

文章编号: 1671-6833(2004)01-0077-05

基于Smith 预估的纯滞后系统的控制

朱晓东, 王 军, 万 红

(郑州大学电气工程学院, 河南 郑州 450002)

摘要: 针对具有纯时间滞后的系统, 分析了PID 算法、Dahlin 算法和Smith 预估算法的特点, 指出只要对Smith 预估控制器进行适当的设计, 就能等效于 Dahlin 算法. 提出了利用Smith 预估器补偿时间滞后, 按 Dahlin 算法原理设计控制器的 Dahlin-Smith 算法和将Smith 预估系统结构变换为内模控制器(IMC) 结构, 再按内模控制原理进行控制器设计的IMC-Smith 算法. 在模型匹配和失配情况下进行了仿真研究, 结果表明, 所提出的新控制算法在系统模型不精确的情况下, 具有更好的稳定性和鲁棒性, 对于大时间滞后系统是一种比较实用的控制方法.

关键词: 纯滞后; Dahlin 算法; Smith 预估; 内模控制(IMC)

中图分类号: TP 273+.5

文献标识码: A

0 引言

在多数工业过程当中, 控制对象普遍存在纯时间滞后的现象, 如化工、热工过程等. 这种滞后时间的存在, 对系统的控制是极为不利的, 严重时甚至会破坏系统的稳定性. 滞后时间与系统时间常数的比值越大, 其对系统性能的影响也就越大. 因此长期以来, 纯滞后系统就一直是工业过程中的难控制系统, 人们也对它进行了大量的研究. 对于纯滞后系统, 常用的控制方法主要是PID 控制、Dahlin 控制算法和Smith 预估算法^[1], 本文对这几种控制方法进行了仿真, 比较了它们的特点, 分析了其内在的联系, 同时给出了将 Dahlin 算法和 Smith 预估相结合的控制算法和内模预估算法.

1 纯滞后系统常用的控制算法

1.1 PID 控制算法^[2]

在工业过程控制中, 应用最广泛的就是PID 控制器, 即比例(P)、积分(I)、微分(D)控制器. 其结构简单, 参数易于调整, 在长期应用中积累了丰富的经验. PID 控制算法的原理可以表示为:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt},$$
控制器的传递函数为 $G_c(s) = K_p + K_i/s + K_d s$.

PID 控制器的参数 K_p , K_i 和 K_d 可以根据过

程的动态特性进行整定^[3], 其整定方法很多, 如工程整定法、理论值计算法、经验法等, 在工程中一般采用实验经验法, 如 Ziegler-Nichols 整定公式^[4]、Cohen-Coon 整定公式^[5]等等.

1.2 Dahlin 控制算法^[6]

Dahlin 算法是克服纯滞后的主要控制方法之一, 其设计原理是使整个闭环系统所期望的传递函数等价于一个延迟环节和一个惯性环节相串联, 即

$$G(s) = \frac{K}{(Ts + 1)} e^{-\tau s}$$
式中, K 为系数; T 为时间常数; τ 为时滞常数. 控制系统结构如图 1 所示.

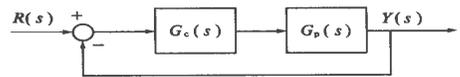


图 1 Dahlin 控制系统的结构图

Fig. 1 Block diagram of Dahlin control system

图 1 中, $G_c(s)$ 为控制器传递函数. 设被控对象的数学模型为 $G_p(s) = \frac{K}{Ts + 1} e^{-\tau s}$, 按照 Dahlin 控制算法的原理, 整个闭环系统所期望的传递函数 $G(s)$ 为

$$G(s) = \frac{1}{(Ts + 1)} e^{-\tau s} \quad (1)$$

按图 1 所示, 其闭环系统的传递函数为

收稿日期: 2003-10-07; 修订日期: 2003-12-18

基金项目: 河南省自然科学基金资助项目(0111060300)

作者简介: 朱晓东(1970-), 男, 河南安阳市人, 郑州大学讲师, 硕士, 主要从事计算机控制方面的教学和研究.

