IFAC Conference on Advances in PID Control
PID'12
Brescia (Italy), March 28-30, 2012

Signal Filtering in PID Control ⋆

Tore Hägglund

Department of Automatic Control, Lund University, Box 118, SE 221 00 Lund, Sweden (e-mail: Tore.Hagglund@control.lth.se).

Abstract: The major input signals entering the PID controller are; the setpoint, the process output, and measurable load disturbances. By feeding these signals through suitable filters, the properties of the feedback loop can be improved significantly. This presentation will treat setpoint handling, feedforward from load disturbances, TITO (two input two output) control, noise filtering, and process dynamics compensation. An industrial case from the steel industry is also discussed.

Keywords: PID control, feedforward, filtering, lead-lag filter, TITO control, decoupling, controller structures, steel industry.

1. INTRODUCTION

The PID controller is the standard controller used at the lowest levels in process control configurations. It is also often used at higher levels and in many other engineering areas. Together with a process section, the PID controller forms the basic feedback loop, see Figure 1.

The major input signals to the controller are setpoint r, process output y, and, if available, measurable load disturbance d. There may be several measurable load disturbance signals, but for simplicity it is assumed that only one signal is available. Besides these major signals, there may also be other analog and digital input signals used for tracking, mode switching etc.

The three controller input signals are normally filtered in different ways before they enter the controller. A more detailed description of the basic feedback loop is given in Figure 2 where each input is fed through a filter before it enters controller C. The controller part of the feedback loop can therefore be described according to Figure 3.

Setpoint filter F_r is used to separate the design for setpoint responses from the design of responses to load disturbances, and to reduce the high-frequency variations in controller output u introduced by the setpoint. Process output filter F_y can be used for several purposes. High-frequency noise can be reduced if F_y is designed as a low-pass filter with roll-off at high frequencies. The filter can also be used to modify the dynamics in the loop transfer

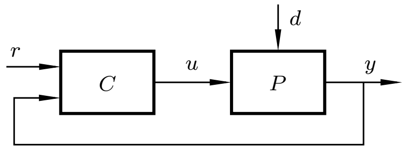

Fig. 1. The basic feedback loop with controller C, process P, and the signals setpoint r, controller output u, process output y, and load disturbance d.

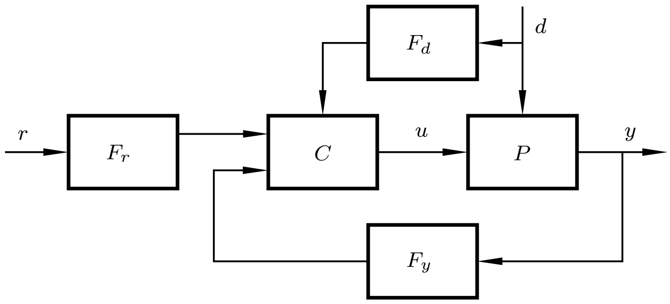

Fig. 2. The basic feedback loop with filters included.

function, and thereby improve the control performance. This feature is useful when the structure of controller C is restricted, e.g. to PID. Load disturbance filter F_d can be used for feedforward compensation and to decouple interacting control loops.

A proper choice of the three filters can improve the performance of the feedback loop considerably. Therefore, it's important to keep these filters in mind during the design procedures.

This paper treats the design of the three filters F_r, F_y, and F_d. An industrial case from the steel industry is also presented, where a proper design of filter F_y turned out to be of great value.

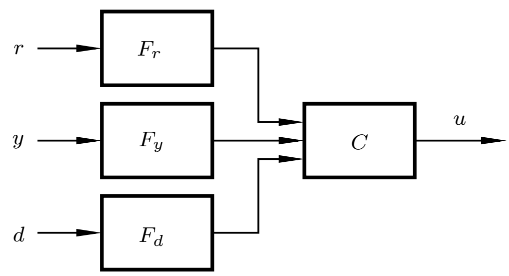

Fig. 3. Controller with the three major input signals and their corresponding filters.

⋆ This work was funded by the Swedish Foundation for Strategic Research through the PICLU center.

2. SETPOINT FILTER F_r

The standard form of the PID controller is

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\,\frac{d}{dt}e(t)\right), \qquad (1)$$

where $e = r - y$ is the control error. With this structure, the controller has only one input signal, the control error. Many of the earlier controllers were built in this way, and there are still controllers today with this structure.

The fact that setpoint r and process output y are treated in the same way causes mainly two problems. Controllers tuned for good load-disturbance response give often overshoots at setpoint changes. For this reason, it is often necessary to detune the controllers if setpoint changes are considered. There are several tuning methods that provide different rules depending on whether load or setpoint changes are considered. See O'Dwyer (2009).

Setpoint r is often changed stepwise. Another problem with the structure (1) is that these abrupt changes in r may result in large variations in controller output u, and these variations may lead to wear in the actuators. This problem may require a detuning of the controller, especially of gain K and derivative time T_d.

Most controllers and control systems have the possibility to treat the setpoint and the process output differently, i.e. they have two degrees of freedom. This freedom can be seen as exploiting a filter F_r on the setpoint according to Figure 2, even though it's not always implemented in this way.

Setpoint filter F_r may be chosen in several different ways. It has normally a low-pass character, but not always. It must have the property $F_r(0) = 1$ to ensure that the process output equals the setpoint in steady state. Some common filters are discussed in the remainder of this section.

2.1 Simple filters

A simple way to reduce overshoots at step changes in the setpoint is to introduce low-pass filters of the forms

$$F_r = \frac{1}{1+sT_f} \qquad F_r = \frac{1}{(1+sT_f)^2}.$$

The first-order filter will also eliminate step changes in the output from the proportional part of the controller, and the second-order filter will eliminate step changes also in the derivative part, at step changes in the setpoint.

2.2 Rate limiters

A traditional way to reduce high-frequency components in the setpoint and also in the controller output is to feed the signals through rate limiters or ramping modules. See Shinskey (1996) and Åström and Hägglund (2005). One way to implement a rate limiter is presented in Figure 4. The output follows the input if the rate of change of the input is smaller than the rate limit. A more sophisticated limiter is the jump and rate limiter shown in Figure 5. The output follows the input for small changes in the input signal. At large changes, the output will follow the input

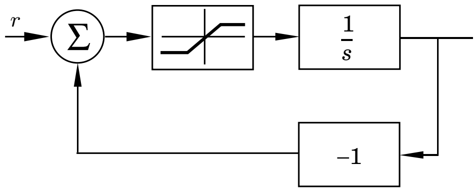

Fig. 4. Rate limiter.

