Open access Journal International Journal of Emerging Trends in Science and Technology

Temperature Control in HVAC Application using PID and Self-Tuning Adaptive Controller

Authors

Swarup D. Ramteke[1], Bhagsen J. Parvat[2]

[1]M.E, Instrumentation & Control Engineering, Pravara Rural Engineering College, Loni
Savitribai Phule Pune University, India

[2]Associate Professor, Instrumentation & Control Engineering, Pravara Rural Engineering College, Loni
Savitribai Phule Pune University, India

Abstract

Conventional Proportional Integral Derivative Controllers are used in many industrial applications due to their simplicity and robustness. The parameters of the various industrial processes are subjected to change due to change in the environment. These parameters may be categorized as steam, pressure, temperature of the industrial machinery in use. Various process control techniques are being developed to control these variables. In this paper, the temperature of a boiler is controlled using conventional PID controller and then optimized using Self-Tuning Adaptive controller. The comparative results show the better results when Self-Tuning Adaptive controller is used.

Keywords: *HVAC, PID controller, Self-Tuning Adaptive controller-STC*

1. Introduction

The Proportional-Integral-Derivative (PID) controllers have been the most commonly used controller in process industries for over 50 years even though significant development have been made in advanced control theory. According to a survey conducted by Japan Electric Measuring Instrument Manufacturers Association in 1989, 90 % of the control loops in industries are of the PID type. The proportional action adjusts controller output according to the size of the error, the integral action eliminates the steady state offset and the future is anticipated via derivative action. These useful functions are sufficient for a large number of process applications and the transparency of the features lead to wide acceptance by the users. Strength of the PID controller is that it also deals with important practical issues such as actuator saturation and integrator windup. PID controllers perform well for a wide class of processes and they give robust performance for a wide range of operating conditions and are easy to implement using analog or digital hardware. Moreover, due to process uncertainties, a more sophisticated control scheme is not necessarily more efficient than a well tuned PID controller [1].

The concept of intelligent control lies with the fact that adaptation of living organisms is imbibed in to the controller architecture so that adaptation can be emulated in the control decision. Originally, adaptation was displayed only by plants and animals, where it is seen in its most varied forms. It is a characteristic of living organisms that they adapt their behavior to their environment even where it is harsh. Each adaptation involves a certain loss for the organism, whether it is material, energy or information. After repeated adaptations to the same changes, plants and animals manage to keep such losses to a minimum. Repeated adaptation is, in fact, an accumulation of experiences that the organism can evaluate to minimize the losses involved in adaptation.

Alongside such systems found in nature there are also technical systems capable of adaptation. These vary greatly in nature, and a wide range of mathematical tools are used to describe them.

Adaptive control systems adapt the parameters or structure of one part of the system (the controller) to changes in the parameters or structure in another part of the system (the controlled system) in such a way that the entire system maintains optimal behavior according to the given criteria Independent of any changes that might have occurred.

The field of adaptive control has undergone significant development in recent years. The aim of this approach is to solve the problem of controller design, for instance where the characteristics of the process to be controlled are not sufficiently known or change over time. Several approaches to solving this problem have arisen. One showing great potential and success is the so-called self-tuning controller (STC). This approach to adaptive control is based on the recursive estimation of the characteristics of the system and disturbances and updating the estimates, so monitoring possible changes. This kind of controller, which identifies unknown processes and then synthesizes control (adaptive control with recursive identification), is referred to in the literature as a self-tuning controller – STC [2].

2. Proportional-Integral-Derivative Controller

A proportional–integral–derivative controller (PID controller) is a generic control loop feedback mechanism (controller) widely used in industrial control systems – a PID is the most commonly used feedback controller. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs. In the absence of knowledge of the underlying process, PID controllers are the best controllers. However, for best performance, the PID parameters used in the calculation must be tuned according to the nature of the system – while the design is generic, the parameters depend on the specific system. The PID controller calculation (algorithm) involves three separate parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. The proportional value determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element. Heuristically, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. By tuning the three constants in the PID controller algorithm, the controller can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the set point and the degree of system oscillation. Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability. Some applications may require using only one or two modes to provide the appropriate system control. This is achieved by setting the gain of undesired control outputs to zero. A PID controller will be called a PI, PD, P or I controller in the absence of the respective control actions. PI controllers are fairly common, since derivative action is sensitive to measurement noise, whereas the absence of an integral value may prevent the system from reaching its target value due to the control action [1].

