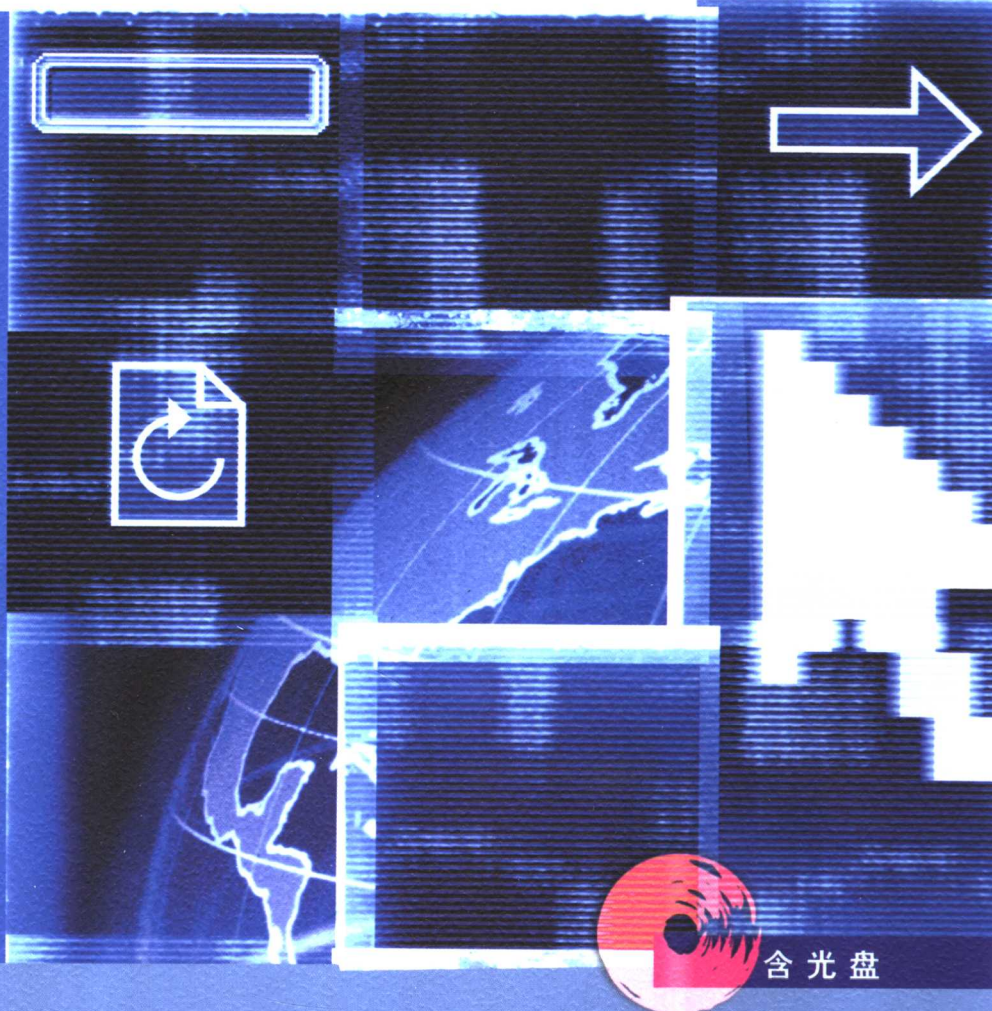


国家自然科学基金 航空基金 资助项目

先进PID控制 MATLAB仿真 (第2版)

刘金琨 著



含光盘



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

国家自然科学基金 航空基金 资助项目

先进 PID 控制 MATLAB 仿真 (第 2 版)

刘金琨 著

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书从 MATLAB 仿真角度系统地介绍了 PID 控制的基本理论、基本方法和应用技术,是作者多年来从事控制系统教学和科研工作的结晶,同时融入了国内外同行近年来所取得的新成果。

全书共分 11 章,包括连续系统和离散系统的 PID 控制;常用数字 PID 控制;专家 PID 和模糊 PID 控制;神经 PID 控制;遗传算法 PID 控制;多变量 PID 控制;几种先进的 PID 控制;灰色 PID 控制;伺服系统 PID 控制;机器人的 PID 控制;PID 实时控制等内容。每种方法都通过 MATLAB 仿真程序进行了说明,所有仿真程序均存储在随书光盘中,读者可以直接调用。本书各部分的内容既相互联系又各自独立,读者可根据需要选择学习。

本书适用于从事生产过程自动化、计算机应用、机械电子及电气自动化领域的工程技术人员和研究生阅读,也可作为高等院校工业自动化、自动控制、机械电子、自动化仪表及计算机应用等专业的教学参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

