第 30 卷 第 9 期 控 制 与 决 策 2015 年 9 月
Vol. 30 No. 9 Control and Decision Sep. 2015

文章编号: 1001-0920 (2015) 09-1685-06 DOI: 10.13195/j.kzyjc.2014.1040

离散时间Itô型跳变系统Lyapunov方程的有限次迭代求解算法

付艳明[1], 崔振茂[1], 刘永信[2]

(1. 哈尔滨工业大学 控制理论与制导技术研究中心，哈尔滨 150001；
2. 内蒙古大学 电子信息工程学院，呼和浩特 010021)

摘 要: 针对离散时间Itô型马尔科夫跳变系统Lyapunov方程的求解给出一种迭代算法. 经证明, 在误差允许的范围内, 该算法可以在确定的有限次数内收敛到系统的精确解, 收敛速度较快, 具有良好的数值稳定性, 并且该算法为显式迭代, 可避免迭代过程中求解其他矩阵方程对结果精度产生的影响. 最后通过一个数值算例对该算法的有效性进行了验证.

关键词: 马尔科夫跳变系统；Itô微分；迭代算法；收敛性

中图分类号: TP273 **文献标志码:** A

Finite iterative algorithm for solving Lyapunov equations of Itô stochastic systems with Markovian jumps

FU Yan-ming[1], *CUI Zhen-mao*[1], *LIU Yong-xin*[2]

(1. Center for Control Theory and Guidance Technology，Harbin Institute of Technology，Harbin 150001，China; 2. College of Electronic Information Engineering，Inner Mongolia University，Hohhot 010021，China．Correspondent: FU Yan-ming，E-mail：fuyanming@hit.edu.cn)

Abstract: An iterative algorithm is given to find an exact solution to the coupled Lyapunov matrix equations of the discrete-time Itô stochastic liner systems with Markovian jumps. It has been proved that the algorithm can obtain the solution within finite steps in absence of round-off errors, and has fast convergence speed and good numerical stability. The algorithm is explicit iteration, which avoids the influence of the errors generated during the process of solving the other matrix equations. Finally, a numerical example is given to illurstrate the effectiveness of the proposed algorithm

Keywords: Markov jump systems；Itô differential；iterative algorithm；convergence

0 引 言

离散时间马尔科夫跳变系统可以被视为系统模态是由一马尔科夫链表示的切换系统. 近些年来, 由于研究已经深入到一些时变的非线性系统领域, 确定性系统模型已无法满足研究的需要. 早在1961年, Krasovskii等[1]就提出了马尔科夫跳变系统模型. 随后, 人们对马尔科夫跳变系统的一些特性做了深入的研究, 并在此基础上针对一些实际问题, 如网络控制[2], 给出了更合理的控制方案.

在对马尔科夫跳变系统的研究中, 稳定性分析一直是研究的重点. 文献[3-4]指出, 离散时间马尔科夫跳变系统的均方稳定的充要条件是其相应的Lyapunov方程存在正定解. 因此, 求解Lyapunov方程便成为众多学者关注的焦点. 文献[5]在假定零初始条件和子系统稳定的前提下, 提出了一种并行算法来求解耦合的Lyapunov方程. Wang等[6]对以上算法进行了分析后指出, 文献[5]中的迭代算法在以上两个假设不成立时, 仍然是收敛的. 文献[7]提出了一种有限次数迭代算法, 可以在有限次数内得到Lyapunov方程的精确解, 但仅仅适用于不含噪声项的马尔科夫跳变系统.

有关Itô型马尔科夫跳变系统的研究在最近几年开始活跃起来. 文献[8]给出了Itô型系统二次稳定的Lyapunov方程的条件. 然而, 目前有关Itô型跳变系统

收稿日期: 2014-06-30；**修回日期:** 2015-01-19.
基金项目: 国家自然科学基金项目(61104059, 61362002).
作者简介: 付艳明(1978−), 男, 副教授, 博士, 从事鲁棒控制理论、航天器控制等研究；刘永信(1955−), 男, 教授, 博士生导师, 从事图像处理与模式识别等研究.

Lyapunov 方程的求解算法还很少. 文献 [9] 提出了一种迭代算法来求解该类Lyapunov方程, 但由于该迭代算法为隐迭代, 每一步迭代过程中都要派生出一个新的矩阵方程, 必须通过解该方程才能继续下一步迭代, 并且派生方程的求解精度极大地影响了最终结果的精度, 这使得该求解算法的使用范围受到极大的制约. 针对文献 [9] 的情况, 本文以Itô型马尔科夫跳变系统为研究对象, 提出一种有限迭代算法来求解其Lyapunov方程. 与文献 [9] 中算法明显不同的是, 该算法能在确定的有限次数内收敛到方程的精确解, 收敛速度较快. 并且该迭代算法为显式迭代, 迭代过程中无需求解其他的矩阵方程, 从而降低了算法的复杂度, 避免了求解其他矩阵方程对结果精度的影响.

