

切换系统Lyapunov指数的算法及应用*

李清都[†] 郭建丽

(重庆邮电大学工业物联网与网络化控制教育部重点实验室, 重庆 400065)

(2013年12月9日收到; 2014年1月9日收到修改稿)

Lyapunov指数是判定系统非线性行为的重要工具, 然而目前的大多算法并不适用于切换系统。在传统Jacobi法的基础上, 提出了一种新算法, 可以直接计算得到 n 维切换系统的 n 个Lyapunov指数。首先, 根据切换面处相邻轨线的动态变化规律, 从相空间几何推导出切换面处轨线变化的Jacobi矩阵; 然后, 对该矩阵进行 QR 分解, 从而利用 R 的对角线元素实现Lyapunov指数的切换补偿; 最后, 将新算法应用到平面双螺旋混沌系统、Glass网络和航天器供电系统三个实例中, 并将计算结果与Poincaré映射方法的计算结果进行比较, 对新算法的有效性进行验证。

关键词: 切换系统, Lyapunov指数, Jacobi矩阵, 切换面

PACS: 05.10.-a, 05.45.-a, 05.45.Pq

DOI: 10.7498/aps.63.100501

1 引言

在非线性动力系统的研究中, Lyapunov指数是判定系统动力学行为类型的重要指标。在研究系统稳定性和混沌时, 与动力系统的拓扑马蹄理论^[1-3]、计算同调论^[4]、面向集合方法^[5]、横截同宿^[6-8]等方法相比, Lyapunov指数计算相对简单高效, 其已成为不可或缺的研究手段。虽然关于光滑动力系统Lyapunov指数的算法已经成熟, 但是对于非光滑动力系统, 因为Jacobi矩阵可能不存在或不连续, 所以传统方法一般并不适用。切换系统作为目前研究较多的一种典型非光滑系统, 在实际中具有广泛应用, 很多复杂系统都可以转换成含有若干个子系统的切换系统来描述^[9-17]。因此, 如何准确地计算出切换系统的Lyapunov指数具有重要的学术意义和工程应用价值。

关于切换系统的Lyapunov指数计算, 目前常用的主要有基于Poincaré映射法和基于时间序列法的两种方法。前者通过选择Poincaré截面, 将系统转化为离散映射, 然后采用映射的Lyapunov指

数计算方法, 以时间代替迭代次数求平均值的方式进行计算。由于Poincaré映射的维数比原系统少一维, 所以 n 维系统只能得到 $n-1$ 个Lyapunov指数。该方法的难点在于Poincaré映射的Jacobi矩阵难以准确求解, 因此计算精度有限, 通常只能相对有效地得出最大Lyapunov指数。此外, 为确保Poincaré映射的连续性, Poincaré截面的选取也是个难题。而后者则通常采用著名的Wolf法计算时间序列的Lyapunov指数^[18]。其突出优点是仅需一个状态变量, 而且这个状态变量也可以是实验得到的数据, 因此后者被广泛应用。但是该方法有两个局限性: 一是只适用于至少有一个正的Lyapunov指数的混沌系统, 其他的系统不适用; 二是不可避免地引入了时间序列噪声, 造成计算精度不够准确。

近年来, 针对某些特定的切换系统, 一些学者提出了若干新的计算方法。Galvanetto^[19]运用离散映射算法计算低维机械爬行(stick-slip)系统的Lyapunov指数。Stefański等^[20-22]利用两个系统间的同步估计出不连续系统的最大Lyapunov指

* 国家自然科学基金(批准号: 61104150)、重庆市杰出青年科学基金(批准号: cstc2013jcyjjq40001)和重庆市教育委员会科学技术研究计划(批准号: KJ130517)资助的课题。

† 通讯作者。E-mail: liqd@cqupt.edu.cn

数. de Souza 和 Caldas^[23] 针对两个特定的有碰撞现象的力学系统, 通过在碰撞瞬间补充一组转换条件, 再利用光滑离散动力系统 Lyapunov 指数的算法计算该系统的 Lyapunov 指数. 虽然这些方法可以准确有效地得出 Lyapunov 指数, 但一般仅针对特定类型的系统, 因此通用性不足.

本文在传统光滑系统的 Jacobi 算法基础上, 通过补偿切换的影响, 提出一种较为准确和通用的新算法, 可以直接计算出 n 维切换系统的 n 个 Lyapunov 指数, 并通过两个典型实例验证了该算法的有效性.

2 切换系统 Lyapunov 指数的算法

为了克服已有算法的不足, 新算法需适用于更多的系统并更具通用性, 因此我们必须以一般性的理论为依据; 另一方面, 新算法还需保证计算的准确性, 因此我们必须以成熟的数值算法为基础. 下面我们在广泛应用的 Jacobi 法上进行改进.

2.1 算法理论

对于一般的 n 维连续时间系统

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \quad (\mathbf{x} \in \mathbf{R}^n),$$

其 Lyapunov 指数可以利用轨线的首次变分方程

$$\dot{\mathbf{u}} = D\mathbf{f}(\mathbf{x})\mathbf{u} \quad (\mathbf{u} \in \mathbf{R}^n)$$

获得, 根据 \mathbf{u} 长度的变化来获取系统在轨线 $\mathbf{x}(t)$ 附近 \mathbf{u} 方向上的拉伸比率或收缩比率. 若计算系统在 $\mathbf{x}(t)$ 附近 n 个方向 $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$ 的 Lyapunov 指数时, 可令 $\mathbf{Q} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]$, 因而变分方程可写为

$$\dot{\mathbf{Q}} = D\mathbf{f}(\mathbf{x})\mathbf{Q}.$$

设 $\mathbf{Q}_0 = \mathbf{Q}|_{t=0}$, \mathbf{Q}_0 为单位方向矩阵, 对 $t = T_{\max}$ 时的积分终值 \mathbf{Q}_{end} 矩阵进行 \mathbf{QR} 分解, 上三角矩阵 \mathbf{R} 的对角线值为 $|\lambda_1|, |\lambda_2|, \dots, |\lambda_n|$, 依次正好为 n 个方向的拉伸比率或收缩比率. 在计算 Lyapunov 指数时, 如果积分时间过长, 容易造成 \mathbf{Q}_{end} 条件数过大. 为降低积累误差, 通常每隔一段时间 T_s ($T_s = T_{\max}/N$), 令分解后的单位矩阵 \mathbf{Q} 为新的单位方向矩阵 \mathbf{Q}_0 进行重新计算, 然后通过

$$l_j = \lim_{N \rightarrow \infty} \frac{1}{NT_s} \sum_{i=0}^N \ln |\lambda_j^i|$$

可以得到系统的第 j 个 Lyapunov 指数值.

