

文章编号:1007-757X(2017)10-0058-05

基于数据驱动的 PID 自整定算法研究

高菲

(宝鸡职业技术学院, 721000)

摘要: 传统的 PID 整定技术以给定试验信号为基础,不能很好的利用这些历史数据,通过在介绍和分析经典 PID 整定技术的基础之上,利用历史数据完成数据驱动辨识方法研究,以灵敏度为性能指标研究 PID 自整定算法,并通过 OPC 接口采集数据完成基于数据驱动的 PID 自整定软件实现。

关键词: 数据驱动; PID 自整定; 系统辨识; 灵敏度; OPC; 自整定软件

中图分类号: TG409

文献标志码: A

Research on Data-driven PID Self-tuning Algorithm

Gao Fei

(Baoji professional technology institute, 721000)

Abstract: The traditional PID setting technology is based on given test signals, and can not make good use of these historical data. In this paper, based on the introduction and analysis of the classical PID setting technology, the historical data are used to complete a data-driven identification method research. The PID self-tuning algorithm is studied with the sensitivity as the performance index. And data is collected through the OPC interface to complete the data-driven PID self-tuning software implementation.

Key words: Data-driven; PID self-tuning; System identification; Sensitivity; OPC; Self-tuning software

0 引言

随时工业生产工艺越来越复杂,工业生产环境越来越恶劣等因素的影响,对 PID 控制器的控制要求提出了更多更复杂的要求,希望 PID 控制器的控制方式更加准确与全自动化操作^[1]。在实际的工业应用中,只有少部分的 PID 控制器能够比较满意的运行,其主要原因是 PID 参数的整定工作没有做好,以传统的 PID 自整定技术需外加试验信号来获取过程模型,如阶跃响应和频率响应等,这种做法一般只适应于在系统运行之初和系统停机调试的情况,当系统正在运行时这种方法往往不能表现了出色^[2]。

在线运行的系统,操作员频繁操作和长时间运行下会积累大量的历史数据,如何通过这些数据进行 PID 自整定研究而不是通过实验的方法来获得过程模型,通过灵敏度作为性能指标既能提高系统的鲁棒性能又能以比较好的方式进行 PID 参数自整定^[3]。本文主要研究了基于数据驱动的 PID 自整定算法及软件实现,利用历史数据完成数据驱动辨识方法研究,以灵敏度为性能指标研究 PID 自整定算法,并通过 OPC 接口采集数据完成基于数据驱动的 PID 自整定软件实现。

1 基于灵敏度的 PID 自整定方法

在工业过程控制的发展中,PID 自整定技术实现了人们从手动调节到计算机计算调节的解脱^[4]。大多数工业过程

处于高温、高压等恶劣的环境中,而处于现场的一些控制器常常需要操作员去现场亲自动手操作,这样的操作往往是粗放的,现场的 PID 控制异常复杂^[5]。面对向火电厂发电机组这样复杂的控制对象,ZN 整定法相对来说效果不那么明显,基于灵敏度的 PID 自整定算法能够更好的表现出复杂控制对象的更多特性,可以更好的完成火电机组的回路优化工作^[6]。

设 $G_1(s) = G_c(s)G_p(s)$, 则灵敏度的定义为式(1)。

$$M_s = \max_{0 \leq \omega \leq \infty} \left| \frac{1}{1 + G_c(j\omega)G_p(j\omega)} \right| \quad (1)$$