1 问题描述

给定概率空间 $(\Omega,\mathcal{F},\mathcal{P})$. 其中: Ω 为样本空间, $\mathcal{F}$ 为代数事件, $\mathcal{P}$ 为定义在 $\mathcal{F}$ 上的测量概率. 考虑如下用Itô差分方程表示的具有马尔科夫跳变性质的离散时间随机系统模型:

$$x(k+1)=A(\theta_k)x(k)+C(\theta_k)x(k)w(k). \tag{1}$$

其中: $x(k)\in\boldsymbol{R}^n$ 为系统的状态序列; 随机参数 $\{\theta_k, k\geqslant 0\}$ 是有限齐次马尔科夫链, 取值在集合 $N=(1, 2,\cdots,s)$ 中; $A(\theta_k)$ 和 $C(\theta_k)$ 是相应维数的系统结构矩阵, 其实际值随参数 θ_k 发生跳变, 当 $\theta_k=i$ 时, $A(\theta_k)=A(i)$, $C(\theta_k)=C(i)$, 分别简记为 A_i、C_i; $w(k)$ 是定义在 $(\Omega,\mathcal{F},\mathcal{P})$ 上独立的广义平稳的二阶矩过程, 且满足 $E\{w(k)\}=0$ 和 $E\{w(i)w(j)\}=\delta_{ij}$. 设随机变量 $\{w(k),k=0,1,\cdots\}$ 与马尔科夫链 $\{\theta_k,k=1,2,\cdots\}$ 相互独立. 马尔科夫链 θ_k 的转移概率为

$$\mathcal{P}(\theta_{k+1}=j|\theta_k=i)=\pi_{ij}.$$

其中: $\pi_{ij}\geqslant 0,\forall i,j\in N$. 定义其转移概率矩阵为

$$\boldsymbol{\Pi}=[\pi_{ij}]_{i,j\in N}, \tag{2}$$

$$\sum_{j=1}^{s}\pi_{ij}=1,\forall i\in N.$$

定义 1 如果对于任意初始状态 $x_0\in\boldsymbol{R}^n$, 有以下条件成立:

$$\lim_{k\to\infty}E\{\|x(k)\|^2\}=0,$$

则离散时间Itô随机系统 (1) 是渐近均方稳定的.

对于随机系统 (1) 的渐近均方稳定性, 有如下结论成立.

引理 1 离散时间Itô随机线性系统 (1) 是渐近均方稳定的, 当且仅当存在矩阵 $V_i>0,i\in N$ 使得如下的Lyapunov矩阵方程成立:

$$A_i^{\mathrm{T}}\Big(\sum_{j=1}^{s}\pi_{ij}V_j\Big)A_i+C_i^{\mathrm{T}}\Big(\sum_{j=1}^{s}\pi_{ij}V_j\Big)C_i-V_i+Q_i=0. \tag{3}$$

其中: $Q_i>0,i\in N$.

在随机系统 (1) 的相关研究中, 求取Lyapunov矩阵方程 (3) 的数值解至关重要. 该类问题的一种有限次迭代算法将在以下主要结果中给出.

2 主要结果

2.1 有限次迭代算法及其收敛性

对于矩阵方程 (3) 的求解问题, 本文提出如下迭代算法, 并证明该算法能在一确定次数内收敛到方程的精确解.

算法 1 有限次迭代算法.

算法步骤如下.

Step 1: 给定初始值 $V_i(0),i\in N$, 令 $k=0$, 计算

$$E_i(0)=Q_i+A_i^{\mathrm{T}}\Big(\sum_{j=1}^{s}\pi_{ij}V_j(0)\Big)A_i+C_i^{\mathrm{T}}\Big(\sum_{j=1}^{s}\pi_{ij}V_j(0)\Big)C_i-V_i(0), \tag{4}$$

$$P_i(0)=E_i(0)-\sum_{j=1}^{s}\pi_{ji}A_jE_j(0)A_j^{\mathrm{T}}-\sum_{j=1}^{s}\pi_{ji}C_jE_j(0)C_j^{\mathrm{T}}. \tag{5}$$

Step 2: 如果 $E_i(k)=0$, 则停止计算, $V_i(k)$ 即为矩阵方程 (3) 的精确解; 如果 $E_i(k)\neq 0$, 则 $k=k+1$.