上述方法能够以较高的精度计算 $D\mathbf{f}(\mathbf{x})$ 连续系统, 因此得到广泛应用. 当用该方法计算切换系统时, 由于切换面处 $D\mathbf{f}(\mathbf{x})$ 发生跳变, 积分结果 \mathbf{Q}_{end} 并不能包容切换信息, 因此其不适用于切换系统.

基于以上依据, 下面我们通过对切换面处的补偿提出一种新的算法. 为了不失一般性, 我们考虑切换系统的方程为

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}_1(\mathbf{x}) & (s(\mathbf{x}) \geq 0), \\ \dot{\mathbf{x}} &= \mathbf{f}_2(\mathbf{x}) & (s(\mathbf{x}) < 0), \end{aligned} \quad (1)$$

式中, $\mathbf{f}_1(\mathbf{x})$ 和 $\mathbf{f}_2(\mathbf{x})$ 均连续且维数为 n ; $s(\mathbf{x}) = 0$ 代表 $n - 1$ 维的切换面, 记为 S .

令 $\dot{\mathbf{x}} = \mathbf{f}_1(\mathbf{x})$ 是子系统 1, $\dot{\mathbf{x}} = \mathbf{f}_2(\mathbf{x})$ 是子系统 2. 设其 Jacobi 矩阵分别为 $\mathbf{J}_{\mathbf{f}_1}$ 和 $\mathbf{J}_{\mathbf{f}_2}$, 相应的轨线和首次变分方程为

$$\dot{\mathbf{x}} = \mathbf{f}_1(\mathbf{x}), \quad \dot{\mathbf{Q}} = \mathbf{J}_{\mathbf{f}_1}\mathbf{Q}; \quad (2)$$

$$\dot{\mathbf{x}} = \mathbf{f}_2(\mathbf{x}), \quad \dot{\mathbf{Q}} = \mathbf{J}_{\mathbf{f}_2}\mathbf{Q}. \quad (3)$$

图 1 给出了本文算法的几何模型. 如图 1 所示, 研究相邻两轨线 $\mathbf{x}_1(t)$ 和 $\mathbf{x}_2(t)$ 从 $s(\mathbf{x}) > 0$ 的一边穿过切换面的过程. 由于 $\mathbf{f}_1(\mathbf{x})$ 和 $\mathbf{f}_2(\mathbf{x})$ 连续, 那么轨线 $\underline{x} \rightarrow \bar{x}$, $x \rightarrow A$ 和 $A \rightarrow x'$ 在极小邻域内都可以看作是直线 (这里 $\underline{x}, \bar{x}, x, A, x'$ 均是轨线上的点). 两条轨线 $\underline{x} \rightarrow \bar{x}$ 和 $x \rightarrow x'$ 的时间间隔均为 Δt . 设初始向量为 $\mathbf{v} = \overrightarrow{\underline{x}\bar{x}}$, 终止向量为 $\mathbf{v}' = \overrightarrow{\bar{x}x'}$. 为考察切换面造成轨线 $\mathbf{x}_1(t)$ 和 $\mathbf{x}_2(t)$ 之间的拉伸或收缩情况, 需要推导 \mathbf{v} 与 \mathbf{v}' 的函数关系. 显然, Jacobi 矩阵为

$$\mathbf{J}_s = \frac{\partial \mathbf{x}'}{\partial \mathbf{x}^T} = \frac{\partial \mathbf{v}'}{\partial \mathbf{v}^T},$$

即为补偿矩阵.

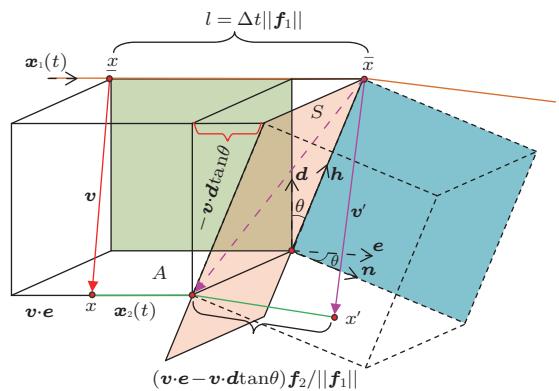


图 1 本文算法的几何模型

下面对图1中对应的各个向量给出解释. e 为轨线 $x_1(t)$ 的前进方向, $e = \mathbf{f}_1(\bar{x})/\|\mathbf{f}_1(\bar{x})\|$; l 为 \underline{x} 经过充分小时间 Δt 后到达与切换面交点 \bar{x} 时的轨线长度, $l = \Delta t \|\mathbf{f}_1(\underline{x})\|$; \mathbf{n} 是与切换面 S 垂直的单位向量, 若 $s(\mathbf{x})$ 可微, 则有

$$\mathbf{n} = \pm \frac{\partial s}{\partial \mathbf{x}} / \left\| \frac{\partial s}{\partial \mathbf{x}} \right\|_{\mathbf{x}=\bar{x}},$$

$$\frac{\partial s}{\partial \mathbf{x}} = \left[\frac{\partial s}{\partial x_1}, \frac{\partial s}{\partial x_2}, \dots, \frac{\partial s}{\partial x_n} \right]^T;$$

