第 10 卷第 5 期
1993 年 10 月

控制理论与应用
CONTROL THEORY AND APPLICATIONS

Vol. 10, No. 5
Oct., 1993

线性定常系统离散相似法仿真的快速算法

金炜东
(西南交通大学电气工程系·成都,610031)

摘要：考虑线性定常系统的数字仿真,状态变量的计算步长为 T,而系统输出的计算间隔常常为 NT. 本文通过以多项式插值函数逼近系统输入,利用增广矩阵法的结果,给出了基于离散相似法的一类仿真算法. 当 N 较大时,与一般同类算法相比,本文算法使计算量显著减小.

关键词：数字仿真；线性定常系统；离散相似法；增广矩阵法

1 引　言

考虑线性定常系统

$$\dot{X}(t) = AX(t) + BU(t), \tag{1}$$

$$Y(t) = CX(t) + DU(t). \tag{2}$$

其中 $X(t)\in R^n, U(t)\in R^r, Y(t)\in R^m, A, B, C, D$ 为相应维数的常数阵. 由(1)式的解可给出离散相似法仿真的常见递推公式[1,2]

$$X((j+1)T) = e^{AT}X(jT) + [\int_0^T e^{A(T-\tau)}B\mathrm{d}\tau]U(jT). \tag{3}$$

(3)式精度较差. 为提高精度,文[3～5]给出了类似形式的一些改进算法. 这些算法及其它一些精度较高的算法每进一步 T 的计算量(指乘法次数)至少是 $n^2+O(n)$[4]. 注意到仿真中往往每进 N 步 T 才计算一次要求的输出 $Y(kNT)$,通常 $N\gg 1$[2]. 上述算法计算一次 $Y(kNT)$ 的计算量为 $Nn^2+NO(n)$. 本文利用线性定常系统的特性给出离散相似法仿真的一类快速算法. 该算法仍保证较高精度,同时将计算一次输出 $Y(kNT)$ 的计算量降为 $NO(n)+n^2$,从而可较大幅度地提高仿真速度.

2 算法原理及计算公式

考虑状态方程(1)的基本计算步长为 T,输出 $Y(t)$ 的计算间隔为 NT.

对于 $U(t)=[u_1(t),\cdots,u_r(t)]^T$,可将 $BU(t)$ 写成 $BU(t)=\sum_{i=1}^{r}B_iu_i(t)$,其中,$B_i\in R^{n\times 1}$ 为 B 中的第 i 个列向量. 可将(1)式的解写成

$$X((k+1)NT) = e^{ANT}X(kNT) + \sum_{i=1}^{r}\int_0^{NT} e^{A(NT-\tau)}B_iu_i(\tau+kNT)\mathrm{d}\tau. \tag{4}$$

记

$$I_a^b = \int_{aT}^{bT} e^{A(NT-\tau)}B_iu_i(\tau+kNT)\mathrm{d}\tau,$$

本文于1992年2月28日收到.

有
$$I_0^N = \sum_{j=0}^{N-1} I_j^{j+1},$$
$$I_j^{j+1} = \int_{jT}^{(j+1)T} e^{A(NT-\tau)} B_i u_i(\tau + kNT)\mathrm{d}\tau,$$
$$\overset{t=\tau-jT}{=} e^{A(N-1-j)T}\int_0^T e^{A(T-t)} B_i u_i(t + jT + kNT)\mathrm{d}t, \tag{5}$$

在$[jT,(j+1)T]$上以 L 次多项式插值函数逼近输入 $u_i(t+jT+kNT)$，即
$$u_i(t + jT + kNT) = \sum_{l=0}^{L} w_{ijl}(k)\frac{t^l}{l!} + R(u_i). \tag{6}$$

其中 $0\leqslant t\leqslant T$，$R(u_i)$为插值余项，$i=1,\cdots,r$；$j=0,1,\cdots,N-1$. 系数 $w_{ijl}(k)$由 $u_i(t)$的某些函数值给出. 于是，舍去插值余项，有
$$I_j^{j+1} = e^{A(N-1-j)T}\sum_{l=0}^{L}\left[\int_0^T e^{A(T-t)} B_i \frac{t^l}{l!}\mathrm{d}t\right] w_{ijl}(k). \tag{7}$$