先进 PID 控制 MATLAB 仿真 / 刘金琨著. —2 版. —北京:电子工业出版社, 2004.9

ISBN 7-121-00325-2

I. 先… II. 刘… III. ①PID 控制 ②计算机辅助计算—软件包, MATLAB IV. ①TP273 ②TP391.75

中国版本图书馆 CIP 数据核字(2004)第 090779 号

责任编辑: 邓小瑜

印 刷: 北京天竺颖华印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 30.25 字数: 774.4 千字

印 次: 2004 年 9 月第 1 次印刷

印 数: 5 000 册 定价: 49.00 元(含光盘 1 张)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010) 68279077。质量投诉请发邮件至 zllts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

PID 控制是最早发展起来的控制策略之一, 由于其算法简单、鲁棒性好及可靠性高, 被广泛应用于过程控制和运动控制中, 尤其适用于可建立精确数学模型的确定性系统。然而实际工业生产过程往往具有非线性、时变不确定性, 难以建立精确的数学模型, 应用常规 PID 控制器不能达到理想的控制效果, 而且在实际生产现场中, 由于受到参数整定方法繁杂的困扰, 常规 PID 控制器参数往往整定不良、性能欠佳, 对运行工况的适应性很差。

计算机技术和智能控制理论的发展为复杂动态不确定系统的控制提供了新的途径。采用智能控制技术, 可设计智能 PID 和进行 PID 的智能整定。

有关智能 PID 控制等新型 PID 控制理论及其工程应用, 近年来已有大量的论文发表。作者多年来一直从事智能控制方面的研究和教学工作, 为了促进 PID 控制和自动化技术的进步, 反映 PID 控制设计与应用中的最新研究成果, 并使广大工程技术人员了解、掌握和应用这一领域的最新技术, 学会用 MATLAB 语言进行 PID 控制器的设计, 作者编写了这本书, 以抛砖引玉, 供广大读者学习参考。

本书是在总结作者多年研究成果的基础上, 进一步理论化、系统化、规范化、实用化而成的, 其特点如下:

(1) PID 控制算法取材新颖, 内容先进, 重点置于学科交叉部分的前沿研究和介绍一些有潜力的新思想、新方法和新技术, 取材着重于基本概念、基本理论和基本方法。

(2) 针对每种 PID 算法给出完整的 MATLAB 仿真程序。这些程序都可以在线运行, 并给出程序的说明和仿真结果。具有很强的可读性, 很容易转化为其他各种实用语言。

(3) 着重从应用领域角度出发, 突出理论联系实际, 面向广大工程技术人员, 具有很强的工程性和实用性。书中有大量应用实例及其结果分析, 为读者提供了有益的借鉴。

(4) 所给出的各种 PID 算法完整, 程序设计、结构设计力求简单明了, 便于自学和进一步开发。

本书共分 11 章。第 1 章介绍连续系统 PID 控制和离散系统数字 PID 控制的几种基本方法, 通过仿真和分析进行了说明。第 2 章介绍常用的数字 PID 控制系统, 主要包括串级计算机控制系统的 PID 控制、纯滞后控制系统 Dahlin 算法和基于 Smith 预估的 PID 控制。第 3 章阐明专家 PID 和模糊 PID 整定的基本算法和程序设计方法, 其中模糊 PID 包括模糊自适应整定 PID 控制和模糊免疫 PID 控制算法, 并进行了仿真分析。第 4 章介绍神经 PID 的几种方法, 包括单神经网络 PID 的设计、神经网络并行 PID 控制、PID 的几种神经网络整定方法, 并通过仿真进行说明。第 5 章引入基于遗传算法的 PID 控制, 主要包括基于遗传算法整定的 PID 控制和基于遗传算法摩擦模型参数辨识的 PID 控制。第 6 章介绍多变量 PID 控制的几种方法, 主要包括 PID 控制、单神经元 PID 控制和基于 DRNN 神经网络整定的 PID 控制。第 7 章阐述几种先进的 PID 控制算法, 包括基于干扰观测器的 PID 鲁棒控制、基于 NCD 优化的非线性 PID 控制、非线性参数整定的 PID 控制、基于重复控制的 PID 高精度控制、基于零相差前馈补偿的 PID 控制和基于卡尔曼滤波的 PID 控制, 每种方法都通过仿真程序进行说明。第 8 章介绍灰色 PID 控制算法和仿真方法, 包括基于连续系统的灰色 PID 控制和基于离散系统的灰色 PID 控制。第 9 章引入伺服系统的 PID 控制, 包括伺服系统在低速摩擦条件下的 PID 控