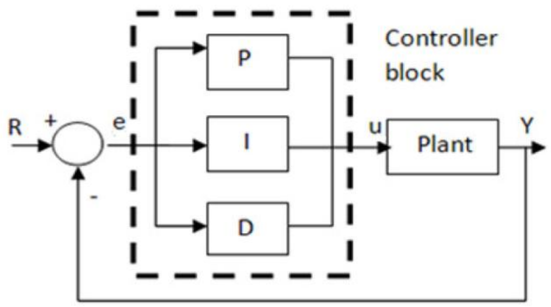

Figure 1: Block diagram of PID controller

3. Self-Tuning Adaptive Controller

The term self-tuning was used to express the property that the controller parameters converge to the controller that was designed if the process was known. Self-tuning controllers belong by their characteristics to the family of adaptive controllers. The aim of adaptive controllers is to solve control problems in cases where the characteristics of the system are unknown or time varying, as with boiler flow. The principle of adaptive control is to change the controller characteristics on the basis of the process change. Typically as with the self-tuning method utilized in this paper, the recursive identification processes is utilized. The task of online adaptive control is to maintain the optimal parameters of a difficulty to control process with time varying characteristics. This presents a complex process concisely explained in the following 3 step cyclic repetition.

1. The process parameters are assumed to be known for current control loop and equal to their current estimation
2. The control strategy is designed based on the previous assumption and controller output is calculated.
3. The following identification step is performed after obtaining new controlled process variables. The parameters of the controlled process are recalculated using Recursive Least Square Method in this case (other Recursive methods can of course be utilized) [3].

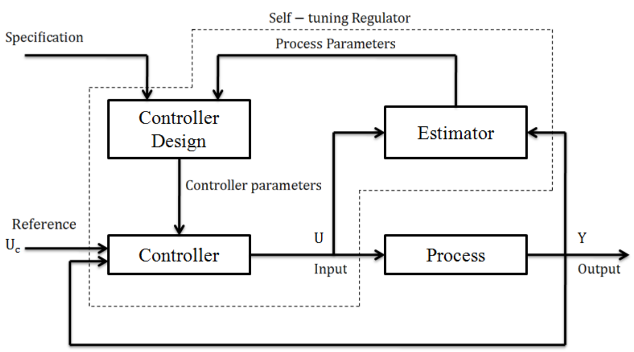

Figure 2: Block diagram of Self-Tuning Adaptive Controller

4. Problem Formulation And Mathematical Modeling

A boiler of an HVAC Application is taken as a case study and the temperature control of the boiler is achieved using conventional PID controller and STC. The comparison of both the controller performance is analyzed in this paper.

Set point

Set point of Temperature = 280 Degree Celsius

The boiler is mathematically modeled using experimental data available and the transfer function of the system is achieved as,

$$G(s) = \frac{5(s+1)}{s(s+1)(s+6)}$$

5. Pid Controller Design And Tuning

The equation of ideal PID controller is

$$u(t) = k_p[e(t) + \frac{1}{\tau_i}\int_0^1 e(t)dt + \tau_d \frac{de(t)}{dt}]$$

$$u(s) = k_p\left[1 + \frac{1}{\tau_i s} + \tau_d s\right]e(s)$$

$$u(s) = k_p\left[\frac{1 + \tau_i s + \tau_i \tau_d s^2}{\tau_i s}\right]e(s)$$

The Ziegler- Nichols (Z-N) methods rely on open-loop step response or closed-loop frequency response tests. A PID controller is tuned according to a table based on the process response test. According to Zeigler-Nichols frequency response tuning Criteria.

$$k_p = 0.6\, k_{cu}, \tau_i = 0.5\, T, \tau_d = 0.125\, T$$

For the PID controller, the values of tuning parameters obtained are K_p =32, T_I =1.5, T_D =0.29 and P= 32, I= 21.2, D=9

Usually, initial design values of PID controller obtained by all means needs to be adjusted repeatedly through computer simulations until the closed loop system performs or compromises as desired. This stimulates the development of "intelligent" tools that can assist the engineers to achieve the best overall PID control for entire operating envelops [1].

