Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Journal of Process Control 14 (2004) 635–650
JOURNAL OF PROCESS CONTROL
www.elsevier.com/locate/jprocont

Revisiting the Ziegler–Nichols step response method for PID control

K.J. Åström, T. Hägglund *

Department of Automatic Control, Lund Institute of Technology, P.O. Box 118, SE-221 00 Lund, Sweden

Abstract

The Ziegler–Nichols step response method is based on the idea of tuning controllers based on simple features of the step response. In this paper this idea is investigated from the point of view of robust loop shaping. The results are: insight into the properties of PI and PID control and simple tuning rules that give robust performance for processes with essentially monotone step responses.

Keywords: PID control; Design; Tuning; Optimization; Process control

1. Introduction

In spite of all the advances in control over the past 50 years the PID controller is still the most common controller, see [1]. Even if more sophisticated control laws are used it is common practice to have an hierarchical structure with PID control at the lowest level, see [2–5]. A survey of more than 11,000 controllers in the refining, chemicals, and pulp and paper industries showed that 97% of regulatory controllers had the PID structure, see [5]. Embedded systems are also a growing area of PID control, see [6]. Because of the widespread use of PID control it is highly desirable to have efficient manual and automatic methods of tuning the controllers. A good insight into PID tuning is also useful in developing more schemes for automatic tuning and loop assessment.

Practically all books on process control have a chapter on tuning of PID controllers, see e.g. [7–16]. A large number of papers have also appeared, see e.g. [17–29].

The Ziegler–Nichols rules for tuning PID controller have been very influential [30]. The rules do, however, have severe drawbacks, they use insufficient process information and the design criterion gives closed loop systems with poor robustness [1]. Ziegler and Nichols presented two methods, a step response method and a frequency response method. In this paper we will investigate the step response method. An in-depth investigation gives insights as well as new tuning rules.

Ziegler and Nichols developed their tuning rules by simulating a large number of different processes, and correlating the controller parameters with features of the step response. The key design criterion was quarter amplitude damping. Process dynamics was characterized by two parameters obtained from the step response. We will use the same general ideas but we will use robust loop shaping [14,15,31] for control design. A nice feature of this design method is that it permits a clear trade-off between robustness and performance. We will also investigate the information about the process dynamics that is required for good tuning. The main result is that it is possible to find simple tuning rules for a wide class of processes. The investigation also gives interesting insights, for example it gives answers to the following questions: What is a suitable classification of processes where PID control is appropriate? When is derivative action useful? What process information is required for good tuning? When is it worth while to do more accurate modeling?

In [32], robust loop shaping was used to tune PID controllers. The design approach was to maximize integral gain subject to a constraints on the maximum sensitivity. The method, called MIGO (M-constrained integral gain optimization), worked very well for PI control. In [33] the method was used to find simple tuning rules for PI control called AMIGO (approximate MIGO). The same approach is used for PID control in [34], where it was found that optimization of integral gain may result in controllers with unnecessarily high

* Corresponding author. Tel.: +46-46-222-8798; fax: +46-46-13-8118.
E-mail addresses: kja@control.lth.se (K.J. Åström), tore@control.lth.se (T. Hägglund).

0959-1524/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2004.01.002

phase lead even if the robustness constraint is satisfied. This paper presents a new method with additional constraints that works for a wide class of processes.

The paper is organized as follows. Section 2 summarizes the objectives and the MIGO design method. Section 3 presents a test batch consisting of 134 processes, and the MIGO design method is applied to these processes. In Section 4 it is attempted to correlate the controller parameters to different features of the step response. It is found that the relative time delay τ, which has the range $0 \leqslant \tau \leqslant 1$, is an essential parameter. Simple tuning rules can be found for processes with $\tau > 0.5$ and conservative tuning rules can be found for all τ. For processes with $\tau < 0.5$ there is a significant advantage to have more accurate models than can be derived from a step response. It is also shown that the benefits of derivative action are strongly correlated to τ. For delay dominated processes, where τ is close to one, derivative action gives only marginal benefits. The benefits increase with decreasing τ, for $\tau = 0.5$ derivative action permits a doubling of integral gain and for $\tau < 0.13$ there are processes where the improvements can be arbitrarily large. For small values of τ there are, however, other considerations that have a major influence of the design. The conservative tuning rules are close to the rules for a process with first order dynamics with time delay, the *KLT* process. In Section 5 we develop tuning rules for such a process for a range of values of the robustness parameter. Section 6 presents some examples that illustrate the results.

2. Objectives and design method

There are many versions of a PID controller. In this paper we consider a controller described by

$$u(t) = k(by_{sp}(t) - y_f(t)) + k_i \int_0^t (y_{sp}(\tau) - y_f(\tau))\,d\tau + k_d\left(c\frac{dy_{sp}(t)}{dt} - \frac{dy_f(t)}{dt}\right) \quad (1)$$

where u is the control variable, y_{sp} the set point, y the process output, and y_f is the filtered process variable, i.e. $Y_f(s) = G_f(s)Y(s)$. The transfer function $G_f(s)$ is a first order filter with time constant T_f, or a second order filter if high frequency roll-off is desired.

$$G_f(s) = \frac{1}{(1 + sT_f)^2} \quad (2)$$

Parameters b and c are called set-point weights. They have no influence on the response to disturbances but they have a significant influence on the response to set-point changes. Set-point weighting is a simple way to obtain a structure with two degrees of freedom [35]. It can be noted that the so-called PI–PD controller [18] is a special case of (1) with parameters $b = c = 0$. See [36].

Neglecting the filter of the process output the feedback part of the controller has the transfer function

$$C(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right) \quad (3)$$

The advantage by feeding the filtered process variable into the controller is that the filter dynamics can be combined with in the process dynamics and the controller can be designed designing an ideal controller for the process $P(s)G_f(s)$.

A PID controller with set-point weighting and derivative filter has six parameters K, T_i, T_d, T_f, b and c. A good tuning method should give all the parameters. To have simple design methods it is interesting to determine if some parameters can be fixed.

