

基于粒子群优化的混沌系统比例-积分-微分控制*

王东风 韩 璞

(华北电力大学自动化系,保定 071003)

(2004 年 11 月 12 日收到;2005 年 9 月 21 日收到修改稿)

基于比例-积分-微分(PID)控制算法的简单性和实用性,但对于复杂非线性系统控制时参数的难以确定问题,运用集群智能中的改进粒子群算法进行 PID 控制器的优化,并应用于若干混沌系统的控制.对 Hénon 混沌、Duffing 混沌、六辊 UC 轧机混沌、Nagumo-sato 神经元混沌、Chen 氏混沌以及永磁同步电动机混沌的控制进行了仿真研究.研究表明:用 PID 进行混沌系统的输出反馈控制是有效的,从而拓宽了 PID 控制的应用范围;用简单方法控制复杂混沌系统是完全可以的,对混沌系统的控制具有较好的参考价值;粒子群优化算法对复杂混沌系统 PID 控制参数的优化不仅算法简单、易于编程,而且十分有效.

关键词:混沌,比例-积分-微分控制,粒子群优化算法

PACC: 0545

1. 引 言

近年来,随着计算机技术的发展,混沌和混沌控制已成为一个特殊的研究领域^[1-4],控制理论和控制工程学科的许多数学方法和智能控制方法被引入混沌系统的控制,国内该方面的相关专著也相继出版.已有的这些控制方法大多需要使用较深的数学工具,有的过于依赖混沌系统本身的数学模型,这使得控制的实用性受到限制.比例-积分-微分(PID)控制是一种广泛使用的控制方法,近年来被引入混沌系统的控制,如文献[5,6]分别用状态比例加积分(PI)调节器实现了连续混沌系统和离散混沌系统的控制.对于一个控制系统,只有在 PID 控制器各参数的恰当配合下才能取得良好的控制效果,对于线性系统有一些经验整定公式,对于非线性系统则没有现成的参数整定公式,因此,我们采用一种简单而又有效的仿生优化技术——粒子群优化(PSO)^[7]算法进行 PID 参数的寻优,以期得到满足非线性混沌系统控制的 PID 参数值.

2. PID 控制

大量实践表明, PID 控制器各项增益控制效果明

确,这种简明的控制分量独立整定效果对于工程应用操作是十分方便而且往往也是十分有效的.对于典型的反馈控制系统,连续形式的 PID 控制器表示为

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}, \quad (1)$$

式中,偏差 $e(t) = R(t) - y(t)$, $y(t)$ 为过程输出测量值, $R(t)$ 为被调量设定值, K_p 为比例系数, K_i 为积分系数, K_d 为微分系数.

用连续域(拉氏变换)传递函数形式描述为

$$G(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_i}{s} + K_d s. \quad (2)$$

在计算机控制系统中往往采用间隔采样控制,此时则采用离散形式的控制算式,离散 PID 控制算式为

$$u(k) = u(k-1) + \Delta u(k), \quad (3)$$

式中, $u(k)$ 为当前 k 时刻控制量, $u(k-1)$ 为前一时刻控制量, $\Delta u(k)$ 为当前时刻控制增量.易得其控制增量算式为

$$\begin{aligned} \Delta u(k) = & K_p [e(k) - e(k-1)] + K_i e(k) \\ & + K_d [e(k) - 2e(k-1) + e(k-2)], \end{aligned} \quad (4)$$

式中 $e(k)$ 为 k 时刻离散系统设定值 $R(k)$ 与测量值 $y(k)$ 之间的偏差,

* 华北电力大学重大项目基金(批准号 20041306)和华北电力大学博士基金(批准号 02060106)资助的课题.

$$e(k) = R(k) - y(k).$$

用离散域 (Z 变换) 传递函数形式描述为

$$G(z) = \frac{u(z)}{e(z)} = \frac{(K_p + K_i + K_d) - (K_p + 2K_d)z^{-1} + K_d z^{-2}}{1 - z^{-1}}. \quad (5)$$

