

寻求探空火箭最佳推力策略的 一种直接优化方法

刘冀湘 甘楚雄

(航天技术系)

摘要 本文运用贝尔曼动态规划方法寻求探空火箭最佳推力策略, 建立了解本文问题的“局部最优性”递推方程, 采用区域性限制系数和非节点处的“单点插值”方案, 大幅度提高了运算速度和精度, 计算过程更趋稳定, 使给定起飞质量的探空火箭在寻优计算中获得较之于间接方法更大的目标效益。同时, 在微机IBMPC/XT上编制了计算程序, 为探空火箭初步设计和方案论证提供了具有较高置信水平的理论依据。

关键词 探空火箭, 最优化方法; 推力程序

分类号 V472

贝尔曼动态规划法是面向过程的直接优化方法。运用贝尔曼法寻求探空火箭最佳推力策略, 是根据该方法的“局部最优性原理”将探空火箭自起飞至升达弹道顶点的整个飞行动态过程分成若干阶段——贝尔曼级, 继而用逐级布置网格的办法在过程必经的状态空间域里构成系统状态参数的空间网格点阵, 并将火箭质量 G 作为此动态过程的独立变量。在各贝尔曼级上运用“局部最优性”递推方程:

$$S(t_i, G_i) = \max_{\substack{G[t_i, t_{i+1}] \\ U[U_1, U_m]}} \left[\int_{t_i}^{t_{i+1}} F(t, v, h, G, U) dt + S(t_{i+1}, G_{i+1}) \right]$$

($i = n, n-1, n-2, \dots, 3, 2, 1$)

(这里, $S(t, G_i)$ 为第 i 贝尔曼级上获得的最大爬升高度函数; $F(t, v, h, G, U)$ 为火箭运动方程组; $G(t, t_{i+1})$ 表示 G 在区间 $[t_i, t_{i+1}]$ 中取值; U 为装药质量秒耗量。 $U[U_1, U_2]$ 表示 U 在区间 $[U_{\min}, U_{\max}]$ 中取值)。在各级网络上逐点搜索和计算, 利用空间平面三点式方程进行非节点上火箭爬升高度插值计算, 如此逆序地进行各级网络上诸节点的推力决策寻优直至第一级, 然后顺序地搜索各级的决策值并逐级构成最佳决策集合(最

佳策略)： $U = \{U_1, U_2, U_3, \dots, U_n\}$

使火箭得以在获取的各级最佳推力决策下随时作变推力飞行，并能以给定的起飞质量升达最大探测高度。

可设： G_p 为有效载荷质量， G_m 为火箭结构质量， Q 为火箭装药质量，且装药质量按贝尔曼分级数分割成 n 等分，有：

$$Q = \sum_{i=1}^n \Delta Q \quad \& \quad \text{火箭质量 } G = G_p + G_m + Q$$

相应地，各贝尔曼级上装药燃烧时间区间值可由下式确定：

$$\Delta t_i = \Delta Q / U_i \quad (i=1, 2, 3, \dots, n)$$

记为 $[t_i, t_{i+1}]$ ，其中， U_i 为确定的每一级的质量秒耗量决策值。而第 $n+1$ 级由整个被动段飞行时间构成，并记为 $[t_k, t_d]$ ，相对于贝尔曼级有：

$$t_n = t_k \quad \& \quad t_{n+1} = t_d$$

决策变量——火箭装药提供的推力：

$$P = V_r \cdot U + A_e \cdot (P_e - P_a) \quad \text{全推力公式}$$

$$P = V_r \cdot U \quad \text{非全推力公式}$$

这里： V_r ——火箭发动机有效排气速度； A_e ——火箭发动机喷口截面积；

P_e ——火箭发动机喷口压力； P_a ——火箭所处高度的外界环境大气压力。

考虑火箭垂直爬升，因在贝尔曼法中采用火箭质量作为求解问题的独立变量，火箭运动方程组可简化为质量微元的表达形式：

$$\left. \begin{aligned} dt &= -\frac{dG}{U} \\ dh &= -\frac{v}{U} dG \\ dv &= -\left(\frac{P-W}{G} - g\right) \frac{1}{U} dG \end{aligned} \right\} (*)$$

h —— t 时刻火箭所处的飞行高度； v —— t 时刻火箭具有的飞行速度；

W —— t 时刻火箭受到的空气阻力； g ——重力加速度，且有 $g = g(h)$ ；

由(*)式可看出，在火箭结构一定的条件下进行推力决策寻优时，只需找到最佳的质量秒耗量 U ($U_{\min} \leq U \leq U_{\max}$) 即可。

