tau)$ 为区间 I_k ($k = 1, \dots, K$) 上的状态变量,那么状态变量可以近似为:

$$x^{(k)}(\tau) \approx y^{(k)}(\tau) = \sum_{j=1}^{N_k+1} x_j^{(k)} l_j^{(k)}(\tau) \quad (14)$$

式中: $y^{(k)}(\tau)$ 为 $x^{(k)}(\tau)$ 的近似函数; $x_j^{(k)}$ 为状态变量在离散节点 T_k 处的取值,表达式 $l_j^{(k)}(\tau)$ 为如下 Lagrange 基本多项式:

$$l_j^{(k)}(\tau) = \prod_{l=1, l \neq j}^{N_k+1} \frac{\tau - \tau_l^{(k)}}{\tau_j^{(k)} - \tau_l^{(k)}} \quad (15)$$

其中:节点 $(\tau_1^{(k)}, \dots, \tau_{N_k}^{(k)})$ 是区间 $I_k = [T_{k-1}, T_K]$ 上的 N_k 个 LGR (Legendre-Gauss-Radau) 配点,但是区间 I_k 的右端点 $\tau_{N_k+1}^{(k)} = T_k$ 不是 LGR 配点。

将式(14)中 $y^{(k)}(\tau)$ 关于 τ 求微分得到:

$$\frac{dy^{(k)}(\tau)}{d\tau} = \sum_{j=1}^{N_k+1} x_j^{(k)} \frac{dl_j^{(k)}(\tau)}{d\tau} \quad (16)$$

将式(10)和式(16)相结合,可得到状态方程在区间 I_k ($k = 1, \dots, K$) 上的离散形式为:

$$\sum_{j=1}^{N_k+1} D_{ij}^{(k)} x_j^{(k)} - \frac{t_f - t_0}{2} f(x_i^{(k)}, u_i^{(k)}, \tau_i^{(k)}) = 0$$

$$(i = 1, \dots, N_k) \quad (17)$$

式中: 区间 I_k 上的 LGR 微分矩阵的元素为 $D_{ij}^{(k)} = dl_j^{(k)}(\tau_i^{(k)})/d\tau$ ($i = 1, \dots, N_k; j = 1, \dots, N_k + 1$); $u_i^{(k)}$ ($i = 1, \dots, N_k$) 为控制变量 $u^{(k)}(\tau)$ 的离散值。

目标函数式(9)的离散形式为:

$$J \approx M(x_1^{(1)}, t_0, x_{N_k+1}^{(K)}, t_f) + \sum_{k=1}^K \sum_{j=1}^{N_k} \frac{t_f - t_0}{2} \omega_j^{(k)} L(x_j^{(k)}, u_j^{(k)}, \pi_j^{(k)}) \quad (18)$$

式中: $\omega_j^{(k)}$ 为 $\tau_j^{(k)}$ 对应的 LGR 权重系数。

端点条件式(11)的离散形式为:

$$e(x_1^{(1)}, \pi_0, x_{N_k+1}^{(K)}, \pi_f) = 0 \quad (19)$$

路径约束式(12)的离散形式为:

$$c(x_i^{(k)}, u_i^{(k)}, \pi_i^{(k)}) \leq 0 \quad (i = 1, \dots, N_k) \quad (20)$$

由于区间 I_k 的右端点 $\tau_{N_k+1}^{(k)} = T_k$ 不是 LGR 配点, 为使得状态变量在相邻区间连续, 需满足:

