

反临拦截弹中制导弹道在线优化设计

李宁波^{1,2†}, 雷虎民¹, 周 颢¹, 邵 雷¹, 王 斌¹

(1. 空军工程大学 防空反导学院, 西安 710051; 2. 西安交通大学 电子信息与工程学院, 西安 710049)

摘 要: 为了实现反临拦截弹中制导段的弹道在线优化, 基于邻域最优控制 (NOC) 理论和改进间接高斯伪谱法 (GPM) 设计一种弹道在线生成算法. 根据临近空间拦截作战特点设计弹道优化模型, 基于 NOC 理论推导关于修正状态量和协态量的两点边值问题. 对横截条件进行二阶变分, 通过逆向递推将终端最优协态修正值表示成跟踪偏差和终端约束修正量的表达式. 将拦截弹的当前状态量作为初始约束值, 终端最优协态修正值作为干扰量, 利用标称弹道数据, 基于改进间接 GPM 求得控制修正量. 仿真结果表明, 所提出的方法可以快速生成一条修正弹道, 且具有较高的终端修正精度.

关键词: 弹道在线优化; 中制导; 邻域最优控制; 间接高斯伪谱法

中图分类号: TP273

文献标志码: A

Design of midcourse trajectory online optimization for near space interceptor

LI Ning-bo^{1,2†}, LEI Hu-min¹, ZHOU Jin¹, SHAO Lei¹, WANG Bin¹

(1. Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China; 2. School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: In order to realize online trajectory optimization for midcourse guidance of the near space interceptor, an online trajectory generation algorithm is designed based on the neighboring optimal control (NOC) theory and improved indirect Gauss pseudospectral method (GPM). A trajectory optimization model is designed according to the features of operation in the near space. The two points boundary value problem is deduced based on the NOC theory. The transversality conditions are further differentiated to second order and the optimal co-state modifications are expressed in terms of current state derivations and terminal constraint modifications by backward recursion. By treating the current states and the optimal co-state modifications as initial constraints and perturbations, the feedback control variables are obtained based on the improved indirect GPM and nominal trajectory information. The simulation results show that this method can generate a modified trajectory effectively with high precision of terminal modifications.

Keywords: online trajectory optimization; midcourse guidance; neighboring optimal control; indirect Gauss pseudospectral method

0 引 言

近年来, 由于临近空间高超声速武器的迅猛发展, 研究反临近空间的先进拦截技术和防御手段已迫在眉睫. 面对高超声速武器这种高速、高机动性目标, 拦截弹应采用复合制导策略远程拦截, 从而保证拦截的成功率. 从初段程序控制飞行结束到末制导捕获目标, 拦截弹大部分时间飞行在中制导段, 因此其中制导段的飞行性能决定了整体性能.

中制导的任务是要使拦截弹在飞行过程中消耗的能量最少, 且以最佳的相对几何关系进入末制

导^[1]. 考虑到各种终端约束和一系列过程特征, 中制导段弹道优化实际上是一个复杂的非线性严格受约束问题^[2]. 此外, 拦截弹在沿标称弹道实际飞行过程中, 空域和速域的变化范围都比较大, 由于受初始条件误差、大气环境变化等因素的影响, 其气动模型、导航设备方面都存在很大的不确定性^[3], 因而会造成跟踪误差; 且在弹目接近的过程中, 对目标的轨迹预测精度会不断提升, 预测交班条件即拦截弹终端约束条件也需要相应调整, 这时就要求所设计的优化弹道具有在线更新的能力.

收稿日期: 2016-12-09; 修回日期: 2017-03-02.

基金项目: 国家自然科学基金项目 (61573374, 61503408); 航空科学基金项目 (20150196006).

作者简介: 李宁波 (1992-), 男, 博士生, 从事反临拦截器轨迹优化与制导的研究; 雷虎民 (1960-), 男, 教授, 博士生导师, 从事飞行器导航、制导与控制等研究.

†通讯作者. E-mail: ln_b_las@163.com

弹道在线生成一直是弹道设计领域所追求的目标,许多国内外学者对此进行了研究. Lin等^[4-5]通过对模型进行简化处理,得到了一种弹道成型制导律,实现了弹道的在线生成; Oleg等^[6-8]基于虚拟域动态逆的方法提出了一种轨迹成型制导律. 通过引入虚拟弧长,将动态方程从时间域转换到了虚拟域,保证了轨迹的优化不受速度影响(速度由发动机推力剖面决定),同时,减少了所需优化变量的个数,且避免了多次迭代或积分过程,收敛性好,可以实现弹道的快速生成.

目前,大多数学者对在线弹道优化问题的研究主要侧重于弹道重新规划,由于完全舍弃了原有弹道数据再次大范围寻优,计算量较大,耗时较长,对弹载计算机的存储容量和计算能力提出了较高的要求,工程上不易实现. 而实际上,根据邻域最优控制理论,在跟踪偏差及终端约束条件变化不大的情况下,可以利用原有弹道数据在标称弹道邻域内快速生成一条满足过程约束和终端约束的修正弹道.