Step 3: 计算

$$V_i(k+1)=V_i(k)+L_1(k+1)P_i(k), \tag{6}$$

$$E_i(k+1)=Q_i+A_i^{\mathrm{T}}\Big(\sum_{j=1}^{s}\pi_{ij}V_j(k+1)\Big)A_i+C_i^{\mathrm{T}}\Big(\sum_{j=1}^{s}\pi_{ij}V_j(k+1)\Big)C_i-V_i(k+1), \tag{7}$$

$$P_i(k+1)=E_i(k+1)+L_2(k+1)P_i(k)+N(k+1). \tag{8}$$

其中

$$L_1(k+1)=\sum_{n=1}^{s}\|E_n(k)\|^2\Big/\sum_{n=1}^{s}\|P_n(k)\|^2, \tag{9}$$

$$L_2(k+1)=\sum_{n=1}^{s}\|E_n(k+1)\|^2\Big/\sum_{n=1}^{s}\|E_n(k)\|^2, \tag{10}$$

$$N(k+1)=-\sum_{j=1}^{s}(\pi_{ji}A_jE_j(k+1)A_j^{\mathrm{T}}+\pi_{ji}C_jE_j(k+1)C_j^{\mathrm{T}}). \tag{11}$$

Step 4: 转 Step 2.

通过以上迭代算法, 便能在有限的迭代次数之后

得到相关方程的精确解. 在证明收敛性之前先给出如下定理.

定理 1 　若序列 $\{V_i(k), i=1,2,\cdots,s\}$, $\{E_i(k), i=1,2,\cdots,s\}$ 和 $\{P_i(k), i=1,2,\cdots,s\}$ 是由算法 1 在一任意初始条件 $\{V_i(0), i=1,2,\cdots,s\}$ 下迭代所得, 且存在整数 $\gamma\geqslant 1$, 使得对于任意的 $k=1,2,\cdots,\gamma$, $E_i(k)\neq 0$, 则有

$$\sum_{i=1}^{s}\mathrm{tr}[E_i(k)^{\mathrm{T}}E_i(l)]=0,$$
$$\sum_{i=1}^{s}\mathrm{tr}[P_i(k)^{\mathrm{T}}P_i(l)]=0.$$

其中: $k,l=1,2,\cdots,\gamma,\ k\neq l$.

定理 1 的证明将在迭代算法的构造过程中给出. 现在给出算法 1 的收敛性证明.

定理 2 　给定 Lyapunov 方程(3), 对于任意初始条件, 算法 1 经有限次迭代后, 能得到方程的精确解.

证明 　首先定义如下内积:

$$\mathrm{tr}\sum_{i=1}^{s}(A_i^{\mathrm{T}}B_i).$$

其中: $(A_1,A_2,\cdots,A_s)\in \boldsymbol{R}^{n\times n}\times\boldsymbol{R}^{n\times n}\times\cdots\times\boldsymbol{R}^{n\times n}$, $(B_1,B_2,\cdots,B_s)\in \boldsymbol{R}^{n\times n}\times\boldsymbol{R}^{n\times n}\times\cdots\times\boldsymbol{R}^{n\times n}$.

当 $k\leqslant sn^2-1$ 时, 若 $E_i(k)=0, i=0,1,\cdots,s$, 则 $V_i(k)$ 即为 Lyapunov 矩阵方程 (3) 的精确解; 若对于任意 $k\leqslant sn^2-1$, $E_i(k)\,(i=0,1,\cdots,s)$ 不全为 0, 则由定理 1 可知, 对于 $k,l=1,2,\cdots,sn^2-1, k\neq l$, $\sum_{i=1}^{s}\mathrm{tr}[E_i(k)^{\mathrm{T}}E_i(l)]=0$. 因此, $(E_1(k),E_2(k),\cdots,E_s(k))\,(k=0,1,\cdots,sn^2-1)$ 是 sn^2 维空间 $\boldsymbol{R}^{n\times n}\times\boldsymbol{R}^{n\times n}\times\cdots\times\boldsymbol{R}^{n\times n}$ 的一组正交基. 又由定理 1 可知,

$$\sum_{i=1}^{s}\mathrm{tr}[E_i(sn^2)^{\mathrm{T}}E_i(l)]=0,\ l=0,1,\cdots,sn^2-1,$$

所以

$$(E_1(sn^2),\cdots,E_s(sn^2))=(0_{n\times n},\cdots,0_{n\times n}).$$

显然, $(V_1(sn^2),V_2(sn^2),\cdots,V_s(sn^2))$ 是矩阵方程 (3) 的一组精确解. □

2.2 　定理 1 的证明及算法构造原理

现在给出定理 1 的证明. 与此同时, 算法 1 的构造过程将会在定理 1 的证明过程中给出.