如图 1 所示。

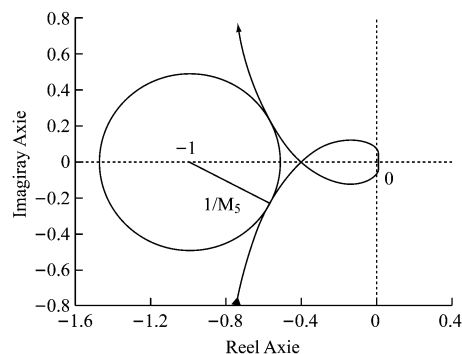


图 1 灵敏度控制器设计方法的几何表示

灵敏度的约束条件是 $G_1(j\omega)$ 的 Nyquist 曲线不能进入

作者简介:高菲(1981-),女,陕西人,硕士研究生,研究方向:计算机教育。

到以临界点 -1 为圆心,以 $1/M_s$ 为半径的圆内,其幅值裕度和相位裕度有如式(2)、(3)。

$$A_m > \frac{M_s}{M_s - 1} \quad (2)$$

$$\varphi_m > 2\arcsin \frac{1}{2M_s} \quad (3)$$

其中, M 一般取 $1.3 \sim 2.0$ 。如取 M_s 为 1.5 则有 $A_m \geq 3.00$ 和 $\varphi_m \geq 38.94^\circ$,相当于在某种程度上同时满足幅值裕度和相位裕度。

给定任一个灵敏度 M_s 值,则对应闭环传递函数 $G_1'(s) = G_c(s)G_p'(s)$ 的 Nyquist 图在相切点与所应满足的条件为式(4)、(5)。

$$G_1(j\omega) = -1 + \frac{1}{M_s}e^{-j\theta} \quad (4)$$

$$\arg \frac{dG_1'(j\omega)}{d\omega} = \frac{\pi}{2} - \theta \quad (5)$$

设 PID 控制器的传递函数可以写为式(6)。

$$G_c(s) = k \frac{(As^2 + Bs + C)}{s} \quad (6)$$

其中, $A = K_d/k, B = K_p/k, C = K_i/k$,设 $A = a, B = b, C = c$,通过前面的公式分析得出式(7)。

$$G_1'(s) = G_c(s)G_p'(s) = \frac{ke^{-sL}}{s} \quad (7)$$

这样便得出了 $G_1'(s)$ 的解析公式,通过前面 $G_1'(s)$ 所满足的幅值和相位条件,利用相应解方程的方法,便可以近似得出 k 的近似式为式(8)。

$$k = \frac{1}{L} \left(1.451 - \frac{1.508}{M_s} \right) \quad (8)$$

则 PID 控制器的参数,可以表示为式(9)

$$\begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} = \frac{1}{L} \left(1.451 - \frac{1.508}{M_s} \right) \begin{bmatrix} b \\ c \\ a \end{bmatrix} \quad (9)$$

这样,PID 控制器的三个参数便可以通过间接的方式求出。当回路的给定值变化时,控制器通过相应的辨识算法便可以计算出二阶加纯滞后模型,通上上式的灵敏度算法便可以完成整个自整定过程。

当人为或者为可抗的因素引起的输入或输出发生变化时,上位机通过通讯接口采集到该过程产生的数据,由于上述方法采用了数据驱动的方式,所以这些数据会自动存档,通过上面的计算方法自动计算出近似的二阶模型,这样,设定灵敏度参数后,算法就会自动整定出满足条件的 PID3 个参数。

2 PID 自整定软件设计

PID 自整定过程通过一定的通讯接口,将下位机数据累积、上传然后在上位机进行分析、优化和决策,实现基于数据流驱动的 PID 自整定过程^[7]。这种 PID 参数自调节系统是一个具有一定自适应能力的系统,在过程数据改变的过程中能够通过前期的数据归档分析、辨识,利用一定的优化控制算法做出决策保证控制系统能够按某一个性能指标运行在最佳状态,一种基于 OPC 的数据驱动 PID 控制器参数自调

整原理图。如图 2 所示。

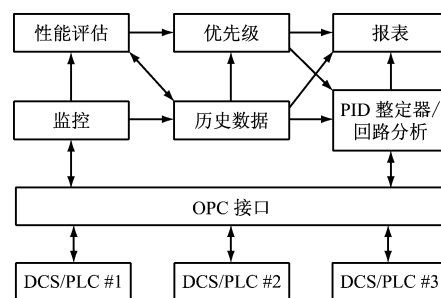


图 2 一种基于 OPC 的数据驱动 PID 控制器参数自调节结构图

基于数据驱动的 PID 自整定软件实施一般应遵循以下步骤:建立与现场控制系统通信(如 PLC、DCS)^[8];选定或新建需整定 PID 参数的回路;获取回路的 PID 参数和采样时间等;记录从此刻开始的现场数据;开始自整定过程;整定结束计算参数和过程模型;确认参数是否符合要求,下载到控制器。