对于连续混沌系统,如 Hénon 混沌、六辊 UC 轧机混沌以及 Nagumo-sato 混沌,将采用连续形式的 PID(即(1)-(2)式)进行控制;对于离散混沌系统,如 Duffing 混沌、Chen 氏混沌以及永磁同步电动机混沌,将采用离散形式的 PID(即(3)-(5)式)进行控制。

3. 改进的 PSO 算法与 PID 参数优化

3.1. PSO 算法

PSO 算法^[7]的提出受鸟群觅食行为的启发,并用于解决优化问题。PSO 算法采用速度-位置搜索模型。每个粒子代表解空间的一个候选解,解的优劣程度由适应度函数决定,其中适应度函数根据优化目标定义。速度 $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$, 决定粒子在搜索空间单位迭代次数的位移。PSO 算法随机初始化为一群粒子,其中第 i 个粒子在 D 维解空间的位置表示为 $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 。每一次迭代,粒子通过动态跟踪两个极值来更新其速度和位置。第一个是粒子从初始到当前迭代次数搜索产生的最优解:个体极值 $p_i = (p_{i1}, p_{i2}, \dots, p_{id})$ 。第二个是粒子种群目前的最优解:全局极值 $g = (g_1, g_2, \dots, g_D)$ 。第 i 个粒子根据以下公式来更新其第 d 维 ($1 \leq d \leq D$) 速度和位置:

$$v_{id} = v_{id} + c_1 r (p_{id} - x_{id}) + c_2 R (g_d - x_{id}), \quad (6)$$

$$x_{id} = x_{id} + v_{id}, \quad (7)$$

式中, r, R 是均匀分布在 $(0, 1)$ 区间的随机数。 c_1, c_2 是学习因子。粒子在解空间内不断跟踪个体极值与全局极值进行搜索,直至达到规定的迭代次数或满足规定的性能指标为止。粒子在每一维飞行的速度不能超过算法设定的最大速度 v_{\max} 。设置大的 v_{\max} 可以保证粒子种群的全局搜索能力, v_{\max} 较小则粒子种群的局部搜索能力加强。

PSO 算法是基于群体智能理论的优化算法,通过群体中粒子间的合作与竞争产生的群体智能指导

优化搜索。PSO 算法是一种相当简单的算法,只需很少的代码和参数,但在各种问题的求解与应用中却展现了它的特点和魅力。该算法为人们提供了如下一种思路^[7]:使智慧出现而不是努力强迫它;模拟自然而不是力图控制它;寻求使事情简单化而不是让它复杂。

与进化算法比较,PSO 算法是一种更高效的并行搜索算法,它保留了基于种群的全局搜索策略,但是其采用的速度-位移模型操作简单,避免了像遗传算法中复杂的遗传操作。它特有的记忆使其可以动态跟踪当前的搜索情况调整其搜索策略。PSO 算法基本都能较快地达到全局最优值,PSO 算法基本不受问题峰数增加的影响,受问题维数的影响也很小^[8]。

与演化规划相比,PSO 算法执行一种有“意识 (conscience)”的变异^[9]。理论上,演化规划有更多的机会在优化点附近开发,而 PSO 算法则有更多的机会更快地飞到有更好解的区域。

3.2. 改进的 PSO 算法

基于以上这些认识,为了进一步提高 PSO 算法的性能,Shi 和 Eberhart^[9]在简单 PSO 算法中引入惯性权重系数 ω ,从而使算法成为

$$v_{id} = \omega v_{id} + c_1 r (p_{id} - x_{id}) + c_2 R (g_d - x_{id}), \quad (8)$$

$$x_{id} = x_{id} + v_{id}. \quad (9)$$

Ozcan 和 Mohan^[10]分析了 PSO 算法在多维搜索空间的能力以及速度步长和 ω 系数对算法的影响,所得出的结论是 ω 并不能显著提高算法的性能。为了有效地控制微粒的飞行速度,使算法达到全局探测和局部开采两者间的有效平衡,Clerc^[11]构造了引入收缩因子的改进 PSO 算法模型,