θ 为在切换面处矢量 \mathbf{n} 与 e 之间的夹角, $\theta = \arccos \mathbf{n} \cdot e$; \mathbf{h} 为切换面 $s(\mathbf{x}) = 0$ 上满足 $\mathbf{h} \cdot e > 0$ 的辅助向量, 显然 $\mathbf{h} \cdot \mathbf{n} = 0$; \mathbf{d} 为与轨道前进方向 e 垂直的单位向量, 即 $\mathbf{d} \cdot e = 0$, \mathbf{d} 与 \mathbf{h} 的夹角为 θ , 且不难导出

$$\mathbf{d} = \frac{(\mathbf{n} \cdot e)\mathbf{e} - \mathbf{n}}{\|(\mathbf{n} \cdot e)\mathbf{e} - \mathbf{n}\|}.$$

根据图1中的几何关系, 由 $\overrightarrow{\underline{x}\bar{x}} = l \cdot e$, $\overrightarrow{\underline{x}\underline{x}'} = \mathbf{v}$, $\overrightarrow{\bar{x}\bar{x}'} = -\mathbf{v}'$ 不难导出

$$\overrightarrow{\underline{x}A} = [l - (-\mathbf{v} \cdot \mathbf{d} \tan \theta) - \mathbf{v} \cdot e] \cdot e.$$

因此 \underline{x} 到 A 之间的轨线经历时间为

$$\Delta t_1 = \frac{\|\overrightarrow{\underline{x}A}\|}{\|\mathbf{f}_1\|} = \Delta t - \frac{(-\mathbf{v} \cdot \mathbf{d} \tan \theta) + \mathbf{v} \cdot e}{\|\mathbf{f}_1\|},$$

因为 $\underline{x} \rightarrow \underline{x}'$ 的时间间隔为 Δt , 所以

$$\begin{aligned} \overrightarrow{\underline{x}\bar{x}'} &= \mathbf{f}_2(\Delta t - \Delta t_1) \\ &= (\mathbf{v} \cdot e - \mathbf{v} \cdot \mathbf{d} \tan \theta) \mathbf{f}_2 / \|\mathbf{f}_2\|. \end{aligned}$$

由于轨线环绕 \underline{x} , A , \underline{x}' , \bar{x} 和 \underline{x} 五个点一周的矢量和为零, 即

$$\overrightarrow{\underline{x}A} + \overrightarrow{\underline{x}\bar{x}'} + \overrightarrow{\bar{x}\underline{x}} + \overrightarrow{\bar{x}\underline{x}'} + \overrightarrow{\underline{x}\underline{x}'} = 0,$$

将各向量值代入后, 可得

$$\begin{aligned} \mathbf{v}' &= \mathbf{v} - (\mathbf{v} \cdot e - \mathbf{v} \cdot \mathbf{d} \tan \theta) \cdot e \\ &\quad + (\mathbf{v} \cdot e - \mathbf{v} \cdot \mathbf{d} \tan \theta) \mathbf{f}_2 / \|\mathbf{f}_2\| \\ &= \mathbf{v} + (\mathbf{v} \cdot e - \mathbf{v} \cdot \mathbf{d} \tan \theta) (\mathbf{f}_2 / \|\mathbf{f}_2\| - e). \end{aligned}$$

因此, 切换面处的补偿 Jacobi 矩阵为

$$\mathbf{J}_s = \frac{\partial \mathbf{x}'}{\partial \mathbf{x}^T} = \frac{\partial \mathbf{v}'}{\partial \mathbf{v}^T}$$

$$= \mathbf{I} + (\mathbf{f}_2 / \|\mathbf{f}_2\| - e)(e - \tan \theta \mathbf{d})^T, \quad (4)$$

式中 \mathbf{I} 为单位矩阵.

2.2 Lyapunov 指数的算法思想与步骤

2.2.1 算法思想

在算法理论的基础上, 针对方程(1)中的切换系统, 下面详述利用切换面处的补偿 Jacobi 矩阵 \mathbf{J}_s 计算其 Lyapunov 指数的具体算法.

算法思想如下: 首先设置初始值 \mathbf{x}_0 和计算参数; 然后求解系统一段时间 (T_{dis}), 使得轨线充分接近系统的吸引子; 当切换事件未发生时, 利用传统连续系统基于 QR 分解的 Jacobi 矩阵方法计算; 否则, 计算切换补偿矩阵 \mathbf{J}_s , 并对乘积 $\mathbf{J}_s \mathbf{Q}$ 做 QR 分解, 然后将分解所得到的 \mathbf{R} 矩阵的对角线值补偿 Lyapunov 指数, 并将分解所得到的 \mathbf{Q} 矩阵作为新单位方向矩阵继续求解.

2.2.2 算法实现的具体步骤

针对方程(1)的切换系统, 下面详述该算法在本系统中的具体实现步骤, 如图2所示. 算法具体步骤如下:

1) 设置最长迭代时间 T_{max} 、预求解时间 T_{dis} 、重整化时间 T_s 等参数; 初始值 $\mathbf{x} = \mathbf{x}_0$ 、初始化 \mathbf{Q} 为 n 阶单位矩阵、 Σ 为 n 维零向量.

2) 先求解切换系统的方程(1)直到时间 $t = T_{\text{dis}}$, 使轨线终点 \mathbf{x}_{end} 充分接近吸引子, 并令 $\mathbf{x}_0 = \mathbf{x}_{\text{end}}$ 、累积时间 $T = 0$.

