

粒子群算法在稳定时滞系统设计中的应用 *

毛鹏轩, 肖 扬

(北京交通大学 信息科学研究所, 北京 100044)

摘 要: 时滞系统稳定性的判断问题一直是控制理论领域的研究热点。针对大多数的理论定理过于复杂,在实际的工程应用中不易应用且不直观的问题,提出一种基于粒子群算法来判定时滞系统的稳定性的简便方法。通过 PSO 搜索拟特征多项式的特征根来判断系统的稳定性;给出了一个应用实例以及路由器设计中实际应用的例子并加以验证。仿真结果表明所提出的算法可以快速有效地判断时滞系统的稳定性。

关键词: 粒子群算法; 时滞系统; 稳定性分析

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2012)01-0214-03

doi:10.3969/j.issn.1001-3695.2012.01.060

Application of PSO algorithm for stable time-delay system design

MAO Peng-xuan, XIAO Yang

(Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China)

Abstract: The stability analysis of the time-delay system is a hot area in recent years. However many theoretical results concerning this problem are too complex to apply in the real application. To address this problem, this paper proposed a simply algorithm based on PSO for the time-delay system stability analysis. This paper also presented an example of designing the router by utilizing the proposed algorithm. The simulation results indicate that the algorithm is effective and simple for the time-delay system stability analysis.

Key words: PSO; time-delay system; stability analysis

0 引言

许多实际系统,如喷汽发动机控制系统、航天器控制系统、核电站控制系统、轧钢机控制系统、船舶稳定控制系统、高速列车控制系统、微波振荡器系统和网络协议的控制系统等,都可以表示成为时滞系统。因此,时滞系统的稳定性检验一直是工程技术界和控制理论界的热点之一^[1~6]。文献[1]提出了基于扫频条件的判断时滞系统稳定性的方法,并给出了系统稳定的充分条件。文献[2]提出了 Khapritonov 定理,但是其结论不适用于区间准多项式,并且其稳定条件也是充分条件^[3]。文献[3]采用频域奈奎斯特图检测的方法对时滞系统的稳定性进行判断,但该方法受频域取样网格密度的限制。文献[4~6]采用经典的 Nyquist 稳定原则对目前流行的网络协议线性时滞系统的稳定性进行分析给出了时滞系统稳定的充分条件。文献[7]结合 Lyapunov-Krasovskii 泛函,提出了 Lyapunov-Krasovskii 稳定性定理,也得到了时滞系统稳定的充分条件,但是其需要结合线性矩阵不等式(LMI)工具进行分析来选取 Lyapunov 函数,证明过程相对较为复杂。以上的理论方法在实际应用中并不方便直观,且多为充分条件,不能直接判断系统为不稳定。

时滞系统稳定的充要条件是其系统的特征多项式根均具有负实部。最基本的时滞系统稳定性检验方法是基于系统特征多项式根的分布来判别稳定性。由于时滞系统的系统特征多项式为拟多项式。难以直接求出此拟多项式的解析解,并根

据其实部的正负性判断系统稳定性。

近年来粒子群算法已经应用到非线性方程组或多项式根的求解,并表现出一定的优势。方程组求解和多项式求根问题可以转换成为粒子群算法的优化问题。由于粒子群在优化问题具有高度的适应性和鲁棒性,可避免其他优化算法的局部性,可以很快搜到方程组和多项式根的最优解。传统的拟多项式根求解算法的局限性和近年来粒子群算法的出现,启发笔者提出用粒子群算法来判断拟多项式的稳定性,进而解决时滞系统的稳定性检验的充分必要条件的问题。目前尚未见发表论文涉及使用粒子群算法搜索时滞系统拟多项式的根,进而解决时滞系统稳定性的检验问题。与文献[1~5]中的结果相比,更加直观、简单、有效、易于应用。

1 时滞系统及其稳定条件

时滞系统可由状态空间模型与递归模型两种形式定义。时滞系统的状态空间模型可以表示为

$$\begin{aligned} \dot{x}(t) &= \sum_{n=0}^N A_n x(t-\tau_n) + \sum_{m=0}^M B_m u(t-\tau_n) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1}$$

