Article:
Model-based PID tuning with Skogestad's method

Finn Haugen
18. October 2009

1 The principle of Skogestad's method

Skogestad's PID tuning method [1] is a model-based tuning method where *the controller parameters are expressed as functions of the process model parameters*. It is assumed that the control system has a transfer function block diagram as shown in Figure 1.

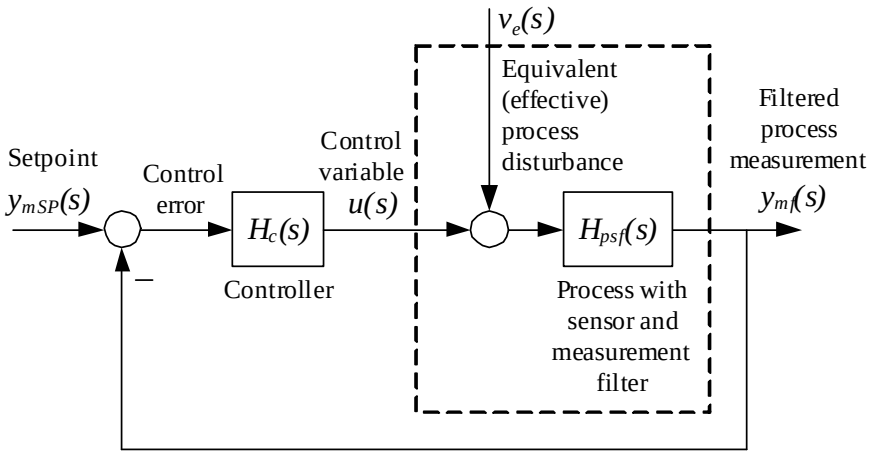

Figure 1: Block diagram of the control system in PID tuning with Skogestad's method

Comments to this block diagram:

- The transfer function $H_{psf}(s)$ is a combined transfer function of the process, the sensor, and the measurement lowpass filter. Thus, $H_{psf}(s)$ represents all the dynamics that the controller "feels". For simplicity we may denote this transfer function the "process transfer function", athough it is a combined transfer function.

[1] Named after the originator Prof. Sigurd Skogestad

- The disturbance acting on the process: In most processes the dominating disturbance influences the process in the same way, dynamically, as the control variable. Such a disturbance is called an *input disturbance* . Here are a few examples:
  - Liquid tank: The control variable controls the inflow. The outflow is a disturbance.
  - Motor: The control variable controls the motor torque. The load torque is a disturbance.
  - Thermal process: The control variable controls the power supply via an heating element. The power loss via heat transfer through the walls and heat outflow through the outlet are disturbances.

The design principle of Skogestad's method is as follows. The control system *tracking transfer function* $T(s)$, which is the transfer function from the setpoint to the (filtered) process measurement, is *specified* as a first order transfer function with time delay:

$$T(s) = \frac{y_{mf}(s)}{y_{m_{SP}}(s)} = \frac{1}{T_C s + 1} e^{-\tau s} \tag{1}$$

where T_C is the time-constant of the control system which *the user must specify*, and τ is the process time delay which is *given* by the process model (the method can however be used for processes without time delay, too). Figure 2 shows as an illustration the response in y_{mf} after a step in the setpoint $y_{m_{SP}}$ for (1).

From the block diagram shown in Figure 1 the tracking transfer function can be found as

$$T(s) = \frac{H_c(s)H_{psf}(s)}{1 + H_c(s)H_{psf}(s)} \tag{2}$$

Setting (2) equal to (1) gives

$$\frac{H_c(s)H_{psf}(s)}{1 + H_c(s)H_{psf}(s)} = \frac{1}{T_C s + 1} e^{-\tau s} \tag{3}$$

Here, the only unknown is the controller transfer function, $H_c(s)$. By making some proper simplifying approximations to the time delay term, the controller becomes a PID controller or a PI controller for the process transfer function assumed.

Skogestad's tuning formulas for several processes are shown in Table 1.[2]

[2] In the table, "min" means the minimum value (of the two alternative values).

