

线性定常系统仿真的两种新算法

蒋 珉 柴 干

(东南大学自动化研究所·南京, 210018)

摘要: 本文提出了两种适合于对线性定常系统进行仿真的新算法. 它们不仅保持了改进转移矩阵法和拟 Adams 法的 A -稳定性、计算量较小和精度较高等优点, 而且弥补了这两种方法在理论上的不严密性和截断误差较大等缺陷, 因而具有更好的数值特性. 仿真结果表明, 文中提出的算法是有效可行的.

关键词: 仿真; 算法; 线性定常系统

1 引 言

考虑线性定常系统

$$\dot{X}(t) = AX(t) + BU(t). \quad (1)$$

其中 $X(t) \in \mathbb{R}^n$, $U(t) \in \mathbb{R}^m$ 分别为状态向量和输入向量, A 和 B 分别为 $n \times n$ 和 $n \times m$ 阶常值矩阵. 由于线性定常系统是自动控制中最常见的系统, 因此研究对该类系统进行仿真的算法有着很重要的理论意义和实用价值.

从理论上讲, 数值分析中所有有关常微分方程(组)数值解的算法都可以对系统(1)进行仿真. 然而, 当涉及到自动控制中某些特殊要求(例如, 快速性和实时性)时, 这些算法能否有效地满足仿真要求就很值得疑问了, 而且随着系统(1)Stiff 比(即矩阵 A 的最大特征值实部的绝对值与最小特征值实部的绝对值之比)的增大, 这种理论上的可行与实际应用要求之间的差异愈加突出. 为此, 已经研究出一些既具有快速性又具有 A -稳定性(即无条件稳定性)的仿真算法^[1]. 在这些算法中, 有一类方法是利用在时间区间 $[kT, (k+1)T]$ 上系统(1)的解析解

$$X[(k+1)T] = e^{AT}X(kT) + \int_{kT}^{(k+1)T} e^{A[(k+1)T-t]}BU(t)dt \quad (2)$$

的性质实现 A -稳定性; 通过将所有与序号 k 无关的计算在仿真开始前预先算好, 从而达到减小每步仿真递推计算量以实现快速性. 作者等在文献[2, 3]中提出的改进转移矩阵法(MTM 法)和拟 Adams 法就是采用了上述原则, 并且将多项式数值逼近的思想运用到(2)式右端积分项的处理中, 从而又实现了可选择计算精度. 因而它们均适合于对系统(1)进行仿真, 并且我们已经将它们成功地应用于实时仿真语言中^[4], 成为该语言中的两种基本积分算法.

然而, 两年多来的仿真实践中我们发现这两种算法在对中度以上的 Stiff 系统(Stiff 比 $> 10^3$)进行仿真时, 均存在着某些缺陷. 本文的工作正是为了弥补这些不足之处而进行的.

2 MTM 法与拟 Adams 法的不足

以 QAM-4^[3]对标量方程

$$\dot{x}(t) = ax(t) + u(t), \quad \text{Re} a < 0 \quad (3)$$

的仿真为例来阐述. 此时局部截断误差可写为

$$E = -\frac{19}{720}T^5[e^{a(T-\xi)}u(kT + t)]_{t=\xi}^{(4)} = -\frac{19}{720}T^5e^{a(T-\xi)}\left[\sum_{i=0}^4 \binom{4}{i} (-a)^{4-i}u^{(i)}(kT + \xi)\right], \\ -2T < \xi < T. \quad (4)$$

由(4)式可以看出 QAM-4 的两点不足之处.

其一,理论上不严密.

众所周知,若不考虑计算机舍入误差,一个 p 阶算法对(3)式进行仿真时,只要 $u(t)$ 为任意 $p-1$ 次以下多项式(包括 $p-1$ 次),仿真结果与解析解应该完全吻合. 而由(4)可以看出,除非 $u(t) \equiv 0$ (即为零输入),不可能有此结果.

其二,当 $|a|$ 很大时截断误差较大.

事实上,当 $|aT| > 1$ ($T < 1$) 时,(4)式中起主导作用的项为

$$E_1 = -\frac{19}{720}a^4T^5e^{a(T-\xi)}u(kT + \xi), \quad (5)$$

显然有

$$|E_1| > \frac{19}{720}T|e^{a(T-\xi)}u(kT + \xi)|$$

成立. 此时, QAM-4 退化成一阶算法公式.

