

The Research of Control System Simulation Technology Based on VC

Tao Geng, Qidan Xiang

Academy of Physics and Electroics, Henan University, Kaifeng
Email: gengtao@henu.edu.cn

Received: Jun. 12th, 2014; revised: Jul. 10th, 2014; accepted: Jul. 18th, 2014

Copyright © 2014 by authors and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper has a detailed analysis of the basic principle of the PID controller algorithm, especially the incremental digital PID controller. Starting from the analysis of its basic principle, the paper expounds the process of the implementation of the incremental PID controller algorithm; this paper also adopts an approximate algorithm of inverse Laplace transform—Stehfest algorithm, which converts S domain Laplace expressions $F(s)$ of time domain $f(t)$ in an approximate method. Finally the MSChart control will display the image of $f(t)$, completing simulation of PID control system. According to the experiment, the program has achieved satisfactory results in the end.

Keywords

VC, PID, Stehfest Algorithm, MSChart Control

基于VC的控制系统仿真实现 技术研究

耿 涛, 相奇丹

河南大学物理与电子学院, 开封
Email: gengtao@henu.edu.cn

收稿日期: 2014年6月12日; 修回日期: 2014年7月10日; 录用日期: 2014年7月18日

摘要

本文首先详细分析了PID控制器算法的基本原理，尤其是数字式PID控制器中的增量式PID控制器，论文从分析其基本原理出发，详细阐述了增量式PID控制器算法的实现过程；本文还采用一种拉普拉斯反变换的近似算法：Stehfest算法，将S域的拉普拉斯表达式 $F(s)$ 近似转换为时域的函数表达式 $f(t)$ ，最后采用MSChart控件将 $f(t)$ 显示出来，完成PID控制系统的仿真，经实验，最终取得满意的结果。

关键词

VC, PID, Stehfest算法, MSChart控件

1. 引言

从 20 世纪 30 年代以来，生产自动化技术取得巨大成就，工业自动化涉及范围很广，过程控制是其中最重要的一个分支，工业生产对过程控制的要求是稳定性、安全性和经济性。过程控制的任务是在了解、掌握工艺流程和生产过程的静态和动态特性的基础上，根据上述三项要求，应用理论对控制系统进行分析和综合，最后采用合适的技术手段加以实现。在很多情况下，只需采用商品化的常规调节器进行 PID 控制就可以实现其控制任务。尽管在控制领域，不断涌现出各种新型控制器，但 PID 控制器还是以其结构简单、易实现、鲁棒性强等优点，在控制领域中处于主导地位，是工业过程控制的基础。根据日本有关调查资料显示，现今使用的各种控制技术中，PID 技术占 84.5%，优化 PID 控制技术占 6.8%，现代控制技术占 1.6%，人工智能技术占 0.6%。如果把优化 PID 控制技术和 PID 控制技术加起来，那么 PID 技术占到整个控制技术的 90% 以上，有资料也指出，工业过程控制中，95% 以上的回路具有 PID 结构[1]。所以，可以毫不夸张地说，随着现代工业化和其他各种先进控制技术的发展，PID 控制技术仍然不过时，而且还站着主导地位。

本文介绍了基于 Microsoft Visual Studio 2010 平台的 VC 的 PID 控制模块设计。可用于控制系统的仿真，控制器代码可以直接移植于目标系统。

2. 使用须知

纸型

PID 控制器由比例单元(P)、积分单元(I)和微分单元(D)组成，是具有比例 - 积分 - 微分控制规律的控制[2]。当系统是连续控制时，其输入 $e(t)$ 与输出 $u(t)$ 的关系为

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

相应的传递函数为

$$G(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = \frac{K_p}{T_i} \times \frac{T_i s^2 + T_i s + 1}{s} \quad (2)$$

