

基于遗传算法的PID控制系统

李玉刚

(南方医科大学, 广东 广州 510515)

摘要: 遗传算法是1962年由美国Michigan大学的Holland教授提出的模拟自然界遗传机制和生物进化论而形成的一种并行随机搜索最优化方法^[1-3], 在自动控制领域中得到了越来越广泛的应用。该文引入了“稳定区域算法”求取闭环系统稳定的PID控制器参数区间, 并以此算法的计算结果限定进化算法的参数寻优区间, 通过仿真试验取得了令人满意的控制效果。

关键词: PID控制; 遗传算法; 稳定区域算法

中图分类号: TP18 文献标识码: A 文章编号: 1009-3044(2009)09-2190-03

Based on Genetic Algorithms' PID Control System

LI Yu-gang

(Southern Medical University, Guangzhou 510515, China)

Abstract: Genetic Algorithms is proposed by the University of Michigan's Professor Holland in the United States in 1962 which is Analog nature of genetic mechanisms and biological theory of evolution and the formation of a parallel random search optimization method [1-3], in the field of automatic control has been more and more wide range of applications. In this paper, the introduction of the "stable region algorithm" to strike a closed-loop system stability of interval parameter PID controller, and the results of this algorithm for calculating the parameters of limited evolutionary algorithm optimization interval, through the simulation achieved satisfactory control effect.

Key words: PID control; genetic algorithm; the stability of regional algorithms

1 PID控制系统设计

1.1 常规的PID控制系统

常规的PID控制系统主要由PID控制器和被控对象组成。通常依据控制器输出与执行机构的对应关系, 将基本数字PID算法分为位置式PID和增量式PID两种。

基本PID控制器的理想算式为:

$$u(t) = K_p[e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{de(t)}{dt}] \quad (1)$$

式中: $u(t)$ ——控制器(也称调节器)的输出; $e(t)$ ——控制器的输入(常常是设定值与被控量之差, 即 $e(t)=r(t)-c(t)$); K_p ——控制器的比例放大系数; T_i ——控制器的积分时间; T_d ——控制器的微分时间。

对所设计的控制系统有两种界定其性能的指标——确定性指标和鲁棒性指标。前者是传统控制系统对性能的基本要求; 后者是指系统在不确定性扰动下具有保持某种性能指标不变的能力。传统PID控制系统包括位置式PID控制算法和增量式PID控制算法。在实际控制中, 增量式算法要比位置式算法应用更广泛。此外还有 Ziegler-Nichols临界比例度法和Cohen-Coon整定法。在实际工程应用中对控制性能要求较为苛刻的情况下, 此类基于经验设计的PID控制器不能很好的满足要求, 因此需要引入更为优秀的PID控制器设计方法

1.2 基于遗传算法的PID控制系统设计

遗传算法模拟了生命的进化机制, 在人工系统中实现了特定目标的优化, 具有天然的增强式学习能力, 且算法简单、鲁棒性强, 特别适用于处理传统搜索算法难以理解的复杂和非线性问题, 近年来在自动控制领域得到了十分广泛的应用。不论是系统的动态性能还是其鲁棒性, 经过遗传算法设计的PID控制器都远远超出了通过经典方法设计的控制器所能达到的性能。如果投入实际的工程应用定将会得到令人满意的控制性能。

2 PID控制器在双容水箱监控系统中的应用

2.1 双容水箱系统的数学建模与辨识

双容水箱平台的监控系统的监控界面包含了三个功能模块: 系统辨识与建模模块、自适应遗传算法PID控制模块和二自由度PID控制模块。本试验所使用的双容水箱图解模型如图1所示。