在定理 1 的证明过程中, 如下公式将被多次用到:

$$\mathrm{tr}[A^{\mathrm{T}}B]=\mathrm{tr}[B^{\mathrm{T}}A]=\mathrm{tr}[BA^{\mathrm{T}}].$$

2.2.1 　定理 1 的证明

Step 1 　在这一步, 将给出如下结论: 对于任意的整数 $k\geqslant 0$, 有

$$\sum_{i=1}^{s}\mathrm{tr}[E_i^{\mathrm{T}}(k)E_i(k+1)]=0,\tag{12}$$

$$\sum_{i=1}^{s}\mathrm{tr}[P_i^{\mathrm{T}}(k)P_i(k+1)]=0.\tag{13}$$

在此, 使用数学归纳法进行证明. 首先证明当 $k=0$ 时, 式 (12) 和 (13) 成立. 将式 (6) 代入 (7) 可知, $E_i(k+1)$ 可以表示为

$$\begin{aligned}&E_i(k+1)=\\&E_i(k)+L_1\Big[A_i^{\mathrm{T}}\Big(\sum_{j=1}^{s}\pi_{ij}P_j(k)\Big)A_i+\\&C_i^{\mathrm{T}}\Big(\sum_{j=1}^{s}\pi_{ij}P_j(k)\Big)C_i-P_i(k)\Big].\end{aligned}\tag{14}$$

由式 (14) 可得

$$\begin{aligned}&\mathrm{tr}(E_i^{\mathrm{T}}(0)E_i(1))=\\&\|E_i(0)\|^2+L_1\mathrm{tr}\Big[\sum_{j=1}^{s}\pi_{ij}P_j(0)A_iE_i^{\mathrm{T}}(0)A_i^{\mathrm{T}}+\\&\sum_{j=1}^{s}\pi_{ij}P_j(0)C_iE_i^{\mathrm{T}}(0)C_i^{\mathrm{T}}\Big]-L_1\mathrm{tr}E_i^{\mathrm{T}}(0)P_i(0),\end{aligned}$$

所以

$$\begin{aligned}&\sum_{i=1}^{s}\mathrm{tr}(E_i^{\mathrm{T}}(0)E_i(1))=\\&\sum_{i=1}^{s}\|E_i(0)\|^2+L_1\mathrm{tr}\Big[\sum_{j=1}^{s}P_j(0)\Big(\sum_{i=1}^{s}\pi_{ij}A_iE_i^{\mathrm{T}}(0)A_i^{\mathrm{T}}+\\&\sum_{i=1}^{s}\pi_{ij}C_iE_i^{\mathrm{T}}(0)C_i^{\mathrm{T}}-E_i^{\mathrm{T}}(0)\Big)\Big].\end{aligned}$$

令

$$\begin{aligned}P_i(0)=E_i(0)-\sum_{j=1}^{s}\pi_{ji}A_jE_j(0)A_j^{\mathrm{T}}-\\\sum_{j=1}^{s}\pi_{ji}C_jE_j(0)C_j^{\mathrm{T}},\end{aligned}\tag{15}$$

且

$$L_1(k+1)=\sum_{n=1}^{s}\|E_n(k)\|^2\Big/\sum_{n=1}^{s}\|P_n(k)\|^2,\tag{16}$$

可得

$$\begin{aligned}&\sum_{i=1}^{s}\mathrm{tr}(E_i^{\mathrm{T}}(0)E_i(1))=\\&\sum_{i=1}^{s}\|E_i(0)\|^2-\frac{\sum_{n=1}^{s}\|E_n(0)\|^2}{\sum_{n=1}^{s}\|P_n(0)\|^2}\sum_{i=1}^{s}\|P_i(0)\|^2=0.\end{aligned}$$