$$G(s) = \frac{Y(s)}{R(s)} = \frac{G_p(s) G_c(s)}{1 + G_p(s) G_c(s)} \quad (2)$$

使式(1)和式(2)相等,可以求出 Dahlin 控制器 $G_c(s)$ 的传递函数:

$$\begin{aligned} G_c(s) &= \frac{G(s)}{G_p(s)(1-G(s))} \\ &= \frac{1}{K} \frac{(T_s + 1)e^{-(\tau - T_1)s}}{(T_1 s + 1)e^{-T_1 s}} \end{aligned} \quad (3)$$

式中: T 为对象的时间常数; K 为对象的增益, τ 为对象的时间滞后常数; T_1 为期望闭环传递函数的时间常数; τ_1 为期望闭环传递函数的时间滞后常数. 当时间趋向 $+\infty$ 时, 即 $s \rightarrow 0$, 有 $G_c(0) \rightarrow \infty$, 可见控制器中含有积分环节, 可以有效地消除误差. 取 $T_1 = aT$, $\tau = \tau_1$, 式(3)变为

$$G_c(s) = \frac{1}{K} \frac{T_s + 1}{aTs + 1 - e^{-s}} \quad (4)$$

通过调节 a 值, 可以调节控制器的参数, 这样设计的控制器, 只要原控制对象是稳定的, 则闭环系统就一定是稳定的.

设 $U(s)$ 为控制器的输出, $E(s)$ 为控制器的输入, 则式(4)可以变换为

$$G_c(s) = \frac{U(s)}{E(s)} = \frac{1}{K} \frac{T_s + 1}{aTs + 1 - e^{-s}}$$

变换有 $U(s) = \frac{1+Ts}{aTKs} E(s) - \frac{1}{aTs} (1 - e^{-s}) U(s)$, 由此可见, 式中的第一部分相当于一个 PI 调节器, 第二部分可以看作是对滞后部分的预估项, 正是这一预估项, 使得 Dahlin 算法能够有效地克服系统的纯滞后影响.

1.3 Smith 预估算法^[4]

Smith 预估算法是克服纯滞后的另一个有效的控制方法. 其基本原理是通过预估对象的动态特性, 用一个预估模型来进行时间滞后的补偿, 补偿器与被控对象共同构成一个没有时间滞后的广义被控对象. 这样, 控制器相当于对一个没有时间滞后的系统进行控制, 从而有效地克服了纯滞后的影响. 其结构图如图 2 所示.

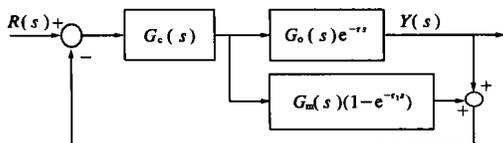


图 2 Smith 预估控制系统的结构图

Fig. 2 Block diagram of Smith predictor control system

图中 $G_o(s)$ 为对象中不含时间滞后的部分, $G_m(s)(1 - e^{-\tau s})$ 为 Smith 预估器. 理想情况下

$G_o(s) = G_m(s)$, $\tau = \tau_1$, $G_c(s)$ 为控制器. 由上图可以得到系统的闭环传递函数为

$$\begin{aligned} G(s) &= \frac{G_c(s) G_o(s) e^{-s}}{1 + G_m(s) G_c(s) + G_o(s) [G_c(s) e^{-s} - G_m(s) e^{-\tau s}]} \\ &= \frac{G_c(s) G_o(s) e^{-s}}{1 + G_m(s) G_c(s)} \end{aligned} \quad (5)$$

从式(5)可见, 时间滞后部分 e^{-s} 已不包含在系统的特征方程里, 闭环系统等效为无时滞部分与时滞部分的串联. 当预估模型与控制对象一致时, Smith 预估控制算法从理论上提供了将含有纯滞后的对象简化为不含纯滞后的对象, 进而再进行控制的方法, 其过渡过程的曲线及其性能指标均与对象特性为 $G_o(s)$ (不含纯滞后的对象部分) 时完全相同, 仅仅是在时间上有一个 τ 的延迟.

