

微分代数控制问题的实时计算

费景高

(北京计算机应用和仿真技术研究所, 100854)

摘 要

本文提出微分代数控制问题的一种实时计算方法, 进行误差估计, 给出采样间隔与控制误差的关系. 对于一个模型问题进行了稳定性分析, 画出稳定区域, 给出保证控制稳定的采样间隔的取值范围.

关键词: 微分代数问题, 实时控制, 稳定性.

一、前 言

许多空间飞行器的轨道控制问题都可以表示成微分代数问题. 飞行器的运动由常微分方程组

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}, \mathbf{u}) \quad (1)$$

给出, 飞行轨道约束由附加的代数方程组

$$\mathbf{g}(\mathbf{y}) = 0 \quad (2)$$

给出, 其中 \mathbf{y} 和 \mathbf{f} 的维数为 l , \mathbf{u} 和 \mathbf{g} 的维数为 m . 称 \mathbf{y} 为状态变量, 它表示飞行器的位置、速度或质量等状态. 称 \mathbf{u} 为控制量或代数变量, 它可以是攻角 α 、侧滑角 β 或者飞行程序角等控制量. 轨道控制问题为求控制 \mathbf{u} , 使轨道 $\mathbf{y}(t)$ 满足代数方程(2), 即求解微分代数问题(1)(2).

最近许多文献构造了求解微分代数问题的算法. 文献 [1,2] 利用向后微分公式构造的算法求解空间往返飞船再入问题, 使飞行器沿着过渡轨道飞入安全的空中走廊. 文献 [3] 提出构造计算微分代数控制规律的一种新途径, 构造了一些算法, 并且证明构造的算法是 A_0 和 $A(\alpha)$ -稳定的, 控制误差可达到 $O(h^3)$, 其中 h 为采样间隔. 但是, 上述的算法都是非实时的.

文献 [4] 提出一个实时计算方法, 但其控制误差只能达到 $O(h^2)$. 本文将文献 [4] 的算法进行改进, 使控制误差可达到 $O(h^3)$ 的精度阶, 并对模型问题进行稳定性分析. 最后, 给出应用本文算法计算耗尽关机控制问题的一些结果.

二、实时控制算法

在实时控制计算时,状态变量 \mathbf{y} 的值由系统的实际运动状态采样得到. 设采样间隔为 h , 则采样时间序列为 $t_n = t_0 + nh$, $n = 0, 1, 2, \dots$. 设在区间 $[t_n, t_{n+1}]$ 上的控制值为 $\mathbf{u}(t) = \mathbf{u}_n$ 和在 t_n 处的采样值为 \mathbf{y}_n , 则在 t_{n+1} 处的采样值 \mathbf{y}_{n+1} 为初值问题

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}, \mathbf{u}_n), \quad \mathbf{y}(t_n) = \mathbf{y}_n \quad (3)$$

的解 $\mathbf{y}(t)$ 在 t_{n+1} 处的值, 即 $\mathbf{y}_{n+1} = \mathbf{y}(t_{n+1})$. 所谓实时控制算法, 是指在时刻 t_{n+1} 到来之前, 完成系统在区间 $[t_{n+1}, t_{n+2}]$ 上的控制值 \mathbf{u}_{n+1} 的全部计算的算法. 它可利用的信息只能是 t_{n+1} 以前得到的值. 文献[1]~[3]中的算法不是实时的, 因为在计算 \mathbf{u}_{n+1} 时, 它们都用到了 t_{n+1} 处的采样值 \mathbf{y}_{n+1} .

下面给出微分代数问题(1)(2)的一个实时控制算法. 设已知 t_n 处的采样值 \mathbf{y}_n , 预估值 \mathbf{y}_n^p , 控制 \mathbf{u}_n 和 $\mathbf{g}(\mathbf{y}_{n+2})$ 的估计值 \mathbf{w}_n . 按下面的计算步骤计算 \mathbf{y}_{n+1}^p , \mathbf{u}_{n+1} 和 \mathbf{w}_{n+1} .

(1) 由 $\mathbf{y}_n, \mathbf{u}_n$ 计算 \mathbf{y}_{n+1} 的预估值 \mathbf{y}_{n+1}^p

$$\mathbf{y}_{n+1}^p = \mathbf{y}_n + hf(t_n, \mathbf{y}_n, \mathbf{u}_n). \quad (4)$$

(2) 由 \mathbf{y}_n 和 \mathbf{y}_n^p 对 \mathbf{y}_{n+1}^p 的预估误差进行校正

$$\mathbf{y}_{n+1}^R = \mathbf{y}_{n+1}^p + \Delta\mathbf{y}_n^p, \quad \Delta\mathbf{y}_n^p = \mathbf{y}_n - \mathbf{y}_n^p. \quad (5)$$

(3) 对未知量 \mathbf{u} 求解方程

$$\mathbf{g}_y(\mathbf{y}_{n+1}^R)\mathbf{f}(t_{n+1}, \mathbf{y}_{n+1}^R, \mathbf{u}) = 0. \quad (6)$$

记其解为 \mathbf{u}_{n+1}^p .