6. Boiler Control Using Self-Tuning Adaptive Controller

PID controller is a standard control structure for classical control theory. But the performance is greatly distorted and the efficiency is reduced due to nonlinearity in the process plant.

The Ideal Textbook version of a continuous-time PID controller is usually given in the form,

$$u(t) = K_p\,[e(t) + \frac{1}{T_1}\int_0^t e(\tau)d\,\tau + T_D\,\frac{de(t)}{dt}] \qquad (1)$$

Using the Laplace transform it is possible to convert Equation (1) into the form

$$U(s) = K_p\,[\,1 + \frac{1}{T_1 s} + T_D s\,] \qquad (2)$$

From equation (2) we can determine the transfer function of the PID controller

$$G_R(s) = \frac{U(s)}{E(s)} = K_p\,[\,1 + \frac{1}{T_1 s} + T_D s\,]$$

To obtain a digital version of a continuous-time PID controller we must discretize the integral and derivative components of Equation (1). The simplest algorithm is obtained by replacing the derivative with a difference of the first-order (two-point, backward difference).

$$\frac{de}{dt} \approx \frac{e(k)-e(k-1)}{T_0} = \frac{\Delta e(k)}{T_0}$$

Where e(k) is the error value at the k-th moment of sampling, i.e. at time t = kT_0.

Using the so-called Backward Rectangular method (BRM) yields

$$\int_0^t e(\tau)d\tau \approx T_0\sum_{i=1}^{k} e(i)$$

So that the equation for a discrete PID controller has the form

$$u(k) = k_p\left\{e(k) + \frac{T_0}{T_1}\sum_{i=1}^{k} e(i) + \frac{T_D}{T_0}\,[e(k) - e(k-1)]\right\} \qquad (3)$$

for steps *k and k − 1, we obtain the recurrent relation*

$$u(k) = \Delta u(k) + u(k-1)$$

$$\Delta u(k) = k_p\{e(k) - e(k-1) + \frac{T_0}{T_1}e(k) + \frac{T_D}{T_0}[e(k) - 2ek-1+ek-2\}$$

and in general form

$$u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) + u(k-1) \qquad (4)$$

The incremental algorithm deduced from Equation (3) (BRM)

$$u(k) = k_p\{e(k) - e(k-1) + \frac{T_0}{T_I}e(k) + \frac{T_D}{T_0}[e(k) - 2ek-1+ek-2\}+u(k-1) \qquad (5)$$

Comparing with the general form, we get

$$q_0 = k_p(1 + \frac{T_0}{T_I} + \frac{T_D}{T_0})$$

$$q_1 = -k_p(1 + \frac{2T_D}{T_0})$$

$$q_2 = k_p\frac{T_D}{T_0}$$

Instead of using error $e(k)$ in the derivative component, we can use process output $y(k)$ to decrease the larger changes in the controller output resulting from set point changes. In this case algorithm (Eq. 5) has the form

$$u(k) = k_p\{e(k) - e(k-1) + \frac{T_0}{T_I}e(k) + \frac{T_D}{T_0}[2y(k-1) - yk-yk-2\}+u(k-1) \qquad (6)$$

In this way we can achieve a significant decrease in the controller output at the moment of a set point change and then a decrease in the limitation on the controller output and the movements of the final control element into an area of nonlinearity. Usually, the rise time of the process output is slowed down and overshoot is significantly decreased while the settling time remains roughly the same. The adjustment of the parameters for controller (Eq.6) to changes in control and disturbance differs little from the adjustment of a controller using error in the derivation.

Changes in controller output amplitude decrease further if the reference signal $w(k)$ is substantial only in the integral component.

$$u(k) = k_p\{-y(k) + y(k-1) + \frac{T_0}{T_I}[w(k) - y(k)] + \frac{T_D}{T_0}[2y(k-1) - y(k) - y(k-2)]\} + u(k-1) \qquad (7)$$

Changing the process output to the reference signal is then mainly regulated by the integral component. This can, however, be a fairly slow process. To decrease larger changes in the controller output (as a result of the reference change) it capacity filter, or a change limiter, or to employ term $\beta w(k) - y(k)$ instead of term $w(k) - y(k)$, in the proportional component, where weighting factor β is determined by the dynamics of the system and is chosen from the interval $0 < \beta < 1$. It was proved that a good characteristic of the process dynamics is the so-called normalized gain k, which is defined as the product of the gain of the controlled process k_s and critical proportional gain k_{pu}, where the control loop is on the point of stability.

$$k = k_s k_{pu} \quad (8)$$

Then it is possible to change PID controller parameters k_p , T_I and T_D in relation to the size of normalized gain k. In order to reduce the maximum overshoot of the process output, the reference signal w in the proportional component (Eq.8) can be weighted using the factor β so that a change of the normalized gain k is achieved.