$$v_{id} = K [v_{id} + c_1 r (p_{id} - x_{id}) + c_2 R (g_d - x_{id})], \quad (10)$$

$$x_{id} = x_{id} + v_{id}, \quad (11)$$

式中 K 称为收缩因子,

$$K = \frac{2}{|2 - C - \sqrt{C^2 - 4C}|},$$

$C = c_1 + c_2$ 且 $C > 4$ 。Carlisle 和 Dozier^[12]综合了 PSO 算法参数选择问题上的各种研究成果,为 PSO 算法提供了一组较完善、合理的参数,他们给出的参数最佳取值为 $c_1 = 2.8, c_2 = 1.3$,微粒群规模为 30。

通过对文献 [9] 的改进模型以及对文献 [12] 的

改进模型的分析,并通过对大量的高维、复杂非线性函数的优化测试,我们发现将以上两种改进模型相结合,则对优化问题的求解十分有效.因此,在本文中我们就是采用混合改进模型进行混沌 PID 控制器的参数优化,即采用如下模型:

$$v_{id} = K[\omega v_{id} + c_1 r(p_{id} - x_{id}) + c_2 R(g_d - x_{id})], \quad (12)$$

$$x_{id} = x_{id} + v_{id}, \quad (13)$$

且惯性权重系数 ω 随优化的进程从 1.1 逐渐调整到 0.4,即所谓的变权重系数.变权重系数的方法有多种,如根据目标函数或者按照进化代数调整.本文是按照进化代数调整,假定 n 是进化代数,则调整算法如下:

If $\omega > 0.4$

$$\omega = 1.1 - 0.7 * n/100$$

End

其他参数则如上所述,即 $c_1 = 2.8$, $c_2 = 1.3$,微粒群规模为 30.

3.3. PSO 算法对 PID 参数优化的步骤

本文优化问题的适应度函数 $F(p)$ 定义为控制系统常用的反映系统调节品质的 ITAE 指标,ITAE 表示绝对误差的一阶矩积分.对连续混沌系统采用适应度函数

$$F(p) = \int_0^{\infty} t |e(t)| dt, \quad (14)$$

对离散混沌系统采用适应度函数

$$F(p) = \sum_{j=1}^{\infty} j |e(j)|. \quad (15)$$

适应度函数 $F(p)$ 值越小,表明相应粒子 p 越靠近全局最优解.适应度函数是随着学习系统的运行而逐步减小的,采用的学习方式类似于控制理论中的迭代学习控制系统,蕴涵着反馈的调节原理.下面给出运用 PSO 算法优化混沌系统 PID 控制参数的算法流程.

第 1 步:初始化一群微粒,即 PID 的 3 个参数的若干随机位置和速度,以及群体规模.

第 2 步:对研究的混沌系统,基于计算机仿真计算每个微粒的适应度.

第 3 步:对每个微粒,将其适应值与其经历过的最好位置 p_i 作比较,如果较好,则将其作为当前的最好位置 p_i .

第 4 步:对每个微粒,将其适应值与全局所经历

的最好位置 g 作比较,如果较好,则重新设置 g 的索引号.

第 5 步:根据方程(12)(13)改变微粒速度与当前位置,即 PID 参数的学习优化.

第 6 步:如未达到结束条件(通常为足够好的适应值或达到一个最大演化代数),则返回第 2 步.

