文章编号: 1001-0920 (2011) 05-0794-03

基于交叉熵算法的PID控制器设计

李 洁[1], 柴天佑[1], 宫经宽[2]

(1. 东北大学 流程工业综合自动化教育部重点实验室，沈阳 100819；2. 北京航空工程技术研究中心，北京 100076)

摘 要: 交叉熵优化方法是一种新型高效的随机优化算法, 算法控制参数简单, 鲁棒性强. 将交叉熵优化算法用于PID控制器的参数设计, 并与基于遗传算法的PID控制器设计进行对比, 结果表明, 交叉熵优化算法不仅所获结果较优, 而且计算复杂度也明显小于遗传算法.

关键词: 交叉熵；优化；PID控制器

中图分类号: TP18 **文献标识码:** A

Design of PID controller using cross entropy method

LI Jie[1], *CHAI Tian-you*[2], *GONG Jing-kuan*[2]

(1. Key Laboratory of Integrated Automaton for Process Industry，Northeastern University，Shenyang 110819，China; 2. Beijing Aeronautical Engineering and Technology Research Center，Beijing 100076，China. Correspondent：LI Jie，E-mail：lj2010079@hotmail.com)

Abstract: The cross entropy(CE) is a kind of efficient stochastic optimization method, which has simple control parameter setting and high robustness. CE method is applied to the design of PID controllers which is compared with the genetic algorithm(GA) based PID controller design methods. The result shows that CE algorithm has significant low computation complexity.

Key words: cross entropy；optimization；PID controller

1 引 言

PID控制是工业过程控制中应用最广的策略之一. PID控制器具有简单固定的形式, 性能优良, 易于实现. 同时, 其控制品质对被控对象特性的变化不太敏感, 在很宽的操作条件范围内都能保持较好的鲁棒性. 由于具有这些优点, PID控制被广泛应用于工业与民用领域, 并取得了很好的经济效果. 但PID参数整定过程一直困扰着工程技术人员, 用经验规则加试凑的方式来调整PID控制参数, 往往费时且难以满足控制要求, 因此PID控制器参数的优化成为人们关注的问题. 目前PID参数的优化方法有很多, 如梯度法和爬山法等. 近年来广泛采用启发式优化算法, 如遗传算法 (GA)[1,5-8]和粒子群优化 (PSO) 等[1]. 但这类启发式算法的计算复杂度较高, 同时需要设置多个控制参数, 并要求使用者具备较高的算法调试经验.

本文将交叉熵 (CE)[2-4,9-10]算法引入PID控制, 为PID参数的优化整定提供了新的途径. 交叉熵算法是近年来出现的一种新型随机优化算法, 它只依赖于适应度函数, 不需梯度等信息, 即使在对象模型不确定的情况下, 仍可根据对象的输出情况对PID控制器中的参数进行优化, 而且交叉熵算法具有计算复杂度低、算法控制参数少、鲁棒性强、易于编程等特点, 适合于工程实践应用.

2 PID控制器

PID控制器本身是一种基于“过去”、“现在”和“未来”信息估计的简单控制算法. PID控制器根据给定值$r(t)$和实际输出值$y(t)$构成控制偏差$e(t)=r(t)-y(t)$, 将偏差按比例、积分和微分通过线性组合构成控制量, 对被控对象进行控制, 其控制规律为

$$u(t)=K_{\rm p}\Big[e(t)+\frac{1}{K_{\rm i}}\int_0^t e(t){\rm d}t+K_{\rm d}\frac{{\rm d}e(t)}{{\rm d}t}\Big]. \quad (1)$$

3 交叉熵算法

交叉熵方法是一种新型随机优化方法, 最先用于模拟小概率事件, 后来扩展到求解最优化问题[2-4,9-10]. 交叉熵方法广泛应用于组合优化、连续多目标优化和机器学习等领域的若干实际问题中. 交叉熵方法的基

收稿日期: 2010-04-14；**修回日期:** 2010-07-12.

基金项目: 国家863计划项目(2006AA060202)；国家科技支撑计划项目(2007BAE25B01).

作者简介: 李洁(1974−), 女, 博士生, 从事企业优化管理的研究；柴天佑(1947−), 男, 中国科学院院士, 教授, 博士生导师, 从事智能解耦控制、自适应控制等研究.