类似情况在 Stiff 系统的仿真中也会发生,此时系统中总存在某个实部绝对值很大的特征值. 由于局部截断误差中存在与(5)式相似的项,使得步长 T 受到了限制,制约了算法优点的发挥.

对 MTM 法和拟 Adams 法中其它公式的讨论结果相同. 原因在于在这两种算法的推导中,均令

$$F(t) = e^{A[(k+1)T-t]}BU(t), \quad (6)$$

而引入了附加误差项. 因此有必要寻找更好的算法,以满足用大步长对 Stiff 系统仿真的需要.

3 新算法的推导

3.1 若干有用的公式

定义

$$G_i(T) \triangleq \frac{1}{T^i}e^{AT} \int_0^T t^i e^{-At} dt, \quad i = 0, 1, 2, \dots, \quad (7)$$

则下列递推公式成立(限于篇幅,证明省略):

$$e^{AT} = AG_0(T) + I, \quad (8)$$

$$G_{i-1}(T) = \frac{T}{i}[AG_i(T) + I], \quad (9)$$

$$G_i(T) = i!T \sum_{j=i+1}^{\infty} \frac{(AT)^{j-i-1}}{j!}, \quad (10)$$

$$G_i(2T) = \frac{1}{2^i} [e^{AT} G_i(T) + \sum_{j=0}^i \binom{i}{j} G_j(T)]. \quad (11)$$

3.2 算法1

令 $t = kT + s$, 则(2)式改写为

$$X[(k+1)T] = e^{AT} X(kT) + e^{AT} \int_0^T e^{-As} BU(kT + s) ds, \quad (12)$$

将区间 $[kT, (k+1)T]$ l 等分, 并令 $h = \frac{T}{l}$. 取插值基点

$$t_0 = kT, \quad t_1 = t_0 + h, \quad t_2 = t_0 + 2h, \quad \dots, \quad t_l = t_0 + lh = (k+1)T,$$

用 Newton 前差公式^[5]表示 $U(t)$ 在 $[kT, (k+1)T]$ 上的 l 次插值多项式

$$\begin{aligned} N_f(t) &= U(t_0) + \frac{t-t_0}{1!h} \Delta U(t_0) + \frac{(t-t_0)(t-t_1)}{2!h^2} \Delta^2 U(t_0) + \dots \\ &\quad + \frac{(t-t_0)(t-t_1)\dots(t-t_{l-1})}{l!h^l} \Delta^l U(t_0) \\ &\stackrel{t=kT+s}{=} U(kT) + \frac{s}{1!h} \Delta U(kT) + \frac{s(s-h)}{2!h^2} \Delta^2 U(kT) + \dots \\ &\quad + \frac{s(s-h)\dots[s-(l-1)h]}{l!h^l} \Delta^l U(kT) \\ &= U(kT) + \sum_{i=1}^l \frac{1}{i!h^i} \left[\sum_{j=0}^{i-1} d_{j,i} s^{i-j} \right] \Delta^i U(kT) \\ &= U(kT) + \sum_{i=1}^l \frac{1}{i!T^i} \left[\sum_{j=0}^{i-1} g_{j,i} s^{i-j} \right] \Delta^i U(kT). \end{aligned} \quad (13)$$

其中

$$\Delta U(t_0) = U(t_1) - U(t_0), \quad \Delta^2 U(t_0) = \Delta U(t_1) - \Delta U(t_0), \dots; \quad d_{0,i} = 1,$$

$$d_{1,i} = (-1)h \sum_{j=1}^{i-1} j, \quad d_{2,i} = h^2 \sum_{1 \leq r < j}^{i-1} pr, \quad d_{3,i} = (-1)h^3 \sum_{1 \leq r < s < j}^{i-1} prs, \dots,$$

$$d_{i-1,i} = (-1)^{i-1} h^{i-1} (i-1)!; \quad g_{j,i} = \frac{i-j T^j d_{j,i}}{h^j}, \quad i = 1, 2, \dots, l, \quad j = 0, 1, \dots, i-1.$$

插值余项为

$$\begin{aligned} R_f(t) &= \frac{U^{(l+1)}(\xi)}{(l+1)!} \prod_{i=0}^l (t - t_i), \quad t_0 < \xi < t_l \\ &\stackrel{t=kT+s}{=} \frac{U^{(l+1)}(kT + \eta)}{(l+1)!} \prod_{i=0}^l (s - \frac{i}{l}T), \quad 0 < \eta < T. \end{aligned} \quad (14)$$