3) 执行主循环, 循环条件 $T < T_{\text{max}}$, 否则算法结束:

① 若 \mathbf{x}_0 位于子系统 1, 即 $s(\mathbf{x}_0) \geq 0$, 用上述基于 QR 分解的 Jacobi 矩阵方法, 以 T_s 为重整化步长, 计算子系统 1 的 Lyapunov 指数, 将 $[\ln |\lambda_1|, \ln |\lambda_2|, \dots, \ln |\lambda_n|]$ 累积计入 Σ 中; 求解过程中, 当遇到切换面, 即 $s(\mathbf{x}) = 0$ 且 $\dot{s}(\mathbf{x}) < 0$ 时, 计算从子系统 1 到子系统 2 的切换补偿矩阵 \mathbf{J}_{s12} , 并对 $\mathbf{J}_{s12} \mathbf{Q}$ 做 QR 分解, 将对角线值取自然对数后计入 Σ 中. 最后, 令 $\mathbf{x}_0 = \mathbf{x}_{\text{end}}$ 继续;

② 若 \mathbf{x}_0 位于子系统 2, 即 $s(\mathbf{x}_0) < 0$, 用同样的方法计算方程(3), 当 $s(\mathbf{x}) = 0$ 且 $\dot{s}(\mathbf{x}) > 0$ 时, 计算从子系统 2 到子系统 1 切换补偿矩阵 \mathbf{J}_{s21} 进行补偿.

4) 将最终的 Σ 除以最终轨线时间 T , 即得到系统的 Lyapunov 指数. 算法结束.

上述算法是针对含两个子系统的切换系统. 实际上, 该算法可以很容易地推广到多个子系统切换的情形. 从算法步骤和流程不难看出, 每增加一个子系统只需增加一个 \mathbf{x}_0 位于相应子系统的步骤即

可。针对这种多子系统情形，将在下面列举的 Glass 网络实例中给予详细说明。

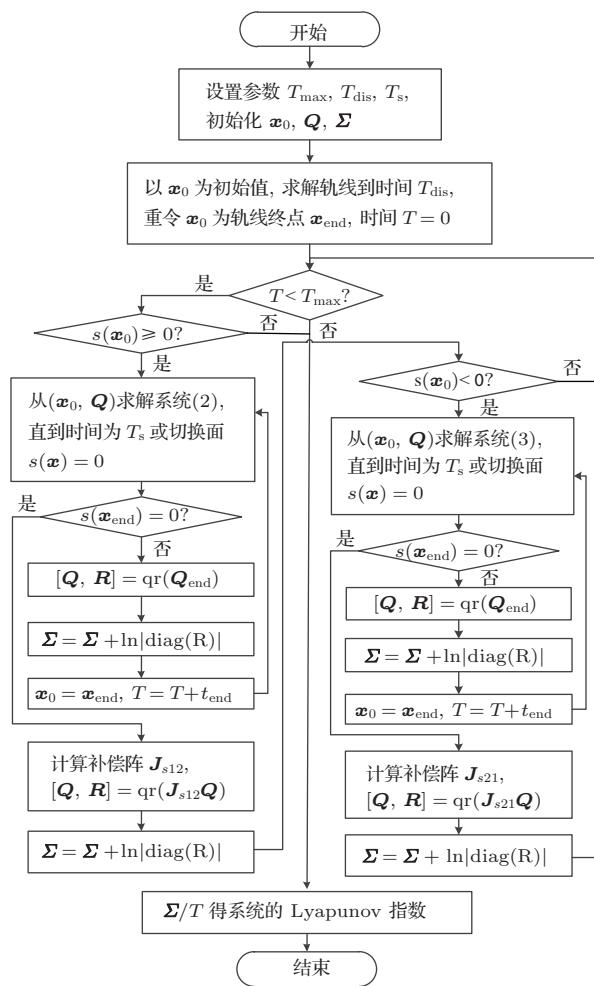


图 2 本文算法的具体流程图

与常用的 Poincaré 映射法相比, 本文算法无须将系统转为 Poincaré 映射, 就能算出 n 维切换系统的 n 个 Lyapunov 指数, 更加直接。因为不涉及 Poincaré 映射的 Jacobi 矩阵的数值求解, 因此更为准确。与基于时间序列的 Wolf 法相比, 本文算法对系统的行为类型没有要求, 由于直接从系统模型出发, 从而避免了时间序列噪声的引入, 因此更加通用、更为准确。

3 算法在切换系统中的应用

下面我们分别以平面双螺旋混沌系统、Glass 网络以及航天供电系统为例, 验证本文算法的有效性。

3.1 平面双螺旋混沌系统

平面双螺旋混沌系统是二维的切换系统, 含有两个子系统, 其方程^[24]如下: 系统 S_{A1} 的设计,

$$\dot{\mathbf{x}} = \mathbf{g}_1(\mathbf{x}) = \begin{pmatrix} 1 & 10 \\ -10 & 0 \end{pmatrix} \left(\mathbf{x} - \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right); \quad (5)$$

系统 S_{A2} 的设计,

$$\dot{\mathbf{x}} = \mathbf{g}_2(\mathbf{x}) = \begin{pmatrix} 1 & 10 \\ -10 & 0 \end{pmatrix} \left(\mathbf{x} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right). \quad (6)$$

其切换机制如下: 在系统 S_{A1} 下, 当 x_2 正向穿过 $x_2 = 1$ 切换面时启用动力系统 S_{A2} ; 在系统 S_{A2} 下, 当 x_2 负向穿过 $x_2 = -1$ 切换面时启用动力系统 S_{A1} 。

平面双螺旋混沌系统有一个双螺旋混沌吸引子, 计算机仿真结果如图 3 所示, 其中实线代表子系统 S_{A1} 的轨线, 虚线则代表子系统 S_{A2} 的轨线。因为子系统 S_{A1} 和 S_{A2} 都是线性系统, 且具有相同的系统 Jacobi 矩阵, 其特征值为 $0.5 \pm 0.5\sqrt{399}i$ 。如果用传统 Jacobi 算法, 则会得到两个约为 0.5 的 Lyapunov 指数, 实际算得的结果分别约为 0.5167 和 0.5172。因为系统存在一个吸引子, 至少有一个 Lyapunov 指数为 0, 所以上述结果显然是错误的。