对于 $i=1,\cdots,r$；$j=0,1,\cdots,N-1$；$l=0,1,\cdots,L$，令
$$Z_i(l,T) = \int_0^T e^{A(T-t)} B_i \frac{t^l}{l!}\mathrm{d}t, \tag{8}$$
$$Z_{ij}(l,T) = e^{A(N-1-j)T} Z_i(l,T). \tag{9}$$

$Z_i(l,T)\in \mathrm{R}^{n\times 1}$，$Z_{ij}(l,T)\in \mathrm{R}^{n\times 1}$，则(4)式的递推公式为
$$X((k+1)NT) = e^{ANT}X(kNT) + \sum_{i=1}^{r}\sum_{j=0}^{N-1}\sum_{l=0}^{L} Z_{ij}(l,T) w_{ijl}(k), \tag{10}$$

给出 $Z_i(l,T)$，即可由(9)，(10)，(2)式算出要求的输出 $Y((k+1)NT)$.

可利用增广矩阵法[1,5,6]的结果给出 $Z_i(l,T)$的计算公式. 为此，考察与(1)式相应的单输入系统
$$\dot{Z}_i(l,t) = AZ_i(l,t) + B_i\frac{t^l}{l!}. \tag{11}$$

令 $Z_i(l,0)=0$，则 $t=T$ 时(11)式的解 $Z_i(l,T)$即为(8)式. 对(11)式由增广矩阵法建立计算模型可得
$$\tilde{Z}_i(l,T) = e^{\tilde{A}_l T}\tilde{B}_l. \tag{12}$$

式中 $\tilde{Z}_i(l,T) = \begin{bmatrix} Z_i(l,T) \\ \cdots\cdots \\ Z_a(T) \end{bmatrix} \in \mathrm{R}^{n+l+1}$，$\tilde{A}_l = \begin{bmatrix} A & B_i & 0 \\ 0 & 0 & I_l \\ 0 & 0 & 0 \end{bmatrix}$，$\tilde{B}_l = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$，

其中 I_l 为 l 级单位阵.

根据 $\tilde{A}_l$的特点，可得出矩阵指数 $e^{\tilde{A}_l T}$为
$$e^{\tilde{A}_l T} = \begin{bmatrix} e_{11}(T) & e_{12}(i,l,T) \\ 0 & e_{22}(l,T) \end{bmatrix}_{(n+l+1)\times(n+l+1)}. \tag{13}$$

其中
$$e_{11}(T) = e^{AT} \in \mathrm{R}^{n\times n},\quad e_{12}(i,l,T) \in \mathrm{R}^{n\times(l+1)},$$

$$e_{22}(l,T)=\begin{bmatrix}1 & T & \cdots & T^l/l!\\ & \ddots & \ddots & \vdots\\ & 0 & \ddots & T\\ & & & 1\end{bmatrix}_{(l+1)\times(l+1)}.$$

并且，注意到 $\tilde{B}_l$ 的特点，由(12)式可得，$e_{12}(i,l,T)$的最后一列即为列向量 $Z_i(l,T)$. 易见，令 $l=L$，则有

$$e_{12}(i,L,T)=[Z_i(0,T),Z_i(1,T),\cdots,Z_i(L,T)]_{n\times(L+1)}. \tag{14}$$

于是，计算出 $e^{\lambda_i L T}$，即可由其中的 $e_{12}(i,L,T)$的各列给出 $Z_i(l,T)$，$l=0,1,\cdots,L$.

由(6)式可见，对于形如(10)式的仿真计算模型，选用不同的代数插值方法及阶次，则系数 $w_{ijl}(k)$随之确定，相应地可给出各种具体的递推公式. 这里，分别利用 Newton 插值表末公式和 Hermite 样条函数构造三次多项式插值函数，在$[jT,(j+1)T]$上逼近 $u_i(t+jT+kNT)$，给出 $w_{ijl}(k)$的两组具体计算公式.