2.1. Requirements

Controller design should consider requirements on responses to load disturbances, measurement noise, and set point as well as robustness to model uncertainties.

Load disturbances are often the major consideration in process control. See [10], but robustness and measurement noise must also be considered. Requirements on set-point response can be dealt with separately by using a controller with two degrees of freedom. For PID control this can partially be accomplished by set-point weighting or by filtering, see [37]. The parameters K, T_i, T_d and T_f can thus be determined to deal with disturbances and robustness and the parameters b and c can then be chosen to give the desired set-point response.

To obtain simple tuning rules it is desirable to have simple measures of disturbance response and robustness. Assuming that load disturbances enter at the process input the transfer function from disturbances to process output is

$$G_{yd}(s) = \frac{P(s)G_f(s)}{1 + P(s)G_f(s)C(s)}$$

where $P(s)$ is the process transfer function $C(s)$ is the controller transfer function (3) and $G_f(s)$ the filter transfer function (2). Load disturbances typically have low frequencies. For a controller with integral action we have approximately $G_{yd}(s) \approx s/k_i$. Integral gain k_i is therefore a good measure of load disturbance reduction.

Measurement noise creates changes in the control variable. Since this causes wear of valves it is important that the variations are not too large. Assuming that measurement noise enters at the process output it follows that the transfer function from measurement noise n to control variable u is

$$G_{un}(s) = \frac{C(s)G_f(s)}{1 + P(s)C(s)G_f(s)}$$

Measurement noise typically has high frequencies. For high frequencies the loop transfer function goes to zero and we have approximately $G_{un}(s) \approx C(s)G_f(s)$. The variations of the control variable caused by measurement noise can be influenced drastically by the choice of the filter $G_f(s)$. The design methods we use gives rational methods for choosing the filter constant. Standard values can be used for moderate noise levels and the controller parameters can be computed without considering the filter. When measurement noise generates problems heavier filtering can be used. The effect of the filter on the tuning can easily be dealt with by designing controller parameters for the process $G_f(s)P(s)$.

Many criteria for robustness can be expressed as restrictions on the Nyquist curve of the loop transfer function. In [32] it is shown that a reasonable constraint is to require that the Nyquist curve is outside a circle with center in c_R and radius r_R where

$$c_R = \frac{2M^2 - 2M + 1}{2M(M-1)}, \quad r_R = \frac{2M-1}{2M(M-1)}$$

By choosing such a constraint we can capture robustness by one parameter M only. The constraint guarantees that the sensitivity function and the complementary sensitivity function are less than M.

2.2. Design method

The design method used is to maximize integral gain subject to the robustness constraint given above. The problems related to the geometry of the robustness region discussed in [34] are avoided by restraining the values of the derivative gain to the largest region that $\partial k_i/\partial k \geqslant 0$ in the robustness region. This design gives the best reduction of load disturbances compatible with the robustness constraints.

There are situations where the primary design objective is not disturbance reduction. This is the case for example in surge tanks. The proposed tuning is not suitable in this case.

3. Test batch and MIGO design

In this section, the test batch used in the derivation of the tuning rules is first presented. The MIGO design method presented in the previous section was applied to all processes in the test batch. The controller parameters obtained are presented as functions of relative time delay τ.

3.1. The test batch

PID control is not suitable for all processes. In [33] it is suggested that the processes where PID is appropriate can be characterized as having essentially monotone step responses. One way to characterize such processes is to introduce the monotonicity index

$$\alpha = \frac{\int_0^\infty h(t)\,dt}{\int_0^\infty |h(t)|\,dt} \tag{4}$$

where h is the impulse response of the system. Systems with $\alpha = 1$ have monotone step responses and systems with $\alpha > 0.8$ are consider essentially monotone. The tuning rules presented in this paper are derived using a test batch of essentially monotone processes.

The 134 processes shown in Fig. 1 as Eq. (5) were used to derive the tuning rules. The processes are representative for many of the processes encountered in process control. The test batch includes both delay dominated, lag dominated, and integrating processes. All processes have monotone step responses except P_8 and P_9. The parameters range for processes P_8 and P_9 were chosen so that the systems are essentially monotone with $\alpha \geqslant 0.8$. The relative time delay ranges from 0 to 1 for the process P_1 but only from 0.14 to 1 for P_2. Process P_6 is integrating, and therefore $\tau = 0$. The rest of the processes have values of τ in the range $0 < \tau < 0.5$.

3.2. MIGO design

Parameters of PID controllers for all the processes in the test batch were computed using the MIGO design

$$\begin{aligned}
P_1(s) &= \frac{e^{-s}}{1+sT}, \\
&\quad T = 0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 1, \\
&\qquad 1.3, 1.5, 2, 4, 6, 8, 10, 20, 50, 100, 200, 500, 1000 \\
P_2(s) &= \frac{e^{-s}}{(1+sT)^2}, \\
&\quad T = 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 1, \\
&\qquad 1.3, 1.5, 2, 4, 6, 8, 10, 20, 50, 100, 200, 500 \\
P_3(s) &= \frac{1}{(s+1)(1+sT)^2}, \\
&\quad T = 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 2, 5, 10 \\
P_4(s) &= \frac{1}{(s+1)^n}, \\
&\quad n = 3,\ 4,\ 5,\ 6,\ 7,\ 8 \\
P_5(s) &= \frac{1}{(1+s)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}, \\
&\quad \alpha = 0.1, 0.2, 0.3, 0, 4, 0.5, 0.6, 0.7, 0.8, 0.9 \\
P_6(s) &= \frac{1}{s(1+sT_1)}e^{-sL_1}, \\
&\quad L_1 = 0.01, 0.02, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0, \quad T_1 + L_1 = 1 \\
P_7(s) &= \frac{T}{(1+sT)(1+sT_1)}e^{-sL_1}, \qquad T_1 + L_1 = 1, \\
&\quad T = 1, 2, 5, 10 \qquad L_1 = 0.01, 0.02, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0 \\
P_8(s) &= \frac{1-\alpha s}{(s+1)^3}, \\
&\quad \alpha = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1 \\
P_9(s) &= \frac{1}{(s+1)((sT)^2 + 1.4sT + 1)}, \\
&\quad T = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
\end{aligned} \tag{5}$$

Fig. 1. The test batch.

with the constraints described in the previous section. The design parameter was chosen to $M = 1.4$.

