

新型动态粒子群算法及其在工业控制中的应用

陈俊

(安徽机电职业技术学院电气工程系,安徽 芜湖 241000)

摘要:粒子群优化算法是一种在复杂优化问题的空间域探求最优解的启发式搜索方法。为了优化传统的 PSO 算法,缩短其运行时间,提出了一种动态收缩型的粒子群优化。收缩型粒子群算法在初始阶段含大量的粒子,随着迭代次数的增加,粒子数量不断减少,仿真结果显示,此方法相比于传统的 PSO 算法可减少近 60% 的运算时间。运用此改进的粒子群算法优化理论,实现了对 PID 控制参数的自适应调节。结果表明,新型的 PSO 算法可以使得 PID 控制参数调整速度更快,产生超调量小。

关键词:动态粒子群;工业控制;优化;PID

中图分类号:TP181

文献标识码:A

引言

常规 PID 控制是目前工业过程控制中应用最为广泛的策略之一,其控制结构简单,主要适用于线性系统控制。而工业环境中被控对象往往具有高度非线性、时变不确定性和纯滞后等特点,常规 PID 控制器存在参数整定不良、工况适应性差等问题。为此人们将模糊控制与 PID 控制结合,然而一般模糊控制器均表现为 PD 控制器的性能,难以获得积分误差的规则,故系统存在稳态误差^[1]。如何有效的整定其参数,以达到所期望的动态性能的问题,仍值得探讨。

对 PID 参数整定的要求,不仅要依赖于对象的数学模型,同时还要具有实时调节的功能。优化技术是一种以数学为基础,用于求解工程问题最优解的技术,在科研工程领域发挥着极其重要的作用。粒子群优化最早是由 Kennedy 和 Eberhat 于上世纪九十年代中期提出,它是受到自然界生物的群体行为的启发而发展起来,在这里,群体指的就是 swam,而具体的群体里面的个体就是我们这里的 particles,每一个粒子都代表了一个可能的最优解,经过迭代,每一个粒子都加速向各自独立的最优解的方向移动,因而全局最优解就会在这些粒子中,此算法目前已广泛应用于目标函数优化、神经网络

训练等领域,可以解决所有的优化问题。

一般典型的 PSO 算法的执行往往从较为适度的 40 个粒子开始,这些粒子都已经被初始化为此空间里的任意一个解,随着迭代工作的进行,这些粒子便逐步趋向于一个全局的最优解^[2]。一般情况下,优化过程将重复大约 10 000 次,才能确认粒群已经聚集到全局最优。如果适应函数比较复杂,那么每次迭代的时间和优化过程将趋于延长,直到粒子群收敛聚集,在这种情况下,若绝大多数的粒子群都已经聚集到非常近的位置,那么对于整体 40 个粒子的反复迭代过程就不是很必要了^[3]。

针对此问题,本文提出一种加速 PSO 收敛的优化算法,更好的实现 PID 控制中参数的实时调整和整定。

1 传统粒子群算法基本描述

用 s 表示群体的大小,每一个粒子 $1 \leq i \leq s$ 在这里都会被定义三种属性:(1)粒子位置向量 X_i ;(2)粒子移动向量 V_i ;(3)在此局部范围内得到的最优个体粒子的位置 L_i 。另外,用 G 表示粒子群里当前全局的最优粒子^[4]。

第一步:初始化 X_i 和 V_i ,并设定在 $1 \leq i \leq s$ 的范围内,局部最优 $L_i = X_i$ 。

第二步:在 $1 \leq i \leq s$ 的范围内,估求每个 X_i 的值。

第三步:使得 G 为 $\{L_1, L_2, \dots, L_s\}$ 集合里的最优

收稿日期:2011-5-26

基金项目:安徽省高等学校优秀青年人才基金项目(2009SQRZ204)

作者简介:陈俊(1982-),男,安徽芜湖人,讲师,硕士,主要从事智能控制和自动检测方面的研究,(E-mail) cj11@163.com

粒子。

第四步:这样,在 $1 \leq i \leq s$ 的范围内,粒子移动向量 V_i 的优化更新可以表示为:

$$V_i = wV_i + c_1 r_1 (L_i - X_i) + c_2 r_2 (G - X_i)$$

这里, w 称为惯性权重,传统的 PSO 算法,这里的 w 的值默认为 1,然而,经过后来学者们的研究发现,如果在迭代过程中,让 w 从 0.5 ~ 0.1 之间递减,可以显著提高收敛速度。 c_1 和 c_2 分别称为自学习因子和群学习因子, r_1 和 r_2 是两个处于 0 ~ 1 之间的随机的两个数。粒子的位置优化根据公式:

$$X_i = X_i + V_i$$

如果发现 X_i 比 L_i 好,那么就会有: $L_i = X_i$ 。

第五步:重回第三步直至粒子群收敛完成^[5]。

PSO 原理上十分简单,所需参数比较少,易于实现。然而传统的 PSO 算法最致命的缺陷在于,每次迭代过程中,存在着每个粒子过多的循环计算,按传统的方法,初始化的 40 个粒子将重复消耗 10 000 次迭代,那么总计将需要计算 400 000 次,如果可以在不影响运算结果的前提下,用某种方法将运算次数减少,就可以显著提高 PSO 的运算速度,加快收敛。

2 动态收缩型粒子群算法(DP-PSO)

对于复杂的优化问题,在粒子群优化算法的初始阶段,可以给定一个远大于 40 的初始粒子数量,以保证在空间域中有最广阔的搜索域,涵盖具有全局最优值的区域。一旦此区域被搜索到,算法将很好的调整所有粒子的位置,以向全局最优值收敛。所以,此策略可以初始化众多的粒子,以确保此算法能识别出含有全局最优值的区域,然而在后期的迭代过程中,当搜索域已经缩小到较小的区域时,就不需要那么多的粒子了。

对于此策略来说,对于粒子性能好坏的鉴别工作,相当于是额外的工作,这部分的消耗必须充分考虑到^[6]。显然,这个工作对于此策略来说是必不可少的,在整个运行过程中,这样对粒子性能进行的鉴定分类只需要几次,一般四五次就可以了。而这仅有的几次额外的分类工作,每一次都可以十分有效的减少后面进行迭代的粒子数量,显然,这样的“额外工作”相比于大量减少的迭代工作而言,是近乎可以忽略的。

粒子群算法计算优化 PID 参数的基本方式有离线计算和在线计算两种。离线计算在确定参数后,其参数是不能变的,极大制约了 PID 控制性能;所以,这里我们采用在线计算,在动态过程的每一步都进行 PID 参数寻优计算,使之能够跟随系统的动态性能变化,从而具有专家 PID 控制特性,满足动静态性能要求。

3 实例仿真

对于 DP-PSO,在迭代次数达到总次数的四分之一时,根据粒子的适应度,使用大量的分类算法对粒子进行性能鉴别和分类,舍弃其中性能较差的一半粒子群,然后对剩下的另一半粒子继续进行迭代计算,同样的,再经过总次数的四分之一时,重复一次粒子性能鉴定,并舍弃其中一半的粒子,如此往复下去,直到迭代结束。

为了验证 DP-PSO 相比于传统 PSO 的优势,使用一个常用来评价优化算法性能的无约束最大值求解函数

$$\max f(x, y) = 0.5 - \frac{(\sin \sqrt{x^2 + y^2})^2 - 0.5}{(1.0 + 0.001(x^2 + y^2))^2}$$

通过 Matlab 软件仿真,分别使用 DP-PSO 和 PSO 进行计算,分析最终的结果。此验证函数拥有一个很特别的最优值,就是在(0,0)点的时候,另外,它具有很多局部最优解,所以对于一些传统的优化算法,这个问题是很困难的,在 Matlab 环境中,使 x 从 -100 ~ 100 之间变化,保持 y 处于最优点(即 $y=0$),就可以得到图 1 的函数图形。

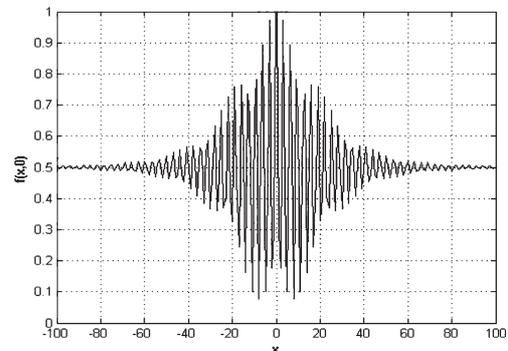


图 1 仅 x 变化而 y 不变时得到的函数图

分别使用 DP-PSO 和 PSO,初始化粒子数同为 32 个,随机分布在解空间内。选定初始位置同样的 10 个样本,可以得到表 1 数据。

表 1 DP-PSO 和 PSO 的运行时间比较

样本	运行时间(秒)		提速量
	PSO	DP-PSO	DP-PSO
1	2.0317	0.9795	52%
2	1.9901	0.8514	57%
3	2.0094	1.2845	36%
4	1.9846	0.8734	56%
5	1.9794	0.8672	56%
6	2.0880	1.0408	50%
7	1.9768	0.8614	56%
8	2.3213	0.8757	62%
9	1.9297	0.8740	55%
10	1.9348	0.8449	56%
平均值	2.0246	0.9353	54%