其中 K_p 为比例系数； T_i 为积分时间常数； T_d 为微分时间常数。

使用中只需设定三个参数(K_p , T_i 和 T_d)即可。在很多情况下，并不一定需要全部三个单元，可以取其中的一到两个单元，但比例控制单元是必不可少的。各种控制作用(即比例作用、微分作用和积分作用)

的实现表达式(1)、(2)中表达得很清楚,相应的控制器参数包括比例系数 K_p , 积分时间常数 T_i , 微分时间常数 T_d 。

1) 比例控制

比例控制是一种最简单的控制方式,可以及时地反应控制系统的偏差信号,一旦系统出现了偏差,比例调节作用即产生调节作用,减小系统偏差的变化。具有比例控制规律的控制器,称为 P 控制器[3]。P 控制器实质上是一个具有可调增益的放大器,在信号变换过程中,P 控制器只改变信号增益而不影响其相位。在串联校正中,加大控制器增益,可以提高系统的开环增益,减小系统稳态误差,从而提高系统的控制精度,但会降低系统的相对稳定性,甚至可能造成闭环系统不稳定。因此,在系统校正设计中,很少单独使用比例控制规律[4]。

2) 积分控制

具有积分控制规律的控制器,称为 I 控制。I 控制器的输出信号 $u(t)$ 与其输入信号 $e(t)$ 的积分成正比,即

$$u(t) = K_i \int_0^t e(t) dt \quad (3)$$

其中, K_i 为可调系数。由于 I 控制器的积分作用,当其输入 $e(t)$ 消失后,输出信号 $u(t)$ 有可能是一个不为零的常量。

在串联校正时,采用 I 控制器可以提高系统的型别,有利于系统稳定性的提高,但积分控制使系统增加了一个位于原点的开环极点,使信号产生 90° 的相角滞后,不利于系统的稳定性[5]。如果 I 控制器和之前提到的 P 控制器结合起来,便构成了具有比例 - 积分控制规律的控制器,称 PI 控制器,其输出信号 $u(t)$ 同时成比例的反应输入信号 $e(t)$ 及其积分:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (4)$$

其中, K_p 为可调比例系数; T_i 为可调积分时间常数。在串联校正时,PI 控制器相当于在系统中增加了一个位于原点的开环极点,同时也增加了一个位于 s 左半平面的开环零点。位于原点的极点可以提高系统的型别,以消除或减小系统的稳态误差,改善系统的稳态性能;而增加的负实零点则用来减小系统的阻尼程度,缓和 PI 控制器极点对系统稳定性及动态过程产生的不利影响[6]。只要积分时间常数 T_i 足够大,PI 控制器对系统稳定性的不利影响可大为减弱。在控制工程实践中,PI 控制器主要用来改善控制系统的稳态性能。

3) 微分控制

微分控制器的输出与输入误差信号的微分(即误差的变化率)成正比关系,可以改善控制系统的响应速度和稳定性,微分作用能预见偏差变化的趋势,并能在偏差信号变得太大之前,在系统中引入一个有效的早期修正信号,反应系统偏差变化的规律,能产生超前的控制作用,从而加快系统的动作速度,减少调节时间。简单来说,微分作用能在偏差还没有形成之前,就可以消除误差,改善系统的动态性能[7]。

微分作用取决于微分时间常数 T_d , T_d 越大,控制器的微分作用表现得就越强, T_d 偏大时,超调量会较大,调节时间也会较长; T_d 越小,控制器的微分作用越弱, T_d 偏小时,超调量也较大,相应调节时间也会较长; T_d 调节合适时,系统的超调量和调节时间可以有效的减小。微分作用反映的是输入信号的变化率,当偏差没有变化时,微分作用的输出为零。此外,从滤波器角度来说,微分控制器相当于一个高通滤波器,对噪声干扰有放大作用,这对系统是非常不利的,因此我们就不能过多的增加微分调节[8] [9]。