由火箭飞行动态过程可知，在相邻两贝尔曼级——第 i 级和第 $i+1$ 级有：

$$G_{i+1} \leq G_i \quad (\text{当 } \Delta Q = 0 \text{ 时有 } G_{i+1} = G_i)$$

$$v_{B_i} = V_r \cdot \ln \frac{G_0}{G_{i+1}} \leq v_{B_{i+1}} = V_r \cdot \ln \frac{G_i}{G_{i+1}} \quad (\text{不计重力、空气阻力时火箭飞行速度})$$

$$h_{B_i} = V_{B_i} \cdot \frac{i \cdot \Delta Q}{U_{\min}} \leq h_{B_{i+1}} = V_{B_{i+1}} \cdot \frac{i \cdot \Delta Q}{U_{\min}} \quad (\text{火箭飞行最大估值高度})$$

$$(i=1, 2, 3, \dots, n)$$

这样，设置各级网格的区域时可分别确定 v 和 h 的上、下限 4 个系数：

$$a_1, a_2, b_1, b_2 \quad (0 \leq a_1, a_2, b_1, b_2 \leq 1)$$

$$\begin{aligned} \text{使} \quad & V_{B_{i+1}} = a_1 \cdot V_{B_i} \\ & h_{B_{i+1}} = b_1 \cdot h_{B_i} \\ & V_{B_{i+2}} = a_2 \cdot V_{B_i} \\ & h_{B_{i+2}} = b_2 \cdot h_{B_i} \end{aligned}$$

分别构成第 i 级网格中状态参数的上下限。使下一级网格诸节点状态参数, 经过计算, 其积分结果落在上一级网格区域之中, 即满足:

$$V_{B_i} \text{终} \in [V_{B_{i+1}}, V_{B_{i+2}}] \quad \& \quad h_{B_i} \text{终} \in [h_{B_{i+1}}, h_{B_{i+2}}]$$

便能使相邻两网格形成较为理想的嵌套衔接(如图1示), 使我们有可能以较小的各级局部网, 适应性地覆盖住过程的必经区域, 既保证了计算精度的要求, 又使运算过程尽可能地避免了许多不必要的运算, 达到了简单而快捷的目的。

在贝尔曼法的实际计算中, 由于各级网格均是根据火箭在各级的质量 $G(t)$ 建立的一种估值网格, 而级间的弹道积分运算, 却是根据这种估值网格诸节点上的状态参数进行的实际运算。因此, 由第 $i-1$ 级网格某节点起始进行积分运算后, 由于推力 P 的作用, 该级终点处状态参数一般不会恰好落在第 i 级的估值参数网格节点上。本文通过一系列计算, 设计了两种网格插值计算方案。

第一种方案是: 第 $n-1$ 级完成工作后进行的第 n 级网格上的插值计算。由于火箭在 $[t_n, t_{n+1}]$ 区间作被动段弹道飞行, 爬升高度值仅为主动段终点状态参数的函数, 因此, 整个网格诸节点上计算得到的爬升高度值将出现一种规律——爬升高度值依网格节点顺序递增。这时若弹道积分值 $e(i', j')$ 落在网格节点 $a(i, j)$ 附近(如图2示), 则可取图中 $a(i, j)$, $b(i+1, j)$ 和 $c(i, j+1)$ 三点来构成一空间平面, 并认为非节点上的爬升高度值落在构造的平面上。于是可运用平面方程:

$$S = S_1 + [v_1 \cdot (h_3 \cdot S_2 - h_2 \cdot S_3) + h_1 \cdot (v_2 \cdot S_3 - v_3 \cdot S_2)] / (v_2 \cdot h_3 - v_3 \cdot h_2) \quad (3)$$