(4) 计算 $\mathbf{g}(\mathbf{y}_{n+1}^R)$, 并取 \mathbf{w}_{n+1} 为

$$\mathbf{w}_{n+1} = \mathbf{g}(\mathbf{y}_{n+1}^R) + \mathbf{w}_n. \quad (7)$$

(5) 对未知量 $\Delta\mathbf{u}$ 解线性方程

$$\mathbf{g}(\mathbf{y}_{n+1}^R) + \mathbf{w}_{n+1} + h\mathbf{g}_y(\mathbf{y}_{n+1}^R)\mathbf{f}_u(t_{n+1}, \mathbf{y}_{n+1}^R, \mathbf{u}_{n+1}^p)\Delta\mathbf{u} = 0. \quad (8)$$

(6) 取

$$\mathbf{u}_{n+1} = \mathbf{u}_{n+1}^p + \Delta\mathbf{u}. \quad (9)$$

由这个算法看出, 它只用到 t_{n+1} 以前得到的信息. 图 1 给出采样时间间隔 $[t_n, t_{n+1}]$ 内实时控制算法的计算示意图.

上述算法的计算复杂性随具体问题的复杂程度而定. 由算法的计算步骤可以看出计算量主要集中在式(4)中 $\mathbf{f}(t_n, \mathbf{y}_n, \mathbf{u}_n)$ 的计算和方程(6)(8)的求解上. 下一节的误差估计得出公式(4)得到的 \mathbf{y}_{n+1}^p 和方程(6)(8)中用到的量都带有一定阶的误差. 因此函数值 $\mathbf{f}(t_n, \mathbf{y}_n, \mathbf{u}_n)$ 的计算和方程(6)(8)的求解不需要十分精确, 只要其计算误差与下一节的误差估计相适应. 对于具体问题, 可以构造函数 $\mathbf{f}(t, \mathbf{y}, \mathbf{u})$ 和 $\mathbf{g}(\mathbf{y})$ 的近似计算模型, 减少每一个采样间隔内算法的计算复杂性.