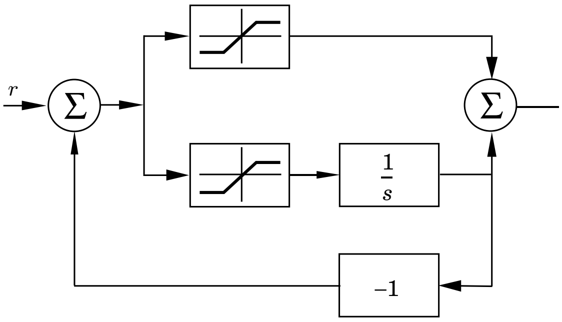

Fig. 5. Jump and rate limiter.

with a limited rate. The rate limiters are nonlinear filters. They have low-pass character and cause delays. They have also the property $F_r(0) = 1$, which means that the outputs are equal to the inputs in steady state.

2.3 Setpoint weighting

An efficient and common way to separate setpoint and load disturbance responses in modern control systems is to use setpoint weighting. A PID controller with setpoint weighting is given by

$$u(t) = K\left(br - y + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\,\frac{d}{dt}(cr - y)\right), \quad (2)$$

where b and c are the setpoint weights, see Åström and Hägglund (2005). The controllers obtained for different values of b and c will respond to load disturbances and measurement noise in the same way. The response to setpoint changes will, however, depend on the values of b and c. The weight in the derivative part is normally set to $c = 0$ to avoid large abrupt changes in u at fast setpoint changes.

The controller (2) can be interpreted as a controller C with error feedback and a filter F_r, with transfer functions

$$C = K\frac{1 + sT_i + s^2T_iT_d}{sT_i}$$
$$F_r = \frac{1 + bsT_i + cs^2T_iT_d}{1 + sT_i + s^2T_iT_d},$$

where C is the transfer function of the standard form (1).

Example – Setpoint weighting The properties of a system where the controller has set-point weighting is illustrated in Figure 6. The figure shows responses to step changes in setpoint and load in a system composed of a PI controller and the process $P(s) = (s+1)^{-3}$.

The figure illustrates the effect of changing b. The overshoot at set-point changes is smallest for $b = 0$, which is the case where the reference is only introduced in the integral term, and increases with increasing b. In this ex-

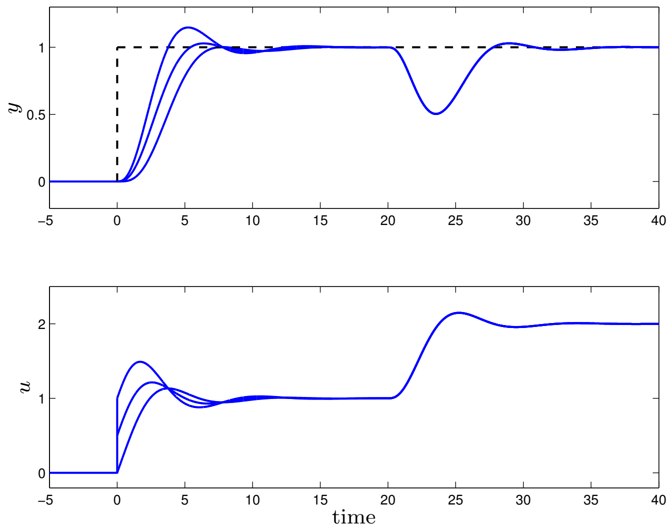

Fig. 6. PI control of the process $P(s) = (s + 1)^{-3}$, using setpoint weights $b = 0$, $b = 0.5$, and $b = 1$. The figure shows responses to a step change in r followed by a step change in the load.

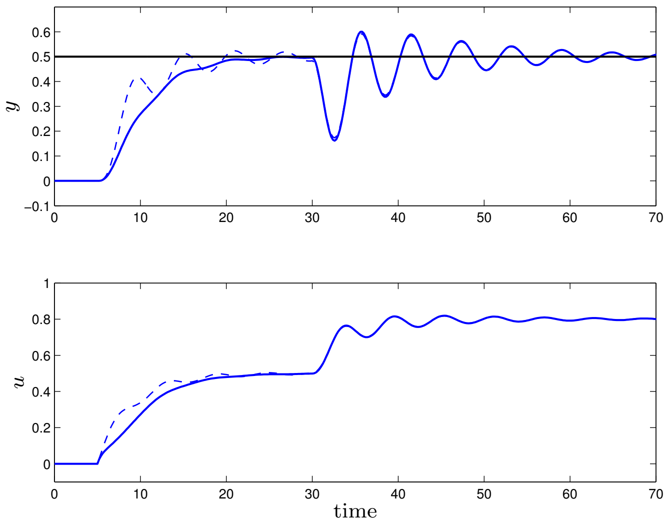

Fig. 7. Responses to set points and load disturbances of the process $P(s) = 1/(s^2 + 0.4s + 1)$ without notch filter (dashed lines) and with notch (solid lines).

ample, responses to setpoint and load disturbances become similar when the setpoint weight $b = 0$ is chosen. □

Even if the setpoint weight normally is chosen in the interval $0 \leq b \leq 1$ to reduce overshoots in the set-point response, there are also situations when $b > 1$ is used to speed up the response in a sluggish control loop.

2.4 Notch filtering

In mechanical constructions, such as industrial robots, a sudden change in the setpoint may induce oscillations in the control loop. One way to avoid this is to feed the setpoint through a notch filter before it enters the PID controller. A common structure of the noise filter is

$$F_r = \frac{s^2 + 2\zeta\omega s + \omega^2}{(s + \omega)^2} \qquad \zeta \ll 1$$

where ω is the resonance frequency of the system. This transfer function is close to one for all frequencies except those corresponding to the oscillatory modes where it has low gain. The transfer function thus blocks signals that can excite the oscillatory modes.

Example – Notch filtering Consider a system with the transfer function

$$P(s) = \frac{1}{s^2 + 0.4s + 1}.$$

The oscillatory mode has a relative damping $\zeta = 0.2$, which is quite low.