4. 仿真实例

4.1. Hénon 混沌系统

受控的 Hénon 混沌系统

$$x(n+1) = 1 - ax(n)^2 + bx(n-1) + u(n). \quad (16)$$

当未施加控制作用,即 $u(n) = 0$ 时,在 $a = 1.4$, $b = 0.3$ 的情况下,系统处于混沌态^[3].选择离散 PID 控制器,基于 PSO 算法运行 60 步后的控制器参数优化结果为 $K_p = 7.4$, $K_i = 0.9$, $K_d \approx 0$,因此只使用 PI 作用.初始条件为 $x(0) = 1$,基于 Matlab 6.1/Simulink 编程,当 $n \geq 50$ 时加入控制作用,目标是控制 $x(n)$ 到 0,系统的控制效果如图 1 所示.

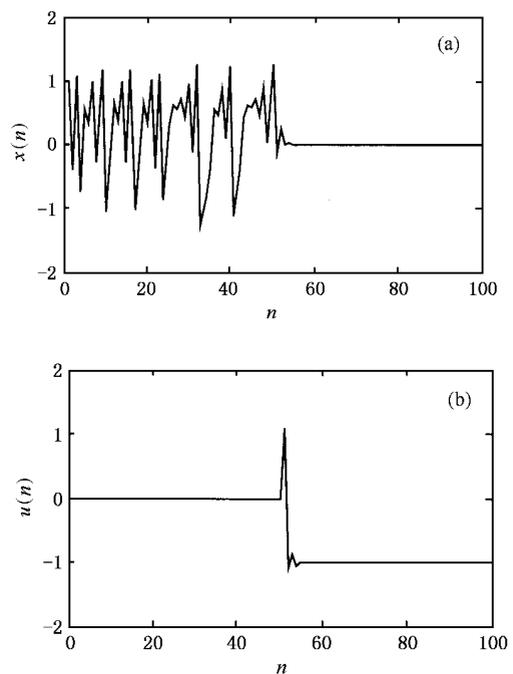


图 1 Hénon 混沌系统的 PI 控制结果 (a) 系统输出, (b) 控制作用

4.2. Duffing 混沌系统

受控的 Duffing 混沌系统

$$\frac{d^2x}{dt^2} + a \frac{dx}{dt} + bx + cx^3 = d \sin(\omega t) + u. \quad (17)$$

当未施加控制作用,即 $u = 0$ 时,在 $a = 0.4$, $b = -1.1$, $c = 1$, $d = 2.1$, $\omega = 1.8$ 的情况下,系统处于混沌态^[13].选择连续 PID 控制器,基于 PSO 算法运行 100 步后的控制器参数优化结果为 $K_p = 502.3$, $K_i = 10.3$, $K_d = 5.1$.初始条件为 $x(0) = 0$,基于 Matlab 6.1/ Simulink 编程,当 $t \geq 50$ 时加入控制作用,目标是控制 $x(t)$ 到 0,系统的控制效果如图 2 所示.