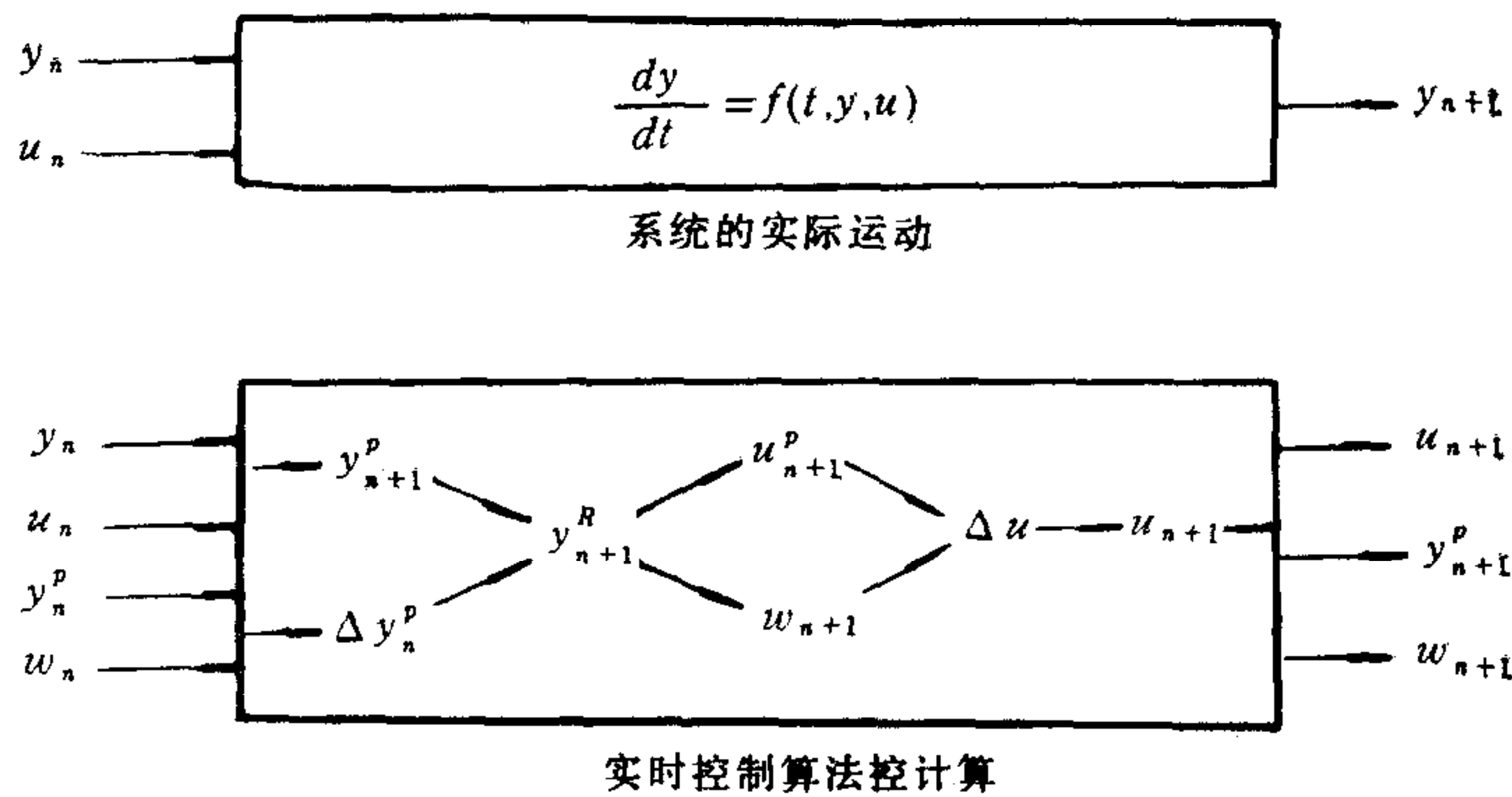


图 1 $[t_n, t_{n+1}]$ 内实时控制算法计算示意图

三、实时控制算法的误差估计

由于 y_n 是系统实际运动的采样值, 方程(1)是满足的. 为了估计问题(1)(2)的实时控制算法的误差, 只需估计序列 $g(y_n)$, $n = 0, 1, \dots$ 的值. 在下面的估计中, 假定函数 f, g 对其变量具有用到的各阶导数, 并且连续有界, 矩阵 g, f_u 具有有界逆矩阵 $[g, f_u]^{-1}$.

由于式(4)恰好是欧拉公式, 有估计式

$$y_{n+1} - y_{n+1}^p = \frac{1}{2} y''(t_n) h^2 + O(h^3).$$

所以有

$$y_{n+1} - y_{n+1}^R = \frac{1}{2} (y''(t_n) - y''(t_{n-1})) h^2 + O(h^3) = O(h^3). \tag{10}$$

由式(6)和(8)

$$\begin{aligned} & g(y_{n+1}^R) + h g_y(y_{n+1}^R) f(t_{n+1}, y_{n+1}^R, u_{n+1}) \\ &= g(y_{n+1}^R) + h g_y(y_{n+1}^R) f(t_{n+1}, y_{n+1}^R, u_{n+1}^p) \\ & \quad + h g_y(y_{n+1}^R) f_u(t_{n+1}, y_{n+1}^R, u_{n+1}^p) \Delta u + \frac{1}{2} h g_y f_{uu} (\Delta u)^2 \\ &= -w_{n+1} + \frac{1}{2} h g_y f_{uu} (\Delta u)^2. \end{aligned}$$

在上式中, 我们省略了 $g_y f_{uu}$ 取值点的表示(下同). 再由式(10), 我们得到

$$\begin{aligned} & g(y_{n+1}) + h g_y(y_{n+1}) f(t_{n+1}, y_{n+1}, u_{n+1}) \\ &= -w_{n+1} + \frac{1}{2} g_y f_{uu} (\Delta u)^2 + O(h^3). \end{aligned} \tag{11}$$

方程(8)的解 Δu 可表示成

$$\Delta u = -\frac{1}{h} A_{n+1}^{-1} (g(y_{n+1}^R) + w_{n+1}) + O(h^2).$$

式中

$$A_{n+1} = g_y(y_{n+1}^R) f_u(t_{n+1}, y_{n+1}^R, u_{n+1}^p).$$

现在,我们应用上述各式来估计 $g(y_{n+2})$.