Reasonable PI controller parameters for the system are $K = 0.2$ and $T_i = 0.7$. Set-point weighting with $b = 0$ is also used. A suitable notch filter is

$$F_r = \frac{s^2 + 0.4s + 1}{(s + 1)^2}.$$

Figure 7 shows the response of the system to setpoint changes and load disturbances. The set-point response is improved substantially by the use of the notch filter. The load disturbance response is still poor, which reflects the fact that PI control is not appropriate for a highly oscillatory system. □

The example shows that the use of a notch filter F_r is an efficient way to avoid inducing oscillations by the setpoint in oscillatory systems. However, it is also evident from Figure 7 that oscillations caused by variations in the load have to be reduced using active damping in the control loop, i.e. more advanced controllers than the PID has to be used.

In the notch filter, frequency ω should correspond to the resonance frequency of the process. This frequency may vary if the load varies. In this case, it is necessary to retune the filter or include some kind of adaptation in the filter.

2.5 More advanced filters

There are more advanced structures that can be used to introduce the setpoint in the control loop. Instead of just using a filter F_r, one can feed the setpoint forward directly to the controller output, after feeding it through a suitable filter, see e.g. Åström and Hägglund (2005). The filters mentioned in this section can also be combined. For example, in the notch filtering example, setpoint weighting with $b = 0$ was used to reduce high frequencies in the setpoint.

3. PROCESS OUTPUT FILTER F_y

Filter F_y is used to improve the information about the process provided by measurement signal y. Noise can be removed from the signal by giving F_y low-pass character, and undesired features of the dynamics can be removed by giving F_y compensating features.

3.1 Noise filtering

It is well known, that the derivative part of the PID controller requires low-pass filtering to limit the high-frequency gain. Many PID controllers have just a filter

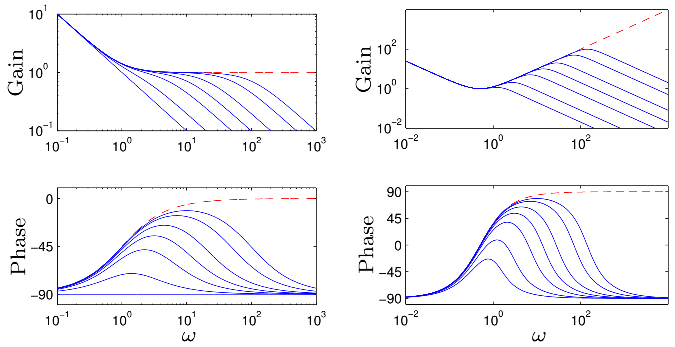

Fig. 8. Bode plot of the transfer function of a PI controller with a first-order noise filter (left) and a PID controller, with $T_i = 4T_d$, with a second-order noise filter (right). The time constants of the filters are proportional to T_i for the PI controller and to T_d for the PID controller, with factors 100, 50, 20, 10, 5, 2, and 1. The dashed lines show plots without noise filters.

in the derivative term, and these filters are often of first order. It means that these controllers lack roll-off.

It has been shown that roll-off improves performance, see Larsson and Hägglund (2011). This means that for P and PI control, there should be a filter of at least first order, and for controllers with derivative action, there should be a filter of at least second order. In Larsson and Hägglund (2011) it is shown that there is not much to gain in using filters of higher order. The following filter structures are suggested.

$$\begin{aligned} F_y &= \frac{1}{1+sT_f} \\ F_y &= \frac{1}{1+sT_f+s^2T_f^2/2}. \end{aligned} \tag{3}$$

The filters have only one tuning parameter, filter-time constant T_f. Note that the second-order filter has complex poles with damping ratio $\zeta = 0.707$.

Figure 8 shows Bode plots of a PI controller with a first-order noise filter, and a PID controller with a second-order noise filter. The figure shows that the filters cause a significant phase reduction unless the filter-time constants are significantly shorter than the controller times T_i and T_d. This illustrates why the noise filter should be taken into account in the controller design.

Most design methods for PID controllers do not take measurement noise into account, and it is often suggested to choose the filter-time constant as a fraction of the derivative time, i.e. $T_f = T_d/N$. The fact that this solution has severe drawbacks, and that the design of T_f should be integrated in the design of the other controller parameters, was pointed out in Isaksson and Graebe (2002). Examples of such methods are given in Kristiansson and Lennartson (2006), Garpinger (2009), Sekara and Matausek (2009), and Larsson and Hägglund (2011).

3.2 Dynamics compensation

In controller design, it is often assumed that the process dynamics are given, and the design procedure starts with

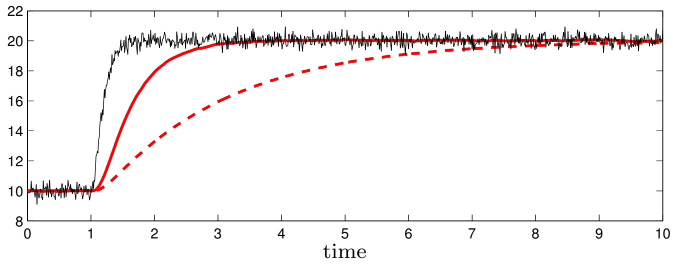

Fig. 9. Lead/lag-filtering of temperature measurement. The thin line shows the real temperature, the dashed line the output from the sensor, and the solid thick line the output from lead/lag-filter F_y.

this assumption. However, one can often modify the process dynamics in several ways. The actuators can often not be replaced, but their characteristics may be modified, e.g., to make them linear. If this is not possible, linearization blocks or gain scheduling can be used.

The sensors include normally a low-pass filter. The time constant of this filter is chosen as a compromise between accuracy and ability to react to fast changes in the measured variable. Accuracy has often the highest priority, which means that the filter-time constant is long. For control applications, it is however important to have fast responses, which means that it often is desired to have a shorter filter-time constant. Sometimes, it's possible to adjust the filter-time constant. If this is not possible, filter F_y can be used to reach the same goal.

Suppose that the sensor has a low-pass filter with time-constant T_{lp}, and that it is desired to change the time-constant to T_{lp}^{new}. This is accomplished using the lead/lag-filter

$$F_y = \frac{1+sT_{lp}}{1+sT_{lp}^{\text{new}}}.$$

Example – Lead-lag filter to speed up sensor data Figure 9 shows a temperature that is increased rather quickly from 10% to 20%. The measurement signal is noisy, and the sensor has a low-pass filter with filter-time constant $T_{lp} = 2s$, resulting in the dashed line in the plot. The output from the filter has a very low noise level, but the response to the temperature change is slow.