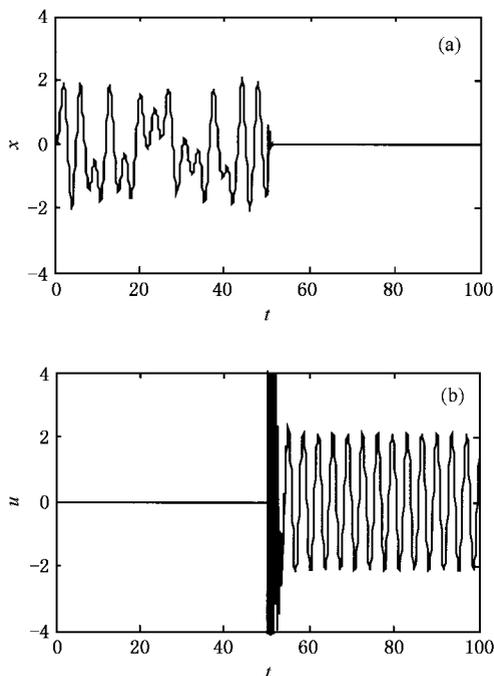


图 2 Duffing 混沌系统的 PID 控制结果 (a) 系统输出, (b) 控制作用

4.3. 六辊 UC 轧机混沌系统

受控的六辊 UC 轧机混沌系统

$$x(n+1) = rx(n) \exp(1 - x(n)^2) + u(n). \quad (18)$$

当未施加控制作用,即 $u(n) = 0$ 时,在 $r \geq 1.5$ 的情况下,系统处于混沌态^[14].选择离散 PID 控制器,本文选取 $r = 1.55$,基于 PSO 算法运行 100 步后的控制器参数优化结果为 $K_p = -1.6992$, $K_i = 0.8924$, $K_d \approx 0$,因此只使用 PI 作用.初始条件为 $x(0) = 1.45$,基于 Matlab 6.1 编程,当 $n \geq 50$ 时加入控制作用,目标是控制 $x(n)$ 到平衡点 1.199273,系统的控制效果如图 3 所示.

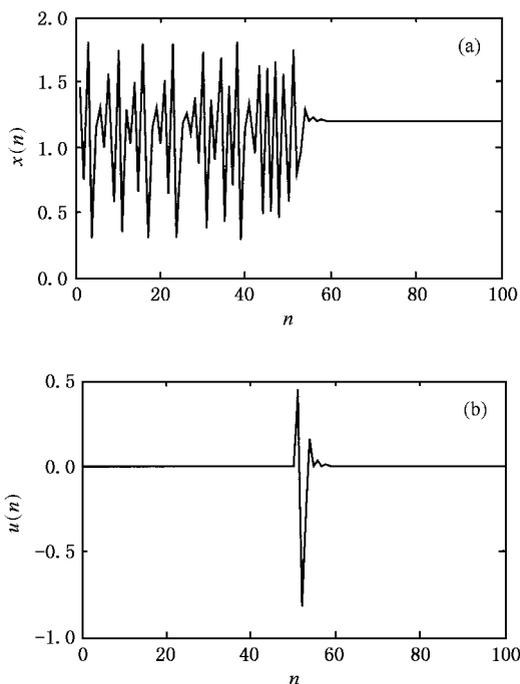


图 3 六辊 UC 轧机混沌系统的 PI 控制结果 (a) 系统输出, (b) 控制作用

4.4. Nagumo-sato 神经元混沌系统

受控的修改和扩展的 Nagumo-sato 混沌模型为

$$\begin{aligned} x(t+1) &= kx(t) - a[f(x(t))] + b + u(t), \\ y(t+1) &= f[x(t+1)], \end{aligned} \quad (19)$$

$$f(x) = (|x + \epsilon| - |x - \epsilon|)4\epsilon + 1/2.$$

当未施加控制作用,即 $u(t) = 0$ 时,在 $k = 0.5$, $a = 1$, $b = 0.3$, $\epsilon = 0.5$ 的情况下,系统处于混沌态^[15].选择离散 PID 控制器,基于 PSO 算法运行 100 步后的控制器参数优化结果为 $K_p = -0.9953$, $K_i = 0.2803$, $K_d \approx 0$,因此只使用 PI 作用.初始条件为 $x(0) = 0.2$, $y(0) = -0.25$,基于 Matlab 6.1 编程,当 $t \geq 50$ 时加入控制作用,目标是控制 $y(t)$ 到 0,系统的控制效果如图 4 所示.

4.5. Chen 氏混沌系统

受控的 Chen 氏混沌系统

$$\begin{aligned} \frac{dx}{dt} &= a(y - x) + u_1, \\ \frac{dy}{dt} &= (c - a)x - xz + cy + u_2, \\ \frac{dz}{dt} &= xy - bz + u_3. \end{aligned} \quad (20)$$

当未施加控制作用,即 $u_1 = 0$, $u_2 = 0$, $u_3 = 0$ 时,在

$a = 35, b = 3, c = 28$ 的情况下,系统处于混沌态^[4].选择连续 PID 控制器,基于 PSO 算法运行 200 步后的控制器参数优化结果为 $K_{p1} = 31.70, K_{i1} = 69.75, K_{p2} = 51.03, K_{i2} = 8.74, K_{p3} = 13.47, K_{i3} =$

$15.41, K_{d1} \approx 0, K_{d2} \approx 0, K_{d3} \approx 0$,因此只使用 PI 作用.初始条件为 $(-18, 3, 12)$,基于 Matlab 6.1/ Simulink 编程,当 $t \geq 10$ s 时加入控制作用,目标是控制系统到平衡点 $(0, 0, 0)$,系统的控制效果如图 5 所示.