本思想是首先将优化问题

$$S(x^*)=\gamma^*=\min_{x\in\gamma}S(x) \tag{2}$$

同与其相关的概率估计问题进行关联. 在γ上定义一组概率密度函数$\{f(x,v),\ v\in V\}$和指示函数$\{I_{S(x)\leqslant\gamma}\}$, 对于某个$u\in V$, 式 (2) 优化问题的相关估计为

$$l(\gamma)=P_u(S(x)\leqslant\gamma)=\int_{x\in\gamma}I_{S(x)\leqslant\gamma}f(x,u)\mathrm{d}x=E_u[I_{S(x)\leqslant\gamma}]. \tag{3}$$

当式 (3) 中γ取值接近最优解时, 概率l的值会非常小, $S(x)\leqslant\gamma$为一小概率事件. 直接用 Monte Carlo 方法求解式 (3), 需要有足够大的样本数量. 为此引入重要采样 (IS) 密度$g(x)$, 则式 (3) 变为参数估计式

$$l(\gamma)=\frac{1}{N}\sum_{i=1}^{N}I_{S(x^{(i)})\leqslant\gamma}\frac{f(x^{(i)};u)}{g(x^{(i)})}. \tag{4}$$

其中: $x^{(i)}$是由$g(x)$生成的样本, $l(\gamma)$的估计取决于$g(x)$的选择. 为求得最优的重要采样密度

$$g^*(x)=\frac{I_{S(x)\leqslant\gamma}f(x,u)}{g(x^{(i)})}, \tag{5}$$

需选择分布族$\{f(x,v),v\in V\}$中的概率密度函数, 并通过优化参数V, 使得$f(x,v)$与$g^*(x)$之间的交叉熵最小 (即 Kullback-Leiber 距离最小). 两个分布密度函数之间的交叉熵定义为

$$\mathrm{KL}[g,f]=\int g(x)\log\frac{g(x)}{f(x)}\mathrm{d}x, \tag{6}$$

使得$g^*(x)$与$f(x,v)$交叉熵最小的等价式为

$$\min_v \mathrm{KL}[g^*(x),f(x,v)]=\min_v E_uI_{S(x)\leqslant\gamma}\log f(x,v). \tag{7}$$

交叉熵方法采用了一个多级更新算法, 即构造一个分布参数序列$\{v_t,t\geqslant 0\}$和一个级序列$\{\gamma_t,t\geqslant 0\}$, 因此式 (7) 可变为

$$\min_v \mathrm{KL}[g^*(x),f(x,v)]=\min_v E_{v_{i-1}}I_{S(x^{(i)})\leqslant\gamma}\log f(x,v); \tag{8}$$

$$\min_v \mathrm{KL}[g^*(x),f(x,v)]=\min_v \frac{1}{N}\sum_{i=1}^{N}I_{S(x^{(i)})\leqslant\gamma}\log f(x^{(i)},v). \tag{9}$$

式 (9) 为 (8) 的参数估计形式, 其中$x^{(i)}$是由$f(x,v_{t-1})$生成的样本.

4 基于交叉熵算法的 PID 控制器参数优化

在 PID 控制器中, 需要优化的参数有 3 个: 比例系数K_p, 积分时间常数K_i和微分时间常数K_d, 即$x=[K_\mathrm{p},\ K_\mathrm{i},\ K_\mathrm{d}]$.

1) 样本编码及其概率分布的选择. 对于 3 个需要优化的参数, 采用实数值编码. 概率分布函数的选择是交叉熵算法中一个重要的环节. 在组合优化中, 常用的概率分布函数是二项分布; 而针对连续值优化问题, 常用的概率分布函数是正态分布函数$N(\mu,\varSigma)$. 根据优化的问题, 本文选用三元正态分布作为抽样用的概率密度函数, 即$f(x;v)\equiv N(x\mid\mu,\varSigma)=(2\pi)^{-3/2}\mid\varSigma\mid^{-1/2}\mathrm{e}^{-(x-u)^\mathrm{T}\varSigma^{-1}(x-u)/2}$, 其中正态分布的均值$\mu$和方差$\varSigma$的值会根据交叉熵迭代算法自适应地调整. 当算法接近收敛时, μ接近全局或是局部最优值, 而$\varSigma$会变得非常小, 此时从$N(\mu,\varSigma)$中抽样, 将会以大概率得到一个非常接近全局或是局部最优值的数值, 从而达到优化目的.