可以看到,虽然迭代次数相同,因为 DP-PSO 的粒

子数量的递减,即便增加了粒子性能优劣鉴定的工作,DP-PSO 粒子迭代的总次数仍得到了大幅的减少,从而加快了向全局最优解收敛的速度,平均速度提高了54%。据此,我们将 DP-PSO 算法引入调节 PID 控制器的增益。将 PID 控制器的一组参数 (K_p, K_i, K_d) 作为 PSO 中的一个粒子的位置 x_i , 把 PID 控制器的参数整定问题转化为三维的 PSO 算法问题,而后,定义一个性能指标作为目标函数,用于评价 PSO 算法中粒子个体的适应度。以二阶系统研究基于自适应在线 DP-PSO 算法整定的 PID 控制。在一定的采样时间内,选取足够多的个体,用性能指标评估每个个体的适应度,通过 PSO 优化,选定适应度大的个体对应的 PID 参数作为采样时间下 PID 的控制参数。常用的性能指标主要包含时间和误差两个变量,误差以绝对值的形式出现。

假定二阶系统为:

$$G(s) = \frac{400}{s^2 + 50s}$$

基于 Matlab/Simulink 环境进行仿真,比较传统 PSO 和 DP-PSO 在 PID 控制器在线调节中各自的表现。种群大小均设为 30, 参数 K_p 的取值范围设定为 $[0, 20]$, 参数 K_i 以及 K_d 的取值范围均设定为 $[0, 1]$ 。取 $\omega_1 = 0.999, \omega_2 = 0.01, \omega_3 = 2.0, \omega_4 = 100$, 最大迭代次数为 100。结果见表 2。

表 2 DP-PSO 和 PSO 在二阶系统上的比较

方法	超调 (%)	迭代次数	整定参数的结果			收敛速度
			K_p	K_i	K_d	
DP-PSO	0.03	68	18.65	0.26	0.01	0.010s
PSO	0.03	65	18.65	0.26	0.01	0.016s

从表 2 可以看出,DP-PSO 与传统 PSO 相比,在性能相同的前提下,其迭代次数相差不多,而收敛速度却

远远高于传统 PSO,减少了将近 60% 的时间,超调小,上升时间短。因此,效果更佳。

4 结束语

本文在理论上分析了 PSO 算法并对其进行了进一步的深究和探索,使其更好的应用于 PID 控制器的参数整定,实现参数的优化整定。仿真结果表明,该方法可以加快 PSO 的收敛速度,提高其实时性效果,具有一定的应用价值。

参考文献:

- [1] 张明君,张化光.基于遗传算法优化的神经网络 PID 控制器[J].吉林大学学报:工学版,2005,35(1):91-96.
- [2] Kennedy J, Eberhart R C. Particle swarm optimization [C]. In: Proc. of the IEEE Conf. on Neural Networks, IV. Perth: IEEE Press, 1995, 1942-1948.
- [3] 吕振肃,志荣.自适应变异的粒子群优化算法[J].电子学报,2004,32(3):16-20.
- [4] Abraham A, Nath B. Hybrid Intelligent Systems Design: A Review of Decade of Research [R]. Technical Report Series, School of Computing and Information Technology, Faculty of Information Technology, Monash University, Australia, 2000, 1-37.
- [5] Van den Bergh F, Engelbrecht A P. A Cooperative Approach to Particle Swarm Optimization [C]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 6.225-239.
- [6] 武志峰,杨蓓.一种改进的粒子群优化算法[J].郑州大学学报:理学版,2007,9(3):109-112.

Application of Evolutionary Dynamic PSO in Industry Control

CHEN Jun

(Department of Electric Engineering, Anhui Professional University of Mechanic and Electrician, Wuhu 241000, China)

Abstract: Particle Swarm Optimization (PSO) is a heuristic search method for the exploration of solution spaces of complex optimization problems. With the aim of reducing execution time, a dynamic population size improvement for classical PSO is explored. Diminishing Population PSO (DP-PSO) starts with a large number of particles and iteratively reduces the swarm size. The simulation result shows this improvement produce almost 60% reduction in the execution time compared to the classical PSO. The evolutionary optimization theory of Particle Swarm Algorithm is automatically applied to tune the parameters of PID controllers. The result indicates that the new-style PSO makes the tuning of PID controller parameters more quickly, and the overshoot small.

Key words: dynamic PSO; industry control; optimization; PID