由此可知, 当 $k=0$ 时式 (12) 成立. 且只有当 $P_i(0)$ 与 $L_1(k+1)$ 满足式 (15) 和 (16), 即算法 1 中式 (5) 和 (9) 时, 才有以上结论成立. 这便是算法 1 中式 (5) 和 (9) 的由来. 进一步, 由式 (8) 可知

$$P_i(1)=E_i(1)+L_2(1)P_i(0)+N(1),$$

因此

$$\sum_{i=1}^{s}\operatorname{tr}(P_i^{\mathrm{T}}(0)P_i(1))=$$
$$\sum_{i=1}^{s}\operatorname{tr}[P_i^{\mathrm{T}}(0)(E_i(1)+L_2P_i(0)+N(1))]=$$
$$\sum_{i=1}^{s}L_2\|P_i(0)\|^2+\operatorname{tr}\Big(\sum_{i=1}^{s}P_i^{\mathrm{T}}(0)E_i(1)\Big)+$$
$$\operatorname{tr}\Big(\sum_{i=1}^{s}P_i^{\mathrm{T}}(0)N(1)\Big).\tag{17}$$

令

$$N(k+1)=-\sum_{j=1}^{s}(\pi_{ji}A_jE_j(k+1)A_j^{\mathrm{T}}+$$
$$\pi_{ji}C_jE_j(k+1)C_j^{\mathrm{T}}),\tag{18}$$

则

$$\operatorname{tr}\sum_{i=1}^{s}P_i^{\mathrm{T}}(0)E_i(1)+\operatorname{tr}\sum_{i=1}^{s}P_i^{\mathrm{T}}(0)N(1)=$$
$$\operatorname{tr}\sum_{j=1}^{s}E_j(1)\Big[P_j^{\mathrm{T}}(0)-\sum_{i=1}^{s}\pi_{ji}(A_j^{\mathrm{T}}P_i^{\mathrm{T}}(0)A_j+$$
$$C_j^{\mathrm{T}}P_i^{\mathrm{T}}(0)C_j)\Big]=$$
$$\frac{1}{L_1}\Big[\operatorname{tr}\sum_{j=1}^{s}E_j(1)E_j(0)-\operatorname{tr}\sum_{j=1}^{s}\|E_j(1)\|^2\Big]=$$
$$-\frac{1}{L_1}\operatorname{tr}\sum_{j=1}^{s}\|E_j(1)\|^2.\tag{19}$$

由式(16)可得

$$\sum_{i=1}^{s}\operatorname{tr}[P_i^{\mathrm{T}}(0)P_i(1)]=0,$$

所以, 当$k=0$时式(13)成立. 同时, 只有当$N(k+1)$满足式(18), 即算法1中式(11)时, 才有以上结论成立. 这便是算法1中式(11)的由来.

假设当$k=t-1$时, 式(9)和(10)成立. 下面证明当$k=t$时, 式(12)和(13)成立. 求证过程如下:

$$\sum_{i=1}^{s}\operatorname{tr}[E_i^{\mathrm{T}}(t)E_i(t+1)]=$$
$$\sum_{i=1}^{s}\|E_i(t)\|^2+L_1\operatorname{tr}\sum_{j=1}^{s}P_j(t)\Big[\sum_{i=1}^{s}\pi_{ij}A_iE_i^{\mathrm{T}}(t)A_i^{\mathrm{T}}+$$
$$\sum_{i=1}^{s}\pi_{ij}C_iE_i^{\mathrm{T}}(t)C_i^{\mathrm{T}}-E_j^{\mathrm{T}}(t)\Big].$$

由假设和算法1可知

$$\sum_{i=1}^{s}\operatorname{tr}[E_i^{\mathrm{T}}(t)E_i(t+1)]=$$
$$\sum_{i=1}^{s}\|E_i(t)\|^2-L_1\sum_{j=1}^{s}\|P_j(t)\|^2=0.$$

这表明当$k=t\geqslant 0$时, 式(12)成立. 由式(8)可得

$$\sum_{i=1}^{s}\operatorname{tr}[P_i^{\mathrm{T}}(t)P_i(t+1)]=$$
$$L_2\sum_{i=1}^{s}\|P_i(t)\|^2+\operatorname{tr}\sum_{i=1}^{s}P_i(t)^{\mathrm{T}}E_i(t+1)+$$
$$\operatorname{tr}\sum_{i=1}^{s}P_i^{\mathrm{T}}(t)N(t+1).\tag{20}$$