因为一般的 PID 控制器分为自动和手动模式,这样就涉及到对 PID 整定时两种模式所采用的整定方法不同^[9]。一般情况下,选择自动模式,则直接对控制对象进行整定,而控制器的输入值不参与整定工作,此时调节控制器输入设定值是无效的;对于自动模式,则系统完全运行在闭环的情况下,这种模式下的 PID 参数自整定,对控制器设定值做出相当操作,观察系统响应,从而辨识过程模型,计算出 PID 控制器运行最优参数。

根据整个软件需满足的性能指标,整个软件需满足 3 个基本块功能:

(1)界面设计模块,应包括 P、I、D 三个参数、滤波系数以及采样时间 T_s 值的更新与设置,手动/自动切换按钮,自整定启动按钮,历史数据保存,模型辨识窗口,通讯设置窗口,趋势图框架等。

(2)通讯接口开发模块,主要是现场总线通讯协议的开发,本文利用 OPC 通讯接口,OPC 接口以 Server/Client 为主要通讯方式,OPC Server 通过一组一组的接口提供服务,在实际的架构上,OPC Server 共分为三层:分别是 OPCServer, OPCGroup 和 OPCItem 三种。

(3)自整定算法开发模块,自整定算法的开发是整个自整定软件实施的核心步骤,确保 PID 回路优化性能好坏的重要指标。

在程序运行前需要先开辟一个缓冲区,用来存放所有的被控变量的参数,利用此函数可以实现了 PID 的迭代计算,在需要时可以方便调用。

辨识与整定算法:整定算法包括两部分的内容,其一是辨识方法的设计,主要分为两大类方法,一种是基于阶跃响应和频率响应方法的辨识方法,这类方法用于比较简单的控制回路;另一种是基于数据驱动的辨识方法,利用这种方法一般需要先记录当前运行的一段过程数据,然后才能辨识出被控对象的过程模型,用于复杂回路的整定过程。

其二是参数整定方法,PID 参数自整定算法也同样分为两种,一种与基于阶跃和频率响应的辨识方法相对应,主要

利用经典 ZN 整定方法和继电反馈方法,同样用于比较简单的回路当中。另一种是基于灵敏度的 PID 自整定算法,用于复杂回路的整定工作。

3 基于灵敏度自整定算法设计

基于灵敏度的 PID 自整定算法,是建立以数据驱动为基础的,首先需要在程序运行时,启动归档功能,这样软件便可以记录下一段时间内的过程运行数据并保存到本地磁盘中,当需要进行自整定计算时,可以选择从保存下的任何一段数据进行辨识,得出辨识模型后系统便显示在软件上面。这样,通过基于灵敏度的 PID 自整定算法便能实现对 PID 参数的整定。

实际软件运行时,上位机软件通过 OPC 接口与下位机设备如 PLC、DCS 进行通信,获取下位机数据并以 TAG 的形式表现,选择需要进行监控的过程值,上位机会显示出选择的过程值的曲线,一般选择 SP、PV 和 CV 3 个参数作为监控对象。上位面显示过程曲线的过程,操作员会根据实验曲线和当前设备运行状况判断是否应用 PID 参数自整定功能,确定应用后的 PID 参数通过 OPC 接口写入功能以 TAG 形式传送到下位机执行的。其软件实现流程如图 3 所示。

在建立好回路与 OPC 通讯设置完成后,进入到自整定软件的操作,此时当前运行的 PID 回路参数便可以通过 OPC 采集上来,显示在当前值栏中。这样如果需运行自整定

操作,在选择好辨识方法和参数整定计算后,自整定计算便可以计算出新的 PID 参数显示在整定值,同时自整定曲线也会画出供操作员判断是否下载到 PID 回路运行,如图 4 所示。