制、单质量伺服系统 PID 控制和二质量伺服系统 PID 控制，并进行了仿真说明。第 10 章介绍 PID 在机器人控制中的应用实例，包括确定性单臂机械手、不确定性单臂机械手、 N 关节机器人的 PID 控制。第 11 章阐述 PID 在实时控制中的应用实例，并给出 PID 控制的 MATLAB 程序和相应的 Borland C++ 语言实时控制程序。

本书是基于 MATLAB 环境下开发的，各个章节的内容具有很强的独立性，读者可以结合自己的方向深入地进行研究。

本书第 2 版在第 1 版的基础上主要增加了以下内容：基于 S 函数的连续系统 Simulink 仿真；基于 S 函数的离散系统 Simulink 仿真；基于一种离散微分-跟踪器的 PID 控制；基于 Ziegler-Nichols 方法的 PID 整定；基于 Hopfield 神经网络的 PID 控制；模糊 RBF 网络的 PID 整定；实时遗传算法优化的 PID 控制；基于 Anti-windup 的 PID 控制；基于 PD 增益自适应调节的模型参考自适应控制。并增加了新的一章：机器人 PID 控制。针对某些以 M 语言实现的程序进行了 Simulink 设计，并针对第 1 版中的某些错误进行了修改。

北京航空航天大学尔联洁教授在伺服系统设计方面提出了许多宝贵意见，东北大学徐心和教授和薛定宇教授给予了大力支持和帮助，薛定宇教授在 S 函数设计和 Simulink 仿真方面给作者提供了很多的指导。

作者在仿真研究中，得到实验室许多同仁的帮助。在神经网络设计方面得到扈宏杰博士的帮助，在遗传算法和零相差设计等方面得到刘强博士的帮助，在灰色系统设计方面得到李水清硕士的帮助，在机器人控制器设计方面得到卢宇硕士的帮助，在 PID 实时控制方面得到刘涛硕士的帮助，在图表制作中得到了邬强硕士的帮助，在此一并表示感谢。

感谢郝瑞霞、郝春霞、刘海荣在本书的撰写及整理工作中给予的帮助。

本书的研究工作得到了国家自然科学基金（编号：69874037）和航空基金（编号：00E51022）的资助。

由于作者水平有限，书中难免存在一些不足和错误之处，欢迎广大读者批评指正。

刘金琨
北京航空航天大学
2004 年 7 月 1 日

目 录

第 1 章 数字 PID 控制	(1)
1.1 PID 控制原理	(1)
1.2 连续系统的模拟 PID 仿真	(2)
1.2.1 基本的 PID 控制	(2)
1.2.2 线性时变系统的 PID 控制	(7)
1.3 数字 PID 控制	(10)
1.3.1 位置式 PID 控制算法	(10)
1.3.2 连续系统的数字 PID 控制仿真	(11)
1.3.3 离散系统的数字 PID 控制仿真	(16)
1.3.4 增量式 PID 控制算法及仿真	(23)
1.3.5 积分分离 PID 控制算法及仿真	(25)
1.3.6 抗积分饱和 PID 控制算法及仿真	(29)
1.3.7 梯形积分 PID 控制算法	(33)
1.3.8 变速积分 PID 算法及仿真	(33)
1.3.9 带滤波器的 PID 控制仿真	(36)
1.3.10 不完全微分 PID 控制算法及仿真	(43)
1.3.11 微分先行 PID 控制算法及仿真	(47)
1.3.12 带死区的 PID 控制算法及仿真	(51)
1.3.13 基于前馈补偿的 PID 控制算法及仿真	(55)
1.3.14 步进式 PID 控制算法及仿真	(58)
1.3.15 PID 控制的方波响应	(61)
1.3.16 一种离散微分-跟踪器	(65)
第 2 章 常用的 PID 控制系统	(71)
2.1 单回路 PID 控制系统	(71)
2.2 串级 PID 控制	(71)
2.2.1 串级 PID 控制原理	(71)
2.2.2 仿真程序及分析	(72)
2.3 纯滞后系统的大林控制算法	(76)
2.3.1 大林控制算法原理	(76)
2.3.2 仿真程序及分析	(76)
2.4 纯滞后系统的 Smith 控制算法	(78)
2.4.1 连续 Smith 预估控制	(79)
2.4.2 仿真程序及分析	(80)
2.4.3 数字 Smith 预估控制	(81)