$$\begin{aligned} g(y_{n+2}) &= g(y_{n+1}) + g_y(y_{n+1})(y_{n+2} - y_{n+1}) \\ &\quad + \frac{1}{2} g_{yy}(y_{n+1})(y_{n+2} - y_{n+1})^2 + O(h^3) \\ &= g(y_{n+1}) + h g_y(y_{n+1}) f(t_{n+1}, y_{n+1}, u_{n+1}) + \frac{1}{2} g_y(y_{n+1}) y''(t_{n+1}) h^2 \\ &\quad + \frac{1}{2} g_{yy}(y_{n+1}) [f(t_{n+1}, y_{n+1}, u_{n+1})]^2 h^2 + O(h^3) \\ &= -w_{n+1} + R_{n+1} + \frac{1}{2} h g_y f_{uu} (\Delta u)^2 + O(h^3). \end{aligned} \quad (12)$$

式中

$$R_{n+1} = \frac{1}{2} [g_y(y_{n+1}) y''(t_{n+1}) + g_{yy}(y_{n+1}) [f(t_{n+1}, y_{n+1}, u_{n+1})]^2] h^2.$$

由式(10)和(7)类推,再利用式(12),有

$$\begin{aligned} w_{n+2} &= g(y_{n+2}^R) + w_{n+1} = g(y_{n+2}) + w_{n+1} + O(h^3) \\ &= R_{n+1} + \frac{1}{2} h g_y f_{uu} (\Delta u)^2 + O(h^3). \end{aligned} \quad (13)$$

将 $g(y_n)$ 和 w_n 分别表示成

$$g(y_n) = \mu_n h^2, \quad w_n = \lambda_n h^2, \quad (14)$$

得到 Δu 的表示式

$$\Delta u = -A_{n+1}^{-1} (\mu_{n+1} + \lambda_{n+1}) h + O(h^2).$$

再由式(12)(13),我们得到递推式

$$\mu_{n+2} = -\lambda_{n+1} + \epsilon_{n+1} + \frac{1}{2} g_y f_{uu} [A_{n+1}^{-1} (\mu_{n+1} + \lambda_{n+1})]^2 h + O(h), \quad (15)$$

$$\lambda_{n+2} = \mu_{n+2} + \lambda_{n+1} + O(h). \quad (16)$$

式中的 ϵ_{n+1} 为量 R_{n+1} 的 h^2 项的系数.

设在所考虑的区域中存在估计式

$$\begin{aligned} \|\epsilon_{n+1}\| &\leq \beta_1, \\ \|A_{n+1}^{-1}\| &\leq \beta_2, \\ \|g_y f_{uu}\| &\leq \beta_3, \\ \|O(h)\| &\leq \beta_4 h. \end{aligned}$$

$\beta_1 - \beta_4$ 为常数. 记

$$\delta_n = \|\lambda_n\| + \|\mu_n\|.$$

由式(15)(16),

$$\begin{aligned} \|\lambda_{n+2}\| &\leq \beta_1 + 2\beta_4 h + \frac{1}{2} \beta_3 \beta_2^2 h \delta_{n+1}^2, \\ \|\mu_{n+2}\| &\leq \|\lambda_{n+1}\| + \beta_1 + \beta_4 h + \frac{1}{2} \beta_3 \beta_2^2 h \delta_{n+1}^2 \\ &\leq 2\beta_1 + 3\beta_4 h + \frac{1}{2} \beta_3 \beta_2^2 h \delta_{n+1}^2 + \frac{1}{2} \beta_3 \beta_2^2 h \delta_n^2. \end{aligned}$$

将这两个不等式相加, 得到递推式

$$\delta_{n+2} \leq \alpha_0 + \alpha_1 \delta_{n+1}^2 + \alpha_2 \delta_n^2.$$

式中

$$\alpha_0 = 3\beta_1 + 5\beta_4 h,$$

$$\alpha_1 = \beta_3 \beta_2^2 h,$$

$$\alpha_2 = \frac{1}{2} \beta_3 \beta_2^2 h.$$

容易证明, 对于任何满足

$$\max\{\|\lambda_0\| + \|\mu_0\|, \|\lambda_1\| + \|\mu_1\|, 3\beta_1\} < \delta$$

的正数 δ , 可找到 $h_0 > 0$, 当 h 满足 $0 < h \leq h_0$ 时, 有

$$\delta_n < \delta.$$

这表示 $\|\lambda_n\|$ 和 $\|\mu_n\|$ 都以正数 δ 为上界. 从而, 由式(13)推得

$$w_{n+1} = R_n + O(h^3). \quad (17)$$

再由函数连续可微性的假定, 有

$$R_{n+1} = R_n + O(h^3). \quad (18)$$

将 w_{n+1} 的式(17)代入式(12), 并考虑式(18), 得到估计式

$$g(y_{n+2}) = O(h^3).$$

此式表明本文给出的实时控制算法可达到 $O(h^3)$ 精度阶.