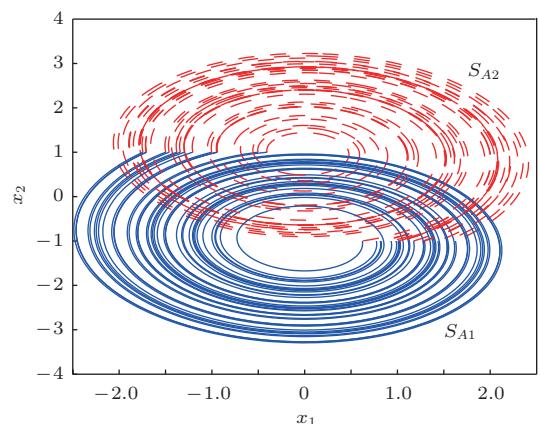


图 3 利用本文算法得到的平面双螺旋混沌系统吸引子的仿真结果

用本文算法计算 Lyapunov 指数, 首先设置参数 $T_{\max} = 1000$, $T_{\text{dis}} = 30$ 和 $T_s = 0.1$ 。当子系统 S_{A1} 切换到子系统 S_{A2} 时, 选择 $x_2 = 1$ 作为切换面, 即交点 \mathbf{x} 满足 $s(\mathbf{x}) = x_2 - 1 = 0$, $\dot{s}(\mathbf{x}) > 0$, 这时 $\mathbf{f}_1 = \mathbf{g}_1$, $\mathbf{f}_2 = \mathbf{g}_2$, 方向向量 $\mathbf{n} = \mathbf{n}_{12} = [0, 1]^T$ 。当子系统 S_{A2} 切换到子系统 S_{A1} 时, 选择 $x_2 = -1$ 作为切换面, 即交点 \mathbf{x} 满足 $s(\mathbf{x}) = x_2 + 1 = 0$,

$\dot{s}(\mathbf{x}) < 0$, 这时 $\mathbf{f}_1 = \mathbf{g}_2$, $\mathbf{f}_2 = \mathbf{g}_1$, 方向向量 $\mathbf{n} = \mathbf{n}_{21} = [0, -1]^T$.

由于 $\mathbf{e} = \mathbf{f}_1(\mathbf{x})/\|\mathbf{f}_1(\mathbf{x})\|$, 可求出

$$\theta = \arccos \mathbf{n} \cdot \mathbf{e},$$

$$\mathbf{d} = \frac{(\mathbf{n} \cdot \mathbf{e})\mathbf{e} - \mathbf{n}}{\|(\mathbf{n} \cdot \mathbf{e})\mathbf{e} - \mathbf{n}\|},$$

然后将上述各向量代入(4)式即可求得子系统 S_{A1} 切换到子系统 S_{A2} 的补偿 Jacobi 矩阵 \mathbf{J}_{s12} , 以及子系统 S_{A2} 切换到子系统 S_{A1} 的补偿 Jacobi 矩阵 \mathbf{J}_{s21} .

这样, 用本文算法进行计算得到的 Lyapunov 指数为 0.9997 和 0.0003. 显然第一个大于 0, 第二个在误差范围内等于 0, 由此可以判定该系统是混沌的. 为了验证计算结果的正确性, 我们利用 Poincaré 截面方法将系统的维数降低一维, 计算得到一个约为 0.9995 的 Lyapunov 指数. 显然, 这两种方法所得结果非常接近, 从而验证了本文方法的正确性.

3.2 Glass 网络

Glass 网络系统的动力学特性可以用以下常微分方程^[25]描述:

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}) = -\text{diag}(\mathbf{a})\mathbf{x} + \mathbf{B} \cdot \text{sgn}(\mathbf{x}), \quad (7)$$

式中, $\mathbf{a} = [a_1, a_2, \dots, a_n]^T$ 为所有元素为正的 n 维向量, $\mathbf{B} = (b_{ij})$ 为 $n \times n$ 的权值矩阵. Li 和 Yang^[26]研究发现, 当参数

$$\mathbf{a} = [1, 1, 2]^T,$$

$$\mathbf{B} = \begin{pmatrix} 0.3 & -0.8 & 0 \\ 0.8 & 0.3 & 0.4 \\ 0 & 0.23 & 0.2 \end{pmatrix}$$

时, 系统存在一个混沌吸引子(图 4), 并用拓扑马蹄理论给予严格证明.

$\text{sgn}(\mathbf{x})$ 只能取值 $\{-1, 0, 1\}$, 而实际上该系统由 8 个线性子系统构成. 因为系统矩阵都为 $-\text{diag}(\mathbf{a})$, 所以每个子系统均稳定. 因此, 传统 Jacobi 法得到的 Lyapunov 指数全为负, 实际计算结果分别为 $-2, -2$ 和 -4 , 这显然也是错误的.

下面我们采用本文算法进行计算, 初始化参数为 $T_{\max} = 2000$, $T_{\text{dis}} = 30$ 和 $T_s = 100$. 由于存在三个切换面 $x_1 = 0$, $x_2 = 0$ 和 $x_3 = 0$, 因此存在如下六个切换子事件: $x_1 = 0$, $\dot{x}_1 > 0$,