采用上述修正算法仍然需要解决两点边值问题(TPBVP)^[9],其复杂的迭代和积分过程给在线实施带来了困难. 最近, Fahroo等^[10-11]在求解TPBVP时,提出了一种间接伪谱法,通过状态量偏差的初始值和标称最优轨迹求解所有状态量和协态量的偏差值,进而求得反馈控制修正量,形成了实时的闭环反馈控制. 本文首先基于NOC理论推导关于修正状态量和协态量的两点边值问题;其次,在求解的过程中,借鉴间接伪谱法的思想,结合逆向递推算法,通过终端协态修正值和标称最优轨迹求解所有状态量和协态量的偏差值,进而实时得到反馈控制修正量;最后,进行相关的仿真验证.

1 标称最优弹道

1.1 拦截弹运动模型

从利于拦截角度考虑,选择目标飞行相对平稳的滑翔段或巡航段,即临近空间30~40 km高度范围进行拦截作战. 该段有利于预测命中点的计算和拦截交会,且此高度范围内的大气内存在稀薄大气,有利于对拦截弹实施气动控制.

在离线生成标称最优弹道时,拦截弹应是朝向预测命中点飞行的,所以,考虑如下纵向平面内的拦截弹质点运动模型^[12]:

$$\dot{v} = \frac{P \cos \alpha - C_x q S}{m} - g \sin \theta, \quad (1)$$

$$\dot{\theta} = \frac{P \sin \alpha + C_y q S}{mv} - \frac{g \cos \theta}{v}, \quad (2)$$

$$\dot{x} = v \cos \theta, \quad (3)$$

$$\dot{y} = v \sin \theta. \quad (4)$$

其中: v 为拦截弹速度; θ 为弹道倾角, x 、 y 分别为横程和纵程; m 为拦截弹质量; P 为作用在拦截弹上的推力; q 为动压; S 为参考面积; g 为重力加速度; α 为攻角,作为模型的控制输入量; C_x 、 C_y 分别为阻力系数和升力系数,可分别表示为马赫数和攻角 α 的函数

$$C_y = C_y^\alpha \alpha, \quad (5)$$

$$C_x = C_{x0} + K \cdot C_y^2, \quad (6)$$

C_{x0} 为零升阻力系数, C_y^α 为升力对于攻角的偏导数, K 为诱导阻力系数. 对大气密度的假设是

$$\rho = \rho_0 \exp(-y/y_0). \quad (7)$$

其中: $\rho_0 = 1.2250 \text{ kg/m}^3$ 为地面空气密度; ρ 为飞行器高度处的空气密度.

1.2 弹道优化模型

设 t_0 表示初始时刻, t_f 表示终端时刻,为保证拦截弹的杀伤效果,一般将终端时刻的速度最大作为优化指标 J ,即

$$J = \phi(v(t_f), t_f) = -v_f. \quad (8)$$

由预测命中点给出终端时刻位置约束(x_f, y_f),由末制导捕获条件给出终端时刻的弹道倾角约束 θ_f . 那么标称弹道的终端约束条件如下所示:

$$\psi_f = [\theta - \theta_f, x - x_f, y - y_f]^T = \mathbf{0}, \quad (9)$$

其中 $\mathbf{0}$ 表示具有相应维度的全零矩阵.

为了保证飞行过程中弹体结构的稳定性以及弹道的可实现性,设置如下的控制量、动压、过载及热流^[13]约束:

$$\|\alpha\| \leq \alpha_{\max}; \quad (10)$$

$$q = 0.5\rho v^2 \leq q_{\max}; \quad (11)$$

$$n = \frac{C_y q S}{mg} \leq n_{\max}; \quad (12)$$

$$\begin{cases} Q = Q_a Q_r \leq Q_m, \\ Q_a = h_0 + h_\alpha \alpha + h_{\alpha^2} \alpha^2, \\ Q_r = 117700 \sqrt{\rho} (10^{-4} v)^{3.07}. \end{cases} \quad (13)$$

上述弹道优化模型采用GPM离线求解获得标称最优弹道.

2 基于邻域最优的弹道修正

首先,基于最小值原理推导出最优控制的一阶必要条件,引入哈密尔顿方程^[14]

$$\mathbf{H} = \boldsymbol{\lambda}^T \mathbf{f} = \lambda_v \dot{v} + \lambda_\theta \dot{\theta} + \lambda_x \dot{x} + \lambda_y \dot{y}. \quad (14)$$

其中: $\boldsymbol{\lambda} = [\lambda_v, \lambda_\theta, \lambda_x, \lambda_y]^T$ 为与状态变量同维的协态变量, $\mathbf{f} = [\dot{v}, \dot{\theta}, \dot{x}, \dot{y}]^T$ 为系统状态方程.

然后将终端约束条件引入最优性指标中得到增广的优化指标 J' ,即

$$J' = \phi + \mathbf{v}^T \psi. \quad (15)$$

其中: $\mathbf{v} = [\nu_\theta, \nu_x, \nu_y]^T$, 满足终端约束(9)的条件下使性能指标(15)取得最小值. 正则方程、耦合方程和横截条件可以表示为

$$\begin{cases} \dot{\mathbf{x}} = \frac{\partial \mathbf{H}}{\partial \lambda}, \\ \dot{\lambda} = -\frac{\partial \mathbf{H}}{\partial \mathbf{x}}, \end{cases} \quad (16)$$

$$\frac{\partial \mathbf{H}}{\partial \mathbf{u}} = \mathbf{0}, \quad (17)$$

$$\psi(\mathbf{x}(t_f)) = \mathbf{0}, \quad (18)$$

$$\lambda(t_f) = -\left(\frac{\partial \phi}{\partial \mathbf{x}} + \mathbf{v}^T \frac{\partial \psi}{\partial \mathbf{x}}\right)_{t=t_f}. \quad (19)$$