等距节点代数插值的 Newton 表末公式[7]为

$$N_L(x_L-\theta h)=\sum_{\alpha=0}^{L}(-1)^{\alpha}\frac{\theta^{(\alpha)}\Delta^{\alpha}y_{L-\alpha}}{\alpha!} \tag{15}$$

其中　　$0<\theta<1$，　$\theta^{(\alpha)}=\theta(\theta-1)\cdots(\theta-\alpha+1)$；　$\Delta^0 y_L=y_L$，

$\Delta y_{L-1}=y_L-y_{L-1}$，　$\Delta^2 y_{L-2}=\Delta(\Delta y_{L-2})=\Delta y_{L-1}-\Delta y_{L-2},\cdots$.

据此公式，取 $L=3$，可给出(6)式中的多项式系数 $w_{ijl}(k)$为

$$\begin{cases} w_{ij0}(k)=y_2,\\ w_{ij1}(k)=[2(y_3-y_2)+5(y_2-y_1)-(y_1-y_0)]/6T,\\ w_{ij2}(k)=[(y_3-y_2)-(y_2-y_1)]/T^2,\\ w_{ij3}(k)=[(y_3-y_2)-2(y_2-y_1)+(y_1-y_0)]/T^3. \end{cases} \tag{16}$$

其中 $y_l=u_i(kNT+(j-2+l)T)$，$l=0,1,2,3$. 以(16)式计算 $w_{i0l}(0)$和 $w_{i1l}(0)$(若 $N=1$，则为 $w_{il}(0)$和 $w_{il}(1)$)时，要用到 $u_i(-T)$和 $u_i(-2T)$的值. 因此，应按其它公式计算 $w_{i0l}(0)$和 $w_{i1l}(0)$. 这里考虑增加两个插值点，根据 Newton 插值公式[7]构造仍为三次的插值多项式，以保证 $t<2T$ 时的计算具有同阶精度，可给出 $w_{ijl}(0)$，$j=0,1$，为

$$\begin{cases} w_{ij0}(0)=\bar{y}_0,\\ w_{ij1}(0)=\dfrac{1}{2T}(2\bar{y}_3-9\bar{y}_2+18\bar{y}_1-11\bar{y}_0),\\ w_{ij2}(0)=\dfrac{9}{T^2}(-\bar{y}_3+4\bar{y}_2-5\bar{y}_1+2\bar{y}_0),\\ w_{ij3}(0)=\dfrac{27}{T^3}(\bar{y}_3-3\bar{y}_2+3\bar{y}_1-\bar{y}_0). \end{cases} \tag{17}$$

其中

$$\bar{y}_l=u_i(jT+l\frac{T}{3}),\quad l=0,1,2,3.$$

若在 $t=kNT+jT$ 上还可得到 $\dot{u}_i(t)$的值，则可利用三次 Hermite 样条函数构造插值函数[7]. 据此可给出(6)式中 $L=3$ 的多项式系数 $w_{ijl}(k)$为

$$\begin{cases} w_{ij0}(k)=y_2,\\ w_{ij1}(k)=\dot{y}_2, \end{cases}$$

$$\begin{cases} w_{ij2}(k) = \dfrac{2}{T^2}[3(y_3 - y_2) - T(2\dot{y}_2 + \dot{y}_3)], \\ w_{ij3}(k) = \dfrac{6}{T^3}[2(y_2 - y_3) + T(\dot{y}_2 + \dot{y}_3)]. \end{cases} \tag{18}$$

其中 $y_l = u_i(kNT + (j-2+l)T)$，$\dot{y}_l = \dot{u}_i(-kNT + (j-2+l)T)$，$l = 2,3$.

以三次插值函数代替(5)式中的 $u_i(t+jT+kNT)$ 近似计算 I_j^{k+1}，其局部截断误差为 $O(T^5)$，与三次 Newton 插值相比，三次样条插值的逼近性质较好，插值余项的系数较小[7]，且无计算启动问题. 代价是每个插值点另需计算一次 $\dot{u}_i(t+jT+kNT)$.