图中2为电阻加热丝, 1为保持双容水箱体内温度均匀的高功率电扇。箱体外壁为由高性能绝热材料组成的绝热层。实验中双容水箱箱体的温度总高于外界温度, 所以存在从恒温箱体内至体外的热传递。双容水箱箱体内部的热量由处于箱体顶部的电阻加热丝提供。在单位时间内流入恒温箱体内的能量与流出恒温箱体内的能量之差等于双容水箱箱体内存储能量的变化量, 可得到如下等式:

$$C \frac{d(T-T_0)}{dt} = Q_i - HA(T-T_0) \quad (2)$$

式中 C 为双容水箱比热容(KJ/K), T 为箱体内被控温度(K), T_0 为水箱周围的环境温度(K), Q_i 为单位时间内电阻丝产生的热量(KJ/s), H 为箱体内部的传热系数(KJ/s·m²·K), A 为温箱传热面积(m²)。等式的左边为单位时间内温箱体内能量的变化, 等式右边第一项 Q_i 为单位时间内流入温箱的能量, 第二部分 $HA(T-T_0)$ 为单位时间内流出温箱的能量。在实际的环境中存在外界温度变化引起的扰动, 但这些温度变化相对于要控制的高温系统来说可以忽略不计。将上式变为如下式(3)形式:

$$C \frac{dT}{dt} + HAT = Q_i + HAT_0 \quad (3)$$

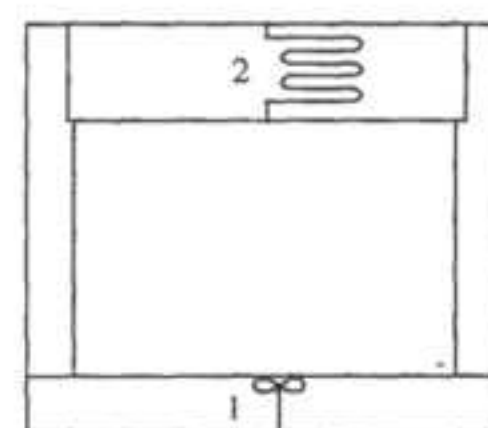


图1 温箱图解模型

此即为双容水箱的数学模型。当双容水箱处于稳定时,有 $dT/dt=0$,箱体内温度 T 与电阻丝产生的热量 Q_i 都达到稳态值: $T=T'$, $Q_i=Q'_i$,则式(2)变为:

$$Q'_i=HA(T'-T_0)$$

当双容水箱处于加热过程中时,有:

$$T=T'+\Delta T, Q_i=Q'_i+\Delta Q$$

将上述三式代入式(3),可得:

$$C \frac{d\Delta T}{dt} + HA\Delta T = \Delta Q$$

此即为式(3)对应的微分方程,若令 $C/HA=\tau$, $1/HA=K$,则可得双容水箱内被控的温度变化量 ΔT 与控制量 ΔQ 之间的传递函数为(s 为时间变量):

$$G(s) = \Delta T / \Delta Q = K / \tau s + 1 \quad (4)$$

式中, K 为恒温箱比例系数, τ 为时间常数。

2.2 双容水箱监控系统软件实现

软件部分包括三个功能模块,系统辨识与建模模块、遗传算法PID控制模块、二自由度PID控制模块。工作流程如图2所示。

模块1是对双容水箱系统进行分析,得出部分参数取值范围,再根据输入输出数据,由系统辨识方法来确定模型结构及参数;模块2通过自适应遗传算法来优化整定PID控制参数,达到好的控制效果;模块3通过改进的遗传算法来对二自由度PID控制器的两组控制参数进行整定。上位机监控界面采用VC++编写,软件中由于用到了曲线拟合、遗传算法等计算较为复杂的模型,且需要进行大量的矩阵运算,在设计过程中通过MATLAB外部程序接口实现了MATLAB功能的调用。