四、实时控制算法的稳定性分析

本节针对模型问题

$$\frac{dy}{dt} = Ay + Bu,$$

$$g(y) = C + y = 0.$$

讨论实时控制算法的控制稳定性, 其中 A, B, C 都是常数, 并且可取复数值.

记 $\mu = Ah$, 将实时控制算法应用到模型问题, 得到递推式

$$\begin{bmatrix} y_{n+1} \\ y_{n+1}^p \\ w_{n+1} \\ u_{n+1} \end{bmatrix} = \begin{bmatrix} e^\mu & 0 & 0 & \frac{Bh}{\mu}(e^\mu - 1) \\ 1 + \mu & 0 & 0 & Bh \\ 2 + \mu & -1 & 1 & Bh \\ -\frac{1}{Bh}(2 + \mu)^2 & \frac{1}{Bh}(2 + \mu) & -\frac{1}{Bh} & -(2 + \mu) \end{bmatrix} \begin{bmatrix} y_n \\ y_n^p \\ w_n \\ u_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ C \\ -\frac{2C}{Bh} \end{bmatrix}.$$

该递推式稳定的充分条件是其系数矩阵的特征方程

$$\lambda^4 + (1 + \mu - e^\mu)\lambda^3 + \left[\frac{1}{\mu}(e^\mu - 1)(2 + \mu)^2 - e^\mu(1 + \mu) - (3 + 2\mu) \right] \lambda^2 + \left[e^\mu(3 + 2\mu) + 1 + \mu - \frac{2}{\mu}(e^\mu - 1)(2 + \mu)(1 + \mu) \right] \lambda$$