将(13)式代入(12)式右端的积分项中, 得

$$\begin{aligned} X(k+1) &= e^{AT} X(k) + e^{AT} \int_0^T e^{-As} B \{ U(k) + \sum_{i=1}^l \frac{1}{i!T^i} \left[\sum_{j=0}^{i-1} g_{j,i} s^{i-j} \right] \Delta^i U(k) \} ds \\ &= e^{AT} X(k) + G_0(T) BU(k) + \sum_{i=1}^l \frac{1}{i!T^i} \left[\sum_{j=0}^{i-1} g_{j,i} T^{i-j} G_{i-j}(T) \right] B \Delta^i U(k) \\ &= e^{AT} X(k) + G_0(T) BU(k) + \sum_{i=1}^l \frac{1}{i!} \left[\sum_{j=0}^{i-1} \frac{T^{i-j} d_{j,i}}{h^j} G_{i-j}(T) \right] B \Delta^i U(k) \end{aligned}$$

$$= e^{At}X(k) + G_0(T)BU(k) + \sum_{i=1}^l \frac{1}{i!} \left[\sum_{j=0}^{i-1} \alpha_{j,i} G_{i-j}(T) \right] B \Delta^i U(k). \quad (15)$$

其中

$$\alpha_{0,i} = 1, \quad \alpha_{1,i} = (-1)^{i-1} \sum_{j=1}^{i-1} j, \quad \alpha_{2,i} = i^{-2} \sum_{1 \leq j < k} p_{jk}, \quad \dots, \quad \alpha_{i-1,i} = (-1)^{i-1} (i-1)!$$

从而可得仿真递推公式及其局部截断误差:

$$X(k+1) = e^{At}X(k) + \sum_{i=0}^l \Phi_i(T)U(k + \frac{i}{l}), \quad (16)$$

$$T(k+1) = \frac{e^{At}}{(l+1)!} \int_0^T \prod_{i=0}^l (s - \frac{i}{l}T) e^{-As} BU^{(l+1)}(kT + \eta) ds. \quad (17)$$

其中 $\Phi_i(T)$ 为 $G_j(T)$ ($j=0, 1, \dots, l$) 与 B 乘积的线性组合.

下面给出两个具体公式及其局部截断误差

$$l = 1,$$

$$X(k+1) = e^{At}X(k) + [G_0(T) - G_1(T)]BU(k) + G_1(T)BU(k+1), \quad (18)$$

$$T_2(k+1) = -\frac{T^3}{12} e^{A(T-\xi)} BU^{(2)}(kT + \eta), \quad 0 < \xi, \eta < T; \quad (19)$$

$$l = 2,$$

$$\begin{aligned} X(k+1) = & e^{At}X(k) + [G_0(T) - 3G_1(T) + 2G_2(T)]BU(k) \\ & + 4[G_1(T) - G_2(T)]BU(k + \frac{1}{2}) \\ & + [2G_2(T) - G_1(T)]BU(k+1), \end{aligned} \quad (20)$$

$$\begin{aligned} T_4(k+1) = & \frac{T^4}{64} [e^{A(T-\xi_1)} BU^{(3)}(kT + \eta_1) - e^{A(T-\xi_2)} BU^{(3)}(kT + \eta_2)] \\ = & \frac{T^5}{64} \theta(k+1) e^{A(T-\xi)} BU^{(4)}(kT + \eta), \quad 0 < \xi, \eta < T. \end{aligned} \quad (21)$$

其中 $\theta(k+1)$ 为一绝对值小于 1 的实数^[5], $0 < \xi_1, \eta_1 < \frac{T}{2}$, $\frac{T}{2} < \xi_2, \eta_2 < T$.