In the Ziegler–Nichols step response method, stable processes were approximated by the simple *KLT* model

$$G_p(s) = \frac{K_p}{1 + sT} e^{-sL} \tag{6}$$

where K_p is the static gain, T the time constant (also called lag), and L the time delay. Processes with integration were approximated by the model

$$G_p(s) = \frac{K_v}{s} e^{-sL} \tag{7}$$

where K_v is the velocity gain and L the time delay. The model (7) can be regarded as the limit of (6) as K_p and T go to infinity in such a way that $K_p/T = K_v$ is constant. The parameters in (6) and (7) can be obtained from a simple step response experiment, see [33].

Fig. 2 illustrates the relations between the controller parameters obtained from the MIGO design and the process parameters for all stable processes in the test batch. The controller gain is normalized by multiplying it either with the static process gain K_p or with the parameter $a = K_p L/T = K_v L$. The integral and derivative times are normalized by dividing them by T or by L. The controller parameters in Fig. 2 are plotted versus the relative dead time

$$\tau = \frac{L}{L + T} \tag{8}$$

The fact that the ratio L/T is important has been noticed before. Cohen and Coon [38] called L/T the self-regulating index. In [39] the ratio is called the controllability index. The ratio is also mentioned in [23]. The use of τ instead of L/T has the advantage that the parameter is bounded to the region $[0, 1]$.

The parameters for the integrating processes P_6 are only normalized with a and L, since K_p and T are infinite for these processes.

The figure indicates that the variations of the normalized controller parameters are several orders of magnitude. We can thus conclude that it is not possible to find good universal tuning rules that do not depend on the relative time delay τ. Ziegler and Nichols [30] suggested the rules $aK = 1.2$, $T_i = 2L$, and $T_d = 0.5L$, but Fig. 2 shows that these parameters are only suitable for very few processes in the test batch.

The controller parameters for processes P_1 are marked with circles and those for P_2 are marked by squares in Fig. 2. For $\tau < 0.5$, the gain for P_1 is typically smaller than for the other processes, and the integral time is larger. This is opposite to what happened for PI control, see [33]. Process P_2 has a gain that is larger and an integral time that is shorter than for the other processes. These differences are explained in the next subsection.

For PI control, it was possible to derive simple tuning rules, where the controller parameters obtained from the AMIGO rules differed less than 15% from those obtained from the MIGO rules for most processes in the

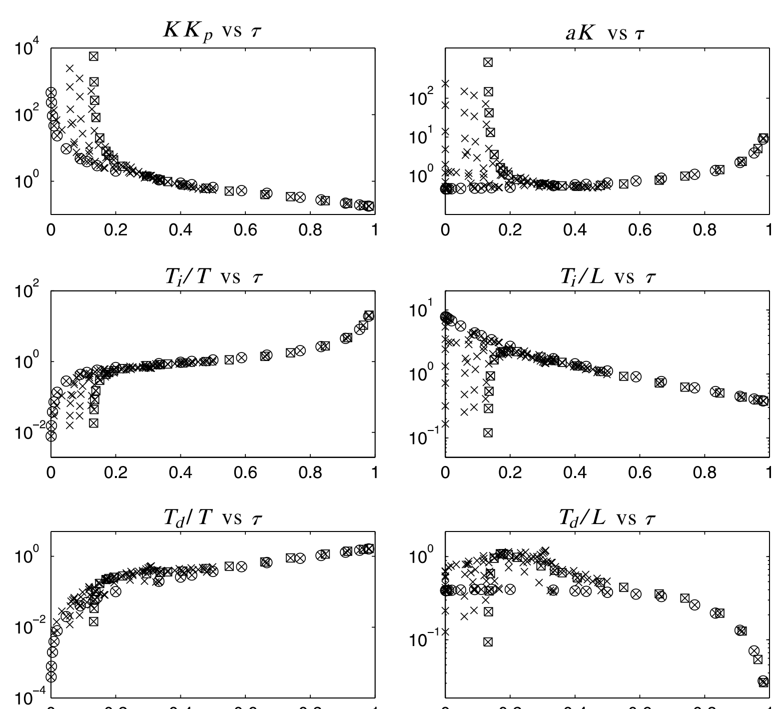

Fig. 2. Normalized PID controller parameters as a function of the normalized time delay τ. The controllers for the process P_1 are marked with circles and controllers for P_2 with squares.

test batch, see [33]. Fig. 2 indicates that universal tuning rules for PID control can be obtained only for $\tau \geqslant 0.5$.

For $\tau < 0.5$ there is a significant spread of the normalized parameters which implies that it does not seem possible to find universal tuning rules. This implies that it is not possible to find universal tuning rules that include processes with integration. This was possible for PI control. Notice that the gain and the integral time are well defined for $0.3 < \tau < 0.5$ but that there is a considerable variation of derivative time in that interval.

Because of the large spread in parameter values for $\tau < 0.5$ it is worth while to model the process more accurately to obtain good tuning of PID controllers. The process models (6) and (7) model stable processes with three parameters and integrating processes with two parameters. In practice, it is not possible to obtain more process parameters from the simple step response experiment. A step response experiment is thus not sufficient to tune PID controllers with $\tau < 0.5$ accurately.

However, it may be possible to find *conservative* tuning rules for $\tau < 0.5$ that are based on the simple models (6) or (7) by choosing controllers with parameters that correspond to the lowest gains and the largest integral times if Fig. 2. This is shown in the next section.

3.3. Large spread of control parameters for small τ

A striking difference between Fig. 2 and the corresponding figure for PI control, see [33], is the large spread of the PID parameters for small values of τ. Before proceeding to develop tuning rules we will try to understand this difference between PI and PID control.