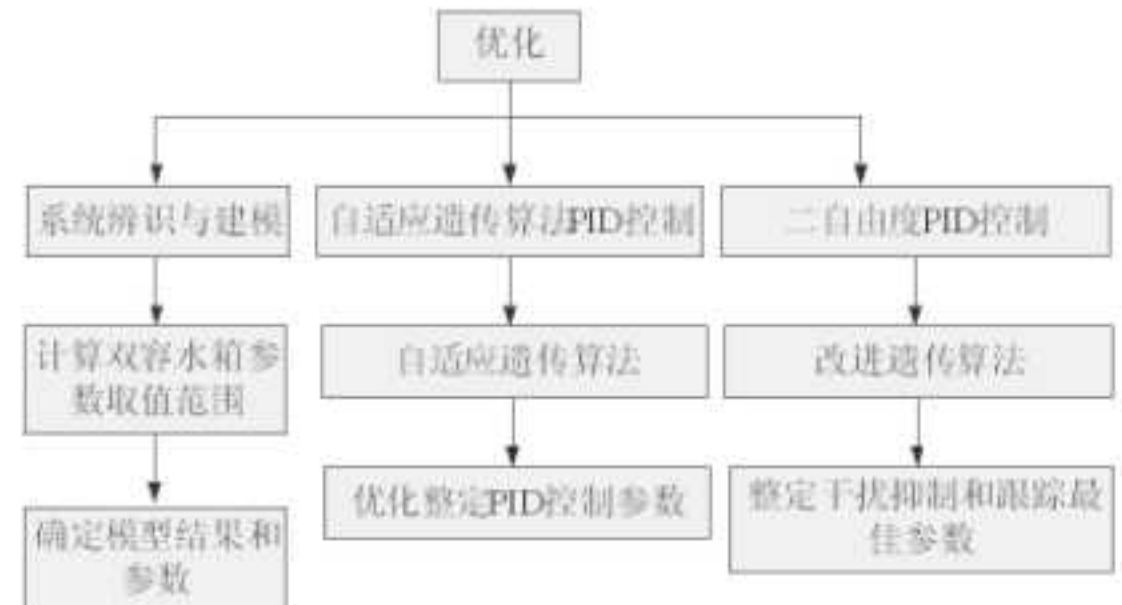


图2 双容水箱监控系统软件工作流程图

2.2.1 MATLAB的外部程序接口

MATLAB的强大矩阵处理功能和图画功能是十分优越的,其他程序设计语言如C、C++等执行速度则比MATLAB快,把它们有机结合能很好地发挥各自优点。

MATLAB外部程序接口主要包含两个部分——MATLAB C (C++) Math Library, MATLAB C (C++) 和 MATLAB Compiler。在VC++中实现MATLAB功能的调用主要有三种方式——通过MATLAB引擎实现与VC的互连、利用MATLAB C (C++) Math Library以动态链接库的形式提供对函数的调用、通过MATLAB的MEX文件实现。结合实际开发条件,本程序中采用的是第二种方式,直接在VC++的编译环境中调用MATLAB的C++函数库。

2.2.2 ADO、OLE DB及数据库访问的软件实现

软件中采用ADO技术访问SQL Server 2000数据库。^[4-6]

数据库的软件实现主要包括以下几个步骤:

1) 初始化OLE/COM库环境

```
CoInitialize(NULL); //初始化OLE/COM库环境
```

程序最后要调用

```
CoUninitialize(); //释放程序占用的COM资源
```

2) 引入ADO库文件

使用ADO前必须在工程的stdafx.h文件最后用直接引入符号#import引入ADO库文件,以使编译器能正确编译。代码如下:

```
#import "c:\Program Files\Common Files\System\ADO\msado15.dll" no_namespace rename("EOF", "EndOfFile")
```

3) 利用智能指针进行数据库操作

```
ConnectionPtr pConnection;
```

```
_RecordsetPtr pRecordset;
```

```
_variant_ t var; //定义变量
```

```
pConnection.CreateInstance(_uuidof(Connection)); //初始化Connection指针
```

```
pRecordset.CreateInstance(_uuidof(Recordset)); //初始化Recordset指针
```