对式(16)和(17)进行二阶变分,可得

$$\delta \dot{\mathbf{x}} = \frac{\partial^2 \mathbf{H}}{\partial \lambda \partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial^2 \mathbf{H}}{\partial \lambda \partial \mathbf{u}} \delta \mathbf{u}, \quad (20)$$

$$\delta \dot{\lambda} = -\frac{\partial^2 \mathbf{H}}{\partial \mathbf{x}^2} \delta \mathbf{x} - \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x} \partial \lambda} \delta \lambda - \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x} \partial \mathbf{u}} \delta \mathbf{u}, \quad (21)$$

$$\mathbf{0} = \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \lambda} \delta \lambda + \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2} \delta \mathbf{u}. \quad (22)$$

如果 $\partial^2 \mathbf{H} / \partial \mathbf{u}^2$ 在整个飞行过程中非奇异,则由式(22)可以得到控制修正量 $\delta \mathbf{u}$ 的表达式为

$$\delta \mathbf{u} = -\left(\frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2}\right)^{-1} \left(\frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \lambda} \delta \lambda\right). \quad (23)$$

将式(23)代入(20)和(21)中, $\delta \mathbf{x}$ 和 $\delta \lambda$ 的动态方程可以表示为

$$\delta \dot{\mathbf{x}} = \mathbf{A}(t) \delta \mathbf{x} - \mathbf{B}(t) \delta \lambda, \quad (24)$$

$$\delta \dot{\lambda} = -\mathbf{C}(t) \delta \mathbf{x} - \mathbf{A}^T(t) \delta \lambda. \quad (25)$$

其中

$$\begin{cases} \mathbf{A}(t) = \frac{\partial^2 \mathbf{H}}{\partial \lambda \partial \mathbf{x}} - \frac{\partial^2 \mathbf{H}}{\partial \lambda \partial \mathbf{u}} \left(\frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2}\right)^{-1} \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \mathbf{x}}, \\ \mathbf{B}(t) = \frac{\partial \mathbf{H}}{\partial \lambda \partial \mathbf{u}} \left(\frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2}\right)^{-1} \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \lambda}, \\ \mathbf{C}(t) = \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x}^2} - \frac{\partial^2 \mathbf{H}}{\partial \mathbf{x} \partial \mathbf{u}} \left(\frac{\partial^2 \mathbf{H}}{\partial \mathbf{u}^2}\right)^{-1} \frac{\partial^2 \mathbf{H}}{\partial \mathbf{u} \partial \mathbf{x}}. \end{cases} \quad (26)$$

式(24)、(25)满足初始条件 $\delta \mathbf{x}(t_0) = \delta \mathbf{x}_0, \delta \lambda(t_0) = \delta \lambda_0$, 与终端条件(18)、(19)构成了一个关于 $\delta \mathbf{x}, \delta \lambda$ 的两点边值问题.

3 弹道修正模型求解

3.1 逆向递推算法

考虑到终端约束条件的变化,对式(18)、(19)再次进行变分,即

$$\delta \psi = \left[\frac{\partial \psi}{\partial \mathbf{x}} \delta \mathbf{x}\right]_{t=t_f}, \quad (27)$$

$$\delta \lambda(t_f) = \left[\left[\frac{\partial^2 \phi}{\partial \mathbf{x}^2} + \mathbf{v}_f^T \frac{\partial^2 \psi}{\partial \mathbf{x}^2}\right] \delta \mathbf{x} + \left(\frac{\partial \psi}{\partial \mathbf{x}}\right)^T \mathbf{d}\mathbf{v}\right]_{t=t_f}. \quad (28)$$

有 $2n + q$ 个未知数,分别是 n 维 $\delta \mathbf{x}$ 变量, n 维 $\delta \lambda$ 变量, q 维 $\mathbf{d}\mathbf{v}$ 变量,由于 q 维 $\mathbf{d}\mathbf{v}$ 变量与 $n - q$ 维终端时刻自由的状态变量为自由变量,并记作向量 $\mathbf{d}\boldsymbol{\mu} = [\mathbf{d}\boldsymbol{\nu}_1, \mathbf{d}\boldsymbol{\nu}_2, \dots, \mathbf{d}\boldsymbol{\nu}_q, \delta \mathbf{x}_{q+1}, \delta \mathbf{x}_{q+2}, \dots, \delta \mathbf{x}_n]$. 对式(27)和(28)进行整理, \mathbf{P} 为 $\delta \mathbf{x}$ 和 $\mathbf{d}\mathbf{v}$ 的系数组成的矩阵,根据 $\mathbf{d}\boldsymbol{\mu}_f$ 、 $\mathbf{d}\boldsymbol{\psi}_f$ 将 \mathbf{P} 写成分块矩阵的形式,构成如下方程组:

$$\begin{bmatrix} \delta \mathbf{x} \\ \delta \lambda \end{bmatrix}_{t=t_f} = \mathbf{P} \begin{bmatrix} \mathbf{d}\boldsymbol{\mu}_f \\ \mathbf{d}\boldsymbol{\psi}_f \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{d}\boldsymbol{\mu}_f \\ \mathbf{d}\boldsymbol{\psi}_f \end{bmatrix}. \quad (29)$$