To speed up the response, a lead/lag filter is added after the low-pass filter with the new time constant $T_{lp}^{\text{new}} = 0.5s$, i.e. the lead/lag-filter is

$$F_y = \frac{1+sT_{lp}}{1+sT_{lp}^{\text{new}}} = \frac{1+2s}{1+0.5s}.$$

The solid line in the figure demonstrates that the output from the filter still has a low noise level, but the ability to react to fast temperature changes has increased. □

4. DISTURBANCE FILTER F_d

Filter F_d is used to reduce the influence of measurable load disturbances. Two cases are treated here, traditional feedforward and TITO control for interacting control loops.

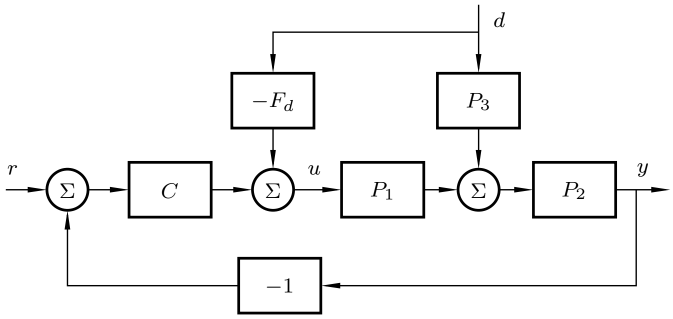

Fig. 10. Block diagram illustrating the feedforward control problem.

4.1 Feedforward control

The response to measurable load disturbance signals can be improved by filtering the disturbance using a feedforward filter F_d, and feeding the filter output forward to the controller. The feedforward structure is shown in Figure 10. Process P is divided into two parts, $P = P_1P_2$, where the relation between P_1 and P_2 is determined by the entrance of the load disturbance. Transfer function P_3 models, together with P_2, the dynamics between disturbance d and process output y.

The goal is to design the feedforward compensator F_d so that the effect of the disturbance d on the process output y is minimized. The optimal solution to the problem is

$$F_d = P_3P_1^{-1}, \qquad (4)$$

since it will eliminate the load disturbance response completely. Unfortunately, the design (4) is not always realizable. It may e.g. lead to F_d being non-causal, unstable, or having infinite high gain because of derivative action. These facts make the design problem non-trivial, and there is a need for design strategies and tuning rules.

It is surprising that there are so few design methods for feedforward compensators presented in the literature. Most basic control textbooks mention the feedforward technique, and present the design philosophy of the ideal compensator (4). They normally also mention the realizabililty problems, but they seldom go any further and present design rules.

In Shinskey (1996), a design procedure for a lead-lag compensator was proposed. The static gain of the compensator is first determined from pure static models. The gain is chosen so that a step change in the load is eliminated in steady state, without any action from the feedback controller. The time constants of the lead-lag filter are then determined with the goal to reach $IE = 0$ with minimized IAE. The effects of the feedback controller are not taken into account in these design calculations.

Seborg et al. (1989) presented a design procedure where the feedforward gain is determined in the same way as in Shinskey (1996). A manual tuning procedure is then suggested to tune the time constants of the lead-lag filter. The tuning is based on repeated step changes of the load, with the static feedforward introduced and the feedback controller in manual.

Coughanowr (1991) presented a tuning procedure that was based on a training film from Foxboro, produced in 1978. The tuning procedure is made in the same way as the one presented in Seborg et al. (1989). The difference is the way the time constants of the lead-lag filter are determined.

All procedures mentioned so far are based on an open-loop design, i.e. the feedback controller is not taken into account when the feedforward compensator is designed. The drawback of not taking the feedback controller into account was noticed in Brosilow and Joseph (2002). It was suggested to eliminate the problem by adding another feedforward component to the control structure, so that a load change not only affects the controller output, but also its input.

Isaksson et al. (2008) pointed out that the feedback controller should be taken into account when designing the feedforward compensator. A design procedure was presented, where the norm of the transfer function from the disturbance to the process output is minimized. The solution is a design scheme consisting of repeated solutions of least-squares problems.

In Guzmán and Hägglund (2011) a simple design procedure with the goal to obtain a load disturbance response without overshoot that minimizes IAE was presented. This procedure is summarized here to demonstrate the design considerations that appear in feedforward control.

It is assumed that the process sections are modeled as

$$P_1 = \frac{K_1e^{-sL_1}}{1+sT_1},\ P_2 = \frac{K_2e^{-sL_2}}{1+sT_2},\ P_3 = \frac{K_3e^{-sL_3}}{1+sT_3}, \qquad (5)$$

and that the feedforward compensator has the lead/lag structure

$$F_d = K_{ff}\frac{1+sT_z}{1+sT_p}e^{-sL_{ff}}.$$

From (4), the optimal feedforward compensator is

$$F_d = \frac{P_3}{P_1} = \frac{K_3}{K_1}\cdot\frac{1+sT_1}{1+sT_3}e^{-s(L_3-L_1)}.$$

This filter is obviously non causal and therefore not realizable when $L_1 > L_3$, which means that one has to look for suboptimal solutions. From now on, it is assumed that $L_1 > L_3$.

A common solution to the problem is to neglect the non-causality problem by simply removing the time delay. This gives the feedforward compensator

$$F_d = \frac{K_3}{K_1}\cdot\frac{1+sT_1}{1+sT_3}. \qquad (6)$$

A drawback with this solution is demonstrated in the following example.

Example – Neglected non-causality problem The process transfer functions are:

$$P_1 = \frac{1}{1+2s}e^{-2s}, \quad P_2 = \frac{1}{1+s}, \quad P_3 = \frac{1}{1+s}e^{-s}. \qquad (7)$$

The controller is tuned using the AMIGO rule, Åström and Hägglund (2005), which gives the parameters $K = 0.32$ and $T_i = 2.85$. From (6), the compensator becomes

$$F_d = \frac{1+2s}{1+s}. \qquad (8)$$

Figure 11 shows the responses to a step change in the load. The figure shows that the process output gets a response

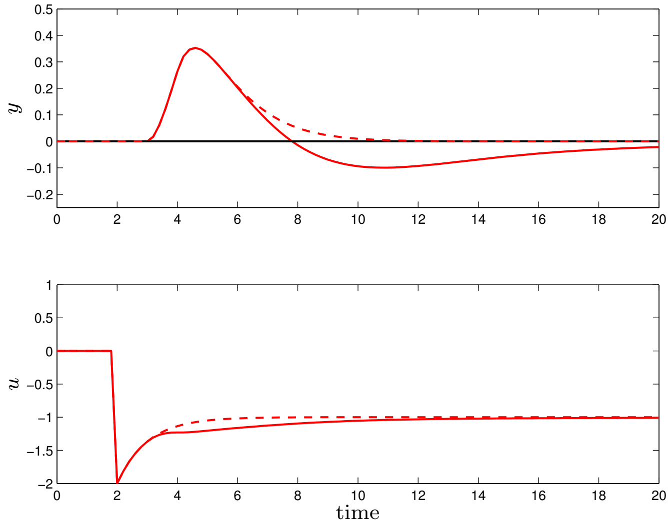

Fig. 11. Responses to a step load disturbance using feedforward compensator (8). The dashed lines show the response when the output from feedback controller C is constant.