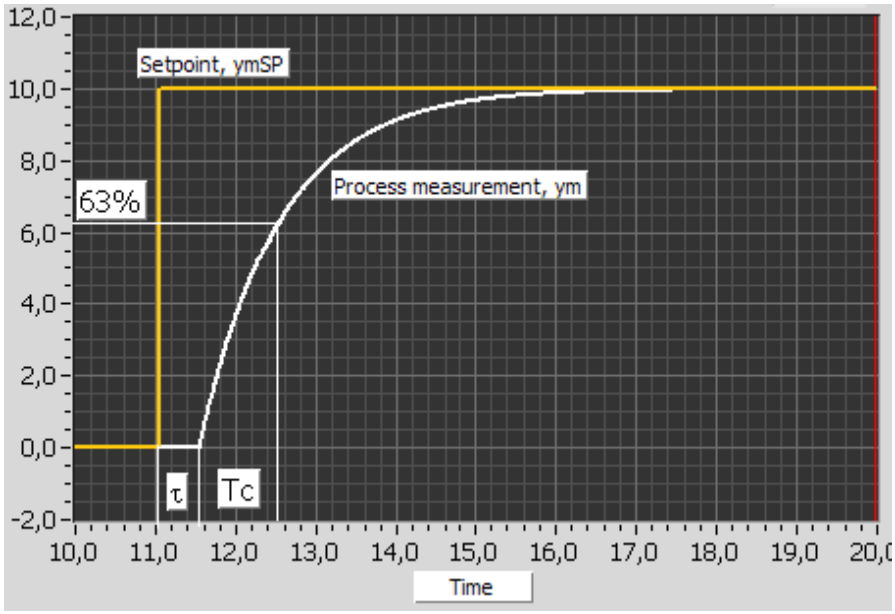

Figure 2: Step response of the specified tracking transfer function (1) in Skogestad's PID tuning method

For the "Two time-constant + delay" process in Table 1 T_1 is the largest and T_2 is the smallest time-constant.[3]

Originally, Skogestad defined the factor c in Table 1 as 4. This gives good setpoint tracking. But the disturbance compensation may become quite sluggish. To obtain faster disturbance compensation, you can use

$$c = 1.5 \tag{4}$$

The drawback of such a reduction of c is that there will be more overshoot

[3] [?] also describes methods for model reduction so that more complicated models can be approximated with one of the models shown in Table 1.

Process type	$H_{psf}(s)$ (process)	K_p	T_i	T_d
Integrator + delay	$\frac{K}{s}e^{-\tau s}$	$\frac{1}{K(T_C+\tau)}$	$c\,(T_C+\tau)$	0
Time-constant + delay	$\frac{K}{Ts+1}e^{-\tau s}$	$\frac{T}{K(T_C+\tau)}$	$\min\left[T,\, c\,(T_C+\tau)\right]$	0
Integr + time-const + del.	$\frac{K}{(Ts+1)s}e^{-\tau s}$	$\frac{1}{K(T_C+\tau)}$	$c\,(T_C+\tau)$	T
Two time-const + delay	$\frac{K}{(T_1s+1)(T_2s+1)}e^{-\tau s}$	$\frac{T_1}{K(T_C+\tau)}$	$\min\left[T_1,\, c\,(T_C+\tau)\right]$	T_2
Double integrator + delay	$\frac{K}{s^2}e^{-\tau s}$	$\frac{1}{4K(T_C+\tau)^2}$	$4\,(T_C+\tau)$	$4\,(T_C+\tau)$

Table 1: Skogestad's formulas for PI(D) tuning.

in the setpoint step respons, and that the stability of the control loop will be reduced.

Note: For the double integrator process I have seen in simulations that the actual time-constant may be several times larger than the specified time-constant T_C. You should simulate the system with different values of T_C to obtain the specified time-constant.

Skogestad suggests using

$$T_C = \tau \tag{5}$$

for T_C in Table 1 – unless you have reasons for a different specification of T_C.