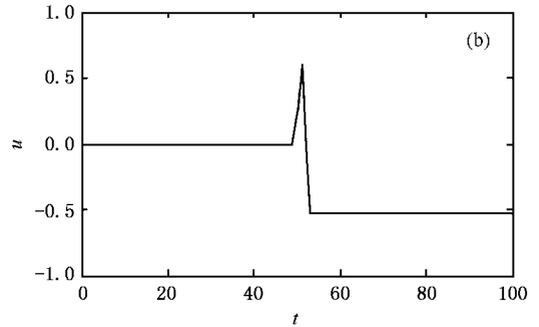
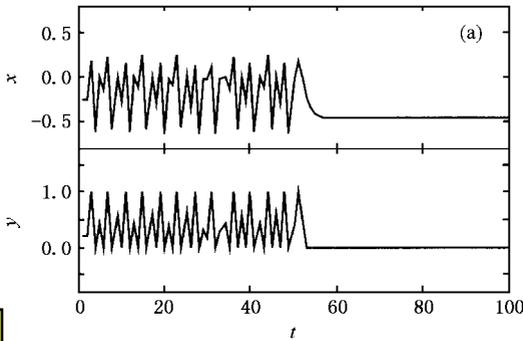


图 4 神经混沌系统的 PID 控制结果 (a)系统输出,(b)控制作用

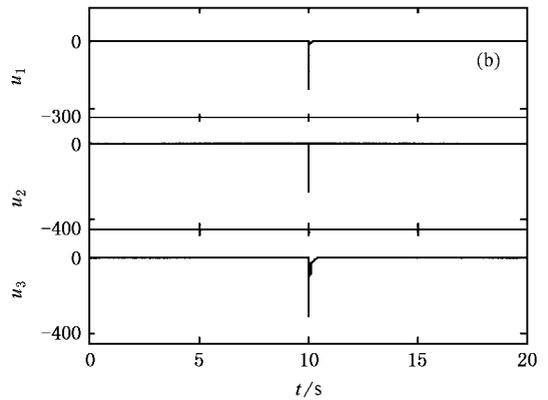
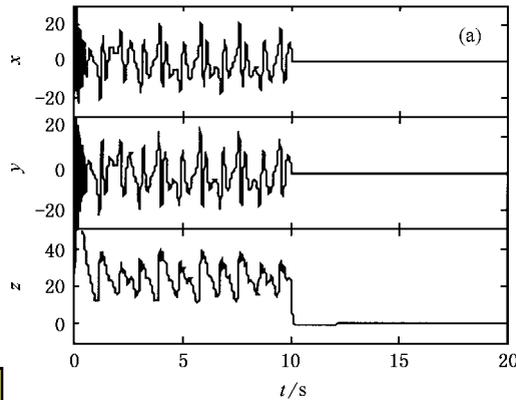


图 5 Chen 氏混沌系统的 PI 控制结果 (a)系统输出 (b)控制作用

4.6. 永磁同步电动机中的混沌

经整理后的受控永磁同步电动机中的混沌数学模型可写为

$$\begin{aligned} \frac{dx}{dt} &= -x + yz + u_1, \\ \frac{dy}{dt} &= -xz - y + \gamma z + u_2, \\ \frac{dz}{dt} &= \alpha(y - z) + u_3. \end{aligned} \quad (21)$$