$$x_{N_k+1}^{(k)} = x_1^{(k+1)} \quad (k = 1, \dots, K-1) \quad (21)$$

采用相同的变量表示 $x_{N_k+1}^{(k)}$ 和 $x_1^{(k+1)}$, 可消除该约束。

采用数值方法求解方程式(17)~式(21)描述的 NLP 问题, 即可得到轨迹优化问题的离散最优解。

2.2 在线轨迹优化程序开发

本文采用基于 SQP 算法开发的求解器 SNOPT^[6] 求解 NLP 问题, 得到轨迹优化问题的离散最优解。在采用 SNOPT 求解 NLP 时, 为 SNOPT 提供 NLP 的一阶偏导数, 即雅克比矩阵(定义为目标函数和全部约束对全部自变量的一阶偏导数矩阵), 能够显著提高计算效率, 但是直接计算偏导数矩阵的计算量比较大。稀疏差分法^[10]通过分析偏导数矩阵的稀疏特性, 将其中的占绝大多数的零元素识别出来, 并且将其中的非零元素的值通过矩阵链式求导运算分解为原始轨迹优化问题的偏导数, 然后采用稀疏差分方法计算, 从而有效减小偏导数的计算量。基于以上思路, 本文针对一般形式的轨迹优化问题, 开发了通用的轨迹优化程序。本文研究工作的一个重要特色之处是采用 C++ 语言编程, 集成了可供 C++ 调用的 NLP 求解器。相对于其他轨迹优化研究中常用的 MATLAB 语言, C++ 编程一方面能够显著提高轨迹优化的计算效率, 更重要的优势是使得开发的轨迹优化程序很容易移植到嵌入式系统平台上运行, 从而具有在线优化轨迹的能力。

2.3 火箭返回着陆轨迹优化策略

本文研究的火箭返回着陆轨迹优化问题的最优推力大小随时间变化是 Bang-Bang 控制, 增加离散节点数量或者引入节点细化技术^[11]都能够取得较高的精度, 但是这种处理方式会增加计算时间, 不利于在线优化。为了能够在尽量短的时间内优化出最优轨迹, 并且具有较高的优化精度, 本文在前期研究工作^[12]的基础上, 引入末段优化策略, 如图 1 所示。

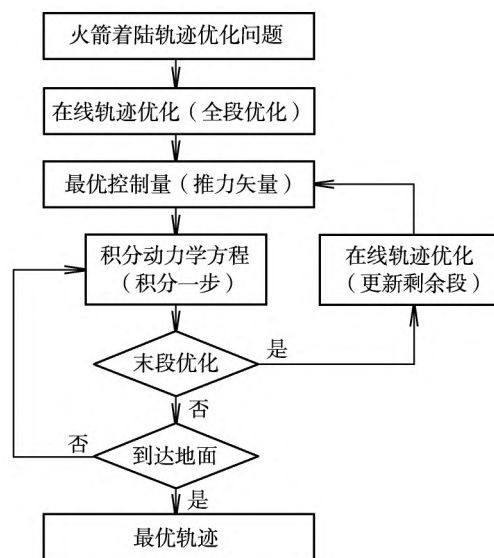


图 1 火箭返回着陆轨迹优化策略

Fig. 1 Strategy for rocket landing trajectory optimization

首先, 采用多区间 Radau 伪谱法对全段返回着陆轨迹进行在线优化, 得到最优推力矢量; 然后, 根据优化出的最优推力矢量采用数值方法积分动力学方程; 当积分轨迹与优化的离散轨迹的误差超过预定值、或者达到预定的更新周期等条件时, 对剩余轨迹重新优化, 更新最优推力矢量; 最后, 根据更新的最优推力矢量继续积分动力学方程至地面。如有必要, 可以多次更新剩余段轨迹。其中, 积分动力学方程的主要目的是对返回着陆轨迹进行模拟仿真。虽然目前只考虑了离散格式误差, 但是无论源于何种误差, 实际飞行轨迹与离散最优轨迹之间存在偏差产生的影响是类似的。在着陆过程中在线更新最优轨迹是本文的一个特色, 也是基于 C++ 语言开发的轨迹优化方法的优势。