2.4.4	仿真程序及分析	(81)
2.5	基于 Ziegler-Nichols 方法的 PID 整定	(86)
2.5.1	连续 Ziegler-Nichols 方法的 PID 整定	(86)
2.5.2	仿真程序及分析	(86)
2.5.3	离散 Ziegler-Nichols 方法的 PID 整定	(89)
2.5.4	仿真程序及分析	(90)
第 3 章	专家 PID 控制和模糊 PID 控制	(94)
3.1	专家 PID 控制	(94)
3.1.1	专家 PID 控制原理	(94)
3.1.2	仿真程序及分析	(95)
3.2	一个典型的模糊控制器的设计	(102)
3.2.1	模糊控制的基本原理	(102)
3.2.2	模糊控制器设计步骤	(104)
3.2.3	模糊控制器设计实例	(106)
3.2.4	模糊控制位置跟踪	(110)
3.3	模糊自适应整定 PID 控制	(115)
3.3.1	模糊自适应整定 PID 控制原理	(115)
3.3.2	仿真程序及分析	(118)
3.4	模糊免疫 PID 控制算法	(129)
3.4.1	模糊免疫 PID 控制算法原理	(129)
3.4.2	仿真程序及分析	(130)
3.5	基于 Sugeno 的模糊控制	(134)
3.5.1	Sugeno 模糊模型	(134)
3.5.2	Sugeno 模糊模型的建立	(135)
3.5.3	基于 Sugeno 的倒立摆模糊控制	(137)
3.6	基于控制规则表的模糊 PD 控制	(146)
3.6.1	模糊控制器的原理	(146)
3.6.2	仿真程序及分析	(146)
第 4 章	神经 PID 控制	(153)
4.1	基于单神经网络的 PID 智能控制	(153)
4.1.1	几种典型的学习规则	(153)
4.1.2	单神经元自适应 PID 控制	(153)
4.1.3	改进的单神经元自适应 PID 控制	(154)
4.1.4	仿真程序及分析	(154)
4.1.5	基于二次型性能指标学习算法的单神经元自适应 PID 控制	(158)
4.1.6	仿真程序及分析	(159)
4.2	基于 BP 神经网络整定的 PID 控制	(162)
4.2.1	基于 BP 神经网络的 PID 整定原理	(162)

4.2.2	仿真程序及分析	(165)
4.3	基于 RBF 神经网络整定的 PID 控制	(170)
4.3.1	RBF 神经网络模型	(170)
4.3.2	RBF 网络 PID 整定原理	(172)
4.3.3	仿真程序及分析	(172)
4.4	基于 RBF 神经网络辨识的单神经元 PID 模型参考自适应控制	(178)
4.4.1	神经网络模型参考自适应控制原理	(178)
4.4.2	仿真程序及分析	(178)
4.5	基于 CMAC (神经网络) 与 PID 的并行控制	(183)
4.5.1	CMAC 概述	(183)
4.5.2	一种典型 CMAC 算法及其仿真	(184)
4.5.3	仿真程序及分析	(186)
4.5.4	CMAC 与 PID 复合控制算法	(188)
4.5.5	仿真程序及分析	(189)
4.6	CMAC 与 PID 并行控制的 Simulink 仿真	(193)
4.6.1	Simulink 仿真方法	(193)
4.6.2	仿真程序及分析	(193)
4.7	基于 Hopfield 网络的 PID 模型参考自适应控制	(197)
4.7.1	系统描述	(197)
4.7.2	基于 Hopfield 网络的控制器优化	(198)
4.7.3	仿真程序及分析	(200)
4.8	基于模糊 RBF 网络整定的 PID 控制	(203)
4.8.1	模糊神经网络结构	(203)
4.8.2	仿真程序及分析	(205)
第 5 章	基于遗传算法整定的 PID 控制	(210)
5.1	遗传算法的基本原理	(210)
5.2	遗传算法的优化设计	(211)
5.2.1	遗传算法的构成要素	(211)
5.2.2	遗传算法的应用步骤	(211)
5.3	遗传算法求函数极大值	(212)
5.3.1	二进制编码遗传算法求函数极大值	(212)
5.3.2	仿真程序	(214)
5.3.3	实数编码遗传算法求函数极大值	(216)
5.3.4	仿真程序	(218)
5.4	基于遗传算法的 PID 整定	(220)
5.4.1	基于遗传算法的 PID 整定原理	(221)
5.4.2	基于实数编码遗传算法的 PID 整定	(223)
5.4.3	仿真程序	(224)
5.4.4	基于二进制编码遗传算法的 PID 整定	(228)