$$+ (1 + \mu) \left[\frac{1}{\mu} (e^\mu - 1)(1 + \mu) - e^\mu \right] = 0$$

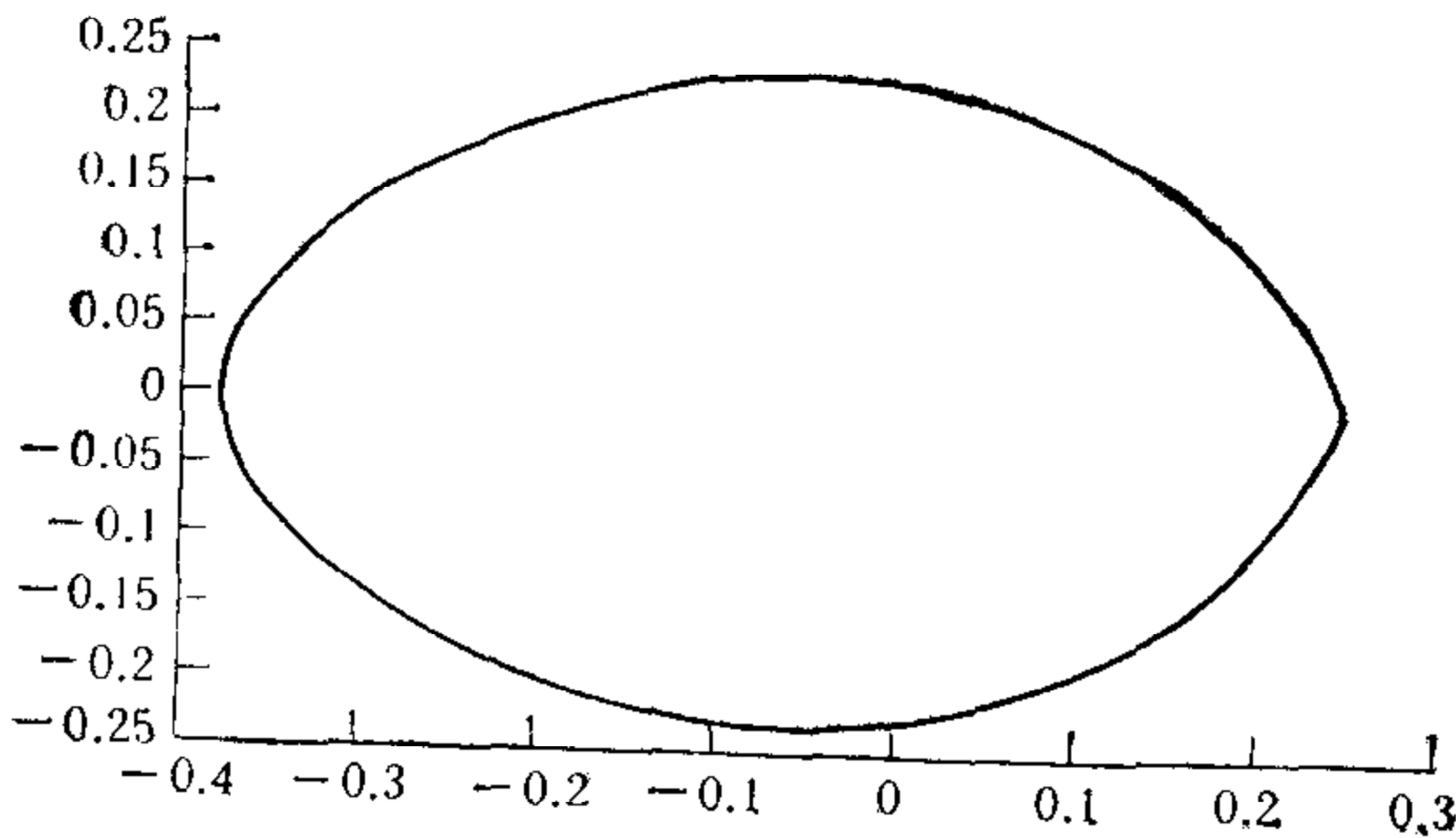


图2 实时控制算法的稳定区域

的根的模都小于1. 当 $\mu = 0$ 时, 特征方程退化为

$$\lambda^4 = 0,$$

其特征根都是零. 由特征方程的根与其系数的连续依赖性, 当 h 较小时, 实时控制算法是稳定的. 在图2中画出实时控制算法的稳定区域. 只要 h 使 μ 落在稳定区域的内部, 系统的运动将是稳定的.

对于非线性方程(1), 考虑矩阵 $h \frac{\partial f}{\partial y}$ 的

特征值 $\mu_i, i = 1, \dots, l$, 若 μ_i 都落在稳定区域的内部, 系统的运动将是稳定的.

五、耗尽关机射程控制的实时计算

这一节给出应用本文算法的一个例子, 并给出一些数字仿真结果.

远程弹道式导弹的射程 L 为其主动段终点(推力终止)时刻的轨道参数 $t, v_x, v_y, v_z, x, y, z$ 的函数

$$L = L(t, v_x, v_y, v_z, x, y, z).$$

所谓耗尽关机射程控制是指在导弹发射后, 当其飞行轨道参数可达到指定的射程 L_0 后, 将导弹的飞行轨道限制在一条等射程轨道上, 从而不需要发动机的关机装置. 在这轨道上的任意瞬间耗尽推进剂而终止推力时, 经被动段飞行所得到的射程都可达到预定的值. 这就需要在导弹的运动方程外, 再引进代数约束方程

$$\Delta L = L(t, v_x, v_y, v_z, x, y, z) - L_0 = 0. \quad (19)$$

导弹运动方程和式(19)组成微分代数控制问题, 其控制量为程序角 φ_{cx} 和偏航角 ψ . 本文只考虑导弹在赤道平面上的运动, 这时只需一个控制量 φ_{cx} .