The criterion used is to maximize integral gain k_i. The fundamental limitations are given by the true time delay of the process L_0. The integral gain is proportional to the gain crossover frequency ω_{gc} of the closed loop system. In [40] it is shown that the gain crossover frequency ω_{gc} typically is limited to

$$\omega_{gc} L_0 < 0.5$$

When a process is approximated by the *KLT* model the apparent time delay L is longer than the true time delay L_0, because lags are approximated by additional time delays. This implies that the integral gain obtained for the *KLT* model will be lower than for a design based on the true model. The situation is particularly pronounced for systems with small τ.

Consider PI control of first order systems, i.e. processes with the transfer functions

$$P(s) = \frac{K_p}{1+sT} \quad \text{or} \quad P(s) = \frac{K_v}{s}$$

Since these systems do not have time delays there is no dynamics limitation and arbitrarily high integration gain can be obtained. Since these processes can be matched perfectly by the models (6) and (7), the design rule reflects this property. The process parameters are $L = 0$, $a = 0$, and $\tau = 0$ and both the design method MIGO and the approximate AMIGO rule given in [33] give infinite integral gains.

Consider PID control of second order systems with the transfer functions

$$P(s) = \frac{K_v}{s(1+sT_1)} \quad \text{and} \quad P(s) = \frac{K_p}{(1+sT_1)(1+sT_2)}$$

Since the system do not have time delays it is possible to have controllers with arbitrarily large integral gains. The first transfer function has $\tau = 0$. The second process has values of τ in the range $0 \leqslant \tau < 0.13$, where $\tau = 0.13$ corresponds to $T_1 = T_2$. When these transfer functions are approximated with a *KLT* model one of the time constants will be approximated with a time delay. Since the approximating model has a time delay there will be limitations in the integral gain.

We can thus conclude that for $\tau < 0.13$ there are processes in the test batch that permit infinitely large integral gains. This explains the widespread of controller parameters for small τ. The spread is infinitely large for $\tau < 0.13$ and it decreases for larger τ. For small τ improved modeling gives a significant benefit.

One way to avoid the difficulty is to use of a more complicated model such as

$$P(s) = \frac{b_1 s + b_2 s}{s^2 + a_1 s + a_2} e^{-sL}$$

It is, however, very difficult to estimate the parameters of this model accurately from a simple step response experiment. Design rules for models having five parameters may also be cumbersome. Since the problem occurs for small values of τ it may be possible to approximate the process with

$$P(s) = \frac{K_v}{s(1+sT)} e^{-sL}$$

which only has three parameters. Instead of developing tuning rules for more complicated models it may be better to simply compute the controller parameters based on the estimated model.

We illustrate the situation with an example.

Example 1 (*Systems with same KLT parameters different controllers*). Fig. 3 shows step responses for systems with the transfer functions

$$P_1(s) = \frac{1}{1+5.57s} e^{-0.54s}, \quad P_2(s) = \frac{1}{(1+s)(1+5s)}$$

If a *KLT* model is fitted to these systems we find that both systems have the parameters $K = 1$, $L = 0.54$ and $T = 5.57$, which gives $\tau = 0.17$. The step responses are quite close. There is, however, a significant difference for small t, because the dashed curve has zero response for

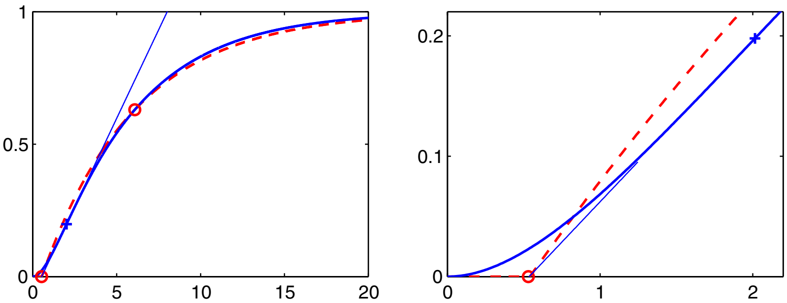

Fig. 3. Step responses of two systems with different dynamics but the same parameters K, L and T. The dashed line represents a system with the transfer function $P_1(s) = e^{-0.54s}/(1 + 5.57s)$ and the full line is the step response of the system $P_2(s) = 1/((1 + s)(1 + 5s))$.

$t < 0.54$. This difference is very significant if it is attempted to get closed-loop systems with a fast response. Intuitively it seems reasonable that controllers with slow response time designed for the processes will not differ much but that controllers with fast response time may differ substantially. It follows from [40] that the gain crossover frequency for P_1 is limited by the time delay to about $\omega_{gc} < 1.0$, corresponding to a response time of about 2. With PI control the bandwidth of the closed loop system for P_2 is limited to $\omega \approx 0.6$. We can thus conclude that with PI control the performances of the closed loop systems are practically the same. Computing controllers that maximize integral gain for $M = 1.4$ gives the following parameters for P_1 and P_2

$$K = 2.97(2.53), \quad T_i = 3.11\ (4.46),$$
$$k_i = 0.96\ (0.57), \quad \omega_{gc} = 0.58(0.47)$$

where the values for P_2 are given in parenthesis.

The situation is very different for PID control. For the process P_1 the controller parameters are $K = 4.9323$, $k_i = 2.0550$, $T_i = 2.4001$ and $T_d = 0.2166$ and $\omega_{gc} = 0.9000$. For the process P_2 the integral gain will be infinite.

Another way to understand the spread in parameter values for small τ is illustrated in Fig. 4 which gives the product of the gain crossover frequency ω_{gc} and the apparent time delay L as a function of τ. The curve shows that the product is 0.5 for $\tau > 0.3$, which is in good agreement with the rule of thumb given in [40]. For smaller values of τ the product may, however, be much larger. There are also substantial variations. This indicates that the value L overestimates the true time delay which gives the fundamental limitations. It should also be emphasized that the performance of delay dominated processes is limited by the dynamics. For processes that are lag dominated the performance is instead limited by measurement noise and actuator limitations, see [40].