3 仿真计算及结果分析

本文仿真研究采用的模型参数来自文献[4]。其中, 推力可调火箭发动机的最大推力

$T_{\max}=300\text{ kN}$,最小推力 $T_{\min}=0.4T_{\max}$,发动机燃料比冲 $I_{\text{sp}}=360\text{ s}$,海平面重力加速度 $g_0=9.8\text{ m/s}^2$,火箭的阻力系数 $C_D=1.3$,气动参考面积 $S_{\text{ref}}=9\text{ m}^2$,大气密度随高度变化特性根据美国 1976 年版本的标准大气模型插值计算。状态变量和时间的边界条件如表 1 所示。

表 1 状态变量和时间的边界条件

Table 1 Boundary conditions of state variables and time

变量	初始条件	末端条件
r_1/m	5 000	0
r_2/m	1 000	0
r_3/m	-1 800	0
$v_1/\text{m}\cdot\text{s}^{-1}$	-200	0
$v_2/\text{m}\cdot\text{s}^{-1}$	40	0
$v_3/\text{m}\cdot\text{s}^{-1}$	150	0
m/kg	23 000	自由
t/s	0	自由

在火箭下降着陆过程中,允许的发动机最大推力倾斜角 θ_{\max} 随火箭高度的变化特性如下:

$$\theta_{\max}=\begin{cases}30 & (r_1\geq 300)\\r_1/10 & (0\leq r_1<300)\end{cases}\quad (22)$$

在火箭下降着陆过程中,允许的火箭最大下滑倾斜角 γ_{\max} 随火箭高度的变化特性如下:

$$\gamma_{\max}=\begin{cases}\inf & (r_1\geq 1\,250)\\2r_1/125 & (1\,000\leq r_1<1\,250)\\r_1/225+104/9 & (100\leq r_1<1\,000)\\20 & (r_1\leq 100)\end{cases}\quad (23)$$

本文仿真采用的计算平台为桌面计算机(处理器 Intel Core i5-4670,内存 4 GB)。本文调用 C++版本的 SNOPT 7 求解 NLP,采用有限差分法为其提供一阶偏导数,编程语言为 C++11。本文中给出的优化耗时为 10 次优化的平均耗时。

3.1 全段优化

本文首先不考虑末段在线优化,在整个时间区间对运载火箭的下降着陆轨迹进行优化。采用 10 个网格区间(每个区间 4 个配点)状态变量的优化初值取初始时刻和终端时刻取值的连线(对于给定终值)或者取初值常值(对于未给定终值),控制变量优化初值第一个分量为 T_{\max} ,其他优化初值取 0。本文开发的 C++程序耗时 0.34 s 完成下降着陆轨迹的优化。优化的推力分量推

值如图 2 所示。

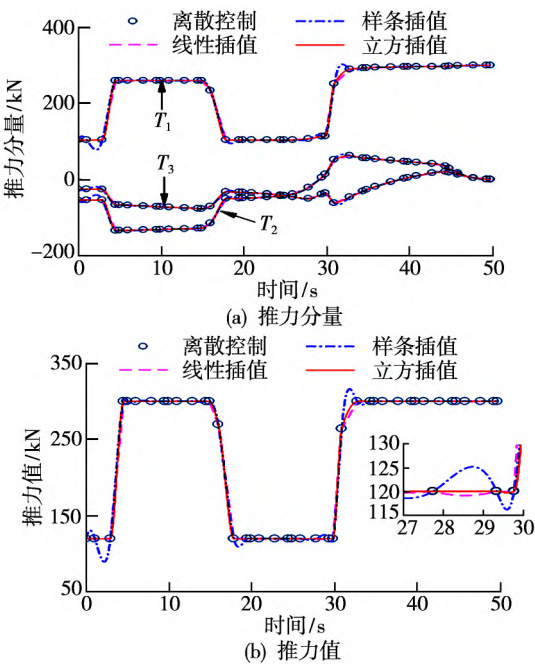


图 2 全段优化下的推力分量及总推力大小

Fig. 2 Thrust components and total thrust magnitude for full segment optimization