其中: A_n 、 B_m 、 C 、 D 分别是 $(N_A \times N_A)$ 、 $(N_A \times N_B)$ 、 $(N_A \times N_C)$ 和 $(N_A \times N_D)$ 矩阵; τ_n 是常数; $u(t)$ 和 $y(t)$ 分别是线性时滞系统的输入信号向量和输出信号向量。

线性时滞系统的递归模型为

$$y(t) = \sum_{m=0}^M \sum_{n=0}^N a_{mn} \frac{\partial^m u(t-\tau_n)}{\partial t^m} - \sum_{m=0}^M \sum_{\substack{n=0 \\ (m,n) \neq (0,0)}}^N b_{mn} \frac{\partial^m y(t-\tau_n)}{\partial t^m} \tag{2}$$

其中: τ_n 为常数; $u(t)$ 和 $y(t)$ 分别是时滞系统的输入信号和输出信号。

对式(2)Laplace 变换,就可得到时滞系统的传递函数为

$$H(s)=\frac{\sum_{n=0}^N\sum_{m=0}^Ma_{mn}s^me^{-\tau_ns}}{\sum_{n=0}^N\sum_{m=0}^Mb_{mn}s^me^{-\tau_ns}}$$

(3)

系统(1)和(2)的特征多项式均有

$$B(s)=\sum_{n=0}^N\sum_{m=0}^{M_A}b_{mn}s^me^{-\tau_ns}$$

(4)

可以看出,系统(1)的特征多项式同样具有式(4)的形式。而且系统(1)的特征多项式的变量 s 和 τ_n 的阶次应分别为 N 和 M_A 。

时滞系统的特征多项式(4)被称为拟多项式^[1~5]。易知时滞系统(1)(2)的渐进稳定性可归结为拟多项式(4)的稳定性。

定义 1^[1] 如果拟多项式(4)满足

$$B(s)=\sum_{n=0}^N\sum_{m=0}^Mb_{mn}s^me^{-\tau_ns}\neq 0,\text{Re}(s)\geq 0$$

(5)

则该拟多项式稳定。

定理 1^[1] 线性时滞系统(2)稳定的充分和必要条件是定义 1 的条件成立。

2 粒子群算法

粒子群算法 (particle swarm optimization, PSO) 是 1995 年 Kennedy 等人受到鸟群觅食的启发,提出的一种新的仿生优化方法。研究者发现鸟群在飞行中经常会突然改变方向,虽然个体行为不可预测,但其整体总是保持一致性,每个个体也同时保持着最适宜的距离。

与其他的优化算法相比,PSO 算法可以不受优化函数形式影响,能在非线性空间内搜索真正解,同时能处理参数空间变化的优化问题,基本不受解的维数的限制。与进化算法的遗传算法相比较,PSO 能更快、更准确地搜索,最终达到全局最优。

3 基于 PSO 的时滞系统稳定性判断算法

本文结合拟多项式的求根问题,可以将对系统稳定性的判断转换为函数的优化,并提出基于 PSO 的拟多项式稳定性检验算法。

根据式(5),设拟多项式 $B(s)$ 的根为 $s=x_1+ix_2,i=\sqrt{-1}$ 。其中: x_1,x_2 分别代表根的实部和虚部; s 为式(5)的任一根。可以看出求解式(5)等价于求解

$$V(s)=|B(s)|$$

(6)

在给定范围内的最小值。设算法的目标函数为

$$V(s_i)=|B(s_i)|\rightarrow 0$$

(7)

其中: $s_i=x_{1,i}+ix_{2,i},|B(s_i)|$ 为

$$|B(s_i)|=\left|\sum_{n=0}^N\sum_{m=0}^Mb_{mn}s_i^me^{-\tau_ns_i}\right|$$

(8)