3. 数字 PID 控制器

PID 控制器是控制系统中应用最为广泛的一种控制器，广泛应用于工业过程控制中。随着微型计算机，特别是现代嵌入式位处理器的普及应用，原先由硬件实现的 PID 控制器的功能都可以用软件来代替实现，形成数值 PID 算法，实现由模拟 PID 控制器到数字 PID 控制器的转变。

在实际应用中，数字 PID 控制器分为两种：位置式 PID 控制器和增量式 PID 控制器。

3.1. 位置式 PID 控制器

计算机实际上是采样控制，它根据采样时刻的偏差值计算控制量。所以式(1)中的积分项和微分项不能直接使用，需要进行离散化处理。具体实现方法是：以 K 作为采样信号， T 作为采样周期，则连续时间对应离散化的采样时间 KT ，用求和的形式代替积分，用增量的形式代替微分，则可作如下的近似变换：

$$\left\{ \begin{array}{l} t \approx KT \quad (K = 0, 1, 2, \dots) \\ \int_0^t e(t) dt \approx T \sum_{j=1}^K e(jT) = T \sum_{j=0}^K e_j \\ \frac{de(t)}{dt} \approx \frac{e(KT) - e[(K-1)T]}{T} = \frac{e_K - e_{K-1}}{T} \end{array} \right. \quad (5)$$

将式(5)代入式(2)中，则得到离散的 PID 控制器表达式为：

$$u_k = K_p \left[e_k + \frac{T}{T_j} \sum_{j=0}^K e_j + \frac{T_D}{T} (e_k - e_{k-1}) \right] + u_0 \quad (6)$$

或：

$$u_k = K_p e_k + K_I \sum_{j=0}^K e_j + K_D (e_k - e_{k-1}) + u_0 \quad (7)$$

式中：

K —— 采样序列号， $K = 0, 1, 2, \dots$ ；

u_k —— 第 K 次采样时刻计算机输出值；

e_k —— 第 K 次采样时刻的输入偏差值；

e_{k-1} —— 第 $K-1$ 次采样时刻的输入偏差值；

K_I —— 积分系数， $K_I = K_p \frac{T}{T_j}$ ；

K_D —— 微分系数， $K_D = K_p \frac{T_D}{T}$ ；

u_0 —— 开始进行 PID 控制时的计算机的输出值。

因此，如果采样周期足够小的话，那么由式(6)和(7)得到的近似结果可获得足够的精确度，此时连续的和离散的 PID 控制过程会十分接近。由于式(6)和(7)的控制算法是直接按式(1)给出的 PID 控制规律进行计算的，所以给出了全部控制量的大小，被称为全量式或位置式 PID 控制算法[10]。

由于位置式 PID 控制器每次输出全部控制量，因此它可以直接作用于执行机构。另一方面，由于这种算法全量输出，每次输出均与过去的状态有关，计算时要对 e_k 进行累加，容易造成积分饱和，计算机运算工作量很大；此外，因为计算机输出的 u_k 对应的是实际的执行机构的位置，如果计算机出现故障， u_k 大幅度变化，这将会引起执行机构位置的大幅度变化，可能造成重大生产安全事故。这就导致这种算法

不适用于生产实践中，催生了增量式 PID 控制器算法。

3.2. 增量式 PID 控制器

增量式 PID 控制器是指 PID 控制器每次输出的只是控制量的增量，当例如步进电机等执行机构，需要的是增量而不是位置量的绝对数值时，就可以使用增量式 PID 控制器进行控制。

由式(7)，根据递推原理可得：

$$u_{k-1} = K_p e_{k-1} + K_j \sum_{j=0}^{k-1} e_j + K_D (e_{k-1} - e_{k-2}) + u_0 \quad (8)$$

用式(8)减去式(7)可得：

$$\begin{aligned} \Delta u_k &= K_p (e_k - e_{k-1}) + K_I e_k + K_D (e_k - 2e_{k-1} + e_{k-2}) \\ &= K_p \Delta e_k + K_I e_k + K_D (\Delta e_k - \Delta e_{k-1}) \end{aligned} \quad (9)$$