5.4.5	仿真程序	(229)
5.4.6	基于自适应在线遗传算法整定的 PID 控制	(233)
5.4.7	仿真程序	(235)
5.5	基于遗传算法摩擦模型参数辨识的 PID 控制	(239)
5.5.1	辨识原理及仿真实例	(239)
5.5.2	仿真程序	(241)
第 6 章	先进 PID 多变量控制	(247)
6.1	PID 多变量控制	(247)
6.1.1	PID 控制原理	(247)
6.1.2	仿真程序及分析	(247)
6.1.3	多变量 PID 控制的 Simulink 仿真	(250)
6.2	单神经元 PID 控制	(254)
6.2.1	单神经元 PID 控制原理	(254)
6.2.2	仿真程序及分析	(254)
6.2.3	多变量单神经元 PID 控制的 Simulink 仿真	(258)
6.3	基于 DRNN 神经网络整定的 PID 控制	(263)
6.3.1	基于 DRNN 神经网络参数自学习 PID 控制原理	(263)
6.3.2	DRNN 神经网络的 Jacobian 信息辨识	(265)
6.3.3	仿真程序及分析	(266)
第 7 章	几种先进 PID 控制方法	(275)
7.1	基于干扰观测器的 PID 控制	(275)
7.1.1	干扰观测器设计原理	(275)
7.1.2	连续系统的控制仿真	(277)
7.1.3	离散系统的控制仿真	(279)
7.2	非线性系统的 PID 鲁棒控制	(284)
7.2.1	基于 NCD 优化的非线性优化 PID 控制	(284)
7.2.2	基于 NCD 与优化函数结合的非线性优化 PID 控制	(286)
7.3	一类非线性 PID 控制器设计	(288)
7.3.1	非线性控制器设计原理	(288)
7.3.2	仿真程序及分析	(289)
7.4	基于重复控制补偿的高精度 PID 控制	(294)
7.4.1	重复控制原理	(294)
7.4.2	基于重复控制补偿的 PID 控制	(294)
7.4.3	仿真程序及分析	(295)
7.5	基于零相差前馈补偿的 PID 控制	(300)
7.5.1	零相差控制原理	(300)
7.5.2	基于零相差前馈补偿的 PID 控制	(301)
7.5.3	仿真程序及分析	(302)

7.6	基于卡尔曼滤波器的 PID 控制	(314)
7.6.1	卡尔曼滤波器原理	(314)
7.6.2	仿真程序及分析	(315)
7.6.3	基于卡尔曼滤波器的 PID 控制	(320)
7.6.4	仿真程序及分析	(321)
7.7	单级倒立摆的 PID 控制	(323)
7.7.1	单级倒立摆建模	(323)
7.7.2	单级倒立摆控制	(325)
7.7.3	仿真程序及分析	(325)
7.8	吊车-双摆系统的控制	(330)
7.8.1	吊车-双摆系统的建模	(330)
7.8.2	吊车-双摆系统的仿真	(331)
7.9	基于 Anti-windup 的 PID 控制	(336)
7.9.1	Anti-windup 的基本原理	(336)
7.9.2	仿真程序及分析	(337)
7.10	基于 PD 增益自适应调节的模型参考自适应控制	(343)
7.10.1	控制器的设计	(343)
7.10.2	稳定性分析	(344)
7.10.3	仿真程序及分析	(345)
第 8 章	灰色 PID 控制	(349)
8.1	灰色控制原理	(349)
8.1.1	生成数列	(349)
8.1.2	GM 模型	(350)
8.2	干扰信号的灰色估计	(350)
8.2.1	灰色估计的理论基础	(350)
8.2.2	仿真实例	(353)
8.3	灰色 PID 控制	(355)
8.3.1	灰色 PID 控制的理论基础	(355)
8.3.2	连续系统灰色 PID 控制	(356)
8.3.3	仿真程序及分析	(358)
8.3.4	离散系统灰色 PID 控制	(362)
8.3.5	仿真程序及分析	(363)
8.4	灰色 PID 的位置跟踪	(367)
8.4.1	连续系统灰色 PID 位置跟踪	(367)
8.4.2	仿真程序及分析	(369)
8.4.3	离散系统灰色 PID 位置跟踪	(372)
8.4.4	仿真程序及分析	(374)
第 9 章	伺服系统 PID 控制	(378)
9.1	基于 Luge 摩擦模型的 PID 控制	(378)