表1列出应用实时控制算法的数字仿真结果, 其中采样间隔 h 取两个值, $L_0 =$

表1 实时控制算法的计算结果

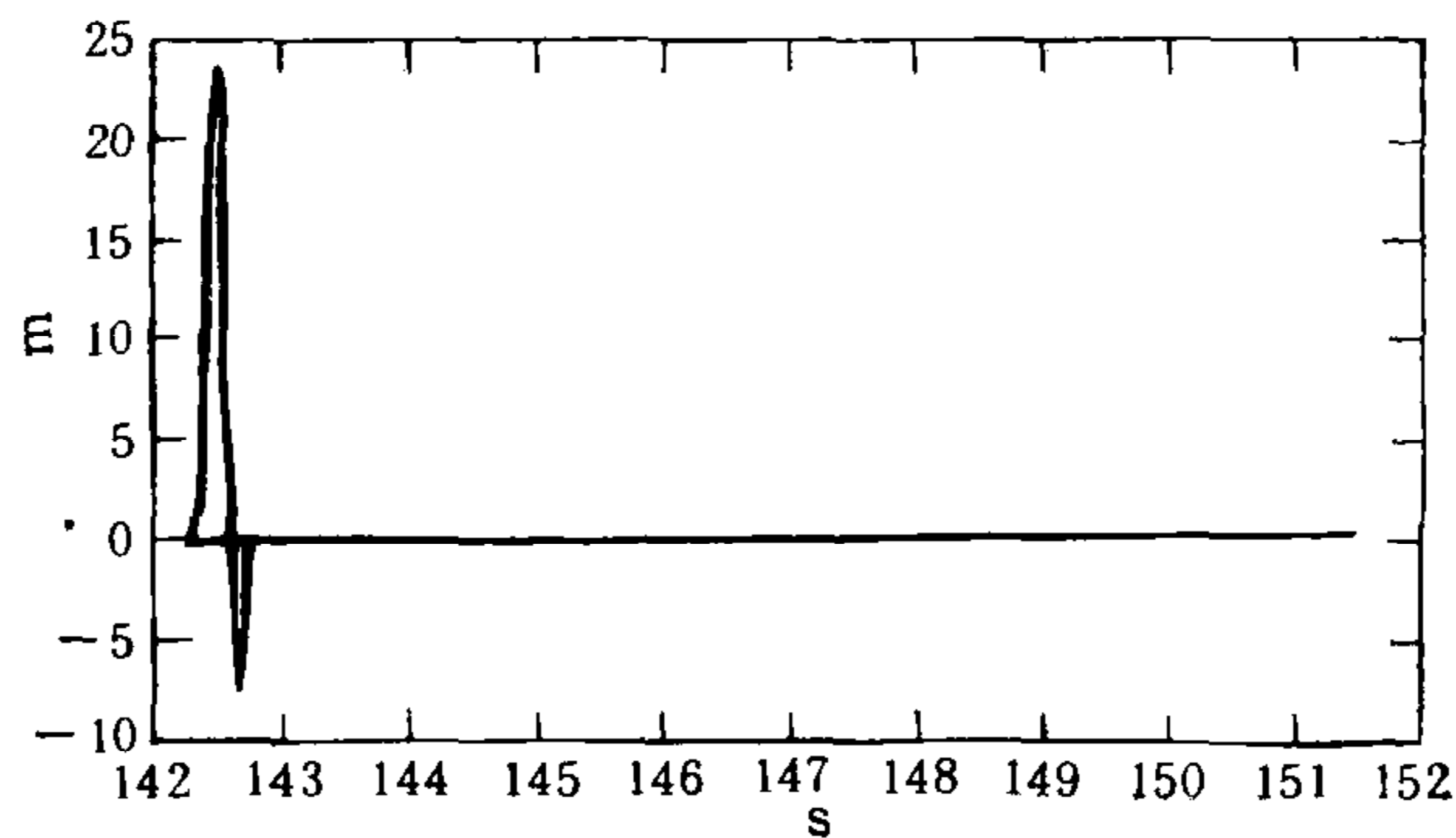
t	$\Delta L(m)$	
	$h = 0.125$	$h = 0.0625$
143	-0.055665918	0.0092385833
144	0.087757123	0.010992614
145	0.10480253	0.013137438
146	0.12587070	0.015790985
147	0.15225041	0.019117013
148	0.18576406	0.023347545
149	0.22904487	0.028818372

表 2 $h = 0.125$ 开始一些步上 ΔL 的值

t	$w_0 = 0$	$w_0 = 7.8, w_1 = 39.89, w_2 = 25.246$
142.25	-0.23432638	-0.23432638
142.375	6.6254352	0.0023282274
142.5	23.559722	0.010337733
142.625	-7.3409383	0.0036726003
142.75	-0.20244188	0.071203250
142.875	0.38699105	0.072665125
143.	-0.055665918	0.073278815
143.125	0.086669938	0.075567290
143.25	0.082709675	0.077053754

6000000 m 。从表 1 所列的结果看出, 当 h 缩小一半时, ΔL 的绝对值将减小到原来的 $\frac{1}{8}$ 。这证实了误差以 $O(h^3)$ 的速度减小的结论。

在计算表 1 所列出的结果时, 取 $w_0 = 0$, 而 $w_n, n = 1, 2, \dots$ 由算法生成。在计算开始的几步, ΔL 值出现振荡现象, 见表 2 列出的 $h = 0.125$ 的一些值。通过选取 w_0, w_1, w_2 的值可以消除这种初始振荡现象。例如取 $w_0 = 7.8, w_1 = 39.89, w_2 = 25.246$, 这种振荡就消失掉了。在表 2 中也列出 w_0, w_1, w_2 经选取后的 ΔL 值。图 3 画出对应于 $w_0 = 0$ 和 $w_i, i = 0, 1, 2$ 经选取后的 $\Delta L(t)$ 的曲线。由曲线看出, 两种结果很快重合。 $w_0 = 0$ 的曲线能很快达到曲线的稳定值。若初始振荡对实际控制的效果影响不大, 可以不进行 w_0, w_1, w_2 的选取。

图 3 w_i 初值选取对实时控制算法计算结果的影响

----- $w_0 = 0$ —— $w_0 = 7.8, w_1 = 39.89, w_2 = 25.246$

六、结 论

本文给出微分代数控制问题的一个实时计算方法, 证明了它的稳定性, 推导了控制误差的理论公式, 证明误差是以 h^3 的速度减小的。仿真例子表明, 这种算法可以达到很高的控制精度。与文[4]中的控制误差相比, 本文算法的误差仅为文[4]的 $\frac{1}{5.88}$ 。另外, 理论推导和仿真结果表明, 算法具有自校正性质。每一步引进的误差, 在下一步能得到校