式中， $\Delta e_k = e_k - e_{k-1}$ 。

式(9)称为增量式 PID 控制算法。将该式整理合并得到：

$$\Delta u_k = A e_k + B e_{k-1} + C e_{k-2} \quad (10)$$

式中，

$$\begin{aligned} A &= K_p \left(1 + \frac{T}{T_I} + \frac{T_d}{T} \right) \\ B &= -K_p \left(1 + \frac{2T_d}{T} \right) \\ C &= K_p \frac{T_d}{T} \end{aligned}$$

它们都是与采样周期、比例增益、积分时间常数和微分时间常数有关的常系数。

根据增量式 PID 算法可以推导出位置式 PID 算法的递推形式，其表达式为：

$$\begin{cases} e_k = r_k - y_k \\ \Delta u_k = K_p (e_k - e_{k-1}) + K_I e_k + K_D (e_k - 2e_{k-1} + e_{k-2}) \\ u_k = u_{k-1} + \Delta u_k \end{cases} \quad (11)$$

其中 r_k 为系统的输入， y_k 为控制对象的测量量。

位置式 PID 控制器实质上与增量式 PID 控制器是一样的，但是增量式 PID 控制器比位置式 PID 控制器有许多优越之处[11]：

- Δu_k 只与 K ， $K-1$ ， $K-2$ 时刻的偏差有关，节省运算时间和内存。
- 每次只做 Δu_k 计算，则计算误差影响很小。
- 由于每次只输出增量 Δu_k ，因此误动作产生的误差很小，并便于实现无扰动切换。

鉴于增量式 PID 控制的种种优点，本论文采用增量式 PID 控制来进行仿真。

3.3. 增量式 PID 控制的算法实现

应用中，增量式 PID 控制器的算法实现如图 1 所示。

4. Stehfest 算法

对于函数 $f(t)$ 定义其拉普拉斯变换为：

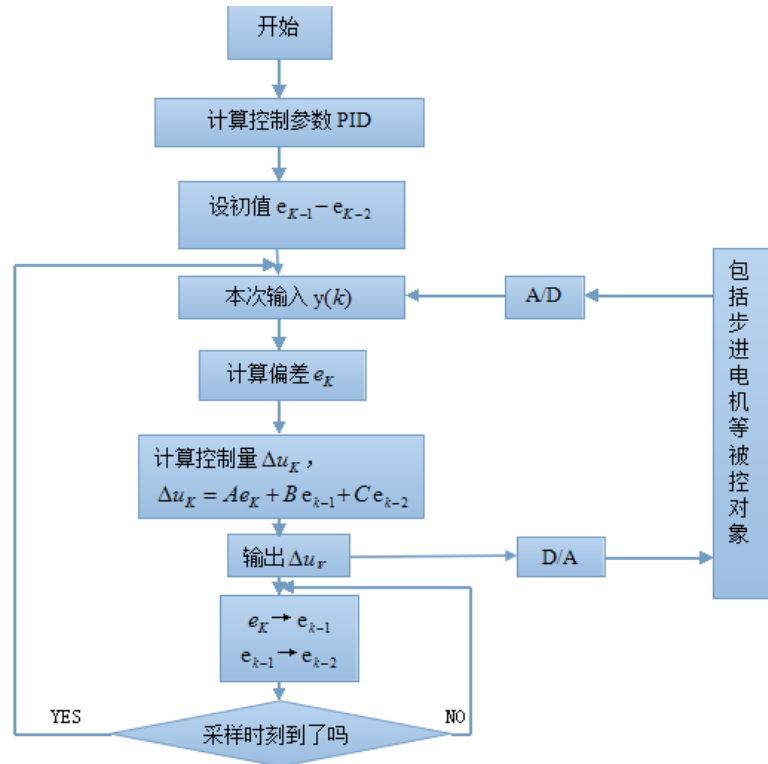


Figure 1. Block diagram of incremental PID

图 1. 增量式 PID 实现框图

$$F(s) = L[f(t)] = \int_0^{\infty} e^{-st} f(t) dt \quad (12)$$

称 $F(s)$ 为函数 $f(t)$ 的拉普拉斯变换像函数。在 VC 中要实现控制系统的输出值的计算，需要将 S 域的拉普拉斯表达式 $F(s)$ 转换为时域的函数表达式 $f(t)$ ，则需要拉普拉斯反变换。