从图 2 中可以看出,样条插值的推力大小发生明显的振荡,线性插值的推力大小也发生振荡。实际上,立方插值有时候也会导致推力大小发生振荡,只是在图 2(b) 所示的案例中立方插值的振荡程度比较轻微。推力大小发生振荡的直接影响是推力会超出允许的变化范围,如本算例要求推力大小介于 T_{\min} 和 T_{\max} 之间,采用插值构造的连续推力一旦超出范围就不是可行解。

为了使得插值得到的连续推力满足约束要求,本文对其大小进行限幅,使其介于 T_{\min} 和 T_{\max} 之间。此外,还需要考虑推力分量与大小之间的一致性,即对推力矢量进行如下处理:

$$\boldsymbol{T}(t)=\boldsymbol{T}(t)s(t)\quad (24)$$

式中: $s(t)$ 为缩放因子,取值方式如下:

$$s(t)=\begin{cases}T_{\min}/\|\boldsymbol{T}(t)\| & (\|\boldsymbol{T}(t)\|<T_{\min})\\1 & (T_{\min}\leq\|\boldsymbol{T}(t)\|<T_{\max})\\T_{\max}/\|\boldsymbol{T}(t)\| & (\|\boldsymbol{T}(t)\|\geq T_{\max})\end{cases}\quad (25)$$

图 3 给出了本文优化的离散最优轨迹与数值积分轨迹对比。图中,圆圈为离散最优轨迹,细实线为数值积分轨迹,积分时采用立方插值方法计算控制变量。从图 3 中可以看到,在轨迹终端,积分轨迹和离散最优轨迹存在较大差异。

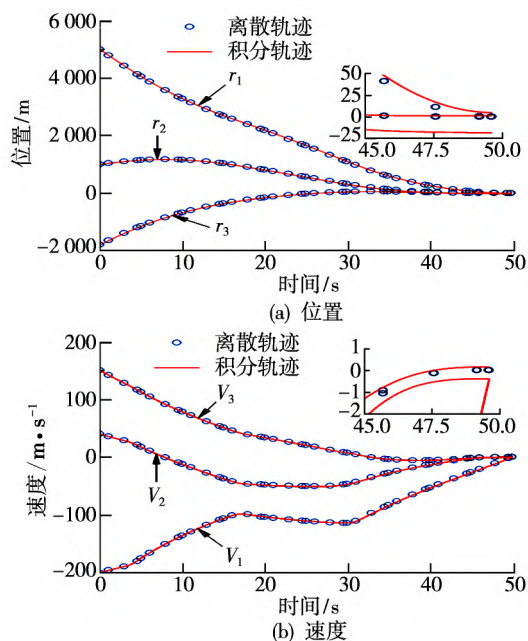


图 3 全段优化下的位置和速度

Fig. 3 Position and velocity for full segment optimization

表 2 给出了状态变量的终端误差,包括采用不同插值方式构建连续控制变量时得到的终端状态误差。

表 2 全段优化下的状态变量终端误差

Table 2 Terminal state errors for full segment optimization

插值方式	位置误差/m			速度误差/m·s ⁻¹		
	Δr_1	Δr_2	Δr_3	Δv_1	Δv_2	Δv_3
线性插值	6.32	-22.04	2.00	-0.48	-0.43	0.23
样条插值	29.18	-31.72	-4.94	0.16	-0.56	-0.01
立方插值	4.93	-18.54	1.32	-0.39	-0.38	0.15

从表 2 中可以看到,无论采用哪种插值方式,数值积分轨迹的终端误差都比较大,其中样条插值对应的误差最大。这是由于两方面原因造成的;一方面,多区间伪谱法的一个特点就是通过牺牲离散精度提高优化速度;另一方面,通过插值构造的连续控制变量在推力大小切换附近发生振荡,需要限幅处理,导致精度降低。为了减小积分轨迹的终端误差,可以增加节点数量或者引入节点细化技术,但是这样会增加轨迹优化时间,不利于在线优化的快速性要求。