3.4. The benefits of derivative action

Since maximization of integral gain was chosen as design criterion we can judge the benefits of derivative action by the ratio of integral gain for PID and PI control. Fig. 5 shows this ratio for the test batch, except for a few processes with a high ratio at small values of τ.

The Figure shows that the benefits of derivative action are marginal for delay dominated processes but that the benefits increase with decreasing τ. For $\tau = 0.5$ the integral gain can be doubled and for values of $\tau < 0.15$ integral gain can be increased arbitrarily for some processes.

3.5. The ratio T_i/T_d

The ratio T_i/T_d is of interest for several reasons. It is a measure of the relative importance of derivative

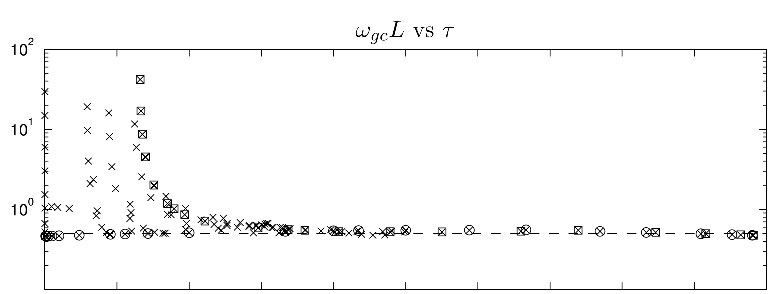

Fig. 4. The product $\omega_{gc}L$ as a function of relative time delay τ. The controllers for the process P_1 are marked with circles and controllers for P_2 with squares.

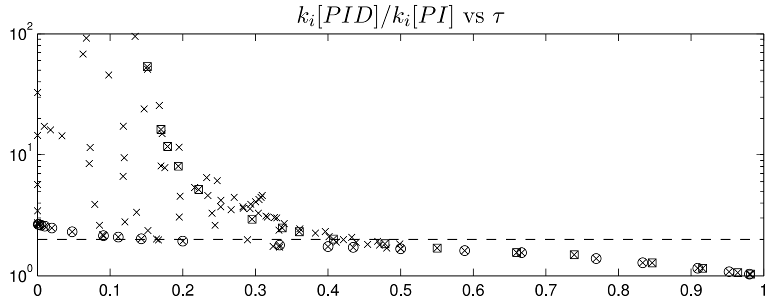

Fig. 5. The ratio of integral gain with PID and PI control as a function of relative time delay τ. The dashed line corresponds to the ratio $k_i[\text{PID}]/k_i[\text{PI}] = 2$. The controllers for the process P_1 are marked with circles and controllers for P_2 with squares.

and integral action. Many PID controllers are implemented in series form, which requires that the ratio is larger than 4. Many classical tuning rules therefore fix the ratio to 4. Fig. 6 shows the ratio for the full test batch. The figure shows that there is a significant variation in the ratio T_i/T_d particularly for small τ. The ratio is close to 2 for $0.5 < \tau < 0.9$ and it increases to infinity as τ approaches 1 because the derivative action is zero for processes with pure time delay. It is a limitation to restrict the ratio to 4. The fact that it may be advantageous to use smaller values was pointed out in [41].

3.6. The average residence time

The parameter T_{63} which is the time when the step response has reached 63%, a factor of $(1 - 1/e)$, of its steady state value is a reasonable measure of the response time for stable systems. It is easy to determine the parameter by simulation, but not by analytical calculations. For the *KLT* process we have $T_{ar} = T_{63}$. The average residence time T_{ar} is in fact a good estimate of T_{63} for systems with essentially monotone step response. For all stable processes in the test batch we have $0.99 < T_{63}/T_{ar} < 1.08$.

The average residence time is easy to compute analytically. Let $G(s)$ be the Laplace transform of a stable system and g the corresponding impulse response. The average residence time is given by

$$T_{ar} = \frac{\int_0^\infty t g(t)\,dt}{\int_0^\infty g(t)\,dt} = -\frac{G'(0)}{G(0)} \tag{9}$$

see [37,42]. Consider the closed loop system obtained when a process with transfer function $P(s)$ is controlled with a PID controller with set-point weighting, given by (1). The closed loop transfer function from set point to output is

$$G_{sp}(s) = \frac{P(s)C_{ff}(s)}{1 + P(s)C(s)}$$

where

$$C_{ff}(s) = bk + \frac{k_i}{s}$$

Straight forward but tedious calculations give

$$T_{ar} = -\frac{G'_{sp}(0)}{G_{sp}(0)} = T_i\left(1 - b + \frac{1}{kK_p}\right) \tag{10}$$

where $T_i = k/k_i$ is the integration time of the controller and $K_p = P(0)$ is the static gain of the system. Fig. 7 shows the average residence times of the closed loop system divided with the average response time of the open loop system. Fig. 7 shows that for PID control the closed loop system is faster than the open loop system when $\tau < 0.3$ and slower for $\tau > 0.3$.

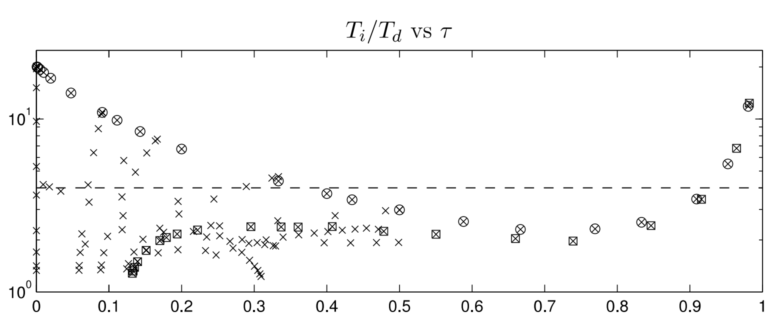

Fig. 6. The ratio between T_i and T_d as a function of relative time delay τ. The dashed line corresponds to the ratio $T_i/T_d = 4$. Process P_1 is marked with circles and process P_2 with squares.