令 $\delta \boldsymbol{\Theta} = (\delta \mathbf{x}, \delta \lambda)^T$, 式(24)和(25)可以表示成如下形式:

$$\delta \dot{\boldsymbol{\Theta}} = \begin{bmatrix} \mathbf{A}(t) & \mathbf{B}(t) \\ -\mathbf{C}(t) & -\mathbf{A}^T(t) \end{bmatrix} \delta \boldsymbol{\Theta} = \mathbf{G} \delta \boldsymbol{\Theta}. \quad (30)$$

对上述微分方程中的变量以时间间隔 Δt 离散为 $M + 1 (k = 0, 1, 2, \dots, M)$ 个点,进行逆向递推,即

$$\delta \boldsymbol{\Theta}_{M-1} = \delta \boldsymbol{\Theta}_M + \mathbf{G}_M \delta \boldsymbol{\Theta}_M (-\Delta t) = (\mathbf{I} - \mathbf{G}_M \Delta t) \delta \boldsymbol{\Theta}_M, \quad (31)$$

$$\begin{aligned} \delta \boldsymbol{\Theta}_k &= \delta \boldsymbol{\Theta}_{k+1} + \mathbf{G}_{k+1} \delta \boldsymbol{\Theta}_{k+1} (-\Delta t) = \\ &(\mathbf{I} - \mathbf{G}_{k+1} \Delta t) \delta \boldsymbol{\Theta}_{k+1} = \\ &(\mathbf{I} - \mathbf{G}_{k+1} \Delta t)(\mathbf{I} - \mathbf{G}_{k+2} \Delta t) \delta \boldsymbol{\Theta}_{k+2}. \end{aligned} \quad (32)$$

同理依次递推到第 $M + 1$ 个点,即

$$\delta \boldsymbol{\Theta}_0 = \mathbf{G} \delta \boldsymbol{\Theta}_M, \quad (33)$$

其中 $\mathbf{G} = (\mathbf{I} - \mathbf{G}_1 \Delta t)(\mathbf{I} - \mathbf{G}_2 \Delta t) \dots (\mathbf{I} - \mathbf{G}_M \Delta t)$. 将式(29)代入(33)中,可得

$$\begin{bmatrix} \delta \mathbf{x}(t_0) \\ \delta \lambda(t_0) \end{bmatrix} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 \\ \mathbf{G}_3 & \mathbf{G}_4 \end{bmatrix} \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{d}\boldsymbol{\mu}_f \\ \mathbf{d}\boldsymbol{\psi}_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Pi}_1 & \boldsymbol{\Pi}_2 \\ \boldsymbol{\Pi}_3 & \boldsymbol{\Pi}_4 \end{bmatrix} \begin{bmatrix} \mathbf{d}\boldsymbol{\mu}_f \\ \mathbf{d}\boldsymbol{\psi}_f \end{bmatrix}. \quad (34)$$

其中

$$\begin{cases} \boldsymbol{\Pi}_1 = \mathbf{G}_1 \mathbf{P}_{11} + \mathbf{G}_2 \mathbf{P}_{21}, \\ \boldsymbol{\Pi}_2 = \mathbf{G}_1 \mathbf{P}_{12} + \mathbf{G}_2 \mathbf{P}_{22}, \\ \boldsymbol{\Pi}_3 = \mathbf{G}_3 \mathbf{P}_{11} + \mathbf{G}_4 \mathbf{P}_{21}, \\ \boldsymbol{\Pi}_4 = \mathbf{G}_3 \mathbf{P}_{12} + \mathbf{G}_4 \mathbf{P}_{22}. \end{cases}$$

若 $\boldsymbol{\Pi}_1$ 可逆,则可以得到 $\mathbf{d}\boldsymbol{\mu}$ 的表达式

$$\mathbf{d}\boldsymbol{\mu}_f = \boldsymbol{\Pi}_1^{-1} (\delta \mathbf{x}_0 - \boldsymbol{\Pi}_2 \mathbf{d}\boldsymbol{\psi}_f). \quad (35)$$

将式(35)代入(29)可以得到

$$\delta \boldsymbol{\lambda}_f = \mathbf{P}_{21} \boldsymbol{\Pi}_1^{-1} (\delta \mathbf{x}_0 - \boldsymbol{\Pi}_2 d\boldsymbol{\psi}_f) + \mathbf{P}_{22} d\boldsymbol{\psi}_f. \quad (36)$$

3.2 基于间接高斯伪谱法求解

本节将根据 $\delta \boldsymbol{\lambda}_f$ 采用改进的间接高斯伪谱法求解修正后的最优弹道. 由于上节在求解 $\delta \boldsymbol{\lambda}_f$ 时, 已经将弹道跟踪偏差考虑在内, 本节求解修正弹道时可以把拦截弹的当前状态量作为初始约束值, 将 $\delta \boldsymbol{\lambda}_f$ 作为干扰量, 利用标称弹道数据求得修正控制量和修正状态量.