由于通过 Smith 预估器的补偿, 广义被控对象中已不含有时滞部分, 因此控制器 $G_c(s)$ 可以采用常规的 PID 控制器. 在实际控制当中, 常常将 Smith 预估器与控制器进行连接, 这样其等效的系统结构图如图 3 所示.

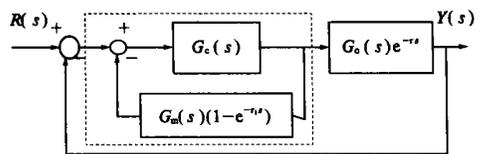


图 3 Smith 预估控制系统的结构图

Fig. 3 Block diagram of Smith predictor control system

1.4 Dahlin-Smith 控制算法

当通过 Smith 预估器对含有时滞的对象进行补偿之后, 再按照 Dahlin 控制器的原理设计控制器, 便可以构成 Dahlin-Smith 控制算法. 设闭环系统的传递函数为式(5), 按照 Dahlin 控制的原理, 令其等效为一个滞后环节和一个惯性环节相串联, 即

$$G(s) = \frac{G_o(s) G_c(s)}{1 + G_m(s) G_c(s)} e^{-s} = \frac{1}{1 + aTs} e^{-\tau_1 + s}$$

其中, T 为对象的时间常数; $G_o(s)$ 为不含纯滞后的对象部分; $G_m(s)$ 为不含时间滞后的对象预测模型, 令 $G_o(s) = G_m(s) = \frac{K}{T_s + 1}$, 因此有:

$$\begin{aligned} G_c(s) &= \frac{e^{-(\tau_1 - \tau)s}}{G_o(s) (aTs + 1 - e^{-\tau_1 - \tau s})} \\ &= \frac{1}{K} \frac{(T_s + 1) e^{-(\tau_1 - \tau)s}}{(aTs + 1) e^{-\tau_1 - \tau s}} \end{aligned} \quad (6)$$

$$U(s) = \frac{1+Ts}{aTKs} e^{-(\tau_1 - \tau)s} E(s) - \frac{1}{aTs} (1 - e^{-(\tau_1 - \tau)s}) U(s)$$

这就是 Dahlin-Smith 控制算法的控制策略, 此时控制器等效为一个带有滞后的 PI 控制器和一个预测项, 其控制策略类似于 Dahlin 算法, 实际上此算法也揭示了 Dahlin 算法和 Smith 预估算法之间的内在联系. 当图 3 的 Smith 预估控制系统中控制器 $G_c(s)$ 取为式 (6), $\tau = \tau_1$ 时, 图 3 中的虚框部分就相当于一个 Dahlin 控制器, 其传递函数与 Dahlin 控制器的传递函数相同, 因此可以说 Dahlin 算法是 Smith 预估算法中的一例. Smith 预估算法中只要对控制器进行适当的设计, 就能实现 Dahlin 算法的功能.

1.5 Smith 预估的内模控制算法(IMC-Smith)

Garcia 于 1982 年提出的内模控制(Internal Model Control)^[3]方法主要用于预测控制系统, 由于其设计简单、跟踪调节性能好、鲁棒性强、能够消除各种不可测干扰的影响, 因此得到了广泛的应用. 而从内模控制的原理分析, 对于纯时滞系统, 内模控制也具有有良好的适应性.

1.5.1 内模控制的基本原理^[3]

IMC 的系统结构如图 4 所示, 图中 $G_p(s)$ 为被控对象, $G_m(s)$ 为内部估计模型, $G_f(s)$ 为前向通道的滤波器, $G_{imc}(s)$ 为内模控制器, $R(s)$ 为输入信号, $Y(s)$ 为输出信号, e_m 为系统输出与模型输出的误差.