目前经常使用的反变换法有部分分式展开法和围线积分法。当 $F(s)$ 表达式比较复杂时，这两种方法都很难实现反变换。这时就需要采用一种新的算法来实现拉普拉斯反变换。Stehfest 算法就是这样的一种拉普拉斯反变换的近似算法。

采用围线积分法实现反变换时，需要计算被积表达式在各极点上的留数。但当表达式比较复杂或存在多重极点时，微分运算用手工实现比较困难；借助计算机运算时，由于数值微分算法具有天生的数值不稳定性，又使运算误差不易控制。Stehfest 算法用一个全新的表达式代替围线积分法中的积分式，可在计算机辅助下近似实现拉普拉斯反变换，其形式如下：

$$f_a(t) = \frac{\ln 2}{t} \sum_{i=1}^n V_i F\left(\frac{\ln 2}{t} * i\right) \quad (13)$$

其中

$$V_i = (-1)^{\binom{n+1}{2}} * \sum_k^j \frac{k^{\binom{n+1}{2}} (2k)!}{\left(\frac{n}{2} - 1\right)! (k!)^2 (i-k)! (2k-i)!} \quad (14)$$

$$k = \text{int}\left(\frac{i+1}{2}\right) \quad (15)$$

$$j = \min(i, n/2) \quad (16)$$

表达式(13)中 $f_a(t)$ 为 $f(t)$ 的近似表达式, 式中的 n 必须是正的偶数, 具体的选值下文将详细说明。Stehfest 算法易于编程, 在程序部分中提供利用 Turbo C 语言编制的该算法程序。接着对其使用条件和适用范围做进一步的讨论。Stehfest 指出, 对于那些随变量 t 的变化振荡较快的函数, 使用该算法误差较大。而通过实际运算, 作者发现, 对于表达式中含有 e^{at} 的函数, 像 $\sinh(at)$ 、 $\cosh(at)$ 甚至 $\exp(at)$ (这里 a 是常数), 当变量 t 较小时(通常在 0~1 之间), 近似效果较好, 当超出这一范围时, 误差明显增大。另外, 当 $F(s)$ 表达式中分母多项式的次数 M 与分子多项式的次数 N 之间存在关系 $M - N > 2$ 时, 近似效果较好。同时, Stehfest 推荐 n 在 4~32 之间选取, 一般情况下取 16。有研究指出, 使用 Stehfest 算法所得结果与实际结果非常接近, 其误差一般都能控制在 $10^{-3} \sim 10^{-4}$ 之间, 因而对于振荡频率较低, 且用部分分式展开法和围线积分法实现拉普拉斯反变换较困难的情况采用 Stehfest 算法进行反变换会取得比较满意的结果[12]。

5. 应用程序实现功能

本设计中在程序中设置的 $F(s) = 1/s + 1$, 通过改变 stehfest 算法任意改变系统的类型。

1) 改变 k_p 的值, k_i , k_d 保持不变。图形如图 2, 图 3 所示。

如图 2 所示, 当 $k_p = 50$, 系统存在一定的稳态误差。相比图 3, 当 $k_p = 100$ 时, 系统达到比较完美的控制精度。所以在串联校正中, 通过加大比例控制参数, 来提高系统开环增益, 减小系统稳态误差, 提高控制精度。