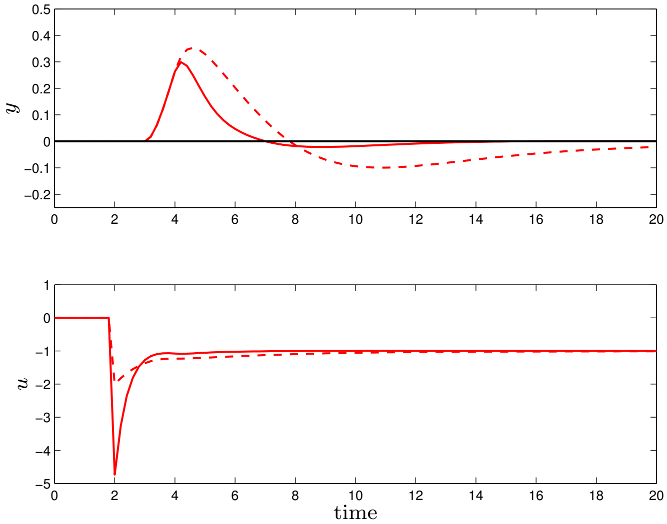

Fig. 12. Responses to a step load disturbance using feedforward tuning rule (9). The dashed lines show the responses obtained using tuning rule (8).

with an overshoot. For comparison, the figure also shows the response that would have been obtained if the output from feedback controller C were constant, i.e. the open-loop case. The comparison shows that the overshoot is caused by the action from feedback controller C, an action that is not taken into account in the design. □

The example illustrates the need for tuning rules that take controller C into account in the design of the feedforward compensator. Such a procedure was presented in Guzmán and Hägglund (2011). The idea is to first reduce gain K_{ff} so that no overshoot in the response is obtained because of the feedback. Then, T_p is determined with the goal to minimize IAE. Time constant T_z is retained as $T_z = T_1$. The tuning rule is

$$\begin{aligned}
L_{ff} &= \max(0, L_3 - L_1)\\
T_z &= T_1\\
T_p &= \begin{cases} T_3 & L_1 - L_3 \le 0\\ T_3 - \dfrac{L_1 - L_3}{1.7} & 0 < L_1 - L_3 < 1.7T_3\\ 0 & L_1 - L_3 > 1.7T_3 \end{cases}\\
K_{ff} &= \frac{K_3}{K_1} - \frac{K}{T_i} IE\\
IE &= \begin{cases} 0 & L_3 \ge L_1\\ K_2K_3(L_1 - L_3 + T_1 - T_3 + T_p - T_z) & L_3 < L_1. \end{cases}
\end{aligned} \tag{9}$$

Note that when $L_1 \le L_3$, these equations are equal to the optimal solution (4). The next example demonstrates the performance of the tuning rule.

Example – Feedforward tuning rule The process transfer functions and the feedback controller are the same as in the previous example. The tuning rule (9) gives the following compensator

$$F_d = 0.956 \frac{1 + 2s}{1 + 0.404s}. \tag{10}$$

The load step responses are shown in Figure 12. The figure shows that the overshoot caused by controller C is almost eliminated, and the disturbance response has been decreased significantly because of the reduction of time T_p. □

Both in Figure 11 and 12, the control signal has a significant initial peak in its response. The magnitude of this peak is proportional to $K_{ff}T_z/T_p$. If this peak is considered too large, it can be reduced in a systematic way by increasing T_p, and consequently also K_{ff}, in the tuning rule (9), see Guzmán and Hägglund (2011).

More advanced structures can be used to handle measurable load disturbances. Brosilow and Joseph (2002) suggests, e.g., that a feedforward signal is not only added to the controller output, but also to the controller input via a filter F_y.

4.2 Inverted decoupling of TITO systems

Figure 13 illustrates the problem of interaction for a system with two inputs and two two outputs (TITO). It is taken from a paper mill and describes a process section where pulp is transported from a pulp tower to a tank. It is obvious from the figure that the pressure controller and the flow controller interact. The problem was solved by detuning the flow controller, since the pressure loop was considered most important. To detune one of the loops is a simple and common solution to the interaction problem. The prize is that one of the loops is detuned and remains coupled to the other one.

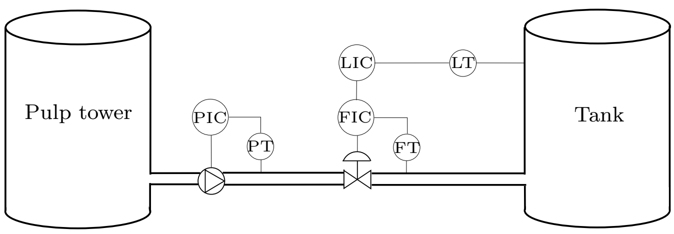

Fig. 13. Pulp flow control in a paper mill.

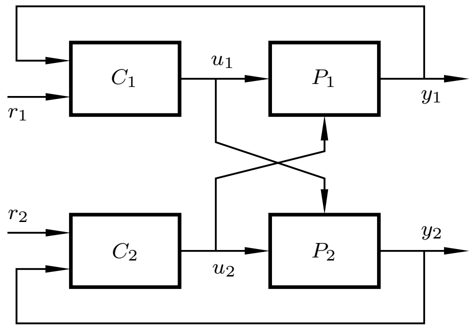

Fig. 14. Two coupled control loops.