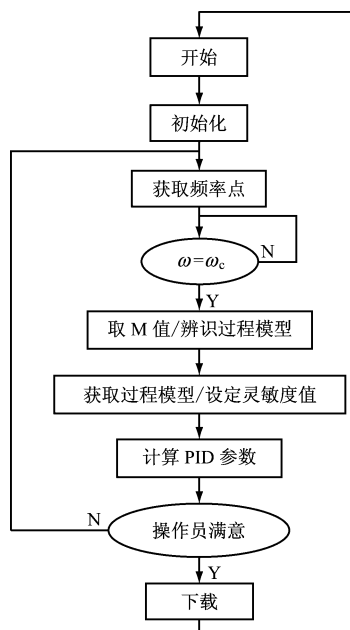


图 3 灵敏度自整定算法执行流程图



图 4 PID 自整定软件面板

4 系统整体仿真实现

选择罗克韦尔 RsLogix5000 软件内 PID 功能块组成一个小型的纯滞后 PID 回路,其中 PID 为 PID 仿真控制器,用于仿真过程控制中的 PID 控制器,DEDT 为仿真死区时间,用于仿真实过程中如压力等,在一定范围内不发生变化的过程;LDLT 为仿真滞后时间,用于温度等大滞后对象的仿真模拟。这里设置 PID 设定值为 30,滞后时间为 40 s,死区时间为 5 s。其回路功能程序如图 4 所示,在设置好 OPC 通讯接口,选择需要获取和整定的值,进入整定界面。在整定开始前通过前面的通讯设置,在进入到整定界面后会出现从

现场采集的过程值的曲线,需要进行自整定时,首选在操作模式中选择手动模式,将整个 PID 回路切换到手动模式,这时修改参考输入值对回路无影响。手动状态下,通过手动给定控制输出,然后选择辨识和整定方法后,按下自整定按钮,这时系统启动自整定过程,软件会收集历史运行过程中从 0 ~ ω_c 个重要的频率特性通过迭代的方法完成过程模型的辨识。

整定过程,软件界面会显示实时运行曲线和整定曲线,通过设定灵敏度参数,软件会自动运行整定计算,同时 PID 参数栏里也会显示整定前和整定后 PID 参数,操作员可以根据辨识模型、整定参数和整定曲线选择是否下载整定数据到

控制器中去,如果整定过程出现不稳定或者振荡等情况,则软件会提示整定过程不成功需重新选择参数进行整定。软件同时提供归档功能,这样可以方便保存运行过程较好曲线的数据,为以后再次进行辨识整定和优化提供参考。

可以看出在 PID 自整定功能运行前,PID 控制器的设定值为 30,当前运行的 PID 参数分别为 $K_p=0.4$, $K_i=0.096$, $K_d=0.0$,启动自整定过程后,选择灵敏度 PID 参数自整定的方法,设定灵敏度参数,当启动自整定过程时,会出现历史数据的选择窗口或者在线记录数据。选择相应的历史数据时,系统会对这段保存或运行的过程数据做出分析,这需要一个过程,当分析得到的数据能够令过程满意时,便可以结束自整定过程,这个时候软件已经通过 OPC 接口与下位机

进行了大量的数据交换工作,通过数据驱动技术在软件辨识方法的基础上便可以生成新的 PID 参数,当然这个需要操作员判断该过程得到的 PID 参数是否满足要求,如果满足要求可以通过下载按钮,将 PID 参数下载到控制器当中去,这里允许操作员对辨识和整定出的 PID 参数做修改,比如可以加大一些 P 参数,这样整定出来的控制器会有更快的响应速度,如图 5 所示。

通过自整定过程运行,软件得到的 PID 参数已经显示在软件的界面上,分别为 $K_p=0.42$, $K_i=0.60$, $K_d=0.0$,通过观察曲线,也可以看出这组参数能够良好的跟踪输入,表明整定效果达到了预期的目标。整个软件的运行测试仿真效果如图 6 所示。