应用式(12)和(14), 可得

$$\operatorname{tr}\sum_{i=1}^{s}[P_i(t)^{\mathrm{T}}E_i(t+1)]+\operatorname{tr}\sum_{i=1}^{s}P_i^{\mathrm{T}}(t)N(t+1)=$$
$$\operatorname{tr}\sum_{j=1}^{s}E_i(t+1)\Big[P_j^{\mathrm{T}}(t)-\sum_{i=1}^{s}\pi_{ji}(A_j^{\mathrm{T}}P_i^{\mathrm{T}}(t)A_j+$$
$$C_j^{\mathrm{T}}P_i^{\mathrm{T}}(t)C_j)\Big]=$$
$$\frac{1}{L_1}\operatorname{tr}\sum_{j=1}^{s}E_i(t+1)[E_j(t)-E_j(t+1)]^{\mathrm{T}}=$$
$$-\frac{1}{L_1}\operatorname{tr}\sum_{j=1}^{s}\|E_j(t+1)\|^2.\tag{21}$$

令

$$L_2(k+1)=\frac{\sum_{n=1}^{s}\|E_n(k+1)\|^2}{\sum_{n=1}^{s}\|E_n(k)\|^2},\tag{22}$$

有

$$\sum_{i=1}^{s}\operatorname{tr}[P_i^{\mathrm{T}}(t)P_i(t+1)]=0.$$

综上所述, 对于任意$k=t\geqslant 0$, 有式(12)和(13)成立. 同时, 只有当$L_2(k+1)$满足式(22), 即算法1中式(10)时, 才有以上结论成立. 这便是算法1中式(10)的由来.

Step 2 在这一步, 将证明

$$\sum_{i=1}^{s}\operatorname{tr}[E_i(k)^{\mathrm{T}}E_i(l)]=0,$$
$$\sum_{i=1}^{s}\operatorname{tr}[P_i(k)^{\mathrm{T}}P_i(l)]=0.\tag{23}$$

式(23)可等价表示为如下形式:

$$\sum_{i=1}^{s}\operatorname{tr}[E_i(k)^{\mathrm{T}}E_i(k+b)]=0,$$
$$\sum_{i=1}^{s}\operatorname{tr}[P_i(k)^{\mathrm{T}}P_i(k+b)]=0,\tag{24}$$

其中$b=l-k\geqslant 1$. 同样, 对其应用数学归纳法进行证明. $b=1$的情形已在Step 1中证明完毕.

假设当$b=m$时, 对于任意$k\geqslant 0$, 式(24)成立. 下面证明当$b=m+1$时, 对于任意$k\geqslant 0$, 式(24)成立.

当$k=0$时, 有

$$\sum_{i=1}^{s}\operatorname{tr}[E_i^{\mathrm{T}}(0)E_i(m+1)]=$$

$$L_1\sum_{j=1}^{s}\mathrm{tr}P_j(m)\Big[\sum_{i=1}^{s}\pi_{ij}A_iE_i^{\mathrm{T}}(0)A_i^{\mathrm{T}}+\sum_{i=1}^{s}\pi_{ij}C_iE_i^{\mathrm{T}}(0)C_i^{\mathrm{T}}-E_j^{\mathrm{T}}(0)\Big].$$

令

$$P_i(0)=E_i(0)-\sum_{j=1}^{s}\pi_{ji}A_jE_j(0)A_j^{\mathrm{T}}-\sum_{j=1}^{s}\pi_{ji}C_jE_j(0)C_j^{\mathrm{T}},\tag{25}$$

可得

$$\sum_{i=1}^{s}\mathrm{tr}(E_i^{\mathrm{T}}(0)E_i(m+1))=L_1\sum_{j=1}^{s}\mathrm{tr}P_j(m)P_j(0)^{\mathrm{T}}=0,\tag{26}$$

且

$$\sum_{i=1}^{s}\mathrm{tr}(P_i^{\mathrm{T}}(0)P_i(m+1))=\sum_{j=1}^{s}\mathrm{tr}E_j(m+1)\Big[P_j^{\mathrm{T}}(0)-\sum_{i=1}^{s}\pi_{ji}A_j^{\mathrm{T}}P_i^{\mathrm{T}}(0)A_j-\sum_{i=1}^{s}\pi_{ji}C_j^{\mathrm{T}}P_i^{\mathrm{T}}(0)C_j\Big]=\frac{1}{L_1}\sum_{j=1}^{s}\mathrm{tr}E_j(m+1)[E_j(0)E_j(1)]^{\mathrm{T}}=0.\tag{27}$$