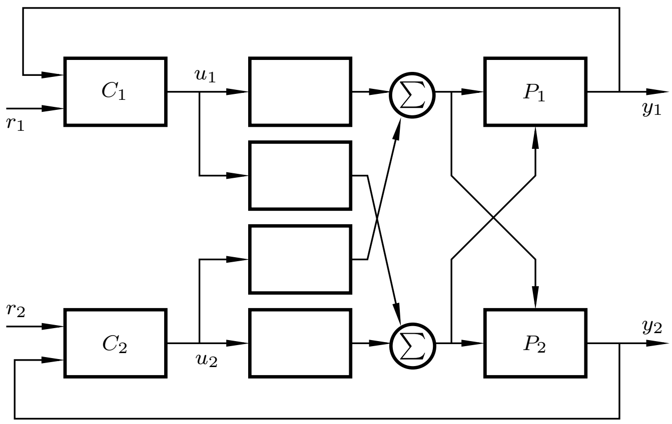

Fig. 15. Conventional decoupling.

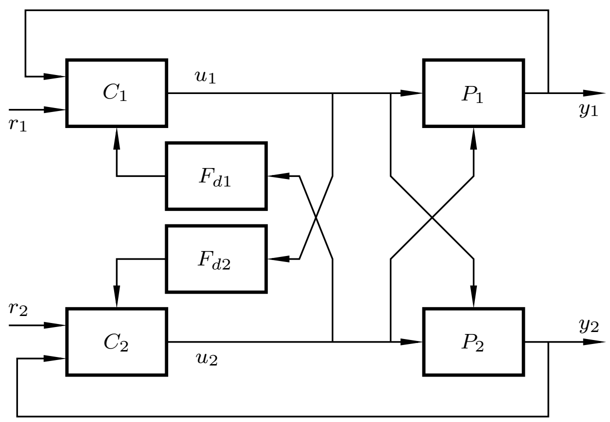

Fig. 16. Inverted decoupling.

The problem of interacting control loops is described by the block diagram in Figure 14. The control signal in one loop acts as a load disturbances on the other control loop. The interaction problem can be solved using a TITO controller that decouples the two control loops. A conventional TITO controller is shown in Figure 15. A drawback with this solution is that the process inputs are formed as combinations of the outputs from the two controllers. This makes the antiwindup solution complicated. This problem is never noticed in simulation studies where control signal limitations are not considered.

Another solution to the interaction problem is shown in Figure 16. This structure is called inverted decoupling and is obtained by feeding the two control signals through filters F_d and then forward to the other controller. With this structure, the antiwindup problem is automatically solved by the antiwindup functions already available in the controllers, since each process input is given by one controller output. Another important advantage is that the decoupling can be obtained by simply using two lead/lag-filters. This means that no special blocks have to be built in the DCS systems.

The differences between the conventional TITO controller and the inverted decoupling does not appear as long as implementation issues like bumpless transfer, limitations, and antiwindup are not considered. This may be one reason why the conventional TITO controller is the most common in textbooks and research papers. Two papers in the late nineties called for interest in the inverted decoupling structure, see Wade (1997) and Gagnon et al. (1998). The technique is nowadays studied by several research groups, see e.g. Chen and Zhang (2007) and Garrido et al. (2011).

5. STEEL BELT POSITION CONTROL

This example describes an industrial case where a properly chosen process output filter F_y is of great value.

5.1 Introduction

At the company Sandvik Process Systems, steel belts of different dimensions are placed around rotating cylinders and welded so that closed belts are obtained. See Figure 17. The lengths of the belts vary between 50m and 170m, and the velocities vary between 40m/min and 80m/min.

The control problem is to position the belt at the centers of the cylinders. With the previous control solution, this could take up to four hours. The goal was to decrease this time so that a maximum settling time corresponding to fifteen revolutions of the belt was obtained, resulting in settling times varying between nine minutes (shortest belt, highest velocity) and 64 minutes (longest belt, lowest velocity). If this goal was obtained, an increase in production of about 25% was expected.

5.2 Process description

Two hydraulic motors are connected at each side of one of the cylinders. Since they are controlled individually, both the total tension and the tension profile in the cross direction can be controlled. The goal is to position the belt at the centers of the two cylinders by controlling the difference between the two forces.

Process output y, measured in mm, is the belt position in the cross direction. The belt position is measured a few decimeters from the lower side of the controlled cylinder,

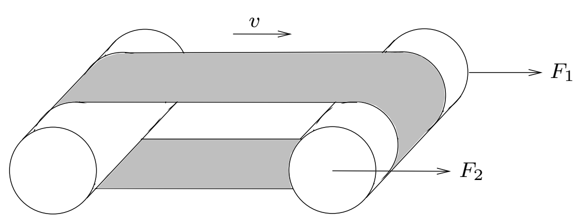

Fig. 17. The steel belt process.

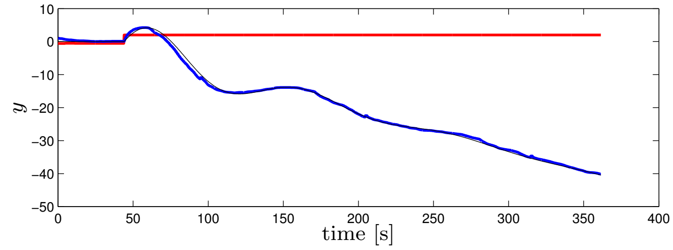

Fig. 18. Open-loop step response of the steel belt process. The thick line is the measured process output, and the thin line is the output from the model (11).

which means that the position is measured a short time after the belt has left this cylinder.

Control signal u is measured in mm, and denotes the position difference between the two sides of the cylinder.

5.3 Modeling

The process was modeled from step response experiments made with steel belts of varying dimensions and velocities. This section shows the results of one experiment, where the belt had the dimensions length $l = 70$m, width $w = 2$m, and thickness $d = 1.2$mm, and the velocity of the belt was $v = 40$m/min.

Figure 18 shows an open-loop step response of the process. Initially, the belt is almost at rest in the cross direction. At time $t = 50$s, a step change of 2mm in u is made.

The step response shows that the process is integrating. If a step change is made when the belt is at rest on the cylinders, the belt starts to move away from the equilibrium. The response also shows that an oscillatory response is added to the integrator. The period of the oscillation corresponds to the revolution time of the belt. Finally, because of the position of the sensor, an inverse response is obtained initially, indicating that there is a zero in the right-half plane.

The model

$$P(s) = \frac{K_v}{s} + \frac{(as+b)\omega^2}{s^2 + 2\zeta\omega s + \omega^2}, \qquad (11)$$

where $K_v = 0.048$, $a = -65$, $b = 1.0$, $\zeta = 0.2$, and $\omega = 0.063$, captured the dynamics of the process well. The thin line in Figure 18 shows the step response from this model.