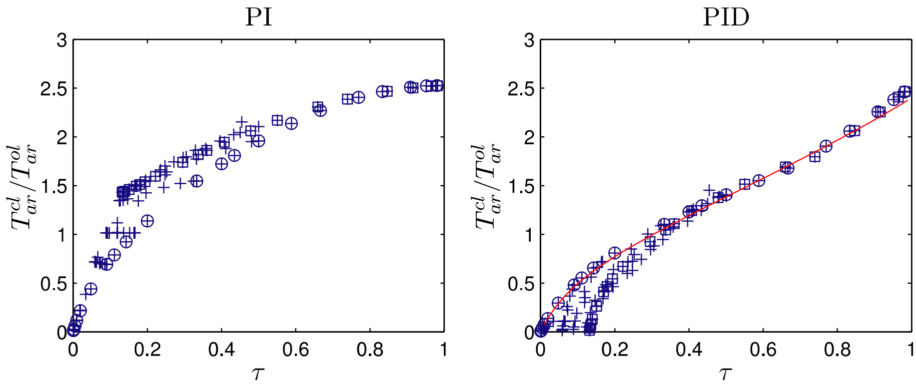

Fig. 7. The ratio of the average residence time of the closed loop system and the open loop system for PI control left and PID control right.

4. Conservative tuning rules (AMIGO)

Fig. 2 shows that it is not possible to find optimal tuning rules for PID controllers that are based on the simple process models (6) or (7). It is, however, possible to find conservative robust tuning rules with lower performance. The rules are close to the MIGO design for the process P_1, i.e. the process that gives the lowest controller gain and the longest integral time, see Fig. 2.

The suggested AMIGO tuning rules for PID controllers are

$$\begin{aligned} K &= \frac{1}{K_p}\left(0.2 + 0.45\frac{T}{L}\right) \\ T_i &= \frac{0.4L + 0.8T}{L + 0.1T}L \\ T_d &= \frac{0.5LT}{0.3L + T} \end{aligned} \tag{11}$$

For integrating processes, Eq. (11) can be written as

$$\begin{aligned} K &= 0.45/K_v \\ T_i &= 8L \\ T_d &= 0.5L \end{aligned} \tag{12}$$

Fig. 8 compares the tuning rule (11) with the controller parameters given in Fig. 2. The tuning rule (11) describes the controller gain K well for process with $\tau > 0.3$. For small τ, the controller gain is well fitted to processes P_1, but the AMIGO rule underestimates the gain for other processes.

The integral time T_i is well described by the tuning rule (11) for $\tau > 0.2$. For small τ, the integral time is well fitted to processes P_1, but the AMIGO rule overestimates it for other processes.

The tuning rule (11) describes the derivative time T_d well for process with $\tau > 0.5$. In the range $0.3 < \tau < 0.5$ the derivative time can be up to a factor of 2 larger than the value given by the AMIGO rule. If the values of the derivative time for the AMIGO rule is used in this range the robustness is decreased, the value of M may be reduced by about 15%. For $\tau < 0.3$, the AMIGO tuning rule gives a derivative time that sometimes is shorter and sometimes longer than the one obtained by MIGO. Despite this, it appears that AMIGO gives a conservative tuning for all processes in the test batch, mainly because of the decreased controller gain and increased integral time.

The tuning rule (11) has the same structure as the Cohen–Coon method, see [38], but the parameters differ significantly.

4.1. Robustness

Fig. 9 shows the Nyquist curves of the loop transfer functions obtained when the processes in the test batch (5) are controlled with the PID controllers tuned with the conservative AMIGO rule (11). When using MIGO all Nyquist curves are outside the M-circle in the figure. With AMIGO there are some processes where the Nyquist curves are inside the circle. An investigation of the individual cases shows that the derivative action is too small, compare with the curves of T_d/L versus τ in Fig. 8. The increase of M is at most about 15% with the AMIGO rule. If this increase is not acceptable derivative action can be increased or the gain can be decreased with about 15%.

4.2. Set-point weighting

In traditional work on PID tuning separate tuning rules were often developed for load disturbance and set-point response, respectively, see [37]. With current understanding of control design it is known that a controller should be tuned for robustness and load disturbance and that set-point response should be treated by using a controller structure with two degrees of freedom. A simple way to achieve this is to use set-point weighting, see [37]. A PID controller with set-point weighting is given by Eq. (1), where b and c are the set-point weights. Set-point weight c is normally set to zero,