考虑 e^{AT} 的计算易获得很高的精度，基本计算步长 T 的选择取决于对 $U(t)$ 的逼近精度，误差较易控制. $U(t)$ 变化较快，T 应取得小些，反之 T 可大些. 此外，当 $U(t)$ 为不高于三次的多项式时，按(18)或(16)式构造的三次插值对输入函数的逼近是准确的，无截断误差. 对于控制系统分析常用的阶跃、斜坡、加速度输入信号，这一性质很有意义. 文[3,4]的算法无此性质. 由于 T 的选取与系统矩阵 A 无关，因此本文算法对 Stiff 系统的仿真计算很有效.

3 编程算法

按(10)式计算 $X((k+1)NT)$，$Z_{ij}(l,T)$ 和 e^{ANT} 只需在递推计算前计算一次，且称预算. $Z_{ij}(l,T)$ 的计算中多次涉及矩阵指数 e^{At} 的计算. 注意 $\bar{A}_u$ 和 $\bar{B}_l$ 的特点设计编程算法可显著减少预算的计算量. 考虑收敛速度，先取 $h=T/N_1$，按级数展开式取前 $p+1$ 项计算 e^{Ah}，再经数次"跳乘"算出 e^{AT} 及 e^{ANT} 等[2]. 算法可归结如下.

1) $E_p = Ih^p/p!$. $E_{p-q} = AE_{p-q+1} + Ih^{p-q}/(p-q)!$，$q = 1,2,\cdots,p$. 则有 $e_{11}(h) = e^{Ah} = E_0$.

2) 设 $p \geqslant L+1$. $Z_i(l,h) = E_{l+1}B_i$，$l = 0,1,\cdots,L$；$i = 1,\cdots,r$. 则 $e_{12}(i,L,h) = [Z_i(0,h), Z_i(1,h),\cdots,Z_i(L,h)]_{n\times(L+1)}$.

3) 由(13)式有

$$e_{11}(t_1+t_2) = e_{11}(t_1)e_{11}(t_2),$$
$$e_{12}(i,L,t_1+t_2) = e_{11}(t_1)e_{12}(i,L,t_2) + e_{12}(i,L,t_1)e_{22}(L,t_2).$$

其中 $e_{22}(L,t_2)$ 可直接给出. 则由 $e_{11}(h)$ 和 $e_{12}(i,L,h)$ 易求得 $e_{11}(T) = e^{AT}$ 和 $e_{12}(i,L,T)$. 则由(14)式知，$Z_i(l,T)$，$l = 0,1,\cdots,L$，亦给出.

4) $Z_{i,N-1}(l,T) = Z_i(l,T)$，$l = 0,1,\cdots,L$.

$Z_{ij}(l,T) = e^{AT}Z_{i,j+1}(l,T)$，$j = N-2,\cdots,1,0$；$l = 0,1,\cdots,L$.

5) 由 e^{AT} "跳乘"算出 e^{ANT}.

于是，即可按(10)式递推计算 $X((k+1)NT)$. 此外，应将 $w_{ijl}(k)$ 中的不变系数(如(16)式中的 $1/6T$ 等)归入 $Z_i(l,T)$，使之为 $Z_i^*(l,T)$，形成 $Z_{ij}^*(l,T)w_{ijl}^*(k) = Z_{ij}(l,T)w_{ijl}(k)$，而 $w_{ijl}^*(k)$ 的计算量较小(如按(16)式，求 $w_{ijl}^*(k)$ 就不必作乘法).

4 算 例

多个算例的编程计算验证了本文算法的有效性. 这里略举一例.

$$\dot{X} = \begin{bmatrix} -10^3 & 1 \\ 0 & -1 \end{bmatrix} X + \begin{bmatrix} 0 & 1 \\ 10 & 0 \end{bmatrix} \begin{bmatrix} \sin\omega t \\ \cos\omega t \end{bmatrix}, \quad Y = [10^4, 0]X,$$

这是一个刚性比为 10^3 的 Stiff 系统. 取 $\omega=10$ 和 $\omega=1$ 的计算结果分别列于表 1 和表 2.

其中，$Y1$ 为根据(16)和(17)式计算，$Y2$ 为根据(18))式计算的结果，Y^* 为精确解. 比较可见：输入函数的周期增大 10 倍，同等精度下相应地计算步长可增大 10 倍；T 较大时，$Y2$ 的精度明显好于 $Y1$.