当未施加控制作用,即 $u_1 = 0, u_2 = 0, u_3 = 0$ 时,在 $\gamma = 20, \sigma = 5.46$ 的情况下,系统处于混沌态^[16].选择连续 PID 控制器,基于 PSO 算法运行 200 步后的控制器参数优化结果为 $K_{p1} = 7.66, K_{i1} = 14.66, K_{p2} = 47.82, K_{i2} = 7.50, K_{p3} = 12.32, K_{i3} = 29.04, K_{d1} \approx 0, K_{d2} \approx 0, K_{d3} \approx 0$ 因此只使用 PI 作用.初始条

件为 $(0.01, 0.01, 0.01)$,基于 Matlab 6.1/Simulink 编程,当 $t \geq 10$ s 时加入控制作用,目标是控制系统到平衡点 $(0, 0, 0)$,系统的控制效果如图 6 所示.

以上只给出了 6 种混沌系统分别在某一初始条件下控制到某一平衡点或非平衡点时的 PID 控制仿真结果.事实上,作者还进行了以下诸多的仿真研究,即在其他初始条件下控制,或将系统控制到其他平衡点或非平衡点,或使用已经优化好的 PID 调节器控制改变了有关参数的混沌系统,所有仿真结果均表明了 PSO 算法优化的 PID 调节器对以上 6 种混沌系统具有良好的控制性能.限于篇幅,在此未能给出其他仿真结果.仿真结果还表明,运用 PSO 算法优化的混沌 PID 控制不需要使用深奥的数学工具,不过于依赖混沌系统本身的数学模型, PID 参数的寻优过程十分简单.

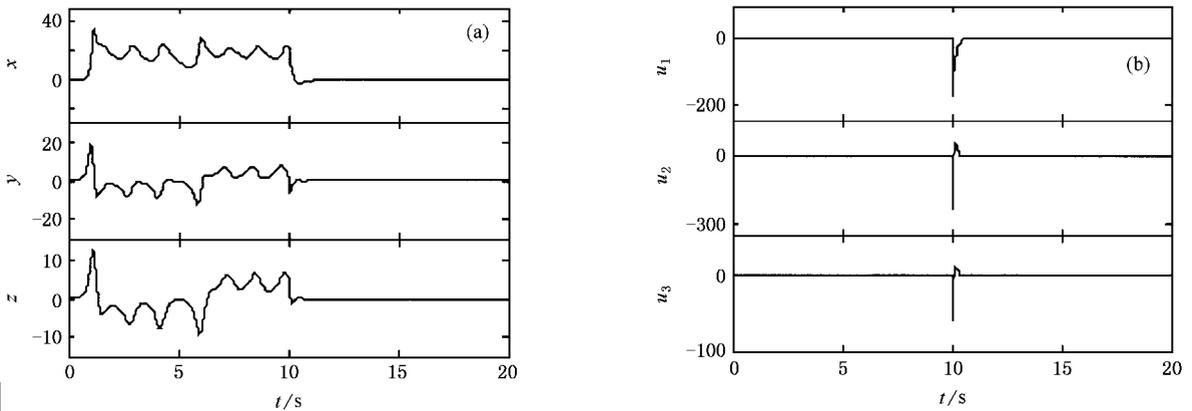


图6 永磁同步电动机中混沌系统的PI控制结果 (a)系统输出,(b)控制作用

5. 结 论

将集群智能算法引入混沌系统的PID控制器优化设计,取得了很好的控制效果,在拓宽PID控制器的应用范围方面是非常有力的证明,同时也再一次表明了使用简单方法控制复杂混沌系统的可能性。然而这仅仅是初步的实验,还有大量的工作需要进

一步深入研究,如PID调节器用于混沌控制的鲁棒性分析和频域分析是值得进一步研究的。推而广之,PID调节器用于其他非线性系统的控制及其分析也值得研究。

另外,PID用于Chen氏混沌系统和永磁同步电动机中混沌的控制消耗能量过大(即控制量 u_1 , u_2 , u_3 过大),如何降低控制过程中的能量消耗将是需要研究的,这也是其实用化的关键。