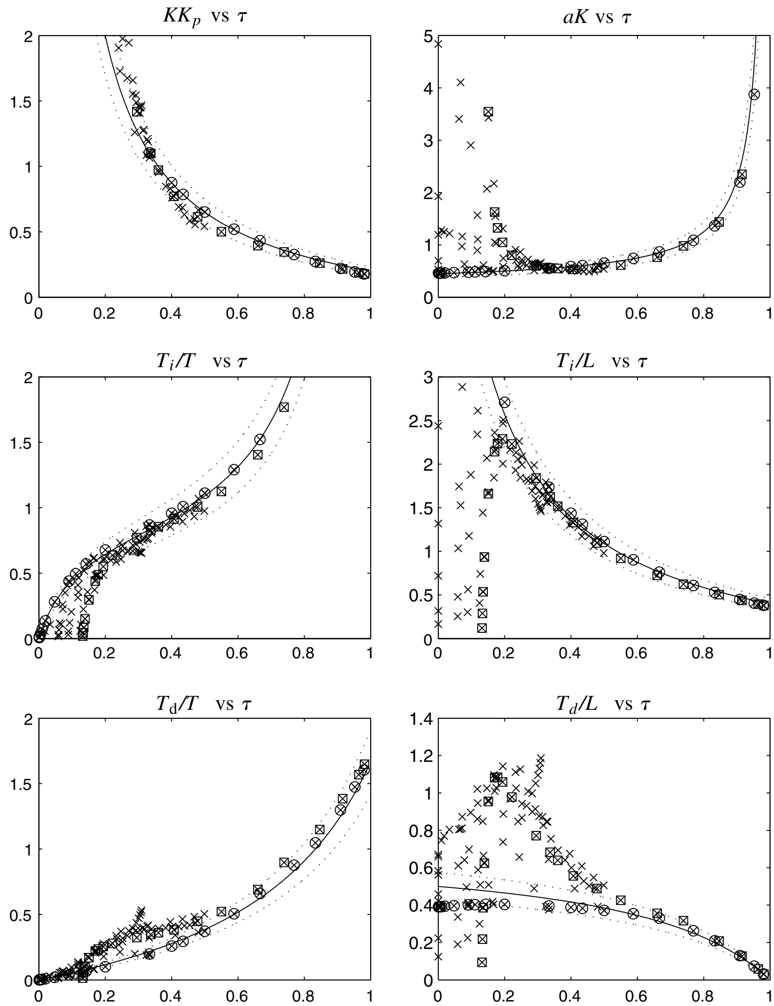

Fig. 8. Normalized controller parameters as a function of normalized time delay τ. The solid line corresponds to the tuning rule (11), and the dotted lines indicate 15% parameter variations. The circles mark parameters obtained from the process P_1, and the squares mark parameters obtained from the process P_2.

except for some applications where the set-point changes are smooth.

A first insight into the use of set-point weighting is obtained from a root locus analysis. With set-point weighting $b = 1$, the controller introduces a zero at $s = -1/T_i$. If the process pole $s = -1/T$ is significantly slower than the zero there will typically be an overshoot. We can thus expect an overshoot due to the zero if $T_i \ll T$. Figs. 2 and 8 show that $T_i \ll T$ for small values of τ. With set-point weighting the controller zero is shifted to $s = -1/(bT_i)$.

The MIGO design method gives suitable values of b. It is determined so that the resonance peak of the transfer function between set point and process output becomes close to one, see [34]. Fig. 10 shows the values of the b-parameter for the test batch (5).

The correlation between b and τ is not so good, but a conservative and simple rule is to choose b as

$$b = \begin{cases} 0 & \text{for } \tau \leqslant 0.5 \\ 1 & \text{for } \tau > 0.5 \end{cases} \tag{13}$$

4.3. Measurement noise

Filtering of the measured signal is necessary to make sure that high frequency measurement noise does not cause excessive control action. A simple convenient approach is to design an ideal PID controller without filtering and to add a filter afterwards. If the noise is not excessive the time constant of the filter can be chosen as $T_f = 0.05/\omega_{gc}$, where ω_{gc} is the gain crossover frequency. This means that the filter reduces the phase margin by 0.1 rad. In Fig. 4 it was shown that for $\tau > 0.2$ we have the estimate $\omega_{gc} \approx 0.5/L$, which gives the filter-time constant $T_f \approx 0.1L$.

For heavier filtering the controller parameters should be changed. This can be done simply by using

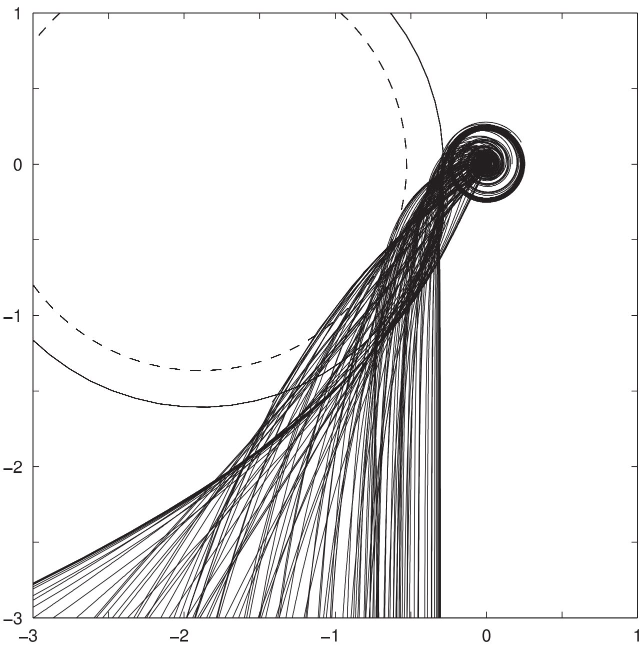

Fig. 9. Nyquist curves of loop transfer functions obtained when PID controllers tuned according to (11) are applied to the test batch (5). The solid circle corresponds $M = 1.4$, and the dashed to a circle where M is increased by 15%.

Skogestads half rule [26] and replacing L and T by $L + T_f/2$ and $T + T_f/2$ in the tuning formula (11).

The effect of filtering on the performance can also be estimated. It follows from (11) that the integral gain is given by

$$k_i = \frac{K}{T_i} = \frac{(0.2L + 0.45T)(L + 0.1T)}{K_p L^2 (0.4L + 0.8T)}$$

Using the half rule and introducing $N = T_d/T_f$ we find that the relative change in integral gain due to filtering is

$$\begin{aligned} \Delta k_i &= \left(\frac{\partial \log k_i}{\partial L} + \frac{\partial \log k_i}{\partial T} \right) \frac{T_d}{2N} \\ &= -\frac{5T(170TL^2 + 197LT^2 + 36T^3 + 26L^3)}{2N(10L + T)(4L + 9T)(L + 2T)(3L + 10T)} \end{aligned} \quad (14)$$

Fig. 11 shows the values of N that give a 5% reduction in k_i for different values of τ. The figure shows that it is possible to use heavy filtering for delay dominated systems. The fact that it is possible to filter heavily without degrading performance is discussed in [41]. Also recall that derivative action is of little value for delay dominated processes.

5. Tuning formulas for arbitrary sensitivities

So far we have developed a tuning formula for a particular value of the design parameter M. It is desirable to have tuning formulas for other values of M. In this section we will develop such a formula for the KLT process (6). It follows from Section 4 that such a formula will be close to the conservative tuning formula given by Eq. (11). Compare also with Fig. 8.