采用目标式(7)搜到其约等于零时后,等价于求出拟多项式(4)的根;然后判断 $B(s_i)$ 是否有根在虚轴的右边。如果 $B(s_i)$ 有根在虚轴右边,则说明系统不稳定。

本文提出的算法考虑在二维目标空间中,有 m 个粒子组成一个群体,其中第 i 个粒子表示为一个二维向量 $x_i=(x_{1,i},x_{2,i}),i=1,2,\cdots,m,x_{1,i},x_{2,i}$ 分别代表根的实部和虚部,则第 i

个粒子在二维空间中的位置(即每个粒子的位置)就是式(7)的一个潜在解。

将 x_i 代入目标函数式(7),即可算出其适应值,根据适应值的大小衡量解的优劣。

粒子群的飞行速度是二维向量 V_i ,记为 $V_i=(V_{1,i},V_{2,i})$ 。本文提出的算法记下当前粒子群搜索到的最优位置为 $P_s=(P_{1,i},P_{2,i})$,记粒子群此前搜索到的最优位置为 $P_g=(P_{1,g},P_{2,g})$ 。

传统的粒子群算法采用下式对粒子操作:

$$v_{n,i}^{t+1}=v_{n,i}^t+c_1r_1(p_s-x_{n,i}^t)+c_2r_2(p_g-x_{n,i}^t)$$

(9)

$$x_{n,i}^{t+1}=x_{n,i}^t+v_{n,i}^t$$

(10)

其中: $i=1,2,\cdots,m;n=1,2$ 。学习因子 c_1 和 c_2 是非负常数; r_1 和 r_2 为伪随机数,服从 $[0,1]$ 上的均匀分布。 $v_{in}\in[-v_{max},v_{max}]$, v_{max} 为常数,由用户设定。

本文将采用已有的改进粒子群算法,其速度和位置更新为

$$v_{n,i}^{t+1}=\omega v_{n,i}^t+c_1r_1(p_s-x_{n,i}^t)+c_2r_2(p_g-x_{n,i}^t)$$

(11)

$$x_{n,i}^{t+1}=x_{n,i}^t+\alpha v_{n,i}^t$$

(12)

式(11)中: ω 为非负数,称为动力常量,控制着前一速度,影响当前速度。 ω 较大时,前一速度影响较大,全局搜索能力较强; ω 较小时,前一速度影响较小,局部搜索能力较强。通过调整 ω 的大小来避免局部极小值。 α 为约束因子,用来控制速度的权值。

迭代终止条件可以设置为目标函数 $V(s)$ 即适应值为零,或者设置最大迭代次数。

本算法的流程如下:

- a)初始化一个规模为 swarm_size 的粒子群,构造粒子结构为 swarm(index,[x_1,x_2]), x_1,x_2 分别代表实部和虚部。设定初始位置、速率和惯性因子 inertia。
- b)计算每一个粒子的适应值 $V(s)=|B(s)|$ 。
- c)每个粒子将其适应值和其经历过的最好位置进行比较。若其值更小,则将其作为当前的最好位置。
- d)每个粒子将其适应值和全局经历过的最好位置的适应值进行比较。若其值更小,则将其作为当前的全局最好位置。
- e)对粒子位置进行更新,对粒子速度进行更新。
- f)满足终止条件,迭代完成,输出解。
- g)根据所得解,得到系统根在 s 平面的位置。
- h)根据是否在右半平面有根,给出结论。

部分主要代码如下:

a)速度更新

$$V(j,:)=\omega\times V(j,:)+c_1\times\text{rand}\times(\text{gbest}(j,:)-\text{swarm}(j,:))+c_2\times\text{rand}\times(\text{zbest}-\text{swarm}(j,:));$$
$$V(j,\text{find}(V(j,:)>V_{\text{max}}))=V_{\text{max}};$$
$$V(j,\text{find}(V(j,:)<V_{\text{min}}))=V_{\text{min}};$$