$\mathbf{n} = \mathbf{n}_{12} = [1, 0, 0]^T$; $x_1 = 0$, $\dot{x}_1 < 0$, $\mathbf{n} = \mathbf{n}_{21} = [-1, 0, 0]^T$; $x_2 = 0$, $\dot{x}_2 > 0$, $\mathbf{n} = \mathbf{n}_{34} = [0, 1, 0]^T$; $x_2 = 0$, $\dot{x}_2 < 0$, $\mathbf{n} = \mathbf{n}_{43} = [0, -1, 0]^T$; $x_3 = 0$, $\dot{x}_3 > 0$, $\mathbf{n} = \mathbf{n}_{56} = [0, 0, 1]^T$; $x_3 = 0$, $\dot{x}_3 < 0$, $\mathbf{n} = \mathbf{n}_{65} = [0, 0, -1]^T$. \mathbf{e} 作为轨道前进的方向可以由系统方程 (7) 直接给出, 即 $\mathbf{e} = \mathbf{g}(\mathbf{x})/\|\mathbf{g}(\mathbf{x})\|$. 最终, 得到的三个 Lyapunov 指数分别为 0.091, 0.000 和 -1.29 . 显然, 第一个大于 0, 这三个 Lyapunov 指数之和小于 0, 表明该系统存在一个混沌吸引子. 这与传统 Jacobi 算法的计算结果截然不同. 此外, 我们还利用 Poincaré 映射法对该系统进行了计算, 结果约为 0.0772 和 -0.3798 , 虽然第一个 Lyapunov 指数为正, 但发现求解 Jacobi 矩阵时存在较大误差, 结果不够准确.

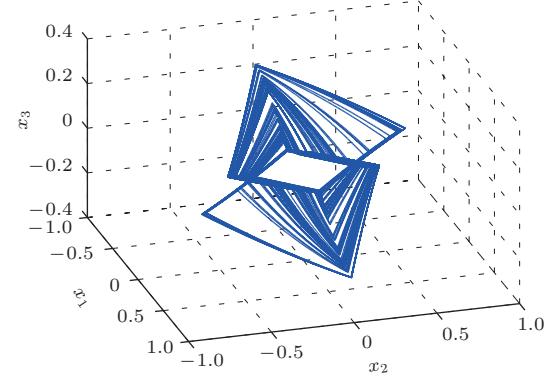


图 4 Glass 系统吸引子的仿真结果^[26]

3.3 航天器供电系统

航天器供电系统是一个复杂的非线性系统, 其动力学特性可以用下列方程描述:

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, w) = \begin{pmatrix} (i_1 - i_L)/C_1 \\ (v_1 - i_L r - v_2)/L \\ (i_L - i_2 w)/C_2 \end{pmatrix}, \quad (8)$$

式中符号的定义详见文献^[27], $\mathbf{x} = [v_1, i_L, v_2]^T$, 伏安特性为

$$\begin{aligned} i_1 &= \alpha I_{\text{SC}} \left(1 - \frac{\exp(v_1/V_{\text{th}}) - 1}{\exp(V_{\text{oc}}/V_{\text{th}}) - 1} \right), \\ i_2 &= P_C/v_2. \end{aligned} \quad (9)$$

$w(t)$ 代表 t 时刻的开关状态, $w(t) \in \{0, 1\}$ (0 表示断开, 1 表示闭合), 满足

$$w(t^+) = \begin{cases} 0 & (v_2 \leq v_{\text{lo}}), \\ 1 & (v_2 \geq v_{\text{hi}}), \\ w(t^-) & (v_{\text{lo}} < v_2 < v_{\text{hi}}). \end{cases} \quad (10)$$

当上述系统短路电流 $I_{SC} = 4$ A, 开路电压 $V_{oc} = 46.2$ V, 太阳能电池阵列端电压 $V_{th} = 1.1385$ V, 开关上下限电压 $v_{lo} = 19$ 和 $v_{hi} = 21$, 负载功率 $P_C = 50$ W, 照射系数 $\alpha = 0.576$, 以及电路参数 $C_1 = 50 \mu\text{F}$, $C_2 = 350 \mu\text{F}$, $L = 700 \mu\text{H}$ 和 $r = 0$ 时, Lim 和 Hamill^[27]通过计算最大Lyapunov指数发现该系统存在一个混沌奇怪吸引子, 如图 5 所示。

利用传统 Jacobi 算法计算其 Lyapunov 指数, 结果为 306.8343, 20.7590 和 19.1718。因为是混沌吸引子, 必须有一个 Lyapunov 指数为 0, 显然该结果不正确。考虑到传统 Lyapunov 指数方法计算切换系统存在误差, Li 等^[28]利用拓扑马蹄理论对系统 Poincaré 映射进行了严格研究, 证明该吸引子实际上为超混沌。因此, 正确的 Lyapunov 指数应该满足前两个为正且最后一个为零的条件。

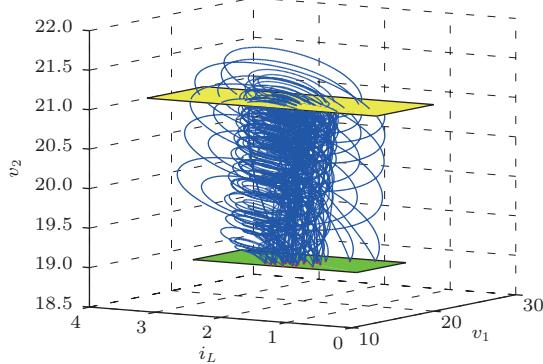


图 5 航天器供电系统吸引子的仿真结果^[27]

下面我们用本文算法进行计算。选取 $w(t) = 0$ 时为子系统 1,

$$\dot{x} = g_1(x) = g(x, 0); \quad (11)$$

选取 $w(t) = 1$ 时为子系统 2,

$$\dot{x} = g_2(x) = g(x, 1). \quad (12)$$

设置参数 $T_{max} = 100$, $T_{dis} = 1$ 和 $T_s = 0.005$ 。

当从子系统 1 切换到子系统 2 时, $s(x) = v_2 - v_{hi} = 0$ 为切换面且满足 $\dot{s}(x) > 0$, 这时 $f_1 = g_1$, $f_2 = g_2$, 方向向量 $n = n_{12} = [0, 0, 1]^T$.