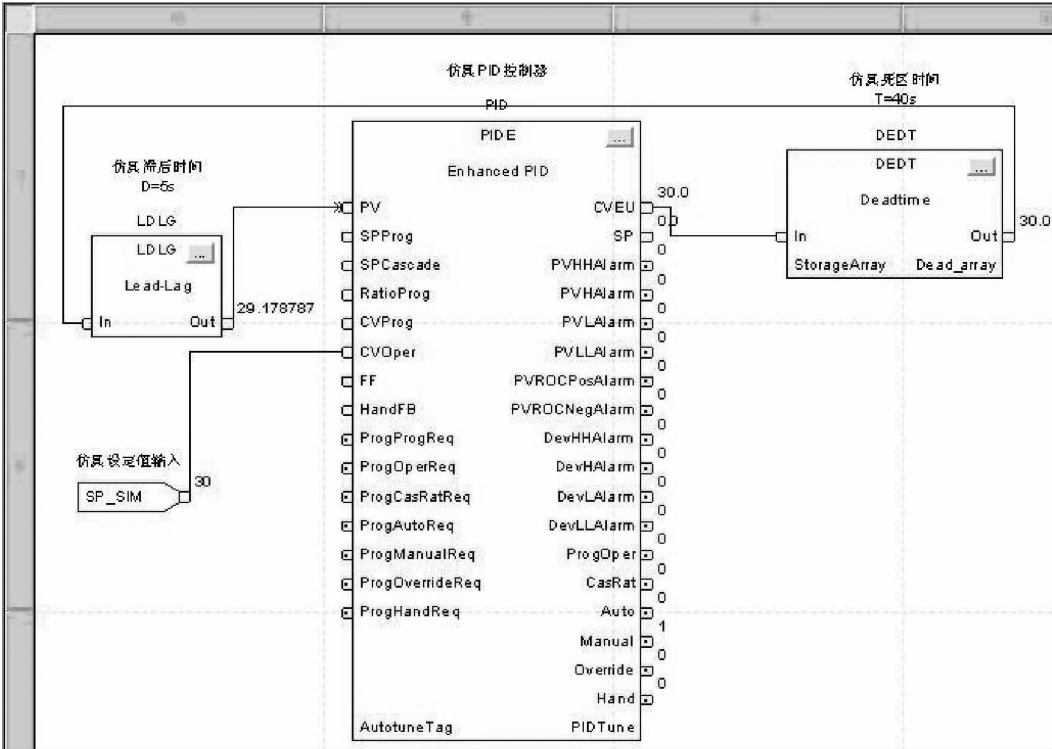


图 5 罗克韦尔功能块程序图

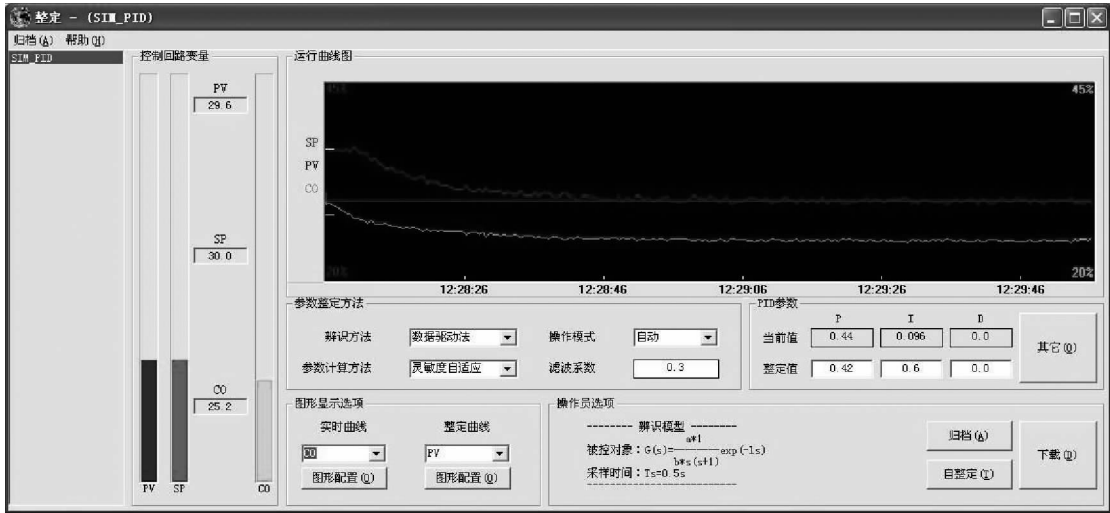


图 6 软件运行测试图

5 总结

PID 自整定技术的研究是一个持续不断的过程,在过去的一些研究成果的基础上,通过软件的形式做出简单的尝试,验证自整定算法的可行性,结合现在比较流行的网络环境,在现场总线技术的架构下,通过总线接口设计,上位面与下位机功能的匹配,设计了基于数据驱动的 PID 自整定软件,在实际的编写过程中利用基于灵敏度的完成 PID 参数自整定,复杂算法的应用最终还是为了解决多参数与多变量带给 PID 自整定过程中的麻烦。通过最后的软件测试,达到了良好的效果。

参考文献

- [1] 胡晚霞,余玲玲,戴义保等. PID 控制器参数快速整定的新方法[J]. 工业仪表与自动化装置, 1996(5): 11-16.
- [2] 赵江东,李娟,马小陆. 基于 RBF 神经网络动态辨识的自整定 PID 控制策略[J]. 工业仪表与自动化装置, 2009(6): 12-15.

- [3] 王亚刚,许晓鸣. 自适应鲁棒最优 PI 控制器[J]. 自动化学报, 2009, 35(10): 1352-1356.
- [4] 黄友锐. 基于遗传神经网络的自整定 PID 控制器[J]. 系统仿真学报, 2003, 15(11): 1628-1620.
- [5] 王介生,王金城,王伟. 基于粒子群算法的 PID 控制器参数自整定[J]. 控制与决策, 2005, 20(1): 73-76.
- [6] 吕宁,姬秋云,夏增刚. 基于混沌理论的 PID 参数自整定锅炉汽包水位控制系统设计[J]. 热能动力工程, 2007, 22(3): 314-316.
- [7] Ziegler J G, Nichols N B. Optimum setting for automatic controllers. Trans [J]. ASME. 1942 (64): 759-768.
- [8] Astrom K J, Hagglund T. Automatic tuning of simple regulators with specifications on phase and amplitude margins[J]. Automatica, 1984, 20(5): 645-651.