上节中的边界条件变为如下形式:

$$\boldsymbol{\psi}' = [\theta - \theta_0, x - x_0, y - y_0]^T = \mathbf{0}, \quad (37)$$

$$\boldsymbol{\lambda}(t_0) = -\left(\mathbf{v}^T \frac{\partial \boldsymbol{\psi}'}{\partial \mathbf{x}}\right)_{t=t_0}. \quad (38)$$

式(37)和(38)可以表示为如下形式:

$$\boldsymbol{\Omega}(\mathbf{x}(t_0), \boldsymbol{\lambda}(t_0)) = \mathbf{0}. \quad (39)$$

对式(39)进行变分可以得到

$$\left[\frac{\partial \boldsymbol{\Omega}}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial \boldsymbol{\Omega}}{\partial \boldsymbol{\lambda}} \delta \boldsymbol{\lambda}\right]_{t=t_0} = \mathbf{0}. \quad (40)$$

下面采用改进间接高斯伪谱法进行求解. 首先将变量在勒让德-高斯(LG)点离散, LG点为 \dot{P}_N 的零点, \dot{P}_N 为勒让德多项式 P_N 的导数^[15], 引入时间变量 τ 进行时域变换, 即

$$\tau = \frac{2t}{t_f - t_0} - \frac{t_f + t_0}{t_f - t_0}. \quad (41)$$

在高斯伪谱法近似中, $\tau_l = 1, 2, \dots, N-1, l$ 为分布在 $(-1, 1)$ 之间的 $N-1$ 个 LG 点, 定义 $\tau_0 = -1, \tau_N = 1$, 状态和控制量近似如下:

$$\delta \mathbf{x} \approx \delta \mathbf{x}^N(\tau) = \sum_{l=0}^{N-1} \delta x(\tau_l) L_l(\tau), \quad (42)$$

$$\delta \boldsymbol{\lambda} \approx \delta \boldsymbol{\lambda}^N(\tau) = \sum_{l=0}^{N-1} \delta \lambda(\tau_l) L_l(\tau). \quad (43)$$

其中: $l = 0, 1, \dots, N-1$;

$$L_l = \prod_{j=0, j \neq l}^{K} \frac{\tau - \tau_j}{\tau_l - \tau_j}. \quad (44)$$

式(44)是 N 阶拉格朗日插值多项式, $\delta \mathbf{x}^N(\tau)$ 和 $\delta \boldsymbol{\lambda}^N(\tau)$ 在 τ_l 点处的导数通过对式(42)和(43)求导获得, 即

$$\begin{aligned} \delta \dot{\mathbf{x}}(\tau_k) &\approx \delta \dot{\mathbf{x}}^N(\tau_k) = \\ &\sum_{l=0}^{N-1} \delta \mathbf{x}(\tau_l) \dot{L}_l(\tau) = \sum_{l=0}^{N-1} \mathbf{D}_{kl} \delta \mathbf{x}(\tau_l), \end{aligned} \quad (45)$$

$$\begin{aligned} \delta \dot{\boldsymbol{\lambda}}(\tau_k) &\approx \delta \dot{\boldsymbol{\lambda}}^N(\tau_k) = \\ &\sum_{l=0}^{N-1} \delta \boldsymbol{\lambda}(\tau_l) \dot{L}_l(\tau) = \sum_{l=0}^{N-1} \mathbf{D}_{kl} \delta \boldsymbol{\lambda}(\tau_l), \end{aligned} \quad (46)$$

其中 \mathbf{D}_{kl} 为 $(N-1) \times N$ 维的差分矩阵, 即

$$\mathbf{D}_{kl} = \begin{cases} \frac{(1 + \tau_k) \dot{P}_K(\tau_k) + P_K(\tau_k)}{(\tau_k - \tau_i)[(1 + \tau_i) \dot{P}_K(\tau_i) + P_K(\tau_i)]}, & i \neq k; \\ \frac{(1 + \tau_i) \ddot{P}_K(\tau_i) + 2\dot{P}_K(\tau_i)}{2[(1 + \tau_i) \dot{P}_K(\tau_i) + P_K(\tau_i)]}, & i = k. \end{cases} \quad (47)$$

将 $\delta \mathbf{x}, \delta \boldsymbol{\lambda}$ 的两点边值问题离散后转换成如下代数方程:

$$\begin{aligned} &\sum_{j=1}^N D_{ij} \delta \mathbf{x}_j(\tau) - \\ &\frac{t_f - t_0}{2} (\mathbf{A}_i(t) \delta \mathbf{x}_i(t) - \mathbf{B}_i(t) \delta \boldsymbol{\lambda}_i(t)) = \mathbf{0}, \end{aligned} \quad (48)$$

$$\begin{aligned} &\sum_{j=1}^N D_{ij} \delta \boldsymbol{\lambda}_j(\tau) + \\ &\frac{t_f - t_0}{2} (\mathbf{C}_i(t) \delta \mathbf{x}_i(t) + \mathbf{A}_i^T(t) \delta \boldsymbol{\lambda}_i(t)) = \mathbf{0}, \end{aligned} \quad (49)$$

$$[\partial \boldsymbol{\Omega}_{\mathbf{x}} \delta \mathbf{x} + \partial \boldsymbol{\Omega}_{\boldsymbol{\lambda}} \delta \boldsymbol{\lambda}]_{t=t_0} = \mathbf{0}, \quad (50)$$