当子系统 2 切换到子系统 1 时, $s(x) = v_2 - v_{lo} = 0$ 为切换面且满足 $\dot{s}(x) < 0$, 这时 $f_1 = g_2$, $f_2 = g_1$, 方向向量 $n = n_{21} = [0, 0, 1]^T$.

这样, 用本文算法计算得到的 Lyapunov 指数约为 145.59, 15.56 和 -0.228。由于是基于实际时间的平均值, 而本系统的时间常数较小, 约为

0.0005 s, 所以 Lyapunov 指数值较大。若乘以该时间常数, 可得到基于无量纲时间的 Lyapunov 指数, 约为 0.0728, 0.0078 和 -0.0001。显然前两个大于 0, 相比第三个在误差范围内为 0, 由此可以判定该吸引子确实超混沌。此外, 我们还利用 Poincaré 截面方法, 将系统的维数降低一维, 计算基于实际时间的 Lyapunov 指数, 结果约为 145.59 和 12.66。显然, 这两种方法所得结果非常接近, 这也验证了本文方法的正确性。

4 结 论

本文在传统 Jacobi 法基础上引入切换补偿矩阵, 提出直接计算 n 维切换系统的 n 个 Lyapunov 指数的算法, 并将该算法应用于平面双螺旋混沌系统、Glass 网络和航天器供电系统, 验证了该算法的有效性, 从而解决了传统 Jacobi 法计算切换系统不准确甚至错误的问题。与基于 Poincaré 映射计算方法相比, 本文算法可以直接计算出全部 Lyapunov 指数。因为不涉及 Poincaré 映射的 Jacobi 矩阵的数值求解, 因此本文算法更为准确。与基于时间序列的 Wolf 法相比, 本文算法对系统的行为不加限制, 因此更具通用性。

附录 本文计算平面双螺旋混沌系统 Lyapunov 指数的 Matlab 程序

```

function LE_ex1
Dbg=1; % 1 动画显示轨线和补偿值, 0 不显示
global A o1 o2 Dim
Dim=2; A=[110; -100];o1=[0; -1];o2=[0; 1];
ns12=[0;1]; ns21=[0; -1]; % 切换线(面)的法向量
Tmax=2000; Tdis=30; Ts=0.1; S=zeros(1,Dim);
opt = odeset('RelTol',1e-8,'AbsTol',1e-8);
ope1=odeset(opt,'Events',@events1);
ope2=odeset(opt,'Events',@events2);
x0=[0.1;zeros(Dim*Dim,1)]; x=x0(:,1); T=0;
while T < Tdis % 丢弃一段时间, 使轨线到达吸引子
    [t,x]=ode45(@fun1,[0,Tdis],x(end,:)',ope1);
    T=t(end)+T;
    [t,x]=ode45(@fun2,[0,Tdis],x(end,:)',ope2);
    T=t(end)+T;
end
if Dbg,clf;grid on;xlabel('x_1');ylabel('x_2');hold on;end
T=0;x0=x(end,1:Dim);Q=eye(Dim);tic;cnt=0; % 循环初始值设置
while(T<Tmax) %Main loop

```

```

ie=[]; % 求解子系统 1 的 Lyapunov 指数
while isempty(ie)
[t,x,~,~,ie]=ode45(@fun1,[0,Ts],[x0(:);Q(:)],ope1);
x0=x(end,1:Dim); T=T+t(end);
[Q,R]=qr(reshape(x(end,Dim+1:end),Dim,Dim));
S=S+log(abs(diag(R))');
if Dbg,plot(x(:,1),x(:,2),'b-');pause(0.01);end
end
%% 补偿子系统 1 到子系统 2 的切换
Js12=JsFun(@fun1,@fun2,x(end,:)',ns12);
[Q,R]=qr(Js12*Q); if Dbg,disp(abs(diag(R).')); end
S=S+log(abs(diag(R))');
ie=[]; % 求解子系统 2 的 Lyapunov 指数
while isempty(ie)
[t,x,~,~,ie]=ode45(@fun2,[0,Ts],[x0(:);Q(:)],ope2);
x0=x(end,1:Dim); T=T+t(end);
[Q,R]=qr(reshape(x(end,Dim+1:end),Dim,Dim));
S=S+log(abs(diag(R))');
if Dbg, plot(x(:,1),x(:,2),'r-');pause(0.01);end
end
%% 补偿子系统 2 到子系统 1 的切换
Js21=JsFun(@fun2,@fun1,x(end,:)',ns21);
[Q,R]=qr(Js21*Q);if Dbg,disp(abs(diag(R).'));end
S=S+log(abs(diag(R))');
cnt=cnt+1;if mod(cnt,10)==0,disp([S/T T/Tmax]);end
end
fprintf('LE=%s\n',mat2str(S/T));
%% 补偿矩阵的计算子函数
function Js=JsFun(hf1,hf2,x,n)
global Dim
f1=hf1(0,x);f1=f1(1:Dim);
f2=hf2(0,x);f2=f2(1:Dim);
e=f1/norm(f1);
theta=acos(n.*e);
d=(n.*e)*e-n;d=d/norm(d);
Js=eye(Dim)+(f2/norm(f1)-e)*(e-tan(theta)*d)';
%% 子系统 1 与切换事件的子函数
function OUT = fun1(t,X)
global A o1 Dim
x=X(1:Dim);Q=reshape(X(Dim+1:end),Dim,Dim);
DX=A*(x-o1); F=A*Q;
OUT=[DX(:); F(:)];
function [value,isterminal,direction] = events1(t,x)
value=x(2)-1; isterminal=1;direction=1;
%% 子系统 2 与切换事件的子函数
function OUT = fun2(t,X)
global A o2 Dim
x=X(1:Dim);Q=reshape(X(Dim+1:end),Dim,Dim);
DX=A*(x-o2); F=A*Q;
OUT=[DX(:); F(:)];
function [value,isterminal,direction] = events2(t,x)
value =x(2)+1; isterminal = 1;direction = -1;