3.3 算法 2

取插值基点

$$t_{k+1} = (k+1)T, \quad t_k = kT, \quad t_{k-1} = (k-1)T, \quad \dots, \quad t_{k-l+1} = (k-l+1)T,$$

用 Newton 后差公式^[5]表示 $U(t)$ 在 $[t_{k-l+1}, t_{k+1}]$ 上的 l 次插值多项式

$$\begin{aligned} N_b(t) = & U(t_{k+1}) + \frac{t - t_{k+1}}{1!T} \nabla U(t_{k+1}) + \frac{(t - t_{k+1})(t - t_k)}{2!T^2} \nabla^2 U(t_{k+1}) + \dots \\ & + \frac{(t - t_{k+1}) \dots (t - t_{k-l+2})}{l!T^l} \nabla^l U(t_{k+1}). \end{aligned} \quad (22)$$

其中 $\nabla U(t_{k+1}) = U(t_{k+1}) - U(t_k)$, $\nabla^2 U(t_{k+1}) = \nabla U(t_{k+1}) - \nabla U(t_k)$, \dots .

$$\text{插值余项为} \quad R_b(t) = \frac{U^{(l+1)}(\xi)}{(l+1)!} \prod_{i=0}^l (t - t_{k+1-i}), \quad t_{k-l+1} < \xi < t_{k+1}, \quad (23)$$

类似算法 1 的推导, 可得仿真递推公式及其局部截断误差:

$$X(k+1) = e^{At}X(k) + \sum_{i=0}^l \Phi_i^*(T)U(k+1-i), \quad (24)$$

$$T^*(k+1) = \frac{1}{(l+1)!} e^{A(T-\xi)} BU^{(l+1)}(kT + \eta) \int_0^T \prod_{s=-1}^{l-1} (s + iT) ds, \quad 0 < \xi < T, \quad (1-l)T < \eta < T. \quad (25)$$

几个具体公式及其局部截断误差如下:

$$l = 0, \quad X(k+1) = e^{AT} X(k) + G_0(T) BU(k+1), \quad (26)$$

$$T_1^*(k+1) = -\frac{T^2}{2} e^{A(T-\xi)} BU^{(1)}(kT + \eta); \quad (27)$$

$$l = 1, \quad X(k+1) = e^{AT} X(k) + G_1(T) BU(k+1) + [G_0(T) - G_1(T)] BU(k), \quad (28)$$

$$T_2^*(k+1) = -\frac{T^3}{12} e^{A(T-\xi)} BU^{(2)}(kT + \eta); \quad (29)$$

$$l = 2, \quad X(k+1) = e^{AT} X(k) + \frac{1}{2} [G_1(T) + G_2(T)] BU(k+1) + [G_0(T) - G_2(T)] BU(k) + \frac{1}{2} [-G_1(T) + G_2(T)] BU(k-1), \quad (30)$$

$$T_3^*(k+1) = -\frac{T^4}{24} e^{A(T-\xi)} BU^{(3)}(kT + \eta); \quad (31)$$

$$l = 3, \quad X(k+1) = e^{AT} X(k) + \frac{1}{6} [2G_1(T) + 3G_2(T) + G_3(T)] BU(k+1) + [G_0(T) + \frac{1}{2}G_1(T) - G_2(T) - \frac{1}{2}G_3(T)] BU(k) + \frac{1}{2} [-2G_1(T) + G_2(T) + G_3(T)] BU(k-1) + \frac{1}{6} [G_1(T) - G_3(T)] BU(k-2), \quad (32)$$

$$T_4^*(k+1) = -\frac{19}{720} T^5 e^{A(T-\xi)} BU^{(4)}(kT + \eta). \quad (33)$$

4 实时公式

如果在算法1的推导中,将最后一个基点 $t_l = (k+1)T$ 去掉;在算法2的推导中,将插值基点向左平移一步,分别可得两种算法的实时递推公式及其局部截断误差:

$$X(k+1) = e^{AT} X(k) + \sum_{i=0}^{l-1} \hat{\Phi}_i(T) U(k + \frac{i}{l}), \quad (34)$$

$$\hat{T}(k+1) = \frac{1}{l!} e^{AT} \int_0^T \prod_{s=0}^{l-1} (s - \frac{i}{l}T) e^{-As} BU^{(l)}(kT + \hat{\eta}) ds, \quad 0 < \hat{\eta} < T, \quad (35)$$

$$X(k+1) = e^{AT} X(k) + \sum_{i=0}^l \hat{\Phi}_i^*(T) U(k-i), \quad (36)$$

$$\hat{T}^*(k+1) = \frac{1}{(l+1)!} e^{A(T-\xi)} BU^{(l+1)}(kT + \hat{\eta}) \int_0^T \prod_{i=0}^l (s + iT) ds, \quad (37)$$

$$0 < \xi < T, \quad -lT < \hat{\eta} < T.$$

限于篇幅,仅分别给出两个具体递推公式及其局部截断误差。
一阶

$$X(k+1) = e^{AT}X(k) + G_0(T)BU(k), \quad (38)$$

$$\hat{T}_1(k+1) = \frac{T^2}{2} e^{A(T-\xi)} BU^{(1)}(kT + \hat{\eta}), \quad (39)$$