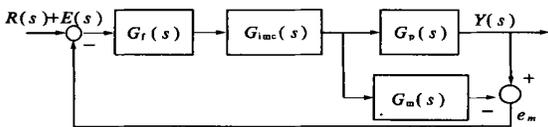


图 4 内模控制系统结构图

Fig. 4 Block diagram of IMC-Smith control system

分析图中结构可知, 系统的闭环传递函数为 $G(s) = \frac{G_{imc}(s) G_f(s) G_p(s)}{1 + G_{imc}(s) G_f(s) [G_p(s) - G_m(s)]}$, 在不考虑扰动的情况下, 当模型与实际对象匹配时, e_m 为零, 系统呈现开环特性, 系统输出为 $Y(s) = G_{imc}(s) G_f(s) G_p(s) R(s)$, 所以只要控制器和对象是稳定的, 则系统就是稳定的. 当模型与对象失配时, 系统呈闭环特性, 反馈信息 e_m 中不仅包含扰动量信息, 还有模型失配的一些信息, 系统的稳定性和鲁棒性可通过 $G_{imc}(s)$ 的设计来保证, 从而有利于控制系统的抗扰设计, 增强系统的鲁棒性.

由图 4 分析可知, 系统闭环误差传递函数为

$$E(s) = R(s) - Y(s) =$$

$$\frac{1 - G_{imc}(s) G_f(s) G_m(s)}{1 + G_{imc}(s) G_f(s) [G_p(s) - G_m(s)]} R(s),$$

当把控制器设计为 $G_{imc}(s) = [G_f(s) G_m(s)]^{-1} F(s)$ 时 ($F(s)$ 为可实现因子), 有 $E(s) = 0$, 即系统的稳态误差为零^[3]. 由此可以看到, 内模控制是一种结构简单, 控制效果显著的控制方法.

1.5.2 Smith 预估控制器的内模控制算法

对于如图 2 所示的 Smith 预估控制系统, 可以看到它与内模控制结构很类似, 当我们将图 2 变换为图 5, 就可以得到 Smith 预估控制系统的内模控制结构, 然后按照内模控制的方法设计 Smith 预估控制器.

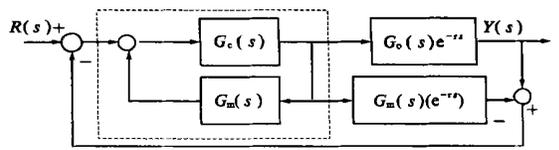


图 5 等效的 Smith 预估控制系统内模控制结构图

Fig. 5 Equivalent block diagram of Smith predictor control system

其中, $G_m(s)$ 为内部估计模型, $G_m(s) = \frac{K_m}{1 + T_m s}$, 不含时间滞后的控制对象传递函数为:

$$G_o(s) = \frac{K}{1 + T_B s}$$

由图可得等效的内模控制器传递函数为: $G_{imc}(s) = \frac{G_c(s)}{1 + G_m(s) G_c(s)}$, 由此可得到控制器的传递函数为: $G_c(s) = \frac{G_{imc}(s)}{1 - G_{imc}(s) G_m(s)}$, 分别代入函数 $G_{imc}(s)$ 和 $G_m(s)$, 取可实现因子 $F(s) = \frac{1}{1 + \lambda s}$, 则控制器传递函数为: $G_c(s) = \frac{G_{imc}(s)}{1 - G_{imc}(s) G_m(s)} = \frac{T_m s + 1}{K_m (\lambda + 1)} \cdot \frac{\lambda + 1}{\lambda} = \frac{1}{\lambda} \cdot \left(\frac{T_m s + 1}{K_m s} \right)$.

可见, Smith 预估控制器可以等效为一个 PI 控制器的设计. 当估计模型与对象模型一致时, e_m 为零, 系统仅在控制器上构成闭环; 当估计模型与对象模型失配时, 则系统的反馈信号中包含了误差信息与失配信息, 可通过控制器的 λ 参数来进行调节, 提高控制系统的稳定性和鲁棒性.