4) 执行SQL命令并取得结果记录集

```
pConnection->Open("DSN=epssystem;uid=sa;pwd=123;", "", "", 0); //接字符串,连接数据库epssystem
```

```
pRecordset->Open("SELECT * FROM cost WHERE (时间 LIKE '%1')", pConnection.GetInterfacePtr(), adOpenDynamic, adLockOptimistic, adCmdText); //要执行的SQL语句,打开表cost,返回“时间 LIKE '%1'”,的记录集
```

5) 记录集的操作

根据要求读取或修改记录集。

```
var=pRecordset->GetCollect(); //读取数据
```

6) 断开连接

```
pConnection=NULL;
```

```
pRecordset=NULL;
```

2.3 双容水箱监控系统界面设计

双容水箱平台的监控系统包括一个软件启动界面和监控界面。监控界面包含了三个功能模块:系统辨识与建模模块、自适应遗传算法PID控制模块和二自由度PID控制模块。如图3所示。

当我们点击“参数设置”按钮时,会弹出图4所示的参数设置对话框要求我们设置相应的遗传算法的参数。

监控界面的左中部是两种PID控制器参数优化的功能界面。图4是自适应遗传算法PID控制器参数优化时的画面(交叉概率 P_c , 常规变异概率 P_m , 小步距变异概率 P_{mm}), 设置好相应的参数后, 点击“参数输入完成”, 然后点击监控界面右上方的“运行监控程序”, 便可得到如图5, Y轴为水箱高度, X轴为时间。

2.4 不同优化方法对应参数和指标比较

为了证明本优化方法的有效性, 选择工程上典型的Ziegler - Nichols法与本文遗传算法优化方法进行比较。本文双容水箱控制系统模型如下:

$$G(s) = \frac{3.81}{1+8.7s} e^{-2.81s}$$

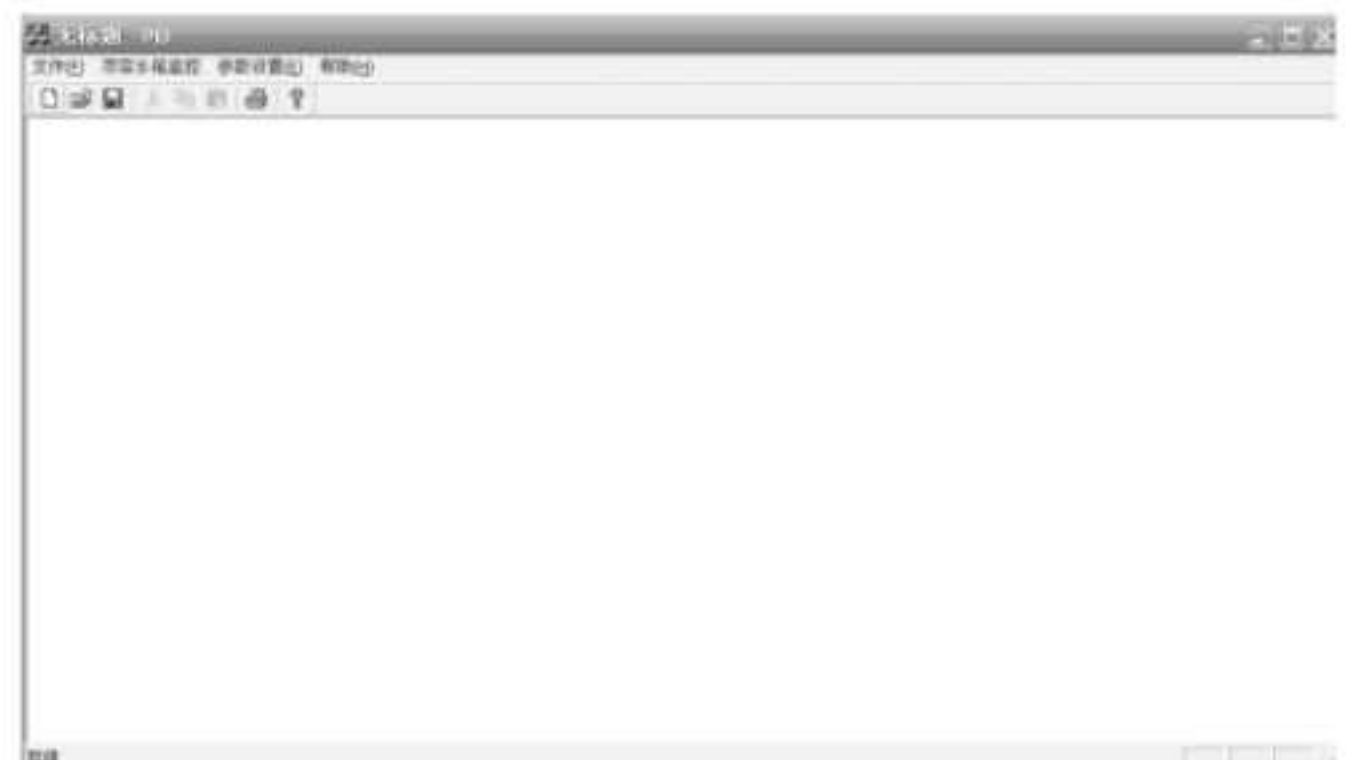


图3 监控界面



图4 参数设置界面

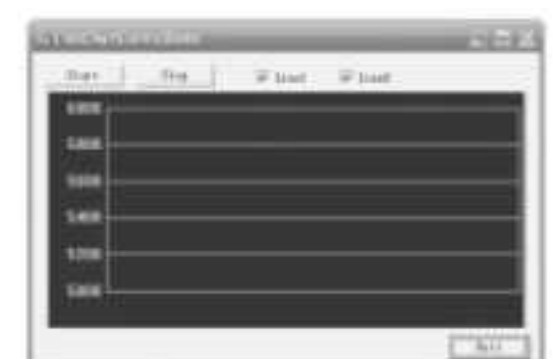


图5 水箱监控结果

三种方法所得结果比较如下(其中 K_p 为控制器的比例放大系数; T_i 为控制器的积分时间; T_d 为控制器的微分时间):

可见,应用本文提出的自适应PID遗传算法优化方法所进行的参数整定可以获得比Z-N法更好的性能指标。

3 总结与展望

本文使用了遗传算法进行PID控制器的设计,引入了控制系统的稳定区域算法计算出可使得控制系统稳定的PID参数稳定范围,以此为基础确定进化算法的参数寻优范围。研究结论在理论上验证了进化算法应用于PID控制系统设计的有效与优越性。但由于时间限制,系统的建模和控制过程不可避免地存在一些问题,如预测模型和控制参数优化的精度问题,以及建模与控制过程的离线/在线问题等。而且监控软件还只是调试版本,对系统运行环境的依赖性还很强,不能保证可移植性,这些都需要再在以后的实验中逐渐的扩充和完善。

参考文献:

- [1] Hung M L, Lin J S, Yan J J, et al. Optimal PID control design for synchronization of delayed discrete chaotic systems[M]. Chaos, Solitons & Fractals, In Press, Corrected Proof, Available online 26 January, 2007.
- [2] Huang L H, Sites J. Automatic control of a microwave heating process for in-package pasteurization of beef frankfurters[J]. Journal of Food Engineering, 2007, 80(1):226-233.
- [3] 邢青青,郝继飞,白文宝.存在常见非线性环节系统PID参数的整定[J].精密制造与自动化,2005(3):56-59.
- [4] 徐克林,卢坚德.液压系统模糊控制技术应用研究[J].内燃机,1997(2):11-14.
- [5] Goldberg D E. Genetic Algorithms in Search, Optimization and Machine learning[M]. Reading, Addison Wesley, 1989.
- [6] Srinivas M, Patnaik L M. Adaptive Probability of Crossover and Mutation in Genetic Algorithm[J]. IEEE Trans. on SMC, 1994, 24(4):656-667.