Figure 19 shows the Bode and Nyquist plots of the model (11). The figure verifies that the process is integrating and the oscillatory mode results in a large peak in magnitude at the frequency $\omega = 0.06$ rad/s, corresponding to the circulation time of the belt.

All experiments made with different dimensions and velocities were well described by the model (11), but the parameters had to be scaled with respect to the belt lengths and the velocities. Gain K_v is proportional to velocity v, and frequency ω corresponds to the revolution time, which is proportional to l/v.

5.4 Control Design

The resonance peak shown in Figure 19 would have caused troubles if the design goal was to position the steel belt in a very short time. However, the design goal was to position the belt within fifteen revolutions, corresponding to a gain crossover frequency of $\omega_c = 0.004$, which is more than an order of magnitude lower than the resonance frequency. This desired cross-over frequency is marked with vertical lines in Figure 19.

Therefore, to simplify the design of controller C, the strategy chosen was to reduce the peak caused by the oscillatory mode and to turn the peak in the Nyquist curve away from $-180°$ by feeding the process output through a low-pass filter F_y. The structure of the low-pass filter was

$$F_y = \frac{1}{1 + sT_f},$$

with the filter-time constant T_f.

Because of the added filter F_y, it was possible to control the steel belt using a pure PI controller

$$C = K\left(1 + \frac{1}{sT_i}\right).$$

Derivative action was not used, since it would have increased the gain at the oscillation frequency.

The three parameters K, T_i, and T_f where designed to minimize the IAE value at step changes in load disturbances, with the robustness requirement that the M_s value was less than 1.4. The design program described in Garpinger and Hägglund (2008) was used to get the parameters $K = 0.15$, $T_i = 365$s, and $T_f = 80$s.

Figure 20 shows the Bode and Nyquist plots of the model (11) combined with the filter F_y. The figure shows that the resonance peak is reduced significantly, with minor changes in the dynamics at the desired gain crossover frequency ω_c .

Figure 21 shows the Bode and Nyquist plots of the loop transfer function with the filter F_y and the PI controller. The diagrams show that the design resulted in a gain crossover frequency $\omega_c = 0.007$ that exceeds the goal $\omega_c = 0.004$, and that the robustness constraint is fulfilled.

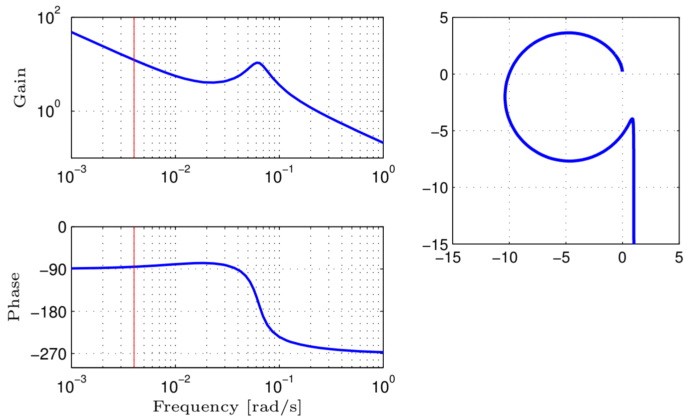

Fig. 19. Bode and Nyquist plots of the steel belt process. The vertical lines denote the desired gain crossover frequency ω_c.

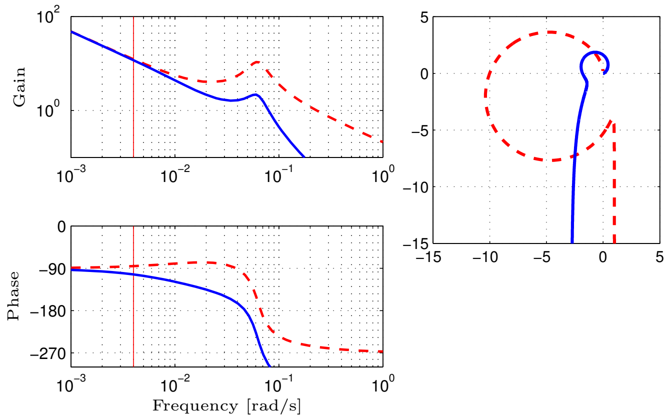

Fig. 20. Bode and Nyquist plots of the steel belt process combined with the filter F_y. The dashed lines are the plots of the unfiltered process, and the vertical lines denote the desired gain crossover frequency ω_c.

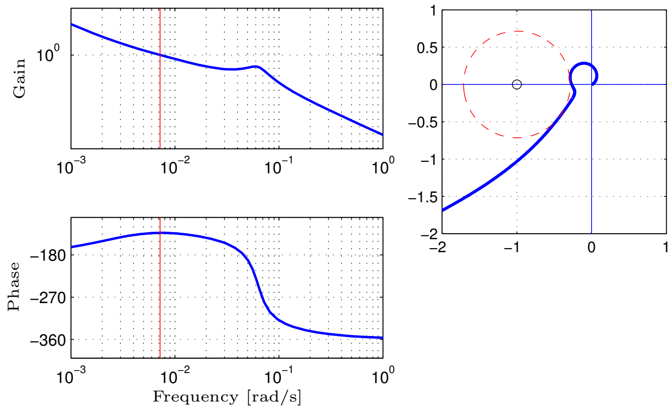

Fig. 21. Bode and Nyquist plots of the loop transfer function. The vertical lines denote the gain crossover frequency ω_c.

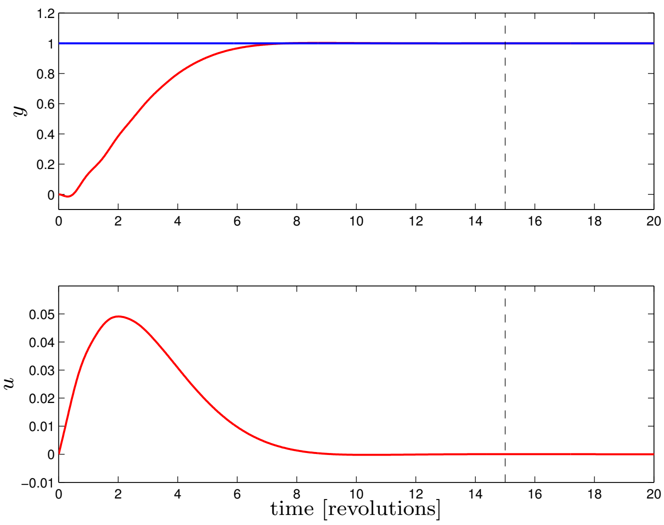

Fig. 22. Simulated closed-loop step response of the steel belt process. The dashed line marks the time when the belt has circulated 15 times.