2) 群体中个体评价, 即确定适应度函数, 然后根据群体每一个个体 (对应一组参数值) 的适应度函数值评价个体的优劣. 对于 PID 参数取值优劣, 可由偏差积分指标衡量. 常用偏差积分指标如下:

$$\mathrm{MSE}=\frac{1}{t}\int_0^z e(t)^2\mathrm{d}t,\ \mathrm{LAE}=\int_0^z e(t)\mathrm{d}t,$$

$$\mathrm{ITAE}=\int_0^z t\mid e(t)\mid^2\mathrm{d}t.$$

注 1　采用不同的积分公式意味着估计整个过渡过程优良程度的侧重点不同. 适应度函数可以定义为偏差积分, 例如$S_{\mathrm{MSE}(x)}=\mathrm{MSE}$. 一组 PID 参数对应的偏差积分越小, 该参数组合适应值越小; 反之则反.

3) 算法的具体实现如下:

Step 1: 初始化. 最大迭代次数为$T=25$, 概率分布为$N(\mu_t,\varSigma_t)$, 初始值$\mu_{0,p}\sim U(0,10)$ (for $P=1,2,3$). 其中: $U(a,b)$表示a与b之间的均匀分布, $\varSigma_t=5I_3$, 样本数为$N=400$, 分位数为$\theta=0.1$, 光滑因子为$\alpha=0.8$.

Step 2: 从概率分布为$N(\mu_t,\varSigma_t)$中抽取N个样本$\{X^{(i)}\}_{i=1}^N$.

Step 3: 计算适应度函数S, 并将样本从小到大排序: $S(1)\leqslant\cdots\leqslant S(N)$. 取适应函数值小的前$\mathrm{Nel}=\theta N$个样本$\{X^{(i)}\}_{i=1}^{\mathrm{Nel}}$.

Step 4: 根据下式更新μ_t, $\varSigma_t$的值:

$$\mu_t=\alpha\mu_{t-1}+\frac{1-\alpha}{\mathrm{Nel}}\sum_{i=1}^{N}\widetilde{X}^{(i)}, \tag{10}$$

$$\varSigma_t=\alpha\varSigma_{t-1}+\frac{1-\alpha}{\mathrm{Nel}}\sum_{i=1}^{N}(\widetilde{X}^{(i)}-\mu_{t-1})(\widetilde{X}^{(i)})^\mathrm{T}. \tag{11}$$

Step 5: 如果$t=T$, 终止程序; 否则, $t=t+1$, 返回 Step 2.

5 仿真实验

被控对象的传递函数为

$$G(s)=\frac{1}{s^4+6s^3+11s^2+6s},$$

优化对象为$K_\mathrm{p},K_\mathrm{i}$和$K_\mathrm{d}$, 其取值范围均限制在 [0,

40] 之间.

除 CE 算法外, 同时实现了基于 GA 的 PID 参数寻优算法. GA 算法控制参数设置如下: 种群规模 100, 交叉概率 0.6, 变异概率 0.05, 最大进化代数 500. 为防止在遗传算法运行过程中搜索到的最优参数值因选择、交叉或者变异操作而破坏, 每一次迭代中的最优参数值均被保留直接进入下一代种群.

按照上文中所述方法, 分别采用 MSE, IAE 和 ITAE 指标对 PID 控制参数进行优化, 每种情况下分别运行 GA 和 CE 算法 50 次, 优化结果统计见表 1. 其中: GA 和 CE 算法的结果为运行 50 次的平均值, 括号内是标准差. 统计结果显示, CE 算法所得结果在各种指标下均优于 GA 的结果. 同时, CE 所得结果方差较小, 说明算法鲁棒性强, 所得结果具有较高的一致性.