- [1] Xue Y J, Yin X H, Feng R P 2000 *Acta Phys. Sin.* **49** 641 (in Chinese) [薛月菊、尹逊和、冯汝鹏 2000 物理学报 **49** 641]
- [2] Ren H P, Liu D 2002 *Acta Phys. Sin.* **51** 982 (in Chinese) [任海鹏、刘 丁 2002 物理学报 **51** 982]
- [3] Chen L, Chen G R 2000 *IEEE Trans. Circ. Syst. (Part I)* **47** 1527
- [4] Lu J H, Chen G R, Zhang S C 2003 *J. Graduate School of the Chinese Academy of Sciences* **20** 123
- [5] Jiang G P, Chen G R, Tang W K 2002 *IEEE Trans. Circ. Syst.* **49** 1820
- [6] Yang Z H, Yao Q H, Yang C H 2003 *Elec. Mach. Contr.* **7** 67 (in Chinese) [杨志红、姚琼荃、杨昌宏 2003 电机与控制学报 **7** 67]
- [7] Kennedy J, Eberhart R 1995 *Proc. IEEE Int. Conf. Neu. Net.* (Perth: IEEE) p1942
- [8] Kennedy J, Spears W M 1998 *Proc. IEEE Int. Conf. Evol. Comp.* (Anchorage: IEEE) p78
- [9] Shi Y H, Eberhart R 1998 *Proc. 7th Annual Conf. Evol. Prog.* (Washington: IEEE) p591
- [10] Ozcan E, Mohan C 1999 *Proc. Congress Evol. Comp.* (Piscataway: IEEE) p1939
- [11] Clerc M 1999 *Proc. Congress Evol. Comp.* (Piscataway: IEEE) p1951
- [12] Carlisle A, Dozier G 2001 *Proc. Workshop PSO* (Indianapolis: Purdue School of Eng. Tech.) p191
- [13] Jiang Z P 2002 *IEEE Trans. Circ. Syst. (Part I)* **49** 244
- [14] Yan D J, Liu X D, Zhu W Y 1999 *Control and Decision* **14** 156 (in Chinese) [鄢德军、刘向东、朱伟勇 1999 控制与决策 **14** 156]
- [15] Aihara K, Takabe T, Toyoda M 1990 *Physica A* **144** 333
- [16] Zhang B, Li Z, Mao Z Y et al 2001 *Contr. Theory Appl.* **18** 796 (in Chinese) [张 波、李 忠、毛宗源等 2001 控制理论与应用 **18** 796]

Proportional-integral-derivative chaotic system control algorithm based on particle swarm optimization *

Wang Dong-Feng Han Pu

(*Department of Automation , North China Electric Power University , Baoding 071003 , China*)

(Received 12 November 2004 ; revised manuscript received 21 September 2005)

Abstract

Proportional-integral-derivative (PID) control algorithm has the property of simplicity and practicability , but it is difficult to be tuned to control a complex nonlinear system. An advanced particle swarm optimization (PSO) was used to optimize the PID controller for controlling chaotic systems. Computer simulation of its application to several chaotic systems has been done. The controlled chaotic systems are the Hénon chaos , Duffing chaos , six-roll UC mill chaos , Nagumo-sato neuron chaos , Chen 's chaos , and permanent magnet synchronous motor chaos. The chaos control simulation results show that : 1) The output feedback based PID control for chaotic systems is effective ; 2) using PID to control complex chaotic systems is feasible ; and 3) using PSO to optimize the PID parameters for chaos control is effective and simple in programming.

Keywords : chaotic system , proportional-integral-derivative control , particle swarm optimization

PACC : 0545

* Project supported by the Major Program Foundation of North China Electric Power University ,China (Grant No. 20041306) and the Doctoral Foundation of North China Electric Power University ,China (Grant No. 02060106).