3.2 全段优化+末段优化

末段优化的初始点可以根据实际轨迹与设计轨迹的偏差大小选取,也可以按照预定的时间周期选取。不同的选取方式对轨迹优化没有本质影响,只影响末段轨迹优化的初始条件。作为示例,本文在火箭下降至离地面高度 300 m 时对末段轨

迹开展在线优化,此时积分轨迹与离散最优轨迹的位置误差为 17.23 m。为了使得最优推力分量连续变化,本文在末段在线优化的初始时刻添加控制变量连续约束,使推力分量与全段优化的结果保持连续。采用 5 个网格区间(每个区间 4 个配点),采用前述全段优化的结果为末段优化提供优化初值。在桌面计算机上,本文开发的 C++ 程序耗时 0.063 s 完成下降着陆轨迹的末段优化。

考虑末端优化的最优推力分量及推力值如图 4 所示。可以看到,末段优化得到的推力分量与全段优化的结果保持连续。

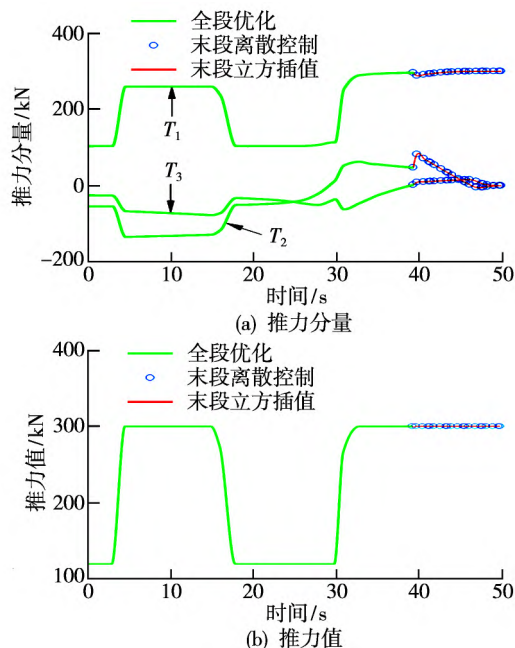
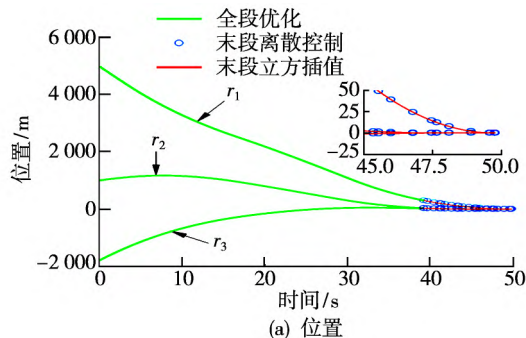


图 4 考虑末端优化的推力分量及推力值

Fig. 4 Thrust components and total thrust magnitude with terminal optimization considered

图 5 给出了本文方法优化的离散最优轨迹与积分轨迹的对比。对比图 3 和图 5 可知,经过末段在线优化,返回着陆轨迹的终端误差显著降低,位置、速度的终端误差分别为 0.06 m、0.14 m/s。