表 1 不同指标下 GA 和 CE 算法 PID 参数优化结果比较

	GA	CE
MSE	0.025 1(3.505 1e-004)	0.024 7(5.345 2e-005)
IAE	16.940 6(1.625 1)	14.141 0(1.002 8)
ITAE	38.967 6(5.269 1)	24.924 4(3.241 1)

由表 1 可知, 在不同指标下会得到不同的优化结果. 这是因为各类指标的侧重点不同, 按照某一种指标寻优后得到的最优结果, 在另一种指标下, 不一定合适. 因此, 在实际中, 需要根据具体的控制要求选择合适的优化指标.

图 1 给出了在 ITAE 指标下, Ziegler-Nichols(ZN), GA 和 CE 的对比运行结果, 其中 CE 算法所得 PID 优化参数为 $K_\mathrm{p}=12.6720$, $K_\mathrm{i}=6.5439$, $K_\mathrm{d}=0.0013$. 如图 1 所示, CE 算法所得结果明显优于 GA 和 ZN 所得结果.

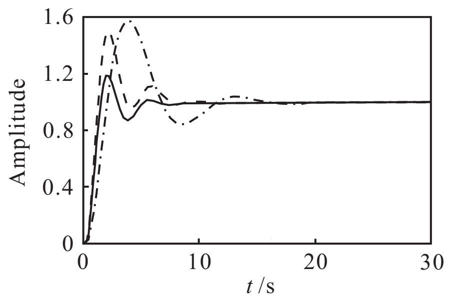

图 1 ZN, GA 和 CE 对比运行结果

CE 算法中抽样、排序等操作与 GA 算法中交叉、变异等操作之间的复杂度难以直接衡量对比, 因此本文从适应度函数的计算次数来衡量两种优化算法的计算复杂度. 在 CE 算法中适应度函数被计算 400×25 次, 而在 GA 算法中适应度函数被计算 100×500 次, 为 CE 算法的 5 倍. 因在多数实际问题中, 算法本身操作的时间如 CE 中抽样、排序等操作要远小于适应度函数的计算时间, 故在实际工程应用中 CE 较 GA 更具有潜力.

6 结 论

本文提出一种应用于 PID 控制器参数值优化的交叉熵算法. 交叉熵算法控制参数少、操作简单、计算复杂度低、鲁棒性强、易于并行化, 是一种高效的优化搜索方式. 通过对比基于遗传算法的 PID 控制器设计, 交叉熵优化算法的计算复杂度明显小于遗传算法, 同时具有更好的寻优能力. 总之, 交叉熵算法是一种适合 PID 控制器设计工程实践应用的高效算法, 值得在实际中推广应用.

参考文献(References)

[1] Iruthayarajan M, Willjuice Baskar S. Optimization of PID parameters using genetic algorithm and particle swarm optimization[C]. IET-UK Int Conf on Information and Communication Technology in Electrical Sciences, 2007: 81-86.

[2] Zdravko I Botev, Dirk P Kroese. The generalized cross entropy method, with applications to probability density estimation[J]. Methodology and Computing in Applied Probability, 2011, 13(1): 1-27.

[3] Zdravko I Botev, Dirk P Kroese. An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting[J]. Methodology and Computing in Applied Probability, 2008, 10(4): 471-505.

[4] Botev Z I, Kroese D P, Taimre T. Generalized cross-entropy methods with applications to rare-event simulation and optimization[J]. Simulation, 2007, 83: 785-806.

[5] 陈祥光. 遗传算法在PID控制器参数寻优中的应用研究[J]. 计算机仿真, 2001, 18(2): 30-32.
(Chen X G. Research of the application of genetic algorithm(GA) to parameters optimization of PID controller[J]. Computer Simulation, 2001, 18(2): 30-32.)

[6] 贺远华, 方彦军. 遗传算法在模糊控制器参数寻优中的应用研究[J]. 电力自动化设备, 2002, 22(12): 14-16.
(He Y H, Fang Y J. Application of genetic algorithm to parameter optimization of fuzzy controller[J]. Electric Power Automation Equipment[J]. 2002, 22(12): 14-16.)

[7] 赵亮, 付兴武, 徐广明. 基于遗传算法的 PID 控制及其 MATLAB 仿真[J]. 微计算机信息, 2004, 20(5): 14-16.
(Zhao L, Fu X W, Xu G M. PID control and its MATLAB simulation based on Genetic Algorithms[J]. Microcomputer Information, 2004, 20(5): 14-16.)

(下转第800页)