Evolving PID tuning rules

A brief history, starting with the earliest PID controllers to the most recent developments. There is more continuity than you might expect.

Willy K. Wojsznis, Terry Blevins

03/13/2013

Looking into the distant past of automation history, one can see the underlying concept of a PID controller exhibited in a nineteenth-century steam engine governor design, with the first real PID-type controller developed by Elmer Sperry in 1911. The first theoretical analysis of a PID controller was published by Nicolas Minorsky in 1922. His observations grew out of efforts to design automatic steering systems for the U.S. Navy. He realized that a helmsman controlled the ship based not only on the current error, but also on past error and current rate of change. Proportional control could provide stability against small disturbances, but it was insufficient for dealing with a steady disturbance, which required adding the integral term. Adding the derivative term further improved control. Currently, the basic principle is applied in many variations, but for our discussion we assume commonly used basic standard ISA PID controller in discrete form:

$$OUT(k) = K_p \left[e(k) + \frac{1}{T_i} \sum e(k) + T_d \Delta e(k) \right] \quad (1)$$

where

OUT(k) – Controller output

K_p – Controller proportional gain

SP - Controlled parameter setpoint

PV – Process value, controlled parameter measurement

e(k) – Control error, *e(k)=SP-PV*

T_i – Controller integral time (reset) in seconds

T_d – Controller derivative time (rate) in seconds

Tuning a control loop is the adjustment of its control parameters (proportional gain K_p, integral time/reset T_i, derivative time/rate T_d) to the optimum values for the desired control response. Stability (bounded oscillation) is a basic requirement, but beyond that, different systems have different behavior, different applications have different requirements, and requirements may conflict with one another. In the following sections we briefly review the basic tuning techniques, focusing on objectives of the tuning and how they were achieved.

Model-free tuning

Model-free tuning techniques don't use a process model for PID controller tuning in a direct way. They are based on the observation of a process which is under control, and became among the first initially applied for controller tuning. The most well-known techniques from this group are manual tuning and Ziegler Nichols tuning. In manual tuning the tuned loop remains in automatic mode. At first integral and derivative actions are removed by setting T_i to infinity (or very high) and T_d to zero. Then, controller proportional gain K_p is increased until the loop oscillates with constant amplitude. After that, the proportional gain K_p should be set to approximately half of that value for a quarter amplitude decay type response. T_i is adjusted until any offset is corrected in sufficient time for the loop operation. It should be noted that too low a value for T_i may cause instability. Finally, T_d is increased until the loop is

acceptably quick to reach its reference after a load disturbance, without excessive overshoot.

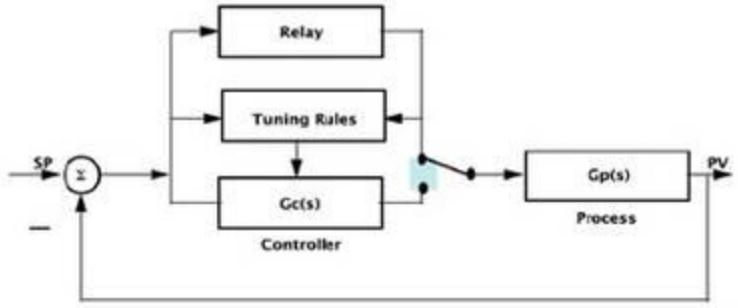
Ziegler Nichols tuning also requires setting proportional-only gain K_p high enough to achieve loop oscillations with constant amplitude, similar to the manual tuning method. The difference is that the period of the oscillation T_u should be measured. Controller gain at the test is named the ultimate gain, K_u. The controller parameters are defined then from the formulas with no need for further loop testing like in the manual tuning shown in Table 1.

Controller type\Parameter	K_p	T_i	T_d
P	$0.5\,K_u$	n/a	n/a
PI	$0.4\,K_u$	$0.8\,T_u$	n/a
PID	$0.6\,K_u$	$0.5\,T_u$	$0.1\,T_u$

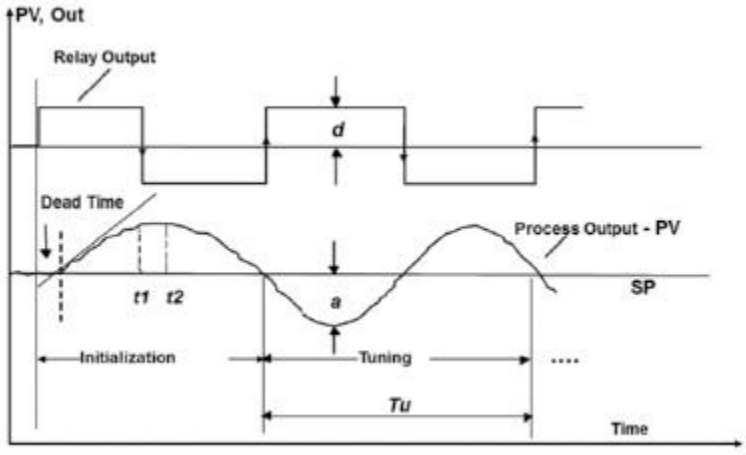
[Table 1: Ziegler-Nichols calculations of P, PI, and PID controller parameters for a standard-form PID.]

There were some obvious drawbacks to these techniques, at least initially. Getting the loop to cycle continuously was a time-consuming process, and there was a risk that the oscillations would grow beyond stability since there was no deterministic way to specify or limit the oscillation magnitude.

The approach became significantly more attractive after introducing relay-oscillation auto-tuning, as described in Figure 1. Relay-oscillation auto-tuning delivers loop oscillation with amplitude limited by the relay step size, shown in Figure 2.



[Figure 1. Relay Oscillation Tuning Diagram]



[Figure 2. Trend plots of relay output and process output during active tuning]

The Ziegler Nichols method identifies the ultimate gain and ultimate period, so controller settings may be determined. The original Ziegler-Nichols tuning rules were designed to provide a quarter amplitude damped response to a load disturbance. Once considered ideal, the underdamped and oscillatory nature of Ziegler-Nichols tuning has been criticized for destabilizing control loops, actually increasing variability instead of reducing it.

There were several options for modifying original Ziegler-Nichols tuning. One way was to calculate a complete first order plus dead time model from the

oscillation test and use model based tuning rules. Another method was to reduce the aggressiveness of the original Ziegler-Nichols tuning rules, which tend to provide the most oscillatory response when dead time is small. Therefore, an improvement can be achieved by making controller gain smaller and integral time larger (i.e., integral gain smaller) for smaller dead times. Dead time, in addition to the ultimate gain and ultimate period, can be easily defined during relay oscillation test shown in Figure 2. An example of such nonlinear adjustment of tuning parameters is given in Advanced Control Foundation (see additional reading list).

Source: http://www.controleng.com/single-article/evolving-pid-tuning-rules/db217d66282b4f4a59fbe7dd554b7cb8.html