The proportional part of controller,

$u(k) = u_p(k) + u_I(k) + u_D(k)$ then takes the form,

$$u_p(k) = k_p[\beta w(k) - y(k)]$$

As a result, the following continuous-time controller algorithm was developed which, as well as using weighting factor β, also makes use of single capacity filter $D(s) = k_p \frac{T_D s}{T_f s+1} E(S)$, $T_f = \frac{T_D}{\alpha}$, $\alpha \in (3 ; 20)$ to filter the derivative component

$$u(t) = K_p\,[\beta w(t) - y(t) + \frac{1}{T_1}\int_0^t e(\tau)d\tau - T_D\,\frac{dy_f}{dt}] \quad (9)$$

Where $y_f(t)$ is the process output filtered by first-order transfer function

$$\frac{y_f(s)}{y(s)} = \frac{1}{1+s\frac{T_D}{\alpha}} \quad (10)$$

where the filter constant α is selected from the interval. The equation for a digital incremental PID controller, taken from equations (9) and (10) after replacing the derivation of the first differential and the approximation of the integral, has the form given.

$$u(k) = u_{pI}(k) + u_D(k)$$

Where,

$$u_{PI}(k) = k_p[y(k-1) - y(k)] + \frac{k_p T_0}{2T_I}[e(k) + e(k-1)] + \beta k_p[w(k) - w(k-1)] + u_{PI}(k-1)$$

$$u_D(k) = k_p \frac{T_D \alpha}{T_D + T_0 \alpha}[y(k-1) - y(k)] + \frac{T_D}{T_D + T_0 \alpha} u_D(k-1)$$

6.1 Parameter Estimation

Parameters estimation is a key element is a self-tuner and is performed on-line. The model parameters are estimated based on the measurable process input, process output, and state signals, a number of recursive parameter estimation schemes are employed for self tuning control.It is described by the following transfer function

$$G(z) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}} z^{-d}$$

The estimated output of the process in step k ($\widehat{y_k}$) is computed on base of the previous process inputs u and outputs y according to the equation.

$$\widehat{y_k} = -\widehat{a_1} y_{k-1} - \cdots - \widehat{a_n} y_{k-n}$$

$$+\widehat{b_1} u_{k-d-1} + \cdots + \widehat{b_m} u_{k-d-m}$$

where $\widehat{a_1}$ ... $\widehat{a_n}$, $\widehat{b_1}$ ... $\widehat{b_m}$ are the current estimations of process parameters. This equation can be also written in vector form, which is more suitable for further work - see equation.

$$\widehat{y_k} = \theta_{k-1}^T . \phi_k$$

$$\phi_{k-1} = [\widehat{a_1} \ldots \widehat{a_n}, \widehat{b_1} \ldots \widehat{b_m}]^T$$

$$\phi_k = [-y_{k-1}, \ldots, -y_{k-n}, u_{k-d-1}, \ldots, u_{k-d-m}]^T$$

The vector θ_{k-1} contains the process parameter estimations computed in previous step and the vector ϕ_k contains output

and input values for computation of current output y_k .

6.2 Recursive Least Square Method

Least square methods are based on minimization of the sum of prediction errors squares:

$$J_k = \sum_{i=1}^{k} (y_i - \theta_k^T \phi_i)^2$$

Where yI is process output in i-th step and the product $\theta_k^T \phi_i$ represents predicted process output. Solving this equation leads to the recursive version of least square method where vector of parameters estimations is updated in each step according to equation

$$\theta_k = \theta_{k-1} + \frac{C_{k-1}.\phi_k}{1+\phi_k^T.C_{k-1}.\phi_k}(y_k - \theta_{k-1}^T\phi_k)$$

The covariance matrix C is then updated in each step as defined by the equation

$$C_k = C_{k-1} - \frac{C_{k-1}.\phi_k.\phi_k^T C_{k-1}}{1+\phi_k^T.C_{k-1}.\phi_k}$$

Initial value of matrix C determines influence of initial parameter estimations to the identification process.

