

# 基于 STM32 的 PID 控制程序设计

唐 飞,徐文权,王陈宁

(安庆师范大学 物理与电气工程学院,安徽 安庆 246133)

**摘 要:** PID 控制是工程领域常用的控制算法。随着嵌入式处理器的不断发展,使用嵌入式设备进行 PID 控制成为趋势。本文基于 STM32 微处理器,利用 C 语言模块化封装性好的优势,设计了直观清晰的 PID 控制程序。利用积分分离算法改进程序,增强了系统的动态特性,并使用 MATLAB 对输出的结果进行仿真,证明了程序设计的正确性。

**关键词:** PID;离散控制;结构体;MATLAB;STM32 **DOI:** 10.13757/j.cnki.cn34-1328/n.2017.01.005  
**中图分类号:** TP273+.5 **文献标识码:** A **文章编号:** 1007-4260(2017)01-0019-04

## Design of PID program based on STM32

TANG Fei, XU Wenquan, WANG Chenning

(School of Physics and Electronics, Anqing Normal University, Anqing 246133, China)

**Abstract:** The PID controller is the most common form of feedback in engineering. As embedded processor's compute speed became more and more quickly, it is possible to complete the calculation of PID controller. This paper describes how to use the advantages of C language, and write a PID controller program using C programming language. This program is improved by the integral separation algorithm. The simulation by using MATLAB showed the validity of the program.

**Key words:** PID; discrete control; structure; MATLAB; STM32

当今的自动控制技术主要是基于反馈概念的闭环控制,控制系统将机构的实际输出值与期望值相比较,用偏差来纠正系统的响应,执行调节控制。在工程实际应用中,最为广泛的调节器控制规律是 PID 控制。对于 PID 控制的程序设计和实现,研究开发平台主要基于 PLC、嵌入式、PC 和工控机等系统,使用的程序设计语言有 MATLAB、汇编语言和 C 语言等。

PID 控制规律简单、稳定性好,对于 PID 的程序设计,广大学者进行了很多研究。何东健等<sup>[1]</sup>使用 MATLAB 软件对 PID 控制器参数整定并进行了仿真,任俊杰等<sup>[2]</sup>基于 PLC 实现了闭环 PID 控制器,刘道兴等<sup>[3]</sup>使用汇编语言设计了水温控制系统的 PID 控制程序,毛雪莲等<sup>[4]</sup>使用 C 语言在 51 单片机上实现了 PID 的温度控制程序,刘亚平

等<sup>[5]</sup>使用 C 语言设计了基于 PID 的液位控制程序,雷慧杰等<sup>[6]</sup>基于 STM32 设计了直流电机的 PID 调速系统。以上对 PID 程序设计方法的研究,或多或少的存在着一些局限性。MATLAB 软件一般运行在 PC 机上,主要作用于算法验证、参数整定和系统仿真,不直接应用于工控系统;PLC 成熟可靠、使用方便,然而浮点运算能力弱,一般用于控制系统后端的控制,不适合进行算法的实现;汇编语言可以作用于嵌入式系统,然而因浮点运算能力差、程序模块化和可读性差、难以移植等缺点无法广泛应用;使用 C 语言进行 PID 程序设计时,很多程序没有充分考虑到程序可读性、模块化的需求,使用较好的数据结构来组织数据,使得程序的可读性和移植性差。基于此,本文提出在嵌入式处理器 STM32 系统上,利用 C 语言浮

\* 收稿日期: 2016-09-25  
基金项目: 安徽省自然科学基金(AQKJ2014B018)。  
作者简介: 唐飞,男,安徽安庆人,硕士,安庆师范大学物理与电气工程学院高级实验师,研究方向为机械制造及其自动化。  
万方数据: mrxyz@163.com

点运算能力强、模块化好、可读性强、便于移植的优势,开发 PID 的控制程序,做到通用性强,并应用 MATLAB 软件验证输出结果。

1 连续和离散 PID 系统的控制规律

PID 控制是一种线性调节器,控制单元由比例单元 P (Proportion)、积分单元 I (Integration)和微分单元 D (Differentiation)组成。PID 控制以其结构简单、稳定性好、工作可靠、调整方便而成为工业控制的主要技术之一<sup>[7]</sup>,其系统结构框图如图 1 所示。

图 1 中  $r(t)$ 是系统的输入值, $c(t)$ 是系统的输出值,  $e(t)$ 是系统输入值与输出值的偏差,  $e(t)=c(t)-r(t)$ , $u(t)$ 是控制器的输出值, $K_p$ 是比例系数,  $T_i$ 是积分时间常数, $T_d$ 是微分时间常数。根据 PID 的控制规律,控制器输出  $u(t)$ 与误差  $e(t)$ 的关系为