2) 改变 k_d 的值, k_i , k_p 保持不变。图形如图 3, 图 4 所示。我们可以在图 4 中看到了波动, 相应的调节时间会偏长。控制器的微分作用能够预见偏差的变化趋势从而产生超前的控制作用, 加快系统的动作速度, 减少调节时间, 且参数越小, 微分作用越弱, 相应的调节时间越长。

6. 结论

PID 控制在工业控制的各个领域得到广泛应用, 论文在 VS2010 环境中用 MSChart 控件实现增量式 PID 控制系统的仿真程序的编写, 并测试成功。本系统中, 用户需自己设置 PID 控制器的各项参数, 也

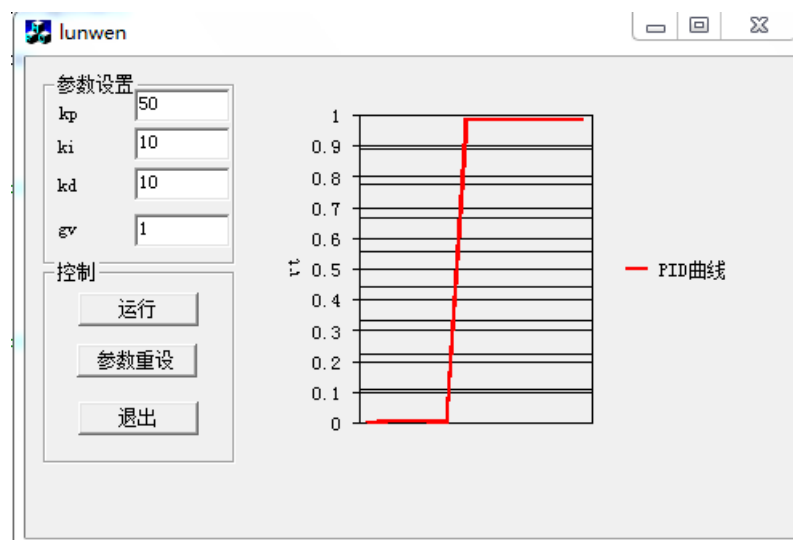


Figure 2. Results of $k_p = 50$, $k_i = 10$, $k_d = 10$

图 2. $k_p = 50$, $k_i = 10$, $k_d = 10$ 程序测试结果

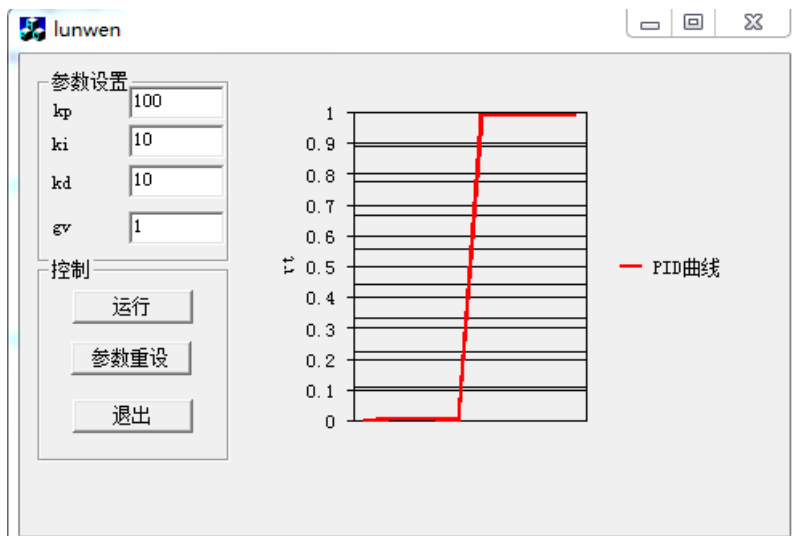


Figure 3. Results of $kp = 100, ki = 10, kd = 10$
 图 3. $kp = 100, ki = 10, kd = 10$ 程序测试结果