(a) 位置

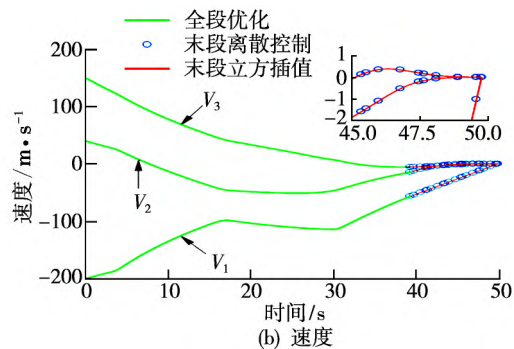


图 5 考虑末段优化的位置和速度

Fig. 5 Position and velocity with terminal optimization considered

图 6 给出了本文方法优化的火箭下降着陆过程的推力倾斜角和视线角约束变化情况。可以看到,采用本文方法优化的下降着陆轨迹满足路径约束要求。

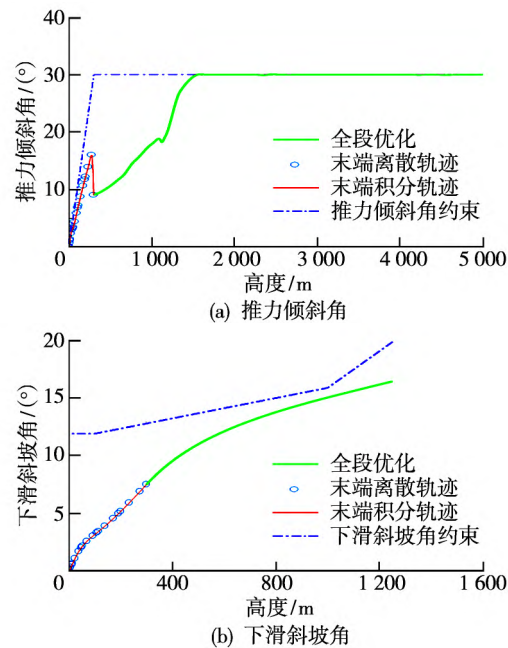


图 6 考虑末段优化的推力倾斜角和下滑斜坡角

Fig. 6 Thrust tilt angle and glide slope angle with terminal optimization considered

表 3 给出了全段优化和末段优化的目标函数、计算耗时以及终端状态误差对比情况。表中还将在全段优化的基础上引入节点细化技术得到的高精度结果作为对比。可以看到,与引入节点细化技术的高精度轨迹相比,全段优化的目标函数与之接近,但是终端位置和速度误差较大。通过对末段轨迹进行在线优化,显著提高了着陆轨迹的落点精度。末段优化的着陆质量为 19 670. 49 kg,比节点细化技术的

结果 19 683. 94 kg 减小 13. 4 kg (0. 068%) 。尽管目前本文中考虑的轨迹终端误差是由于离散格式误差累积产生的,但是误差源的不同没有本质区别,各种轨迹跟踪误差都可以通过末段在线优化予以消除。本算例末段在线优化的计算效率比较高,耗时仅为 63 ms。一方面是因为采用 C++ 编写的代码效率高;另一方面,采用了全段优化结果提供初值,加快了收敛速度。可见,对末段进行在线更新能够以较小的计算代价显著提高火箭着陆轨迹的落点精度。

表 3 最优下降着陆轨迹的状态变量终端误差

Table 3 Terminal state error of optimal descent and landing trajectory

优化算法	着陆	优化	终端位置 误差/m	终端速度 误差/ $\text{m} \cdot \text{s}^{-1}$
	质量/kg	耗时/s		
全段优化	19 683. 94	0. 34	19. 23	0. 57
加入节点 细化	19 682. 57	5. 59	0. 96	0. 02
加入末段 优化	19 670. 49	0. 06	0. 14	0. 02

表 4 给出了本文方法与开源软件 GPOPS-II 优化的目标函数和计算效率对比。对于末段优化,原始 GPOPS-II 不支持直接对控制变量初值进行约束,本文通过修改其底层库函数使其具备这一功能。

表 4 下降着陆轨迹的不同算法性能对比

Table 4 Performance comparison of different algorithms for solving descent and landing trajectory

优化算法	性能指标	本文方法 (C++)	GPOPS-II (MATLAB)
全段优化	着陆质量/kg	19 683. 94	19 683. 94
	优化耗时/s	0. 34	1. 04
加入末段 优化	着陆质量/kg	19 670. 49	19 670. 49
	优化耗时/s	0. 06	0. 73

由表 4 可以看到,本文方法与改进的 GPOPS-II 计算的目标函数均完全一致,这是因为本文方法与 GPOPS-II 程序都是基于多区间伪谱法。本文方法的优化效率比 GPOPS-II 提高了 3~12 倍,其主要原因是 GPOPS-II 采用 MATLAB 语言开发,而本文方法采用 C++ 语言开发,代码执行效率更高。Agamawi 等^[9]最新基于 C++ 语言开发的 CG-POPS 相对于 GPOPS-II 的计算效率提高大约 2~10 倍,本文的效率提升与之基本一致。