其中 $\mathbf{0}$ 为具有相应维度的全零矩阵. 为表达更加简便, 可进行如下替代: $\mathbf{X} = [\delta \mathbf{x}_0^T, \delta \mathbf{x}_1^T, \dots, \delta \mathbf{x}_{N-1}^T]^T$, $\boldsymbol{\Lambda} = [\delta \boldsymbol{\lambda}_0^T, \delta \boldsymbol{\lambda}_1^T, \dots, \delta \boldsymbol{\lambda}_{N-1}^T]^T$. 由此可得

$$\mathbf{E} \mathbf{X} - \frac{t_f - t_0}{2} \mathbf{F} \boldsymbol{\Lambda} = \mathbf{0}, \quad (51)$$

$$\frac{t_f - t_0}{2} \mathbf{G} \mathbf{X} + \mathbf{H} \boldsymbol{\Lambda} = \mathbf{0}. \quad (52)$$

其中 $\mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H}$ 是 $[n(N-1) \times nN]$ 维矩阵, 且有

$$[\mathbf{E}]_{ij} = \begin{cases} \mathbf{D}_{ij} \mathbf{I}_n, & i \neq j; \\ \mathbf{D}_{ij} \mathbf{I}_n - \frac{t_f - t_0}{2} \mathbf{A}_i, & i = j. \end{cases} \quad (53)$$

$$[\mathbf{F}]_{ij} = \begin{cases} \mathbf{0}_n, & i \neq j; \\ \mathbf{B}_i, & i = j. \end{cases} \quad (54)$$

$$[\mathbf{G}]_{ij} = \begin{cases} \mathbf{0}_n, & i \neq j; \\ \mathbf{C}_i, & i = j. \end{cases} \quad (55)$$

$$[\mathbf{H}]_{ij} = \begin{cases} \mathbf{D}_{ij} \mathbf{I}_n, & i \neq j; \\ \mathbf{D}_{ij} \mathbf{I}_n + \frac{t_f - t_0}{2} \mathbf{A}_i^T, & i = j. \end{cases} \quad (56)$$

其中 \mathbf{I}_n 和 $\mathbf{0}_n$ 分别为 $n \times n$ 维的单位矩阵和零矩阵. 将式(50)~(52)归纳为下式:

$$\begin{bmatrix} \mathbf{\Gamma}_1 & \mathbf{\Gamma}_2 \\ \mathbf{E} & -\frac{t_f - t_0}{2} \mathbf{F} \\ \frac{t_f - t_0}{2} \mathbf{G} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \boldsymbol{\Lambda} \end{bmatrix} = \mathbf{V} \mathbf{Z} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \quad (57)$$

其中: $\mathbf{Z}^T = [\mathbf{X}^T, \boldsymbol{\Lambda}^T]$, $\mathbf{\Gamma}_1, \mathbf{\Gamma}_2$ 为 $n \times nN$ 维矩阵, $\mathbf{\Gamma}_1 = [\partial \boldsymbol{\Omega} / \partial \mathbf{x}, \mathbf{0}_n, \dots, \mathbf{0}_n]$, $\mathbf{\Gamma}_2 = [\partial \boldsymbol{\Omega} / \partial \boldsymbol{\lambda}, \mathbf{0}_n, \dots, \mathbf{0}_n]$.

把矩阵 $\mathbf{V} = [\mathbf{V}_e, \mathbf{V}_f]$ 代入式(57)可得

$$\mathbf{V}_e \mathbf{Z}_e + \mathbf{V}_f \delta \boldsymbol{\lambda}_N = \mathbf{0}. \quad (58)$$

其中: V_f 和 V_e 分别为关于矩阵 V 的 $[(2nN - 1) \times n]$, $[(2nN - 1) \times n(2N - 1)]$ 维分块矩阵. 向量 Z_e 定义为 $Z_e = [\delta x_0^T, \delta x_1^T, \dots, \delta x_{N-1}^T, \delta \lambda_0^T, \dots, \delta \lambda_{N-2}^T]^T$, 可通过式(59)求得

$$Z_e = -V_e \setminus V_f \delta \lambda_{N-1} = W \delta \lambda_{N-1}. \quad (59)$$

由 $Z = [\delta x_0^T, \dots, \delta x_N^T, \delta \lambda_0^T, \dots, \delta \lambda_N^T]^T$ 可得

$$Z = \begin{bmatrix} Z_e \\ \delta \lambda_{N-1} \end{bmatrix} = \begin{bmatrix} W \\ I_n \end{bmatrix} \delta \lambda_{N-1} = \begin{bmatrix} W_x \\ W_\lambda \end{bmatrix} \delta \lambda_{N-1}. \quad (60)$$

其中: W_x 和 W_λ 是矩阵 $[W, I_n]^T$ 的分块矩阵, 维数都是 $nN \times n$. 由此可以得到

$$\delta x_k = W_{xk} \delta \lambda_{N-1}, \delta \lambda_k = W_{\lambda k} \delta \lambda_{N-1}. \quad (61)$$

其中: $W_{\lambda k}$ 和 W_{xk} 分别为 W_λ 和 W_x 的 $n \times n$ 维分块矩阵, 下标 k 为第 k 个 LG 点.