b)种群更新

$$\text{swarm}(j,:)=\text{swarm}(j,:)+0.5\times V(j,:);$$
$$\text{swarm}(j,\text{find}(\text{swarm}(j,:)>\text{swarmmax}))=\text{swarmmax};$$
$$\text{swarm}(j,\text{find}(\text{swarm}(j,:)<\text{swarmmin}))=\text{swarmmin};$$

c)适应度值

$$\text{fitness}(j)=\text{fun}(\text{swarm}(j,:));$$

d)个体最优更新

$$\text{if fitness}(j)<\text{fitnessgbest}(j)$$

$$\text{gbest}(j,:)=\text{swarm}(j,:);$$
$$\text{fitnessgbest}(j)=\text{fitness}(j);$$

$$\text{end}$$

```
e) 群体最优更新
if fitness(j) < fitnesszbest
    zbest = swarm(j, :);
    fitnesszbest = fitness(j);
end
```

4 实验与仿真结果

在所提出的方法中, 设学习因子 $c_1 = c_2 = 1.49445$, ω 设置为 0.4, 群体规模数为 50, 最大进化代数 为 300, 用 MATLAB 编程。

例 1 判断特征多项式为 $T(s) = s + e^{-0.01s} - 4$ 系统的稳定性。

首先应用该 PSO 算法来求解此问题, 可以迅速搜索出系统的根。如图 1 所示, 在虚轴右半平面存在特征根, 故系统特征多项式 $T(s) = s + e^{-0.01s} - 4$ 不稳定。且可以从图 2 看出, 该算法在大约进化代数为 10 的时候, 就可以很快搜索出特征多项式的根的位置。这个判断结果与利用文献[1]中的定理判断的结果一致。

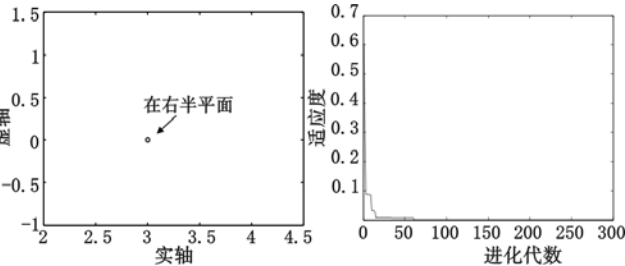


图1 系统 $T(s)=s+e^{-0.01s}-4$ 图2 适应值变化图

例 2 本文算法用来判断常用的路由器网络设计, 以判断 TCP/RED 路由器的稳定性为例。限于篇幅, 直接给出 TCP/RED 路由器在 s 域的系统特征多项式^[8]:

$$T(s) = s^3 + \left(\frac{2N}{R_0^2 C} + \frac{1}{R_0} + \frac{1}{K} \right) s^2 + \left(\frac{2N}{R_0^2 C} + \frac{2NK}{R_0^2 C} + \frac{K}{R_0} \right) s + \frac{2NK}{R_0^2 C} + \frac{KL_{red} C^2}{2N} e^{-sR_0} \quad (13)$$

其中: N 表示 TCP 流的数目; R_0 表示传输时延(s); K 表示滤波器参数; C 表示链路带宽大小(以包为单位); L_{red} 表示 RED 机制参数。这里, 首先设置 $C = 3\,750$ 包/s, $N = 60$, $R_0 = 0.246$ s, $K = 0.005$ 和 $L_{red} = 1.8 \times 10^{-4}$ 。

根据该算法运行 50 次的结果, 如图 3 所示, 可以看出该系统并没有在虚轴右半平面存在特征根, 只在左半平面搜到根, 故可以认为该系统是稳定的。

为了验证结果, 同时给出了该网络实验的实际仿真效果。配置参数和图 3 中的相同 NS2^[9] 仿真结果如图 4 所示。

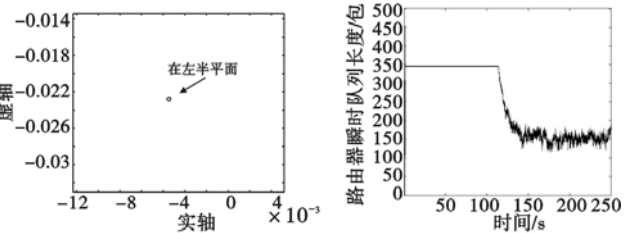


图3 TCP/RED路由器 ($L_{red}=1.8 \times 10^{-4}$) 图4 路由器队列 ($L_{red}=1.8 \times 10^{-4}$)