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

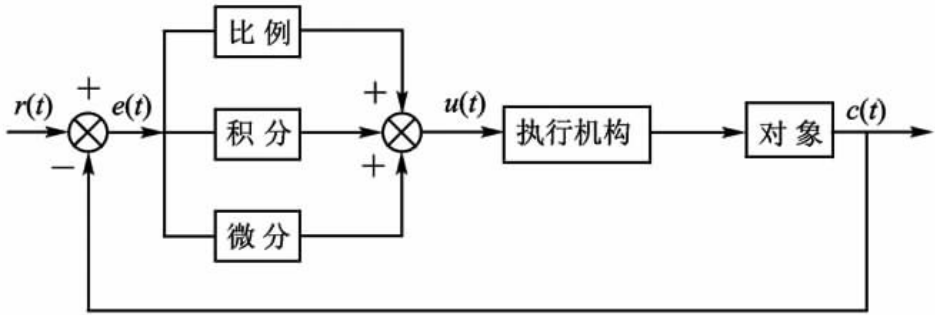


图 1 PID 控制系统框图

2 数字 PID 控制算法的程序实现

数字 PID 控制需要由计算机来实现,传统的高性能计算机在成本、体积和功耗等方面都存在劣势,而 51,AVR 等单片机都难以满足数字 PID 控制系统快速采样和计算的需求。因此,采用基于 ARM 的 Cortex-M3 内核微控制器 STM32 作为系统主控芯片,是一种较好的选择。STM32 具有高性能、低成本、低功耗的特点,且运算能力强<sup>[8]</sup>,适合进行复杂的数值计算;系统体积小、重量轻,适合现代移动设备的需求。

控制系统需要使用嵌入式控制器的定时器,定时时间设为采样周期  $T$ ,每过一个采样周期,执行一次 PID 运算,输出相应的控制量。

分析离散 PID 的计算公式可知,每次计算参与参数有  $K_p$ , $T_i$  和  $T_d$  和误差  $e_i$ ,积分运算通过对误差的累加来实现,差分计算需要记录连续 3 个

上述系统为连续 PID 控制系统,现代控制系统一般采用计算机控制。计算机控制系统是一种离散控制系统,只能根据采样时刻的偏差值来计算控制量,无法实现连续 PID 控制。为此必须对系统进行离散化处理,选定采样周期  $T$ ,用差分方程取代微分方程,用求和代替积分,用差分代替微分,将连续 PID 算式转换为离散 PID 算式:

$$u_n=K_p\left[e_n+\frac{1}{T_i}\sum_{k=0}^ne_k\cdot T+T_d\cdot\frac{e_n-e_{n-1}}{T}\right],$$

此时系统输出  $u_n$  为全量输出,称之为 PID 位置型控制表达式。当执行机构不需要控制量的绝对数值,而是需要增量值时,可以将位置型表达式转换为增量型表达式:

$$\Delta u_i=u_i-u_{i-1}=K_p\left[(e_i-e_{i-1})+\frac{T}{T_i}e_i+\frac{T_d}{T}(e_i-2e_{i-1}+e_{i-2})\right],$$

此时控制器的输出是每一次控制量的增量,避免了绝对量的累计误差,控制效果一般好于位置式。

采样周期的误差并求差。因此,程序设计时需要定义大量变量来存储和传递这些参数,程序变得复杂,可读性和执行效率低。考虑到这些数据同属于一次 PID 运算所需要的参数,而嵌入式 CPU 一般不支持 C++ 语言,因此使用 C 语言中的结构体来组织这些变量,使它们有机地组合起来成为一体,程序结构清晰,传递数据时不易出错。由于 STM32 运算能力很强,因此在程序设计中并不特别注重程序地简化和计算地精简,而是着重于清晰地描述算法,增强程序的可读性。

在程序中定义 PID 的结构体组织运算参数:

```
typedef struct
{
    float r;           // 系统输入值
    float c;           // 系统输出值
    float u;           // 控制器输出值
    float e[3];        // 误差数组
```

```
float kp,ki,kd; // 比例积分微分系数
float integral; // 积分值

}PID;
```

其中变量  $r$  是系统的输入,即系统期望达到的目标值; $c$  是系统的输出; $u$  是 PID 控制器的输出,根据 需要,变量  $c$  和  $u$  可以采取数组的形式,便于进行 迭代计算,数组的大小根据系统阶次而定; $e[3]$ 是 误差数组,该数组包含 3 个变量,用来记录连续 3 个 采样周期的误差值;变量  $integral$  用来累加每次 运算的误差,体现积分的作用; $k_p,k_i$  和  $k_d$  是比例 微分积分的系数。这些变量的数据类型均采用  $float$  型,也可以视精度要求修改为  $double$  型。