Eksempel 1 ***Control of first order system with time delay***

Let us try Skogestad's method and Ziegler-Nichols' closed loop method for tuning a PI controller for the (combined) process transfer function

$$H_{psf}(s) = \frac{K}{Ts+1}e^{-\tau s} \tag{6}$$

(time-constant with time-delay) where

$$K = 1;\ T = 1 \text{ s};\ \tau = 0.5 \text{ s} \tag{7}$$

We use (5):

$$T_C = \tau = 0.5 \text{ s} \tag{8}$$

The controller parameters are as follows, cf. Table 1:

$$K_p = \frac{T}{K\left(T_C+\tau\right)} = \frac{1}{1\cdot(0.5+0.5)} = 1 \tag{9}$$

$$\begin{aligned} T_i &= \min\left[T,\ c\left(T_C+\tau\right)\right] && (10)\\ &= \min\left[1,\ 1.5\left(0.5+0.5\right)\right] && (11)\\ &= \min\left[1,\ 1.5\right] && (12)\\ &= 1 \text{ s} && (13) \end{aligned}$$

$$T_d = 0 \tag{14}$$

Figure 3 shows control system responses with the above PID settings. At time 5 sec the setpoint is changed as a step, and at time 15 sec the disturbance is changed as a step. The responses, and in particular the stability of the control systems, seem ok.

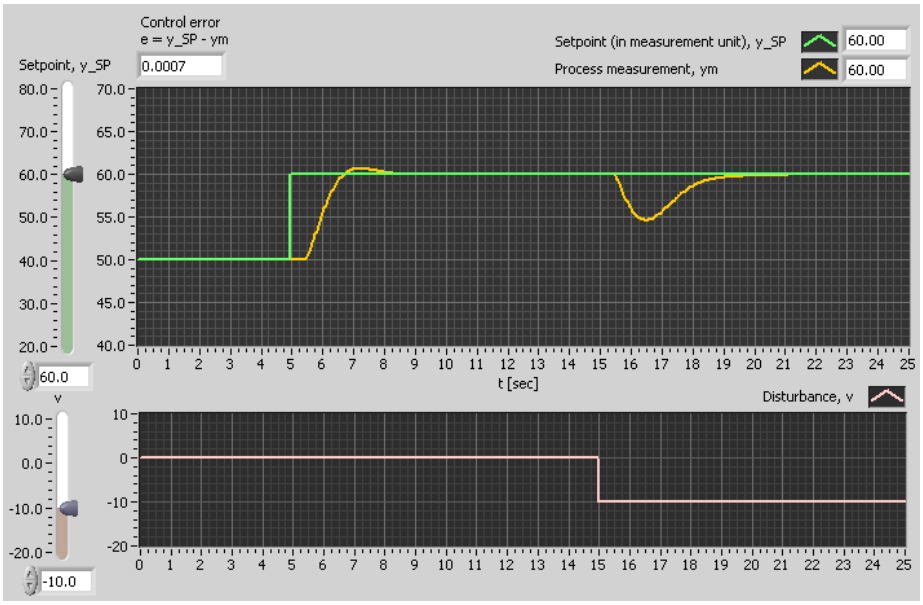

Figure 3: Example 1: Simulated responses in the control system with Skogestad's controller tuning

[End of Example 1]

You may wonder: Given a process model as in Table 1. Does Skogestad's method give better control than if the controller was tuned with some other method, e.g. the Good Gain method or Ziegler-Nichols' method? There is no unique answer to that question, but my impression is that Skogestad's method in general works fine. If you have a mathematical of the process to be controlled, you should always simulate the system with alternative controller tunings. The benefit of Skogestad's method is that you do not have to perform trial-and-error simulations to tune the controller. The parameters comes directly from the process model and the specified control system response time. Still, you should run simulations to check the performance.

2 How to find model parameters from experiments

The values of the parameters of the transfer functions in Table 1 can be found from a mathematical model based on physical principles. The parameter values can also be found from a step-response experiment with

the process. This is shown for the model *Integrator with time-delay* and *Time-constant with time-delay* in the following respective figures.[4]