由图 4 可以看出, 在此参数配置下, 路由器的队列处于稳定状态, 并没有出现剧烈的震荡; 在大约 100 s 之后, 路由器瞬

时队列长度收敛于大约 150 包。该结果与提出算法的判断结果一致。

再选择另外一组参数配置 $N = 240$, $L_{red} = 2 \times 10^{-4}$, $K = 0.000\,028$ 来进行检验。同样用该算法来判断此时系统的稳定性。

如图 5 所示, 根据该算法搜索的结果可以看出, 该系统在虚轴右半平面存在特征根, 故该系统是不稳定的。而文献[8]中给出的稳定条件为充分条件, 无法直接判断系统为不稳定。同样, 给出了网络实验的实际仿真效果。配置参数与图 5 中的相同, NS2 仿真结果如图 6 所示。可以看出在此配置下, 该路由器的瞬时队列有剧烈的震荡现象, 表明此系统处于不稳定状态, 这与提出算法的结论一致。

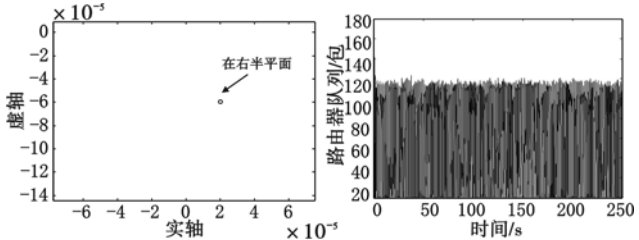


图5 TCP/RED路由器 ($L_{red}=2 \times 10^{-4}$) 图6 路由器队列 ($L_{red}=2 \times 10^{-4}$)

5 结束语

本文提出了一种基于 PSO 算法的判断时滞系统稳定性的方法, 利用 PSO 算法将稳定性的判断问题转换为求解非线性函数超越方程根的优化问题。与其他理论方法比较, 该方法简单、直观, 大量的实验结果表明了算法的有效性, 并且易于设计应用。下一步工作将研究粒子群算法本身的局限性对本方法可能带来的影响。

参考文献:

[1] CHEN Jie, LULIAN S V. Robust stability of quasi-polynomials: frequency-sweeping conditions [C]//Proc of the 42th IEEE Conference on Decision and Control. Berlin:Springer,2003:5048-5053.

[2] KHARITONOV L. Robust stability of time-delay systems [J]. IEEE Trans on Automatic Control,1994,39(12):2388-2397.

[3] FU Ming-yue, ANDRZEJ W. Robust stability for time-delay systems: the edge theorem and graphical tests [J]. IEEE Trans on Automatic Control,1989,34(8):813-820.

[4] LU Zong-tao, ZHANG Shi-jie. Stability analysis of XCP congestion control systems [C]//Proc of WCNC. 2009.

[5] DUKKIPATI N. Rate control protocol (RCP): congestion control to making flows complete quickly [D]. Stanford:Department of Electrical Engineering, Stanford University, 2008.

[6] WU Huang-xing, REN Feng-yuan, PAN Weng-ying. An efficient and fair explicit congestion control protocol for high bandwidth-delay product networks [J]. Computer Communications, 2009, 32(7): 1138-1147.

[7] GU K, KHARITONOV V L, CHEN Jie. Stability of time-delay systems [M]. Boston: Birkhauser, 2003.

[8] HOLLOT C V. Analysis and design of controllers for AQM routers supporting TCP flows [J]. IEEE Trans on automatic Control, 2002, 47(6): 945-959.

[9] <http://www.isi.edu/nsnam/ns>[EB/OL].