把式(61)代入(23), 得到最优修正控制量

$$\delta u(\tau_i) = -H_{uu}^{-1} (f_u^T W_{\lambda i} + H_{ux} W_{x i}) \delta \lambda_N. \quad (62)$$

由于 $\delta \lambda_N = \delta \lambda_f, \delta \lambda_f$ 在上节中已求出, 则 LG 配点处的状态量、协状态量和控制量可以通过式(61)和(62)求得, 节点间变量的值可以通过插值得到. 并且以上求解没有进行任何的积分或者迭代过程, 因而可以保证算法的计算效率. 弹道在线优化的流程框图如图1所示.

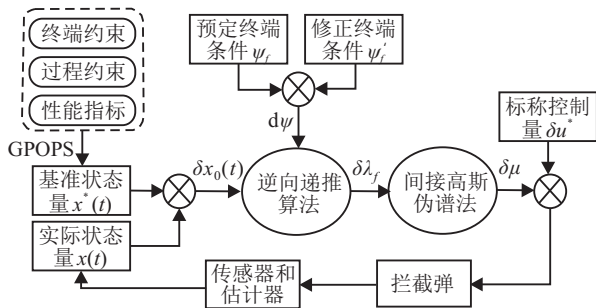


图1 弹道在线优化

4 仿真验证

为了验证本文所设计算法的有效性, 设计如下的拦截作战想定: 假设拦截弹发射 20s 后对终端预测命中点进行修正, 分别将高度增加 2km, 弹道倾角增加 5°, 此时弹道倾角跟踪偏差为 0.1°, 高度跟踪偏差为 10m, 验证弹道修正能力.

GPM 求解最优化问题时具有较高的求解精度, 因此本文将采用 GPM 重新规划的弹道作为基准最优修正弹道进行对比, 仿真中基于 GPM 的求解采用 Matlab 程序包 GPOPS^[16] 实现. 弹道对比与约束量变化曲线如图2和图3所示. 图2中点线表示基于本文算法求解的修正弹道, 实线表示标称弹道, 虚线表示终端约束条件变化后基于 GPM 重新规划的弹道. 图

3中点线表示约束值上限, 虚线表示 Nominal 曲线, 实线表示 Noc 曲线.

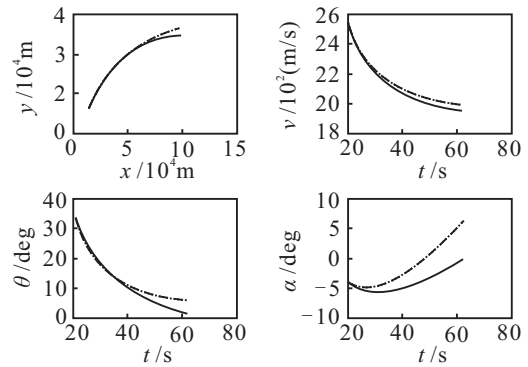


图2 修正弹道对比

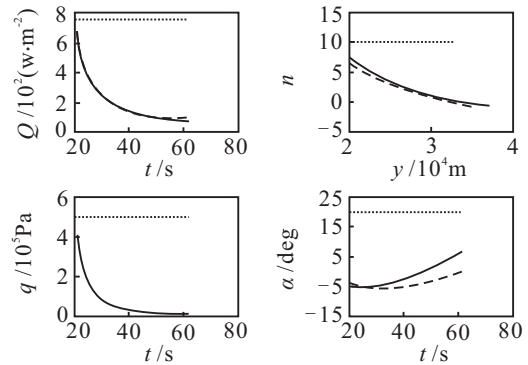


图3 控制量约束及过程约束曲线

由图2可知, 采用本文算法得到的修正弹道与基于 GPM 重新规划的弹道基本重合, 满足终端约束修正的要求, 验证了本文设计算法的有效性. 在修正初始时刻, 基于本文算法得到的攻角与标称弹道的攻角相差不大, 说明攻角没有发生较大的跳变, 保证了控制的稳定性. 由图3可知, 沿本文设计的修正弹道飞行的过程中, 满足控制量约束和过程约束要求.

为了验证改进间接 GPM 具有较高的优化精度和求解效率, 设计如下的对比实验: 在 3.1 节中, 通过逆向递推算法, 根据跟踪偏差和终端约束修正量得到 $\delta x_f, \delta \lambda_f$, 代入第2节的弹道修正模型中逆向积分微分方程并求解反馈控制量, 进而得到修正弹道数据. 积分过程中分别选取 21 个 (采用间接 GPM 求解时使用了 21 个节点) 和 1000 个节点, 弹道倾角修正量曲线和高度修正量曲线如图4和图5所示, 消耗时间和终端时刻修正量对比如表1所示.