控制程序首先定义一个 PID 的结构体实体, 然后将结构体初始化,并给出系统的输入值和比 例微分积分的系数。由于 PID 运算每个采样周期 进行一次,因此将运算的过程编写成函数,便于系 统调用,也保证了程序的封装性。位置型算法的运 算函数如下,

```
float PID_Calc(PID *pid, float ob) //PID 计算函
数, 参数 1 是 PID 结构体指针, 参数 2 是系统目
标值
{
    pid->r=ob; // 输入系统达到的目标
    pid->e[0]=pid->r-pid->c; // 求系统误差
    pid->integral+=pid->e [0]; // 对误差求和,
体现积分作用

    pid->u=pid->kp*pid->e[0]+
    pid->ki*pid->integral+pid->kd*(pid->e[0]-
    pid->e[1]); //PID 运算式
    pid->e[1]=pid->e[0]; // 保存本次误差
    pid->c 和 pid->u 的关系方程式 // 被控
对象输入和输出的关系方程

    return pid->c;
}

如果采用增量式算法,可以省去积分变量,函
数如下,

float PID_Calc(PID *pid,float ob)
{
    pid->r=ob;
    pid->e[0]=pid->r-pid->c;
    pid->u+=pid->kp*(pid->e[0]-pid->e[1])+
pid->ki*pid->e[0]+pid->kd*(pid->e[0]-2*pid->
e[1]+pid->e[2]);
    pid->c=pid->u;
    pid->e[1]=pid->e[0];
```

```
pid->c 和 pid->u 的关系方程式
return pid->c;
}
```

系统运行时使用 STM32 系统的定时器,每个 采样周期调用一次 PID 运算函数,计算并输出结 果。该程序实际应用工控系统时,需要通过实验 或数学分析,建立起被控对象输入量  $u$  和输出量  $c$  之间的函数关系,应用于程序,才能达到实际的 控制效果。

设被控对象的传递函数为  $\frac{C(s)}{U(s)}=\frac{50}{0.125s^2+7s}$ , 系统采样时间为 1 ms,先使用变换将其离散化,  $\frac{C(z)}{U(z)}=\frac{0.000\ 196\ 3z+0.000\ 192\ 7}{z^2-1.946z+0.945\ 5}$ , 则  $c_n=1.946c_{n-1}-0.945\ 5c_{n-2}+0.000\ 196u_{n-1}+0.000\ 193u_{n-2}$ , 这就是被控对象输入与输出函数关系的迭代表 示,在程序中使用循环语句进行迭代,即可得到当 前  $c_n$  值。将 PID 的控制参数设为  $k_p=50,k_i=5, k_d=10$ ,输入为阶跃信号,调用程序进行计算,将输 出结果送入 MATLAB 进行仿真,得到的输出曲线 如图 2 所示,分析可知程序实现了 PID 控制规律。

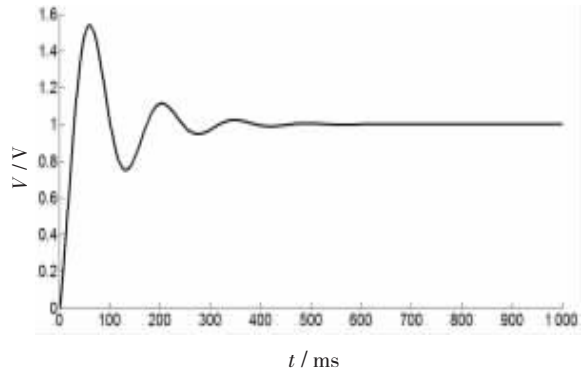


图 2 离散 PID 控制系统输出曲线

3 积分分离 PID 算法的程序实现

在 PID 控制中,引入积分环节的主要目的是 为了消除静差,提高精度。但是在系统起动、结束 或大幅度增减设定时,短时间内系统会有很大的 偏差,造成积分运算的累积,导致控制量超过执行 机构允许的最大控制量,引起较大的超调或震荡<sup>[9]</sup>。 为了解决这个问题,需要引入积分分离的算 法,其设计思想是当系统输出量与设定值偏差较 大时,取消积分作用,当偏差较小时,再引入积分控 制。即系统起动、停止或大幅度改变设定值,误差 较大时,只有比例控制和微分控制参与作用,当误 差变得较小时,再加入积分控制。这种方法更有利