9.1.1	伺服系统的摩擦现象	(378)
9.1.2	伺服系统的Lugre摩擦模型	(378)
9.1.3	仿真程序及分析	(379)
9.2	基于Stribeck摩擦模型的PID控制	(384)
9.2.1	Stribeck摩擦模型描述	(384)
9.2.2	一个典型伺服系统描述	(385)
9.2.3	仿真程序及分析	(385)
9.3	伺服系统三环的PID控制	(396)
9.3.1	伺服系统三环的PID控制原理	(396)
9.3.2	仿真程序及分析	(397)
9.4	二质量伺服系统的PID控制	(402)
9.4.1	二质量伺服系统的PID控制原理	(402)
9.4.2	仿真程序及分析	(403)
9.5	伺服系统的模拟PD+数字前馈控制	(407)
9.5.1	伺服系统的模拟PD+数字前馈控制原理	(407)
9.5.2	仿真程序及分析	(408)
第10章	机器人的PID控制	(411)
10.1	确定性单臂机械手的PD+前馈控制	(411)
10.1.1	单臂机械手的运动方程	(411)
10.1.2	控制器的设计	(411)
10.1.3	仿真程序及分析	(411)
10.2	不确定性单臂机械手的PD+前馈控制	(416)
10.2.1	不确定性单臂机械手的运动方程	(416)
10.2.2	仿真程序及分析	(417)
10.3	不确定性单臂机械手的PD鲁棒控制	(419)
10.3.1	控制器设计	(419)
10.3.2	稳定性分析	(419)
10.3.3	仿真程序及分析	(422)
10.4	基于PD的N关节机器人控制	(425)
10.4.1	N关节机器人运动方程	(426)
10.4.2	PD控制	(426)
10.4.3	PD控制+前馈控制	(426)
10.4.4	PD控制+修正前馈控制	(426)
10.4.5	仿真程序及分析	(426)
10.5	机器人的鲁棒自适应PD控制	(431)
10.5.1	机器人动力学模型及其结构特性	(431)
10.5.2	控制器的设计	(432)
10.5.3	仿真程序及分析	(436)

第 11 章 PID 实时控制的 C++语言设计及应用	(449)
11.1 M 语言的 C++转化	(449)
11.2 基于 C++的三轴飞行模拟转台伺服系统 PID 实时控制	(452)
11.2.1 控制系统构成	(452)
11.2.2 实时控制程序分析	(453)
11.2.3 仿真程序及分析	(456)
参考文献	(469)

第 1 章 数字 PID 控制

自从计算机进入控制领域以来，用数字计算机代替模拟计算机调节器组成计算机控制系统，不仅可以用软件实现 PID 控制算法，而且可以利用计算机的逻辑功能，使 PID 控制更加灵活。数字 PID 控制在生产过程中是一种最普遍采用的控制方法，在机电、冶金、机械、化工等行业中获得了广泛的应用。将偏差的比例（P）、积分（I）和微分（D）通过线性组合构成控制量，对被控对象进行控制，故称 PID 控制器。

1.1 PID 控制原理

在模拟控制系统中，控制器最常用的控制规律是 PID 控制。模拟 PID 控制系统原理框图如图 1-1 所示。系统由模拟 PID 控制器和被控对象组成。

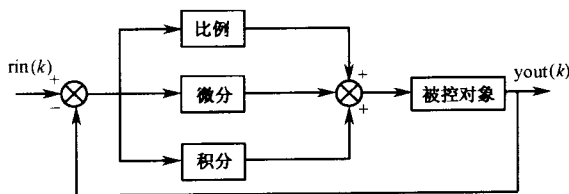


图 1-1 模拟 PID 控制系统原理框图

PID 控制器是一种线性控制器，它根据给定值 $rin(t)$ 与实际输出值 $yout(t)$ 构成控制偏差：

$$error(t) = rin(t) - yout(t) \quad (1.1)$$

PID 的控制规律为：

$$u(t) = k_p \left(error(t) + \frac{1}{T_i} \int_0^t error(t) dt + \frac{T_D derror(t)}{dt} \right) \quad (1.2)$$

或写成传递函数的形式：

$$G(s) = \frac{U(s)}{E(s)} = k_p \left(1 + \frac{1}{T_i s} + T_D s \right) \quad (1.3)$$

式中， k_p ——比例系数； T_i ——积分时间常数； T_D ——微分时间常数。

简单说来，PID 控制器各校正环节的作用如下：

(1) 比例环节：成比例地反映控制系统的偏差信号 $error(t)$ ，偏差一旦产生，控制器立即产生控制作用，以减小偏差。

(2) 积分环节：主要用于消除静差，提高系统的无差度。积分作用的强弱取决于积分时间常数 T_i ， T_i 越大，积分作用越弱，反之则越强。

(3) 微分环节：反映偏差信号的变化趋势（变化速率），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，从而加快系统的动作速度，减少调节时间。

1.2 连续系统的模拟 PID 仿真

1.2.1 基本的 PID 控制

以二阶线性传递函数为被控对象，进行模拟 PID 控制。在信号发生器中选择正弦信号，仿真时取 $k_p = 60$ ， $k_i = 1$ ， $k_d = 3$ ，输入指令为 $\text{rin}(t) = A\sin(2\pi Ft)$ ，其中 $A = 1.0$ ， $F = 0.20 \text{ Hz}$ 。采用 ODE45 迭代方法，仿真时间为 10s。