2 算法仿真

为了检验控制的效果, 利用 MATLAB 对这三种算法进行了仿真^[9]. 控制对象为实验室中的电

阻炉加热的温度系统,通过阶跃响应的辨识方法可以得到对象的传递函数模型为: $G_p(s) = Ge^{-\tau s} / (1 + Ts)$,其中:增益 $K = 30$,时间常数 $T = 600s$,滞后时间常数 $\tau = 300s$.分别采用PID 控制算法、Dahlin 控制算法、Smith 预估算法、Dahlin - Smith 控制算法以及IMC - Smith 控制算法进行仿真.设定目标值为 $100^\circ C$.仿真时为了更符合实际工业状况,对控制器的输出进行了限幅.这是因为在实际中由于执行机构的动作不可能无限大,也不允许过于频繁,因此控制器的输出范围和变化频率都不能过大,仿真中取控制器输出范围为 $-5 \sim 5$,等同于实际控制中的控制器输出电压值;同时为了防止积分饱和,对有积分的控制器均采用了积分分离措施,以提高控制效果.

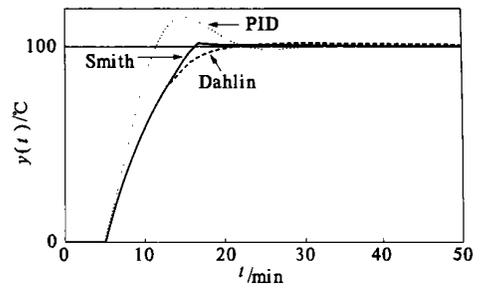
2.1 控制器参数

PID 控制器的参数在通过Ziegler - Nichols 法得到的基础上进行了微调: $K_p = 0.06, K_i = 0.00008, K_d = 4$;同样用上述方法得到Smith 预估PID 控制器参数为: $K_p = 4, K_i = 0.02, K_d = 6$;通过调节,式(4)中Dahlin 控制器的参数 $\alpha = 0.45$,为便于实现,时间滞后项用一阶pade 式来近似 $P(s) = \frac{1 - \tau s/2}{1 + \tau s/2}$,这样控制器为 $G_c(s) = \frac{90000s^2 + 750s + 1}{40500s^2 + 570s}$;Dahlin - Smith 控制器的参数 $\alpha = 1.15, \tau = 305$;IMC - Smith 控制器的参数为 $\lambda = 25$.

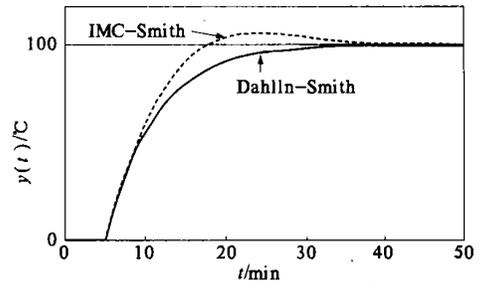
2.2 仿真结果

首先,保持控制对象参数不变,即预测模型与实际模型参数一致,5种控制算法的控制曲线如图6所示.可见PID 算法有较大的超调量,Smith 预估算法和Dahlin 算法均没有超调,Dahlin - Smith 算法和IMC - Smith 算法的调节时间较长.

保持控制器和估计模型的参数不变,改变对象参数和滞后时间,使估计模型与之失配,即将控制对象的参数改变为 $K = 40, T = 400s, \tau = 390s$,5种控制算法的控制曲线如图7所示.从图中可知,PID 算法出现了等幅振荡;Smith 预估算法也有剧烈的宽幅震荡,系统的稳定性被破坏;Dahlin 算法虽然能保持系统的稳定性,但是振荡时间过长;而Dahlin - Smith 算法和IMC - Smith 算法不仅能够保持系统的稳定,而且振荡次数少(仅两次振荡),收敛速度较快,表明其较好的稳定性和鲁棒性.