于改善动态特性、消除静差和提高精度。

程序设计时,针对被控对象的误差,设定一个偏差的门限 $k$ 。当偏差 $e$ 的绝对值大于 $k$ 时,系统不引入积分控制,只使用 PD 控制,当偏差 $e$ 的绝对值小于 $k$ 时,再引入积分控制,此时为 PID 控制。

```
float PID_Calc(PID *pid,float ob)
{
    .....
    if (abs(pid->e[0]<k))    // 判断系统误差是否超限,从而采取不同的控制方式
        pid->u+=pid->kp*(pid->e[0]-pid->e[1])
    +pid->ki*pid->e[0]+pid->kd*(pid->e[0]-2*pid->e[1]+pid->e[2]);
    else pid->u+=pid->kp*(pid->e[0]-pid->e[1])
    +pid->kd*(pid->e[0]-2*pid->e[1]+pid->e[2]);
    .....
}
```

系统实例同上节,输入为阶跃信号,调用积分分离算法的控制程序,将运行结果送入 MATLAB 进行仿真,得到输出曲线如图 3 所示,可知程序实现了积分分离 PID 控制规律。

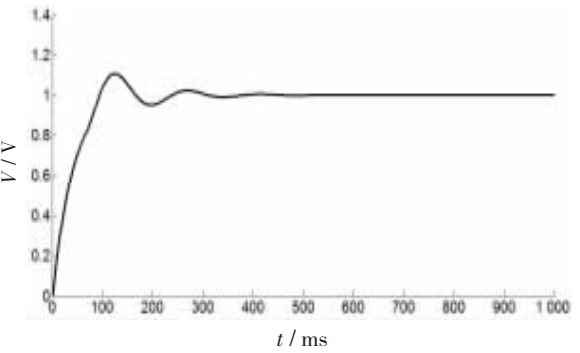


图 3 积分分离的 PID 控制系统输出曲线

将两种 PID 算法的仿真结果进行比较(表 1)可以看出,采用积分分离算法后,系统的调节时间变短,振荡次数减少,并减少了超调量,控制效果优于普通 PID 控制。

表 1 两种 PID 算法控制效果比较

	调节时间 /ms	超调量	振荡次数
普通 PID	620	1.54	4
积分分离 PID	530	1.11	3

4 结束语

PID 控制是工程领域常用的控制方法,其结构简单、稳定性好、工作可靠、调整方便。随着嵌入式处理器的发展,体积越来越小,运算速度越来越快,这使得 PID 控制可以在嵌入式处理器上实现。PID 的计算机控制必须先将模拟 PID 算法离散化;在程序设计时,使用 C 语言的结构体数据类型,组织好运算中的多个变量,使它们成为一个整体,使程序结构清晰;使用函数将 PID 运算过程进行封装,用求和代替积分,用差分代替微分,使用数组记录 3 个采样周期的误差,通过迭代处理好被控对象输入和输出的关系,最终完成 PID 运算。系统工作时,使用 STM32 定时器,每个采样周期调用一次 PID 运算函数,输出运算数据。应用 MATLAB 对输出的数据进行仿真,结果表明,该算法实现了 PID 控制;并将积分分离算法应用于 PID 计算,改善了系统的动态特性、消除了静差和提高了精度。在工程实际系统设计中,只需要建立起被控对象输入量和输出量之间的变换关系,就可以将控制器输出量转化为执行机构的动作量,从而使程序真正得以实用化。

参考文献:

[1] 何东健,刘忠超,范灵燕.基于 MATLAB 的 PID 控制器参数整定及仿真[J]. 西安科技大学学报,2006,12(4):511-514.  
[2] 任俊杰,李永霞,李媛.基于 PLC 的闭环控制系统 PID 控制器的实现[J]. 制造业自动化,2009,31(4):20-23.  
[3] 刘道兴.PID 控制及程序实现方法[J]. 内江师范学院学报,2005,20(6):20-23.  
[4] 毛雪莲. PID 算法的 C 语言实现 [J]. 电脑知识与技术, 2016, 12(14):212-215.  
[5] 卢亚平.基于 C 语言的数字 PID 控制算法及实现[J]. 科技创新新报,2010,30:24-25.  
[6] 雷慧杰.基于 STM32 的直流电机 PID 调速系统设计[J].现代电子技术,2016,39(8):165-167.  
[7] 李元春.计算机控制系统[M]. 2 版.北京:高等教育出版社,2009:213-216.  
[8] 刘火良,杨森.STM32 库开发实战指南[M].北京:机械工业出版社,2013:105-121.  
[9] 张红涛,闫林,徐晓辉,等.基于单片机 PID 算法的无刷直流电机控制系统的研究[J]. 现代电子技术,2007,10:52-54.  
[10] 贾诚安,叶林,葛俊锋.一种基于 STM32 和 ADS1248 的数字 PID 温度控制系统[J]. 传感器与微系统,2015,34(11):103-105.