7. Simulink Result

Simulink is a software package for modeling, simulating and analyzing dynamic systems. It supports linear and nonlinear systems, modeled in continuous time, sampled time, or a hybrid of the two. Boiler control using simulink is modeled as given below.

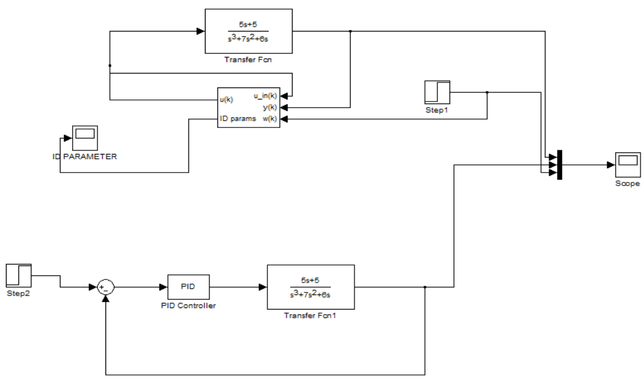

Figure 3: Simulink representation

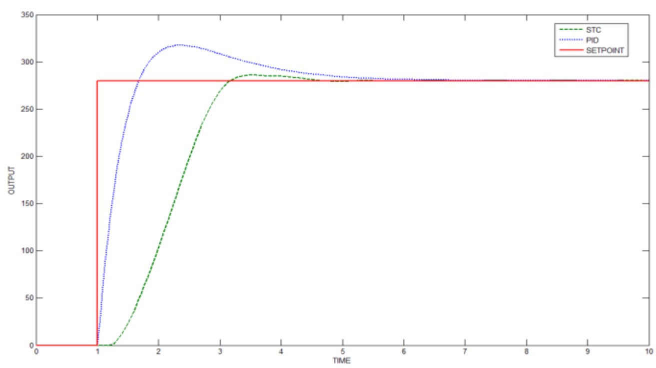

Figure 4: Comparison of PID and STC Response

Table: Comparison of Maximum overshoot and settling time for conventional PID controller and STC

Sr. No.	Parameter	PID	STC
1	Rise Time	0.4846	1.3045
2	Settling Time	4.7528	3.6526
3	Maximum Overshoot	13.4384 %	2.2020 %
4	Undershoot	0	0

8. Conclusion

In this paper a process control case study taking boiler has been implemented. First of all a mathematical model of the system is developed and a conventional PID controller is implemented on it. The PID controller gives a very high overshoot and high settling time. So, Self-Tuning and Recursive Least square method in the controller architecture is proposed and implemented. Results proved that Self-Tuning Adaptive controller gives a much better response than the conventional PID controller. In future scope we can implement this controller on chiller application and can observe the results.

Reference

1. Rahul Malhotra, Rajinder Sodhi (July 2011), “Boiler flow control using PID and fuzzy logic controller,”IJCSET
2. Digital Self-Tuning Controllers by V.Bobal, J.Bohm, J.Fessl and J.Machacek
3. K. J. Astrom, “Theory and applications of adaptive control,” *Automatica*, vol. 19, no. 5, pp. 471–486, 1983.
4. K. J. Astrom and B. Wittenmark, *Adaptive Control*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1989.
5. Marino, R. and P. Tomei (1995). Nonlinear Control Design-Geometric, Adaptive and Robust. Prentice-Hall, Hemel Hempstead, U.K.
6. Kokotovic, P. V. (Ed.) (1991). Foundations of Adaptive Control. Springer-Verlag, Berlin..

7. S. I. Niculescu and A. M. Annaswamy. A simple adaptive controller for positive-real systems with time-delay. In Proc. American Control Conference, Chicago, IL, February 2000.
8. S. Evesque. Adaptive Control of Combustion Oscillations. PhD thesis, University of Cambridge, Cambridge, UK, November 2000.