正,控制误差中仅保留当前步引进的误差。这些误差不会累积起来。在实际应用时,可对方程(6)和(8)的求解进行近似处理,只要求处理误差保持在每一步允许误差的范围内,不需考虑它们对后面传递的影响。

参 考 文 献

- [1] Brennan, K. E., Numerical solution of trajectory prescribed path control problems by backward differentiation formulas, *IEEE Tran. Aut. Control*, **AC-31** (1986), 266—269.
- [2] Brennan, K. E., Campbell, S. L., Petzold, L. R., Numerical solution of initial-value problems in differential-algebraic equations, North-Holland, 1989.
- [3] 费景高,微分代数控制问题的数值计算方法,系统工程与电子技术, (1992), (6), 30—39.
- [4] 费景高,微分代数问题的实时控制计算,航天控制, (1992), (3), 47—56.

THE REAL-TIME COMPUTATION FOR THE CONTROL PROBLEM IN DIFFERENTIAL ALGEBRAIC SYSTEM

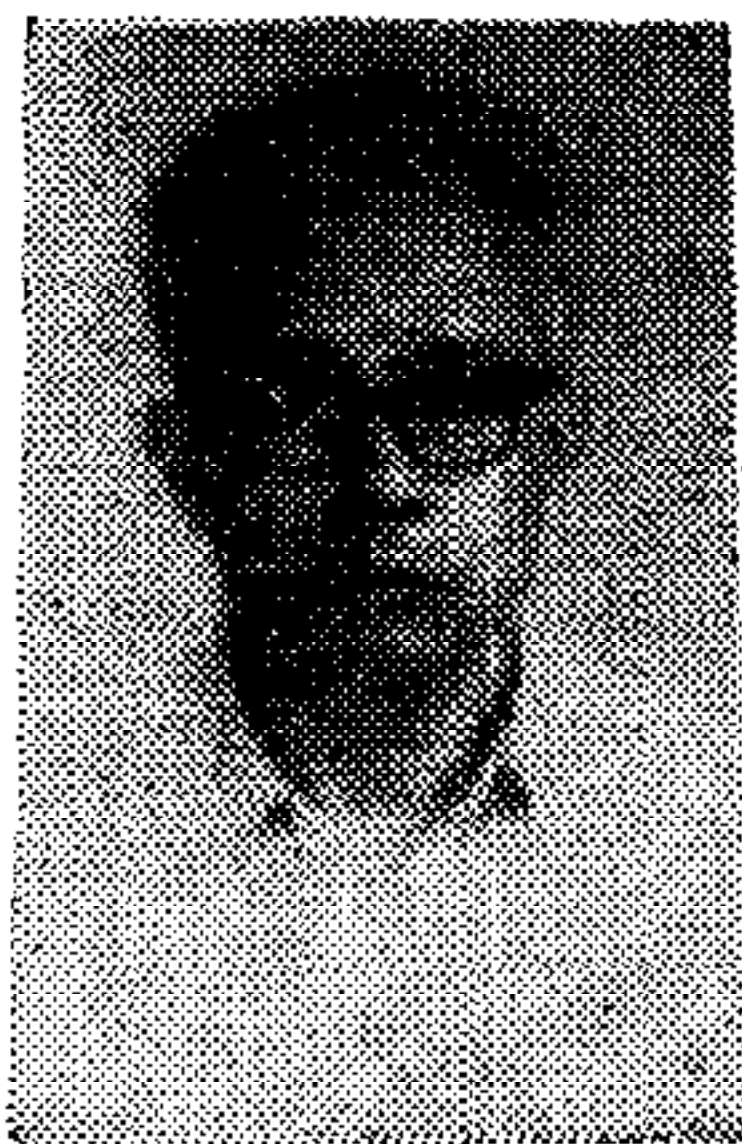
FEI JINGGAO

(Beijing Institute of Computer Application and Simulation Technology, 100854)

ABSTRACT

In this paper, a real-time computation method for the control problem in differential algebraic system is presented. The errors of the method are estimated, and the relation between the sampling stepsize and the controlled errors is given. This paper deals with the stability analysis for a model problem and plots the stability regions which gives the range of the sampling stepsizes for which the stability of control process is guaranteed.

Key words Differential algebraic problem; Real-time control; Stability.



费景高 1938年3月生于江苏无锡,1960年毕业于北京师范大学数学系。现为航空航天部第二研究院研究员。研究领域为数值分析、数字仿真、并行计算、优化和最优控制。