便能进行插值计算了。这里: v 、 h 分别表示火箭在第 $i-1$ 级终点(进入第 i 级网格)时的飞行状态参数; S 则为第 i 级 $v \sim h$ 网格节点上存储的目标函数值——火箭在区间 $[t_i, t_{i+1}]$ 上爬升高度值。

第二种方案是: 从第 $n-2$ 级开始至第一级之间各级的插值计算。由于主动段中推力 P (或质流 U) 的作用, 使自完成了级间网格插值计算后, 虽然各级网格上存储的火箭获得的爬升高度值在整个网格区域上仍有使爬升高度值状态参数递增而增长的趋势, 但局部区域上已造成爬升高度值的波动。因此, 当下一级的弹道计算值 $e(i', j')$ 落在上一级网格点 $a(i, j)$ 附近构成非节点时, 不能采用第一种方案来构造平面, 这时相邻两点 b 和 c 上存储的爬升高度值不一定能与点 a 上的爬升高度值构成递增规律, 以此三点构成平面将降低非节点 $e(i', j')$ 所插值 $S(i', j')$ 的置信度。为此本文设计了一种单点插值方案, 即: 在每一级网格上进行弹道计算时, 对应每一节点 $d(i, j)$, 不仅要计算出该节点状态参数的爬升高度值, 而且要将与该节点相距为 σ (σ 为一较小的数) 的相邻 4 点 d_1 、 d_2 、 d_3 、 d_4 (如图3示) 上的爬升高度值也算出来一并存储。这样, 当下一级网格的计算结果成为该级网格的非节点时, 则可对应该点 $e(i', j')$, 找出与其最近的节点 $d(i, j)$ 及 $d_1 \sim d_4$ 点中与 e 点相近的两点:

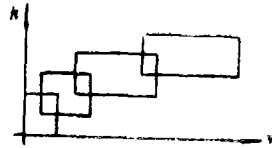


图 1

d_1, d_3 或 d_1, d_4 或 d_2, d_3 或 d_2, d_4

使 d 点连同选出的另两点构成一空间平面，继而进行 e 点在该平面上的插值计算。

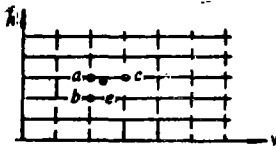


图 2

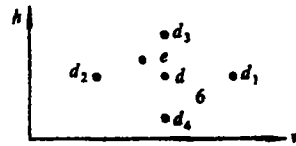


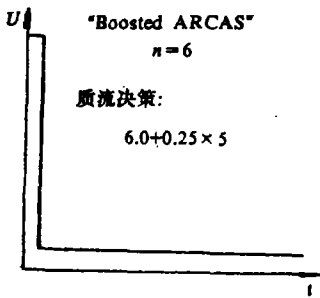
图 3

在网格节点数相同的情况下，第一种方案较第二种方案节省较多的内存空间和计算机时。但第一种方案只能用于被动段($P = 0$)时的弹道参数插值计算。因为在主动段弹道计算和决策选优中，网格诸节点上存储的是不尽相同的 U 值及爬升高度值，因而选出的三个与 e 点相近点 a, b, c 上存储的 U 值亦不尽相同。又 U 值的取值范围为：

$$U_{\min} \leq U \leq U_{\max}$$

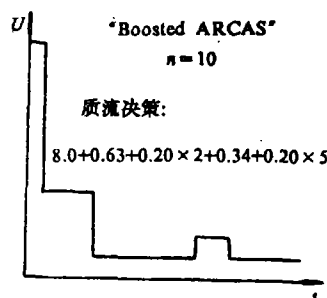
且被等分成 $m + 1$ 个离散值，因此， a, b, c 三点选出的 U 值并无一定的规律，在如此构成的空间平面上进行插值计算必将影响计算程序的稳定性和可靠性。而第二种方案，不仅要计算每一节点的参数，还要计算与每一节点相距为 σ 的相邻 4 点的参数，因此须占用较多的计算机机时和内存。但只要 σ 取得不太大， $d_1 \sim d_4$ 四点上 U 的取值可与 d 点的 U 取同值，构成的平面也就有了共同的基础。由于 $d_1 \sim d_4$ 四点为 d 的辅助点，且 σ 取值不大。因此，在这样构成的平面上作出的插值计算就仅与 d 点有关。所以，它是一种单点插值方案。这种方案不仅能够保证计算的稳定性，而且在一定程度上使计算的简单性和精确性和谐地统一起来了。事实上，通过实际计算，取 $\sigma = 0.05$ 已能保证整个过程中始终都作稳定的插值计算。同时，只要所布置的网格覆盖区域不是太大，也就保证了非节点 e 上获得的爬升高度线性插值的精度。