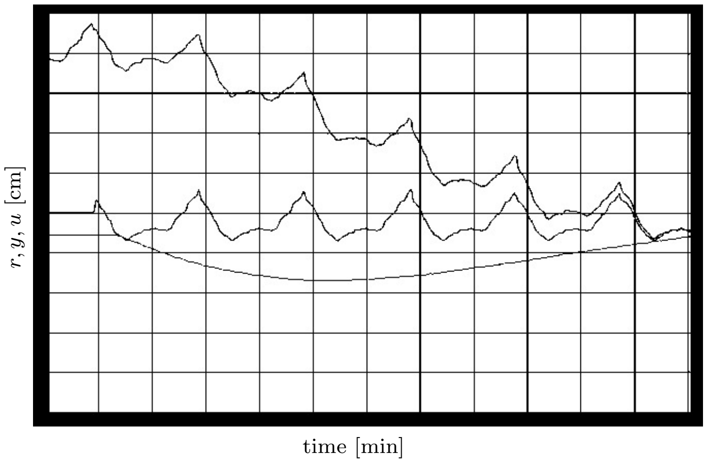

Fig. 23. Transient response of the steel belt process. The curves are band position (upper), setpoint (middle), and controller output (lower).

Because of the integrator in the process, the phase is $-180°$ at low frequencies, reaches a peak at the crossover frequency, and then drops rapidly. This property makes the controller difficult to tune manually.

Figure 22 shows a simulated response to a step change in the setpoint. The settling time is approximately 700s, corresponding to seven revolutions of the belt, i.e. about twice as fast as the requirements.

5.5 Results

The controller structure was implemented at Sandvik Process Systems. Since the dimensions of the belts and the velocities vary, a design rule that takes these variations into account was derived. The following rule was applied:

$$K = 0.15, \quad T_i = 210\, l/v, \quad T_f = 45\, l/v,$$

where the times are given in seconds. This means that the operators have to provide belt length l [m/s] and velocity v [m/min] each time a new belt is processed.

Figure 23 shows data from a case where the belt length was $l = 70$ m and the velocity $v = 80$ m/min, which corresponds to a revolution time of about 2 minutes. The upper curve is the measured belt position, the middle curve is the setpoint, and the lower curve is the controller output. The position sensor, that bears against one side of the belt, records the profile of the belt during the first revolution. This profile is then added to the setpoint. This is the reason why the setpoint in Figure 23 is a is a straight line with the profile of the output signal superimposed. The figure shows that in this case the belt has reached the desired position after about five revolutions.

The new control strategy has been used since 2009. The experience is that it works well and gives settling times that are shorter than the desired maximum time corresponding to fifteen revolutions. This means that the production can be increased by more than 25%. The key to the simple solution was the addition of the filter F_d.

6. CONCLUSIONS

The PID algorithm is the core function in low-level controllers. The controller has three major analog input signals, setpoint r, process output y, and measurable load disturbances d. These signals should be filtered before they enter the the PID controller. The design of these filters is the topic of this paper.

Setpoint filter F_r is used to provide separate handling of the setpoint and the process output, and thereby shape the response to setpoint variations. Load disturbance filter F_d is used to feed the information of load disturbances forward to the controller. Process output filter F_y is used to remove undesired components, such as measurement noise, from the signal, and to compensate for undesired dynamics in the process. A proper choice of these filters is of great importance for the overall performance of the control loop.

REFERENCES

Åström, K.J. and Hägglund, T. (2005). *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709.

Brosilow, C. and Joseph, B. (2002). *Techniques of Model-Based Control*. Prentice-Hall. New Jersey.

Chen, P. and Zhang, W. (2007). Improvement of an inverted decoupling technique for a class of stable linear multivariable processes. *ISA Transactions*, 46, 199–210.

Coughanowr, D.R. (1991). *Process Systems, Analysis and control*. McGraw-Hill. New York.

Gagnon, E., Pomerleau, A., and Desbiens, A. (1998). Simplified, ideal or inverted decoupling. *ISA Transactions*, 37, 265–276.

Garpinger, O. (2009). Design of robust PID controllers with constrained control signal activity. Licentiate Thesis LUTFD2/TFRT--3245--SE, Department of Automatic Control, Lund University, Sweden.

Garpinger, O. and Hägglund, T. (2008). A software tool for robust PID design. In *Proc. 17th IFAC World Congress, Seoul, Korea*.

Garrido, J., Vásquez, F., and Morilla, F. (2011). An extended approach of inverted decoupling. *Journal of Process Control*, 21, 55–68.

Guzmán, J.L. and Hägglund, T. (2011). Simple tuning rules for feedforward compensators. *Journal of Process Control*, 21(1), 92–102.

Isaksson, A., Molander, M., Modén, P., Matsko, T., and Starr, K. (2008). Low-order feedforward design optimizing the closed-loop response. *Preprints, Control Systems*. Vancouver, Canada.

Isaksson, A. and Graebe, S. (2002). Derivative filter is an integral part of PID design. *Control Theory and Applications, IEE Proceedings*, 149(1), 41–45.

Kristiansson, B. and Lennartson, B. (2006). Evaluation and simple tuning of PID controllers with high-frequency robustness. *Journal of Process Control*, 16(2), 91–103.

Larsson, P.O. and Hägglund, T. (2011). Control signal constraints and filter order selection for PI and PID controllers. In *Proc. American Control Conference 2011*. Accepted for publication.

O'Dwyer, A. (2009). *Handbook of PI and PID controller tuning rules*. Imperial College Press, 3rd edition.

Seborg, D., Edgar, T., and Mellichamp, D. (1989). *Process dynamics and control*. Wiley. New York.

Sekara, T. and Matausek, M. (2009). Optimization of PID controller based on maximization of the proportional gain under constraints on robustness and sensitivity to measurement noise. *IEEE Transactions of Automatic Control*, 54(1), 184.

Shinskey, F.G. (1996). *Process-Control Systems. Application, Design, and Tuning*. McGraw-Hill, New York, 4th edition.

Wade, H. (1997). Inverted decoupling: a neglected technique. *ISA Transactions*, 36(1), 3–10.