仿真方法一

在 Simulink 下进行仿真，PID 控制由 Simulink 下的工具箱提供。

仿真程序：chap1_1.mdl，如图 1-2 所示。

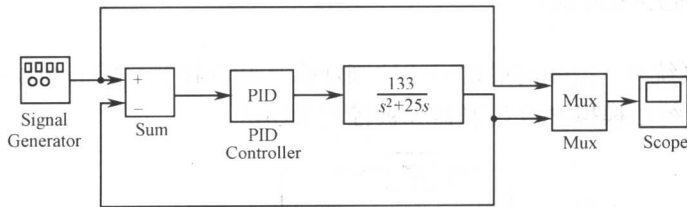


图 1-2 连续系统 PID 的 Simulink 仿真程序

在 PID 控制器采用 Simulink 封装的形式，其内部结构如图 1-3 所示。

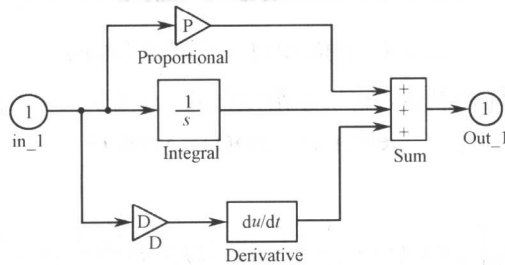


图 1-3 模拟 PID 控制器

连续系统的模拟 PID 控制正弦响应结果如图 1-4 所示。

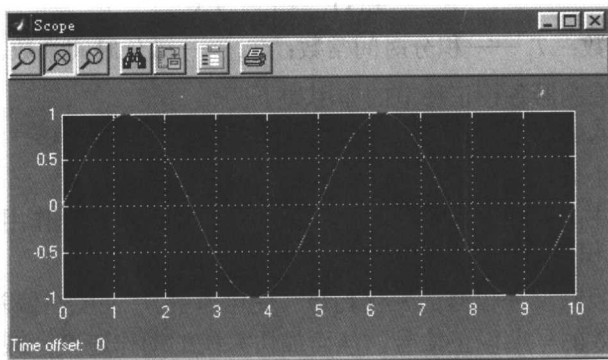


图 1-4 连续系统的模拟 PID 控制正弦响应

仿真方法二

在仿真一的基础上，将仿真结果输出到工作空间中，利用 M 函数作图，仿真结果如图 1-5 所示。

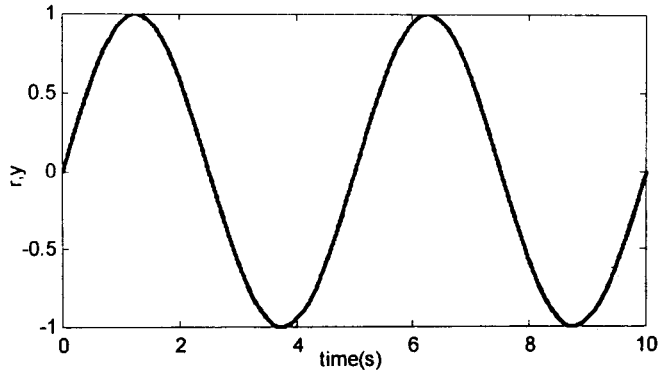


图 1-5 PID 控制正弦响应

仿真程序: chap1_2.mdl。

程序中同时采用了传递函数的另一种表达方式，即状态方程的形式，其中 $A = \begin{bmatrix} 0 & 1 \\ 0 & -25 \end{bmatrix}$,

$B = \begin{bmatrix} 0 \\ 133 \end{bmatrix}$, $C = [1 \quad 0]$, $D = 0$, 如图 1-6 所示。

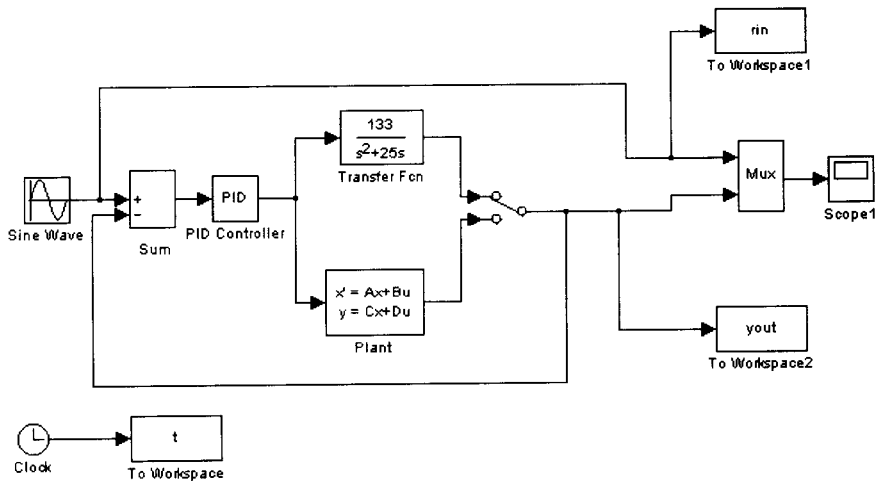


图 1-6 连续系统 PID 的 Simulink 仿真程序

M 函数作图程序: chap1_2plot.m。

```
close all;
```

```
plot(t,rin,'k',t,yout,'k');
```

```
xlabel('time(s)');
```

```
ylabel('r,y');
```