表1 消耗时间和终端时刻修正量对比

方法	GPOPS	NOC	21	1000
t/s	25.68	0.12	0.58	2.02
$\delta \theta_f / \text{deg}$	5.00	4.97	1.80	4.48
$\delta y_f / \text{m}$	2000	2003	-56	1887

由图4、图5及表1可知, 采用本文方法所设计的修正弹道与基于 GPM 重新规划相比, 弹道倾角修正量曲线除刚修正后的几秒有约 0.4° 的偏差外, 其余部

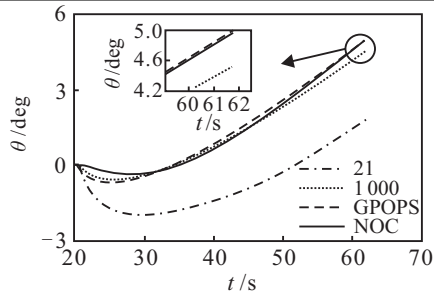


图4 弹道倾角修正量

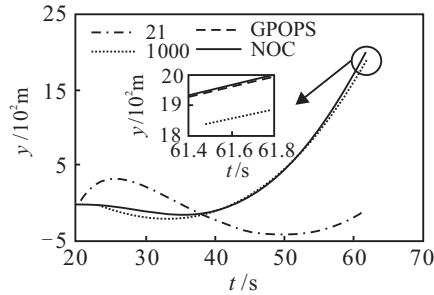


图5 高度修正量

分基本重合,高度修正量曲线除刚修正后的几秒有约100 m的偏差外,其余部分基本重合,且终端时刻的修正量具有相似的精度,消耗时间远小于GPM。

采用基于直接求解微分方程的方法,当选取与本文算法同样的21个节点时,所生成的弹道与基于GPM重新规划相比,两者修正量曲线偏差相对较大,且难以满足改变后的终端约束条件.随着节点数的增加,偏差逐渐减小,终端时刻修正量精度逐渐增加,但消耗时间也相应增加,即使选取1000个节点,其终端时刻修正精度与本文算法相比仍有一定的差距,而消耗时间却大幅增加。

综上,利用标称弹道数据设计修正弹道的思路可以在保证一定修正精度的条件下大幅提升求解效率;基于改进间接GPM的求解方法在精度和效率上都优于直接求解微分方程的方法。

5 结论

本文以拦截临近空间目标为背景,基于邻域最优控制理论和改进间接GPM设计了一种弹道在线优化算法.该算法满足了消除跟踪偏差和修正终端约束条件的要求,在标称弹道邻域内可以快速生成一条满足二阶最优性的修正弹道,实现了弹道的在线生成.本文设计思路可以为临近空间飞行器的轨迹在线生成系统设计提供借鉴,但当终端约束修正量的变化较大,超出标称弹道的邻域范围时,本文设计的在线生成算法不再适用,需要重新寻优规划弹道。

参考文献(References)

[1] Robert W M, Aron P N. Generalized optimal midcourse guidance[C]. 53rd IEEE Conf on Decision and Control. Los Angeles: IEEE Press, 2015: 260-264.

[2] Ghosh S, Ghose D, Raha S. Capturability analysis of a 3-D retro-PN guidance law for higher speed nonmaneuvering targets[J]. IEEE Trans on Control Systems Technology, 2014, 22(5): 1864-1874.

[3] Zhang D Y, Lei H M, Shao L, et al. Interceptor trajectory programming for near space hypersonic target[J]. J of National University of Defense Technology, 2015, 37(3): 91-96.

[4] Lin C F, Tsai L L. Analytical solution of optimal trajectory-shaping guidance[J]. J of Guidance, Control, and Dynamic, 1987, 10(1): 61-66.

[5] Lin C F. Modern navigation guidance, and control processing[M]. New Jersey: Prentice Hall, 1991: 562-591.

[6] Oleg A Y. Direct method for rapid prototyping of near-optimal aircraft trajectories[J]. J of Guidance, Control, and Dynamics, 2000, 23(5): 865-875.

[7] Lukacs J, Yakimenko O. Trajectory-shape-varying missile guidance for interception of ballistic missiles during the boost phase[C]. AIAA Guidance, Navigation and Control Conf and Exhibit. Reston: AIAA, 2007: 6538.

[8] George A B, Marcello R, Oleg A Y. Time-optimal reorientation of a spacecraft using an inverse dynamics optimization method[J]. J of Guidance, Control, and Dynamics, 2011, 34(4): 1197-1207.

[9] Yong E M, Chen L, Tang G J. A survey of numerical methods for trajectory optimization of spacecraft[J]. J of Astronautics, 2008, 29(2): 397-406.

[10] Fahroo F, Ross I. Trajectory optimization by indirect spectral collocation methods[C]. AIAA/AAS Astroynamics Specialist Conf. Reston: AIAA, 2000: 123-129.

[11] Hui Y, Fahroo F, Ross I M. Real-time computation of neighboring optimal control laws[C]. AIAA Guidance, Navigation and Control Conf and Exhibit. Reston: AIAA, 2002: 1-7.

[12] Song E J, Tahk M J. Real-time neural-network midcourse guidance[J]. Control Engineering Practice, 2001, 9(10): 1145-1154.

[13] Afshin R, Krishna D K, Hekmat A. Particle swarm optimization applied to spacecraft reentry trajectory[J]. J of Guidance, Control, and Dynamics, 2013, 36(1): 307-310.

[14] Bryson A E, Ho Y C. Applied optimal control[M]. New York: Halsted Press, 1975: 177-201.

[15] Wang P, Tian X M. Partitioning and simultaneous strategy for dynamic optimization by Gauss pseudo-spectral method[J]. Control and Decision, 2011, 26(11): 1749-1752.

[16] Rao A V, Benson D, Huntingdon G T, et al. Algorithm 902: GPOPS, a Matlab software for solving multiple-phase optimal control problems using gauss pseudospectral methods[J]. ACM Trans on Mathematical Software, 2010, 37(2): 22-39.