当 $k\geqslant 1$ 时, 由假设和算法 1 可知

$$\sum_{i=1}^{s}\mathrm{tr}(P_i^{\mathrm{T}}(k)P_i(k+m+1))=\sum_{j=1}^{s}\mathrm{tr}E_j(k+m+1)\Big[P_j^{\mathrm{T}}(k)-\sum_{i=1}^{s}(\pi_{ji}A_j^{\mathrm{T}}P_i^{\mathrm{T}}(k)A_j+\pi_{ji}C_j^{\mathrm{T}}P_i^{\mathrm{T}}(k)C_j)\Big]=\frac{1}{L_1}\sum_{j=1}^{s}\mathrm{tr}E_j(k+m+1)[E_j(k)-E_j(k+1)]^{\mathrm{T}}=\frac{1}{L_1}\sum_{j=1}^{s}\mathrm{tr}E_j(k+m+1)E_i^{\mathrm{T}}(k),\tag{28}$$

且

$$\mathrm{tr}\sum_{i=1}^{s}(E_i(k+m+1)E_i^{\mathrm{T}}(k))=L_1\mathrm{tr}\sum_{j=1}^{s}P_j(k+m)\Big(\sum_{i=1}^{s}(\pi_{ij}A_iE_i^{\mathrm{T}}(k)A_i^{\mathrm{T}}+\pi_{ij}C_iE_i^{\mathrm{T}}(k)C_i^{\mathrm{T}})-E_j^{\mathrm{T}}(k)\Big)=L_1\mathrm{tr}\sum_{j=1}^{s}P_j(k+m)(P_j(k)-L_2P_j(k-1))^{\mathrm{T}}=-L_1L_2\mathrm{tr}\sum_{j=1}^{s}P_j(k+m)P_j^{\mathrm{T}}(k-1),$$

所以

$$\mathrm{tr}\sum_{i=1}^{s}(E_i(k+m+1)E_i^{\mathrm{T}}(k))=-L_1L_2\mathrm{tr}\sum_{j=1}^{s}P_j(k+m)P_j^{\mathrm{T}}(k-1).\tag{29}$$

重复式 (28) 和 (29), 可以得到确定的 α 和 β 使得

$$\mathrm{tr}\sum_{i=1}^{s}[P_i(k)^{\mathrm{T}}P_i(k+m+1)]=\alpha\mathrm{tr}\sum_{i=1}^{s}[P_i(0)^{\mathrm{T}}P_i(m+1)],$$
$$\mathrm{tr}\sum_{i=1}^{s}[P_i(k)^{\mathrm{T}}P_i(k+m+1)]=\beta\mathrm{tr}\sum_{i=1}^{s}[E_i(0)^{\mathrm{T}}E_i(m+1)].\tag{30}$$

由式 (26) 和 (27) 可知, 当 $b=m+1$ 时式 (24) 成立. 综上所述, 定理 1 是成立的. □

2.2.2　算法 1 的构造原理

由式 (25) 可知, 只有当 $P_i(0)$ 满足式 (25), 即算法 1 中式 (5) 时, 才有结论 (30) 成立. 这样 $E_i(0)$、$P_i(0)$、$L_1(k+1)$、$L_2(k+1)$、$N(k+1)$ 的构造过程已在 2.2.1 的论证过程中给出, 而式 (6) 实质上是解的修正式, (7) 是方程误差计算式, (8) 利用方程误差计算修正项, 从而得到了算法 1.

3　数值算例

本节通过算例来验证本文算法的有效性. 采用文献 [9] 中所用的系统参数. 其中

模态 1:

$$A_1=\begin{bmatrix}-0.2854&-0.0409&0.3464&0.0032\\-0.4621&0.3521&0.0917&-0.0220\\-0.2135&-0.3569&0.4254&-0.4799\\-0.0975&0.4726&-0.4324&0.0353\end{bmatrix},$$

$$C_1=\begin{bmatrix}0.2663&0.2283&-0.1163&0.2558\\-0.1242&-0.2130&-0.0799&-0.0284\\-0.1655&0.0055&0.2577&0.1995\\-0.0447&-0.0318&0.1724&-0.2439\end{bmatrix};$$

模态 2:

$$A_2=\begin{bmatrix}-0.0240&-0.0354&0.2295&-0.0041\\0.1039&0.2905&0.1248&-0.3693\\-0.3534&0.2842&-0.4000&-0.2181\\0.0337&-0.0222&-0.2950&-0.1377\end{bmatrix},$$

$$C_2 = \begin{bmatrix} -0.4347 & 0.4695 & 0.0237 & 0.3432 \\ -0.1249 & -0.1597 & -0.3366 & 0.3062 \\ -0.1265 & -0.2473 & -0.0136 & 0.3587 \\ -0.0160 & 0.0849 & -0.0039 & 0.1098 \end{bmatrix};$$