表 1　($\omega=10$)

Tn	$T=0.01$, $N=100$		Y^*	$T=0.05$, $N=20$	
	$Y1$	$Y2$		$Y1$	$Y2$
1	3.032171	3.032156	3.032136	3.049947	3.031134
2	2.282381	2.282377	2.282375	2.275953	2.282511
3	−0.4719369	−0.4719311	−0.471951	−0.4701886	−0.4717901
4	0.8605064	0.8604956	0.8605001	0.8671864	0.8598546
5	−0.1072692	−0.1072568	−0.1072533	−0.1190552	−0.1064196
6	−0.3623478	−0.3623518	−0.3623576	−0.3488067	−0.3631427
7	0.8323826	0.8323697	0.8323843	0.8216021	0.8328549
8	−0.9914561	−0.9914349	−0.9914445	−0.9868442	−0.9914572
	0.8472561	0.8472442	0.8472414	0.8503236	0.8467846
10	−0.4245414	−0.4245279	−0.4245208	−0.4342886	−0.4237469

表 2　($\omega=1$)

Tn	$T=0.1$, $N=10$		Y^*	$T=0.5$, $N=2$	
	$Y1$	$Y2$		$Y1$	$Y2$
1	38.81318	38.8131	38.81315	38.813152	38.81033
2	68.86865	68.86846	68.86866	68.96187	68.86237
3	49.19122	49.19105	49.19136	49.3088	49.18621
4	−10.7149	−10.7149	−10.71472	−10.67462	−10.71443
5	−58.93133	−58.93117	−58.93123	−59.003	−58.92592
6	−52.29024	−52.29007	−52.29031	−52.40708	−52.28491
7	2.67506	2.675089	2.674902	2.620793	2.675399
8	55.27246	55.27232	55.27235	55.33078	55.2675
9	57.08629	57.08609	57.08633	57.20363	57.08058
10	6.427634	6.42757	6.427783	6.496118	6.426429

5 结　语

本文以代数插值逼近输入函数，利用增广矩阵法的结果给出了线性定常系统的一类快速仿真算法，并容易构造各种精度及形式的计算模型. 该算法保留了同类算法计算稳定性好、精度较高等优点，而平均每进一步 T 的计算量仅为 $r(L+1)n+n^2/N$. 当 N 较大时，与一般同类算法相比[4]，计算量显著减小.

参 考 文 献

[1] 熊光楞. 控制系统数字仿真. 北京:清华大学出版社,1982
[2] 孙增圻,袁曾任. 控制系统的计算机辅助设计. 北京:清华大学出版社,1986
[3] 蒋珉,曹大铸,徐刚. 线性定常系统仿真的改进转移矩阵法. 控制理论与应用,1989,6(1):76—80
[4] 蒋珉,曹大铸. 线性定常系统仿真的拟 Adams 法. 控制与决策,1990, 5(5):24—28
[5] 金炜东. 线性定常系统仿真的广义增广矩阵法. 控制理论与应用,1993,10(2):183—189
[6] Taylor, F. J.. Transient Response Analysis of Linear Differential Systems on Minicomputers. Simulation, 1977, 28(1): 17—21
[7] 李岳生,黄友谦. 数值逼近. 北京:人民教育出版社,1978

A Fast Algorithm Based on Discrete Analog Method for the Simulation of Linear Time-Invariant Systems

JIN Weidong

(Department of Electrical Engineering, Southwest Jiaotong University • Chengdu, 610031, PRC)

Abstract: In course of the simulation of a linear time-invariant system, T is the calculating step of the state variables, and the system output is usually calculated at intervals of NT. Using polynomial interpolating function to approximate the system input and introducing the result obtained with augmented matrix algorithm, a fast simulation algorithm based on discrete analog method is presented in this paper. When N is large, the algorithm has much smaller computing amount than other similar algorithm.

Key words: digital simulation; linear time-invariant system; discrete analog method; augmented matrix algorithm

本文作者简介

金炜东 见本刊 1993 年第 2 期第 189 页.