- [9] 柴天佑,张贵军. 基于给定的相角裕度和幅值裕度 PID 参数自整定新方法[J]. 自动化学报, 1997, 23(2): 167-172.

(收稿日期: 2017. 04. 25)

(上接第 57 页)

联网发展的必然趋势,对互联网和通信领域都会产生积极的推动作用。云计算融合了物联网后,增加了实用性,将会极大促进各个领域的发展。

参考文献

- [1] Alessio Botta, Walter de Donato, Valerio Persico, etc. On the Integration of Cloud Computing and Internet of Things [J]. 2014 International Conference on Future Internet of Things and Cloud. 2014 (14): 23-25.
- [2] Jun Kang, Siqing Yin and Wenjun Meng. An Intelligent Storage Management System Based on Cloud Computing and Internet of Things[J]. Advances in Intelligent Systems and Computing. 2014 (255):

501-502.

- [3] Mohammad Aazam, Imran Khan, Aymen Abdullah Alsaffar, etc. Cloud of Things: Integrating Internet of Things and Cloud Computing and the Issues Involved [J]. Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IB-CAST). 2014(1):414-415.
- [4] Buchholz T, Axel K, Schiffrers M. Quality of Context: What It Is And Why We Need It[J]. 2003(1): 1-14.
- [5] Rveys I S U. A Survey of Quality of Service in Mobile Computing Environments[J]. 1999(4):2-10.

(收稿日期: 2017. 03. 21)