本文采用 C++ 语言开发轨迹优化程序的另一个重要优势是具有在嵌入式系统运行的潜力。目前能够运行 Linux 系统的嵌入式开发板很多, 因此采用 C++ 语言开发的程序容易移植到嵌入式系统上。此外, 由于嵌入式开发的灵活性, 甚至可以根据本文轨迹优化程序的需要自行移植 Linux 系统, 进一步降低对硬件性能的要求。

4 结束语

本文基于多区间 Radau 伪谱法和 C++ 编程技术开发了具有在线优化潜力的轨迹优化程序。在此基础上, 给出了火箭返回着陆轨迹的优化方法, 特别是针对全段优化结果落点误差较大问题, 提出了末段轨迹在线更新的应用模式。

仿真结果表明, 本文方法能够快速更新末段轨迹, 消除累积误差的影响, 显著提高返回着陆轨迹的落点精度。对于运载火箭着陆轨迹优化问题, 本文方法的计算效率比开源软件 GPOPS-II 提高大约 3~12 倍。此外, 本文发展的轨迹优化方法能够移植到嵌入式系统平台上运行, 具有在线轨迹优化的潜力。

参考文献:

- [1] 张志国, 马英, 耿光有, 等. 火箭垂直回收着陆段在线制导凸优化方法 [J]. 弹道学报, 2017, 29(1): 9-16.
- [2] 邵楠, 闫晓东. 火箭垂直回收多阶段最优轨迹规划方法 [J]. 宇航学报, 2019, 40(10): 1187-1196.
- [3] 韦常柱, 琚啸哲, 徐大富, 等. 垂直起降重复使用运载器返回制导与控制 [J]. 航空学报, 2019, 40(7): 197-220.
- [4] YANG R Q, LIU X F. Fuel-optimal powered descent guidance with free final-time and path constraints [J]. Acta Astronautica, 2020, 172: 70-81.
- [5] 宋征宇, 蔡巧言, 韩鹏鑫, 等. 重复使用运载器制导与控制技术综述 [J]. 航空学报, 2021, 42(11): 37-65.
- [6] GILL P E, MURRAY W, SAUNDERS M A. SNOPT: an SQP algorithm for large-scale constrained optimization [J]. SIAM Review, 2005, 47(1): 99-131.
- [7] BIEGLER L T, ZAVALA V M. Large-scale nonlinear programming using IPOPT: an integrating framework for enterprise-wide dynamic optimization [J]. Computers and Chemical Engineering, 2009, 33(3): 575-582.
- [8] PATTERSON M A, RAO A V. GPOPS-II: a MATLAB software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming [J]. ACM Transactions on Mathematical Software, 2014, 41(1): 1-37.
- [9] AGAMAWI Y M, RAO A V. CGPOPS: a C++ software for solving multiple-phase optimal control problems using adaptive gaussian quadrature collocation and sparse nonlinear programming [J/OL]. (2019-03-29) [2022-03-22]. <https://arxiv.org/pdf/1905.11898.pdf>
- [10] PATTERSON M A, RAO A V. Exploiting sparsity in direct collocation pseudospectral methods for solving optimal control problems [J]. Journal of Spacecraft and Rockets, 2012, 49(2): 364-377.
- [11] DARBY C L, HAGER W W, RAO A V. Direct trajectory optimization using a variable low-order adaptive pseudospectral method [J]. Journal of Spacecraft and Rockets, 2011, 48(3): 433-445.
- [12] 赵吉松, 王江华, 王泊乔, 等. 探月返回飞行器跳跃式再入轨迹优化 [J]. 宇航学报, 2021, 42(2): 211-219.

(编辑: 褚江)