```

参考文献

- [1] Yang X S 2009 *Int. J. Bifurcat. Chaos* **19** 1127
- [2] Li Q D, Yang X S 2010 *Int. J. Bifurcat. Chaos* **20** 467
- [3] Li Q D, Tang S 2013 *Acta Phys. Sin.* **62** 020510 (in Chinese) [李清都, 唐宋 2013 物理学报 **62** 020510]
- [4] Kaczyński T, Mischaikow K M, Mrozek M 2004 *Comput. Homol.* **15** 7
- [5] Neumann N, Sattel T, Wallaschek J 2007 *J. Vib. Control* **13** 1393
- [6] Yang F Y, Hu M, Yao S P 2013 *Acta Phys. Sin.* **62** 100501 (in Chinese) [杨芳艳, 胡明, 姚尚平 2013 物理学报 **62** 100501]
- [7] Li Q D, Tan Y L, Yang F Y 2011 *Acta Phys. Sin.* **60** 030206 (in Chinese) [李清都, 谭宇玲, 杨芳艳 2011 物理学报 **60** 030206]
- [8] Li Q D, Zhou H W, Yang X S 2012 *Acta Phys. Sin.* **61** 040503 (in Chinese) [李清都, 周红伟, 杨晓松 2012 物理学报 **61** 040503]
- [9] Zhang H G, Fu J, Ma T D, Tong S C 2009 *Chin. Phys. B* **18** 969

- [10] Wu L F, Guan Y, Liu Y 2013 *Acta Phys. Sin.* **62** 110510 (in Chinese) [吴立峰, 关永, 刘勇 2013 物理学报 **62** 110510]
- [11] Ji Y, Bi Q S 2010 *Acta Phys. Sin.* **59** 7612 (in Chinese) [季颖, 毕勤胜 2010 物理学报 **59** 7612]
- [12] Zhang X F, Chen X K, Bi Q S 2013 *Acta Phys. Sin.* **62** 010502 (in Chinese) [张晓芳, 陈小可, 毕勤胜 2013 物理学报 **62** 010502]
- [13] Gao C, Bi Q S, Zhang Z D 2013 *Acta Phys. Sin.* **62** 020504 (in Chinese) [高超, 毕勤胜, 张正娣 2013 物理学报 **62** 020504]
- [14] Lin C S, Xiong X, Shi L, Liu Y Z, Jiang C S 2007 *Acta Phys. Sin.* **56** 3107 (in Chinese) [林长圣, 熊星, 石磊, 刘扬正, 姜长生 2007 物理学报 **56** 3107]
- [15] Li S R, Jian J G, Geng Y F 2009 *J. Henan Normal Univ. (Nat. Sci. Ed.)* **5** 14 (in Chinese) [李圣荣, 蔡继贵, 耿艳峰 2009 河南师范大学学报 **5** 14]
- [16] Yu Y G, Li H X, Duan J 2009 *Chaos Solitons Fract.* **41** 457
- [17] Chen W H, Guan Z H, Lu X M 2008 *Asian J. Control* **7** 135
- [18] Wolf A, Swift J B, Swinney H L, Vastano J A 1985 *Physica D* **16** 285
- [19] Galvanetto U 2000 *Comput. Phys. Commun.* **131** 1
- [20] Stefański A, Kapitaniak T 2003 *Chaos Solitons Fract.* **15** 233
- [21] Stefański A 2000 *Chaos Solitons Fract.* **11** 2443
- [22] Stefański A, Kapitaniak T 2000 *Discrete Dyn. Nat. Soc.* **4** 207
- [23] de Souza S L T, Caldas I L 2004 *Chaos Solitons Fract.* **19** 569
- [24] Li Q D, Yang X S 2005 *Acta Electron. Sin.* **33** 1299 (in Chinese) [李清都, 杨晓松 2005 电子学报 **33** 1299]
- [25] Kappler K, Edwards R, Glass L 2003 *Signal Process.* **83** 789
- [26] Li Q D, Yang X S 2006 *Chaos* **16** 033101
- [27] Lim Y H, Hamill D C 1999 *Electron. Lett.* **35** 510
- [28] Li Q, Yang X S, Chen S 2011 *Int. J. Bifurcat. Chaos* **21** 1719

Algorithm for calculating the Lyapunov exponents of switching system and its application^{*}

Li Qing-Du[†] Guo Jian-Li

(Key Laboratory of Industrial Internet of Things and Networked Control of Ministry of Education, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

(Received 9 December 2013; revised manuscript received 9 January 2014)

Abstract

Lyapunov characteristic exponent is significant for analyzing nonlinear dynamics. However, most algorithms are not applicable for the switching system. According to the traditional Jacobi method, in this paper we propose a new algorithm which can be used to compute n Lyapunov exponents for an n -dimensional switching system. We first study the geometric dynamics of two adjacent trajectories near the switching manifold, and obtain a compensation Jacobi matrix caused by switching. Then with QR -decomposition of this matrix, we compensate for the diagonal vector of \mathbf{R} to realize the Lyapunov exponent expansion. Finally, we use the algorithm in a two-dimensional double-scrolls system, the Glass network and a spacecraft power system, and show its correctness and effectiveness by comparing the results with the Poincaré-map method.

Keywords: switching systems, Lyapunov exponents, Jacobi matrix, switching manifold

PACS: 05.10.-a, 05.45.-a, 05.45.Pq

DOI: [10.7498/aps.63.100501](https://doi.org/10.7498/aps.63.100501)

* Project supported by the National Natural Science Foundation of China (Grant No. 61104150), the Science Fund for Distinguished Young Scholars of Chongqing, China (Grant No. cstc2013jcyjjq40001), and the Science and Technology Research Program of Education Committee of Chongqing, China (Grant No. KJ130517).

† Corresponding author. E-mail: liqd@cqupt.edu.cn