为了验证贝尔曼法的寻优特性，本文在微型计算机 IBM PC/XT 上独立地编制了可计算二级火箭的 FORTRAN 程序，并根据“织女一号”，“Boosted ARCAS”，“Side-winder ARCAS” 三种探空火箭的实际参数，分别进行了计算，计算结果见附表及图 4~9 所示。



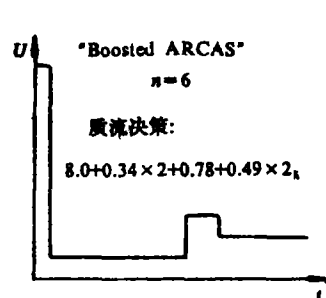
采用全推力公式

图 4



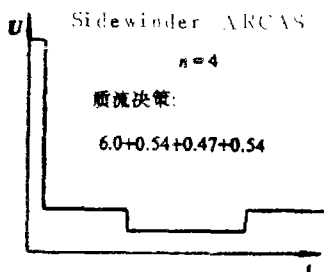
采用全推力公式

图 5



采用非全推力公式

图 6



采用非全推力公式
图 7

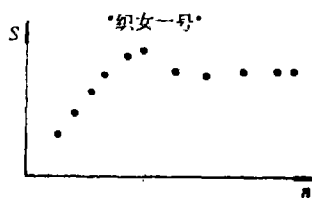


图 8 贝尔曼分级数与爬升高度值关系示意图

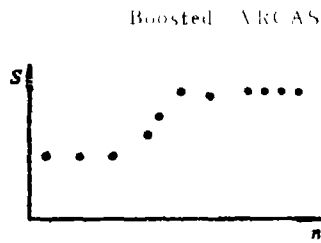


图 9

附 表

		织女一号(761)	Boosted ARCAS	Sidewinder ARCAS
采用全推力公式	S_1	86.84	103.89	196.80
	S_2	118.22	140.61	235.73
	收益	36.13%	35.34%	19.78%
采用非全推力公式	S_1	74.63	91.00	189.72
	S_2	76.61	93.97	189.74
	收益	2.64%	3.27%	0.012%

备注 ① S_1 ——采用实际参数计算得到的垂直爬升高度;
 ② S_2 ——通过分级选优计算获得的最大垂直爬升高度
 ③ 单位: 高度 km

参 考 文 献

[1] [西德]海宁·吐尔. 最优化方法. 国防工业出版社
 [2] Richard Bellman, Dynamic Programming. Princeton University Press, 1957
 [3] [美]伦·库柏, 玛丽·W·库柏. 动态规划导论. 国防工业出版社
 [4] [苏]B.И.瓦弗洛维也夫, М.И.科普托夫. 弹道式导弹设计和试验. 国防工业出版社

A Direct Method to Search out the Best Thrust Plan for Sounding Rocket

Liu Jixiang Gan Chuxiong

Abstract

In this paper, a direct optimal method—the method of Bellman dynamic programming has been used to search out the best thrust plan in a optimization process for sounding rocket. The recurrence formula of local optimization is set up in it. In the paper, because of using the coefficients to restrict the

seeking range in each Bellman stage and the scheme of single spot insertion, the calculation speed and accuracy has been increased greatly and the inserted calculations of non-intersection parameters points have become more stable and relatively credible level can be available.

Moreover, the calculation has been programed in a microcomputer IBM PC/XT. The program can be used directly for rough design of rocket. It will provide a theoretical basis with relatively credible level for the scheme proof.

Key words: sounding rocket, optimization, thrust plan