仿真方法三

S 函数是 Simulink 一项重要的功能, 采用 S 函数可实现在 Simulink 下复杂控制器和复杂被控对象的编程。在仿真一的基础上, 利用 S 函数实现对象的表达、控制器的设计及仿真结果的输出。

在 S 函数中, 采用初始化、微分函数和输出函数, 即 `mdlInitializeSizes` 函数、`mdlDerivatives` 函数和 `mdlOutputs` 函数。在初始化中采用 `sizes` 结构, 选择 2 个输出, 3 个输入, 3 个输入实现了 P、I、D 三项的输入。S 函数嵌入在 Simulink 程序中。系统初始状态为: $x(0) = 0, \dot{x}(0) = 0$ 。仿真结果如图 1-7 所示。

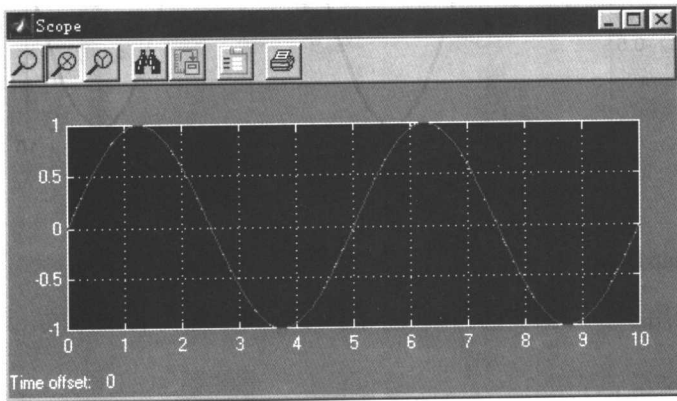


图 1-7 PID 控制正弦响应

仿真程序: `chap1_3.mdl` 基于 S 函数的 Simulink 仿真如图 1-8 所示。

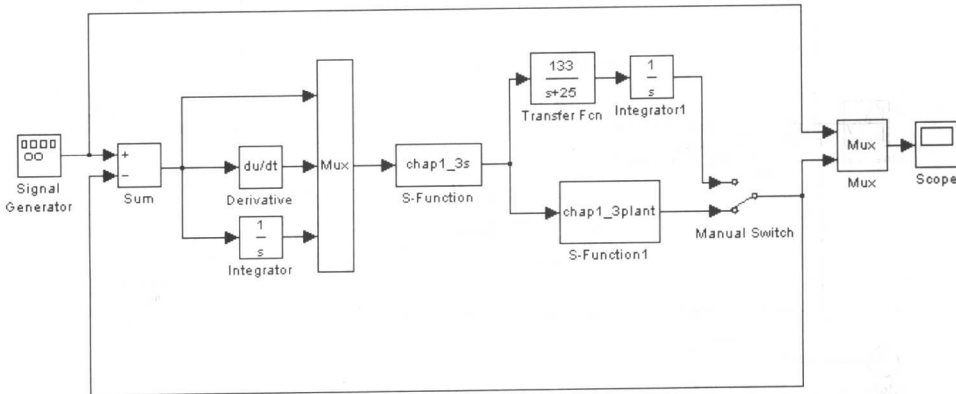


图 1-8 基于 S 函数的 Simulink 仿真

S 函数控制器程序: `chap1_3s.m`。

```
%S-function for continuous state equation
function [sys,x0,str,ts]=s_function(t,x,u,flag)

switch flag,
%Initialization
case 0,
```



```

    [sys,x0,str,ts]=mdlInitializeSizes;
%Outputs
    case 3,
        sys=mdlOutputs(t,x,u);
%Unhandled flags
    case {2, 4, 9 }
        sys = [];
%Unexpected flags
    otherwise
        error(['Unhandled flag = ',num2str(flag)]);
end

```

```

%mdlInitializeSizes
function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 1;
sizes.NumInputs = 3;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 0;

```

```

sys=simsizes(sizes);
x0=[];
str=[];
ts=[];

```

```

function sys=mdlOutputs(t,x,u)
error=u(1);
derror=u(2);
errori=u(3);

kp=60;
ki=1;
kd=3;
ut=kp*error+kd*derror+ki*errori;

```

```

sys(1)=ut;

```

S 函数被控对象程序: chap1_3plant.m.

```

%S-function for continuous state equation

```