这也是算法2的一阶实时递推公式及其误差估计。
二阶

$$X(k+1) = e^{AT}X(k) + [G_0(T) - 2G_1(T)]BU(k) + 2G_1(T)BU(k + \frac{1}{2}), \quad (40)$$

$$\hat{T}_2(k+1) = \frac{T^3}{24} e^{A(T-\xi)} BU^{(2)}(kT + \hat{\eta}), \quad (41)$$

$$X(k+1) = e^{AT}X(k) + [G_0(T) + G_1(T)]BU(k) - G_1(T)BU(k-1), \quad (42)$$

$$\hat{T}_2^*(k+1) = \frac{5}{12} T^3 e^{A(T-\xi)} BU^{(2)}(kT + \hat{\eta}). \quad (43)$$

5 讨 论

下面从稳定性、计算量和精度等三方面对两种新算法进行讨论。

5.1 稳定性

由(2)式可知,右端第一项的数值计算与算法的稳定性和计算精度有关,而第二项则仅与计算精度有关,因此从理论上讲两种新算法都是 A -稳定的. 但当 A 比较复杂时,很难用解析法求解 e^{AT} , 必须采用数值计算. 在目前已有的20多种计算方法中,缩方与乘方法^[1]是较为理想的方法,我们在仿真中选用了这种方法,缩方系数由文献[6]中的结果确定. 根据对几十例 Stiff 比在 $10^3 \sim 10^8$ 之间的系统进行仿真的实践,尚未出现计算不稳定的现象. 不过为了保证足够的仿真精度,我们建议步长 T 的选择应当满足 $\max |R_i| T \leq 1000$ (R_i 为 A 的特征值的实部), 即步长 T 不要超过系统最小时间常数的1000倍.

5.2 计算量

利用(8)~(11)式可将上述诸递推公式中的 e^{AT} , $\Phi_i(T)$, $\Phi_i^*(T)$ 等在仿真递推前预先算好,而每步递推计算量可以大大减小. 以(20)式和(32)式为例,它们每步计算量分别为三次 $U(t)$ 计算及四次矩阵与向量相乘和一次 $U(t)$ 计算及五次矩阵与向量相乘. 事实上,两种新算法的每步计算量分别同阶 MTM 法和拟 Adams 法递推公式的相等. 关于它们与其它方法的计算量的比较见文献[3].

5.3 精 度

以(31)式对(3)式的仿真为例加以说明. 此时

$$T_4(k+1) = -\frac{19}{720} T^5 e^{A(T-\xi)} u^{(4)}(kT + \eta), \quad (44)$$

显然,当 $u(t)$ 为任意三次以下多项式(包括三次)时, $T_4(k+1) \equiv 0$. 此外,将(44)式与(4)式对比后可知,前者仅仅相当于后者中的最后一项,并且恒有

$$|T_4(k+1)| < \frac{19}{720} T^5 |u^{(4)}(kT + \eta)|$$

成立,即(31)式恒为四阶精度公式.

6 仿真实例

图1所示为一中等 Stiff 系统 (Stiff 比为 10^4). 在 Gould Concept 32/67 超级小型计算机上用双精度字长对该系统的仿真结果分别见于表1~表3. 其中 $y(t_n)$ 是解析解,算法1用(20)式,算法2用(32)式.

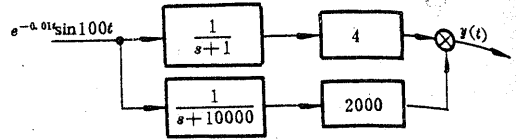


图1 被仿真系统

表1 ($T=0.0001, \max |R_i| T=1$)

t_n	$y(t_n)$	MTM-4	QAM-4	算法1	算法2
2.0	0.35517630	0.35511954	0.35303869	0.35517630	0.35517630
4.0	-0.06862800	-0.06868412	-0.07072588	-0.06862801	-0.06862801
6.0	0.05786331	0.05786424	0.05791346	0.05786331	0.05786331
8.0	0.18408589	0.18414070	0.18614941	0.18408589	0.18408589
10.0	0.12873307	0.12878452	0.13065572	0.12873308	0.12873308