(上接第2181页)

作流的所有工作集合, $E(G_1)=\{e_1^1, e_1^2\}$, $e_1^1=\langle VA_1, VB_1 \rangle$, $e_1^2=\langle VB_1, VC_1 \rangle$, 表示工作的执行活动, 即任务。有向边上的权值集合 $W(G_1)=\{t_1^1, t_1^2\}$ 表示每一个任务预测的执行时间, 如 $t_1^1=3$ 表示任务 e_1^1 的预测执行时间为3个时间单位。 $V(G_2)=\{VA_2, VB_2, VC_2, VD_2\}$ 是 G_2 的顶点集合, 表示工作流的所有工作集合, $E(G_2)=\{e_2^1, e_2^2, e_2^3, e_2^4\}$, $e_2^1=\langle VA_2, VB_2 \rangle$, $e_2^2=\langle VA_2, VC_2 \rangle$, $e_2^3=\langle VB_2, VD_2 \rangle$, $e_2^4=\langle VC_2, VD_2 \rangle$ 。权值集合 $W(G_2)=\{t_2^1, t_2^2, t_2^3, t_2^4\}$ 。

我们把图中的事件结点分为两类: 1) 复杂工作结点, 它有多个前趋结点或者多个后继结点; 2) 简单工作结点, 只有一个前趋或一个后继结点。对于复杂事件结点 VD_2 来说, 它的执行需等待 VB_2, VC_2 均执行完毕后传递必要的结果才能开始执行。

在网格工作流调度中的一个重要的任务是怎样安排DAG图中的顶点和边的优先级。一旦顶点和边的优先级确定, 任务的调度就要考虑以下两个问题来缩短整个工作流的执行时间: 1) 怎样让这些优先级相同的任务去在多个资源上并行执行; 2) 怎样让CP(关键路径)上的任务执行时间尽可能缩短。

算法描述:

- 1) 利用关键路径推算出每个工作流的大约执行时间;
- 2) 根据用户提交工作流的时间和用户要求计算出工作流中每个任务的最后截止时间;
- 3) 根据工作流的最后截止时间和是否为关键路径的任务确定每个任务的优先级;
- 4) 所有的任务按照被赋予的优先级的大小降序放在任务表中;
- 5) 一旦任务要求被处理, 高优先级的任务被首先被执行。在任务被执行时选择从GIS中性价比较高的资源, 直到任务表无任务执行。

为了解决饿死的问题, 任务的优先级值随着时间的消耗, 进一步增加。将低优先级任务提升为高优先级, 再根据上面的定义规则, 进行处理。当某一任务的执行时间超过最后截止时间时, 查找出该任务所在的工作流中的全部任务并从队列中删除, 返回该工作流调度失败信息。

4 结论

把网格工作流抽象为DAG图, 根据每个工作流上任务的执行时间找到关键路径, 利用关键路径来计算每个任务的优先级, 把优先级高的任务分配到性价比较高的资源上。实验结果表明, 基于网格工作流的列表启发式算法可以保证优先级高的任务在截至时间内完成, 利用时间较短, 又能充分利用网格中的资源, 具有较好的性能。

参考文献:

- [1] Ian Foster, Carl Kesselman. The Grid 2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publisher, 2003.
- [2] Graham G E, Evans D, Bertram I. McRunjob: a high energy physics workflow planner for grid [A]. Conference on Computing in High Energy and Nuclear Physics, La Jolla, USA, 2003.
- [3] Jin Hyun Son, Myoung Ho Kim. Improving the performance of time-constrained workflow processing[J]. The Journal of System and Software, 2001, 211-219.
- [4] EunJoung Byuna, SungJin Choia, MaengSoon Baikb. MJSA Markov job scheduler based on availability in desktop grid[J]. Future Generation Computer Systems, 2007, 23:616-622.
- [5] Jia Yu, Rajkumar Buyya and Chen Khong Tham. Cost-based Scheduling of Scientific Workflow Applications on Utility Grids [J]. The Journal of System and Software, 2005:233, 236-242.