传输概率矩阵为

$$\boldsymbol{\Pi} = \begin{bmatrix} 0.3 & 0.7 \\ 0.8 & 0.2 \end{bmatrix}.$$

令 $Q_1 = Q_2 = I_4, V_1(0) = V_2(0) = I_4$, 由算法 1 求解 Lyapunov 方程 (3), 经 25 步迭代, 得到 Lyapunov 方程的解为

$$V_1 = \begin{bmatrix} 1.7472 & -0.0291 & -0.4012 & 0.2622 \\ -0.0291 & 2.2291 & -0.7860 & 0.5811 \\ -0.4012 & -0.7860 & 2.2087 & -0.4741 \\ 0.2622 & 0.5811 & -0.4741 & 1.7891 \end{bmatrix},$$

$$V_2 = \begin{bmatrix} 1.6327 & -0.3632 & 0.2814 & -0.3211 \\ -0.3632 & 1.8400 & 0.0235 & -0.2635 \\ 0.2814 & 0.0235 & 1.7898 & -0.0129 \\ -0.3211 & -0.2635 & -0.0129 & 1.8874 \end{bmatrix}.$$

将以上解代入相应的 Lyapunov 方程后, 所得残差为

$$E_1 = 10^{-3} \times \begin{bmatrix} 0.0412 & 0.3024 & -0.3049 & 0.3646 \\ 0.3024 & -0.6532 & -0.0424 & -0.3853 \\ -0.3049 & -0.0424 & 0.4676 & -0.1565 \\ 0.3646 & -0.3853 & -0.1565 & -0.0406 \end{bmatrix},$$

$$E_2 = 10^{-3} \times \begin{bmatrix} 0.1407 & -0.0008 & 0.2158 & -0.0539 \\ -0.0008 & -0.4781 & -0.1295 & 0.0430 \\ 0.2158 & -0.1295 & -0.6227 & 0.0204 \\ -0.0539 & 0.0430 & 0.0204 & 0.0839 \end{bmatrix}.$$

通过以上算例可以看出, 算法 1 可以在确定的有限步 (即 sn^2) 内求得相应 Lyapunov 方程的解, 并且具有较高的精度.

4 结 论

对于 Itô 型马尔科夫跳变系统, 本文给出的迭代算法可以在一确定的次数内较快地收敛到方程的精确解. 该迭代算法为显式迭代, 迭代步骤中无需求解其他的矩阵方程, 从而降低了算法的复杂度, 提升了算法的精度.

参考文献(References)

[1] Krasovskii N M, Lidskii E A. Analytical design of controllers in systems with random attributes[J]. Automation and Remote Control, 1961, 22(1/2/3): 1021-1025, 1141-1146, 1289-1294.

[2] 祝超群, 郭戈. 事件驱动的网络化系统最优控制[J]. 控制与决策, 2014, 29(5): 802-808.
(Zhu C Q, Guo G. Optimal control for event-triggered networked control systems[J]. Control and Decision, 2014, 29(5): 802-808.)

[3] Costa O L V, Fragoso M D. Stability results for discrete-time linear systems with Markovian jumping parameters[J]. J of Mathematical Analysis and Applications, 1993, 179(1): 154-178.

[4] Kubrusly C S, Costa O L V. Mean-square stability conditions for discrete stochastic bilinear-systems[J]. IEEE Trans on Automatic Control, 1985, 30(11): 1082-1087.

[5] Borno I, Gajic Z. Parallel algorithm for solving coupled algebraic Lyapunov equations of discrete-time jump linear systems[J]. Computers and Mathematics with Applications, 1995, 30(7): 1-4.

[6] Wang Q, Lam J, Wei Y, et al. Iterative solutions of coupled discrete Markovian jump Lyapunov equations[J]. Computers and Mathematics with Applications, 2008, 55(4): 843-850.

[7] Tong L, Wu A G, Duan G R. Brief paper finite iterative algorithm for solving coupled Lyapunov equations appearing in discrete-time Markov jump linear systems[J]. IET Control Theory and Applications, 2010, 4(10): 2223-2231.

[8] Zhang W, Huang Y, Xie L. Infinite horizon stochastic H_2/H_∞ control for discrete-time systems with state and disturbance dependent noise[J]. Automatica, 2008, 44(9): 2306-2316.

[9] Li Z Y, Zhou B, Lam J, et al. Positive operator based iterative algorithms for sloving Lyapunov equations for itô stochastic systems with Markovian jumps[J]. Applied Mathematics and Computation, 2011, 217(1): 8179-8195.

(责任编辑：李君玲)