表2 ($T=0.0005, \max |R_i| T=5$)

t_n	$y(t_n)$	MTM-4	QAM-4	算法1	算法2
2.0	0.35517630	0.33484797	0.19936583	0.35517546	0.35517548
4.0	-0.06862800	-0.08851414	-0.21897938	-0.06862865	-0.06862873
6.0	0.05786331	0.05840168	0.06396656	0.05786350	0.05786340
8.0	0.18408589	0.20370646	0.33437100	0.18408670	0.18408668
10.0	0.12873307	0.14695506	0.26640365	0.12873366	0.12873373

表3 ($T=0.005, \max |R_i| T=50$)

t_n	$y(t_n)$	MTM-4	QAM-4	算法1	算法2
2.0	0.35517630	-0.89935972	-2.68270081	0.35546621	0.35542194
4.0	-0.06862800	-1.26873270	-2.97191879	-0.06846003	-0.06845490
6.0	0.05786331	0.11701066	0.20375080	0.05774530	0.05779884
8.0	0.18408589	1.39362436	3.11287231	0.18381181	0.18385884
10.0	0.12873307	1.22711265	2.78579554	0.12858469	0.12857827

从表中可以看出,两种新算法的仿真精度均比 MTM-4 和 QAM-4 高得多,而且对步长的改变不敏感.这表明它们确实具有更好的数值特性.

7 结 语

理论推导和仿真结果均表明,本文提出的两种新算法均具有数值稳定性好、计算量小

和精度高等优点,非常适合于对系统(1)进行快速和实时仿真.系统的 Stiff 比越大,它们的优越性就表明得越显著,算法 1,2 之间的差别在于,前者为单步法,可以自启动;后者为多步法,不能自启动(一阶公式除外).关于后者的自启动问题,参见文献[3].一般而言,除了一阶公式外,算法 1 的精度要比同阶算法 2 公式的精度高,递推计算量也比后者的略大.

这种利用系统解析解的性质实现算法的 A-稳定性,利用多项式数值逼近提高算法的精度以及通过预计算减小每步仿真递推计算量的思想也可以推广应用到其它系统的仿真的推导中,事实上,我们已经将该思想运用到形如

$$\dot{X}(t) = AX(t) + F[t, X(t)]$$

的一类非线性动态系统的数字仿真中^[7],得到了比较理想的仿真算法.

参 考 文 献

- [1] 熊光楞,肖田元,张燕云.连续系统仿真与离散事件系统仿真.北京:清华大学出版,1991,30-41
- [2] 蒋琨,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
- [3] 蒋琨,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
- [4] 陈宝根,蒋琨,张惠刚,曹大铸.一个以晶片计算机为宿主的实时仿真语言.东南大学学报,1991,21(6):56-62
- [5] 南京大学数学系计算数学专业.数值逼近方法.北京:科学出版社,1978,1-39
- [6] 乔际岳.收缩乘方法的误差分析及收缩常数的决定.应用数学学报,1987,10(4):491-497
- [7] 蒋琨,柴干.一类非线性动态系统的数字仿真.自动化学报,1993,19:

Two New Kinds of Algorithms for the Simulation of Linear Time-Invariant Systems

JIANG Min and CHAI Gan

(Research Institute of Automation, Southeast University • Nanjing, 210018, PRC)

Abstract: Two new kinds of algorithms which are suitable for the simulation of linear time-invariant systems are presented. The algorithms not only maintain the good qualities of Modified Transition Matrix Method and Quasi-Adams Method such as A-stability, small computing effort and higher accuracy, but also make up their shortcomings of not strictness on theory and larger truncational error. Thus they have much better numerical properties. Simulation results show that the presented algorithms are feasible and effective.

Key words: simulation; algorithm; linear time-invariant system

本文作者简介

蒋琨 1957年生.分别于1982年和1988年在南京大学和东南大学获理学学士和工学硕士学位.现任东南大学自动化研究所副教授.主要从事系统仿真、计算机控制与管理的研究.曾获90年度总参科技进步二等奖和92年度光华科技基金三等奖各一次.近年来发表论文20余篇.

柴干 1963年生.分别于1985年和1991年在东北工学院和东南大学获工学学士和硕士学位.现在浙江大学攻读博士学位.主要从事化工过程计算机仿真及仿真器的研究.近年来发表五篇论文.