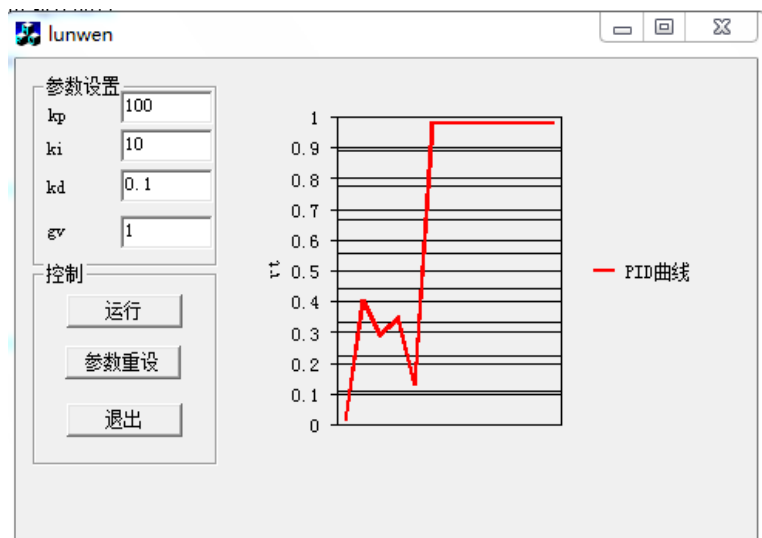


Figure 4. Results of $kp = 100, ki = 10, kd = 0.1$
 图 4. $kp = 100, ki = 10, kd = 0.1$ 程序测试结果

可根据需要改变系统的类型，均可以显示相应的控制图形，由于采用的是近似的拉普拉斯反变换，以及 MSChart 控件数据存储限制，可能会有一定的偏差，但是仍具有一定的学习意义。

另外，各大公司均开发了具有 PID 参数自整定功能的智能调节器，其中 PID 控制器参数的自动调整是通过智能化调整或自校正、自适应算法来实现。有利用 PID 控制实现的压力、温度、流量、液位控制器，能实现 PID 控制功能的 PLC，还有可实现 PID 控制的 PC 系统等等。还有可以实现 PID 控制功能的控制器，如 Rockwell 的 Logix 产品系列，它可以直接与 ControlNet 相连，利用网络来实现远程控制功能，这也是目前的 PID 控制的发展方向。由此可以看出研究 PID 控制很具有现实意义。

参考文献 (References)

[1] 王蕾, 宋文忠 (2004) PID 控制. *自动化仪表*, 4, 7.

- [2] 韩帮华 (2009) PID 控制器参数整定方法及应用研究. 硕士论文, 青岛科技大学, 青岛.
- [3] 白金, 韩俊伟 (2007) 基于 Matlab/Simulink 环境下的 PID 参数整定. *哈尔滨商业大学学报*, **6**, 673-676.
- [4] 李晓丹 (2005) 模糊 PID 控制器的设计研究. 硕士论文, 天津大学, 天津.
- [5] 黄宜庆 (2009) PID 控制器参数整定及其应用研究. 硕士论文, 安徽理工大学, 安徽.
- [6] 金奇, 邓志杰 (2008) PID 控制原理与整定方法. *重庆工学院学报*, **5**, 91-94.
- [7] 胡寿松 (2007) 自动控制原理. 科学出版社, 北京.
- [8] 苏明, 陈伦军, 林浩 (2004) 模糊 PID 控制及其 MatLab 仿真. *现代机械*, **4**, 51-55.
- [9] 林浩 (2006) 模糊 PID 控制器仿真研究. 硕士论文, 贵州大学, 贵州.
- [10] 樊国平 (2005) 智能 PID 控制系统的设计与研究. 硕士论文, 浙江工业大学, 杭州.
- [11] 刘玲玲 (2010) PID 参数整定技术的研究及应用. 硕士论文, 郑州大学, 郑州.
- [12] 刘利强 (2002) 拉普拉斯反变换的一种数值算法. *内蒙古工业大学学报*, **1**, 47-49.