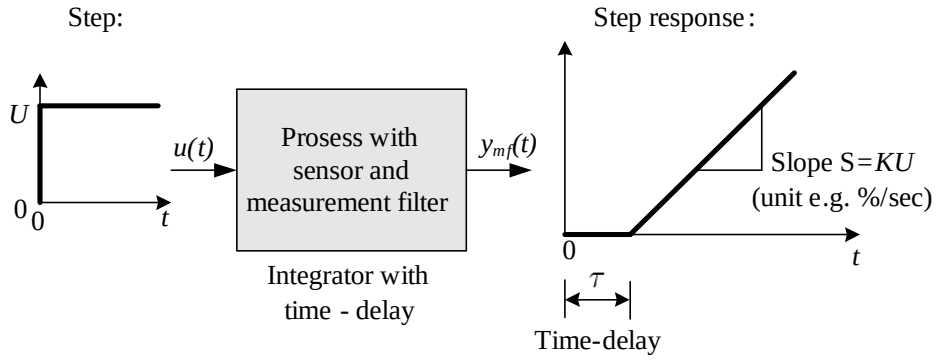

Figure 4: How the transfer function parameters K and τ appear in the step response of an *Integrator with time-delay* prosess

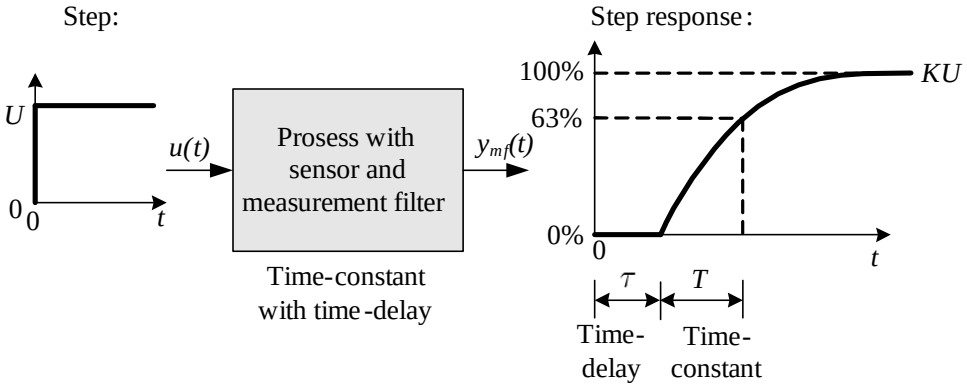

Figure 5: How the transfer function parameters K , T, and τ appear in the step response of a *Time-constant with time-delay* prosess

3 Transformation from serial to parallel PID settings

Skogestad's formulas assumes a *serial* PID controller function (alternatively denoted cascade PID controller) which has the following

[4]The theory of calculating these responses is covered by e.g. my book *Basic Dynamics and Control*.

transfer function:

$$u(s) = K_{p_s} \frac{(T_{i_s} s + 1)(T_{d_s} s + 1)}{T_{i_s} s} e(s) \tag{15}$$

where K_{p_s}, T_{i_s}, and T_{d_s} are the controller parameters. If your controller actually implementes a *parallel* PID controller (as in the PID controllers in LabVIEW PID Control Toolkit and in the Matlab/Simulink PID controllers), which has the following transfer function:

$$u(s) = \left[K_{p_p} + \frac{K_{p_p}}{T_{i_p} s} + K_{p_p} T_{d_p} s \right] e(s) \tag{16}$$

then you should transform from serial PID settings to parallell PID settings. If you do not implement these transformations, the control system may behave unnecessarily different from the specified response. The serial-to-parallel transformations are as follows:

$$K_{p_p} = K_{p_s} \left(1 + \frac{T_{d_s}}{T_{i_s}} \right) \tag{17}$$

$$T_{i_p} = T_{i_s} \left(1 + \frac{T_{d_s}}{T_{i_s}} \right) \tag{18}$$

$$T_{d_p} = T_{d_s} \frac{1}{1 + \frac{T_{d_s}}{T_{i_s}}} \tag{19}$$

Note: The parallel and serial PI controllers are identical (since $T_d = 0$ in a PI controller). Therefore, the above transformations are not relevant for PI controller, only for PID controllers.

4 When the process has no time-delay

What if the process $H_p(s)$ *is without time-delay*? Then you can not specify T_C according to (5) since that would give $T_C = 0$ (zero response time of the control system). You must specify T_C to some reasonable value larger than zero. If you do not know what could be a reasonable value, you can simulate the control system for various values of T_C. If the control signal (controller output signal) is changing too quickly, or often reaches the maximum and minimum values for reasonable changes of the setpoint or the disturbance, the controller is too aggressive, and you can try increasing T_C.