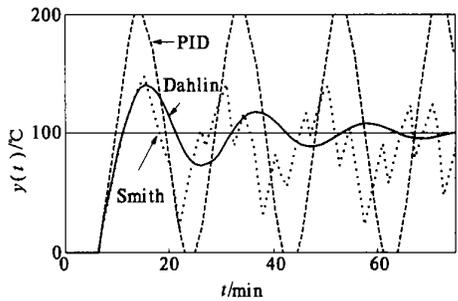
(a) PID、Dahlin、Smith算法仿真曲线



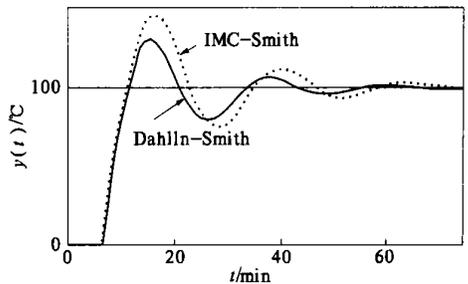
(b) Dahlin-Smith、IMC-Smith算法仿真曲线

图6 模型匹配时控制系统的仿真曲线

Fig. 6 The curve of control system with model matching



(a) PID、Dahlin、Smith算法仿真曲线



(b) Dahlin-Smith、IMC-Smith算法仿真曲线

图7 模型失配时控制系统的仿真曲线

Fig. 7 The curve of control system with model unmatching

3 结论

从以上分析中可以看到,对于具有大时间滞后系统的控制,PID 控制算法仅仅在模型匹配时能够进行稳定的控制,但动态性能不理想.当模型失配时,PID 算法就无能为力了.而Smith 预估算法则在模型匹配时具有好的性能指标,但是由于这种算法严重依赖模型的精确匹配,而在实际中

这是很难做到的,因此模型失配时,Smith 预估算法就难以取得良好的控制效果.相对而言,Dahlin 算法的稳定性和鲁棒性比较好,在模型失配时也能够进行有效地控制,只是振荡过程要长一些.而将Dahlin 算法和Smith 预估算法相结合的 Dahlin - Smith 算法对大时间滞后系统的控制效果比较理想,在模型匹配时,能够保持一定的动态性能(虽然不是最理想的);而在模型失配时则能够快速使系统收敛.IMC - Smith 算法与 Dahlin - Smith 算法类似,只是其振荡幅度要稍大些.这两种算法由于均在反馈信息中包含了模型失配的信息,然后通过控制器进行调节,因此相对于其他算法其鲁棒性能大大增强.

仿真结果表明,IMC - Smith 算法与 Dahlin - Smith 算法在模型失配时,具有良好的稳定性和鲁

棒性,对于大时间滞后系统是一种比较实用的控制方法.

参考文献:

- [1] 王宏华,樊桂林.含有纯滞后系统的控制方法[J].江苏理工大学学报,1994,15(6):87~92.
- [2] 谢剑英.计算机控制技术[M].北京:科学技术出版社,1983.
- [3] 师黎,丁海.PID 控制的参数模糊自整定方法[J].郑州大学学报(工学版),2001,22(3):25~27.
- [4] 何克忠.计算机控制技术[M].北京:清华大学出版社,1981.
- [5] 舒迪前.预测控制系统及其应用[M].北京:机械工业出版社,1998.
- [6] 楼顺天.基于 Matlab 的系统分析与设计[M].北京:电子工业出版社,2000.

Control of the System with Pure Time Delay Based on Smith Predictive Algorithm

ZHU Xiao - dong , WANG Jun , WAN Hong

(College of Electrical Engineering ,Zhengzhou University ,Zhengzhou 450002,China)

Abstract : The characteristic of some control algorithms for the system with pure time delay is analyzed .They are PID , Dahlin and Smith predictor algorithm .The analysis shows that Smith predictor algorithm is equivalent to Dahlin algorithm if Smith predictor controller is designed properly .Two new algorithms are discussed .One is Dahlin Smith algorithm which is designed according to the principle of Dahlin algorithm on the base of Smith predictor .Another one is IMC Smith algorithm whose Smith predictor controller is designed according to IMC structure .The simulation of these control algorithms is studied under the condition of model matching and model unmatching .The result shows that the new control algorithms have a better stability and robustness .And they are appropriate control methods for the system with pure time delay .

Key words : pure time delay ; Dahlin algorithm ; Smith predictive algorithm ; internal model control algorithm(IMC)