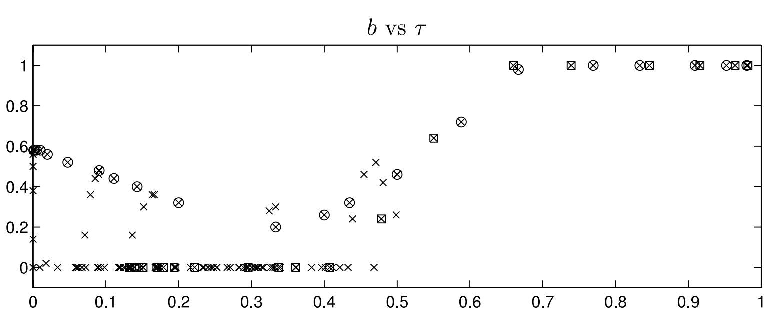

Fig. 10. Set-point weighting as a function of τ for the test batch (5). The circles mark parameters obtained from the process P_1, and the squares mark parameters obtained from the process P_2.

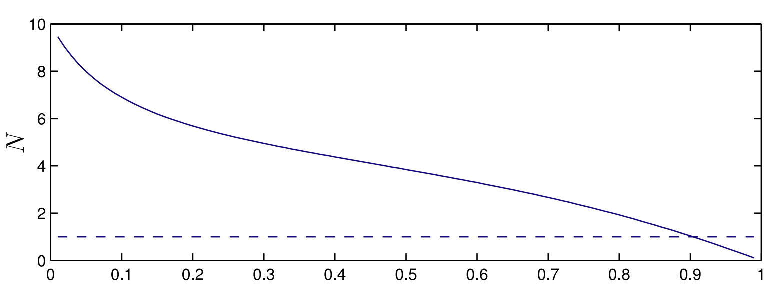

Fig. 11. Filter constants N that give a decrease of k_i of 5%.

Based on Eq. (11) it is natural to represent the controller parameters by

$$\begin{aligned} K &= \frac{\alpha_1 L + \alpha_2 T}{K_p L} \\ T_i &= \frac{\alpha_3 L + \alpha_4 T}{L + \alpha_5 T} L \\ T_d &= \frac{\alpha_6 L T}{L + \alpha_7 T} \end{aligned} \tag{15}$$

To determine the parameters α_i we will compute controller parameters for the processes

$$P^d(s) = e^{-s}, \quad P^b(s) = \frac{1}{s+1} e^{-s}, \quad P^l(s) = \frac{1}{s} e^{-s}$$

which correspond to delay dominated, balanced and lag dominated dynamics. Let these systems have the controllers

$$\begin{aligned} C^d(s) &= K^d \left(1 + \frac{1}{sT_i^d} + sT_d^d\right) \\ C^b(s) &= K^b \left(1 + \frac{1}{sT_i^b} + sT_d^b\right) \\ C^l(s) &= K^l \left(1 + \frac{1}{sT_i^l} + sT_d^l\right) \end{aligned} \tag{16}$$

The formula for controller gain has two parameters α_1 and α_2. To determine these we use the controller parameters computed for delay dominated ($K_p = 1, T = 0, L = 1$), and lag dominated ($T \gg L, K_p/T = 1, L = 1$) processes. Inserting these values in Eq. (15) gives

$$\begin{aligned} \alpha_1 &= K^d \\ \alpha_2 &= K^l \end{aligned} \tag{17}$$

The formula for the integral time has three parameters α_3, α_4 and α_5. To determine these we use the integral times of the controllers for delay dominated ($K_p = 1$, $T = 0$, $L = 1$), balanced ($K_p = 1$, $T = 1$, $L = 1$) and lag dominated ($T \gg L$, $K_p/T = 1$, $L = 1$) processes. Inserting the parameter values in Eq. (15) gives a linear equation for the parameters which has the solution

$$\begin{aligned} \alpha_3 &= T_i^d = 0.3638 \\ \alpha_4 &= \frac{T_i^l (T_i^b - T_i^d)}{T_i^l - T_i^b} = 0.8697 \\ \alpha_5 &= \frac{T_i^b - T_i^d}{T_i^l - T_i^b} = 0.1104 \end{aligned} \tag{18}$$

The formula for the derivative time has two parameters α_6 and α_7. To determine these parameters we use the match the derivative times T_d^d and T_d^l for delay and lag dominated processes. This gives

$$\begin{aligned} \alpha_6 &= T_d^d \\ \alpha_7 &= \frac{T_d^d}{T_d^l} \end{aligned} \tag{19}$$

The derivative time for a pure delay process with $T = 0$ is zero. For finite values of T the derivative gain is limited by the high frequency gain of the loop transfer function. We have for large s

$$P(s)C(s) = \frac{k_d s^2 + ks + k_i}{s(1 + sT)} e^{-sL} \approx \frac{k_d}{T} = \frac{K^d T_d^d}{T}$$

To satisfy the robustness constraint the loop gain must be less than $1 - 1/M$, which implies that the largest derivative time is

$$T_d^d = \left(\frac{1}{M} - 1\right) \frac{T}{K^d} = \frac{M - 1}{K^d M} T$$

Notice that T_d^d goes to zero as T goes to zero.

Table 1 gives the parameters α_i for different values of M. Comparing these values with the values for the tuning formula for conservative tuning, (11) we find that they are very close.

Fig. 12 shows the controller parameters as a function of relative time delay for different values of the tuning parameter. Notice that the gain and integral time varies significantly with M but that the variation in derivative time are much smaller. It follows from Table 1 that the variations in α_6 and α_7 are less than 9% and 3%, respectively. It is thus possible to find values of derivative time that do not depend on the tuning parameter M.

Table 1
Parameters α_i in the tuning formula (15) for different values of M

M	α_1	α_2	α_3	α_4	α_5	α_6	α_7
1.1	0.057	0.139	0.400	0.923	0.012	1.59	4.59
1.2	0.103	0.261	0.389	0.930	0.040	1.62	4.44
1.3	0.139	0.367	0.376	0.900	0.074	1.66	4.39
1.4	0.168	0.460	0.363	0.871	0.111	1.70	4.37
1.5	0.191	0.543	0.352	0.844	0.146	1.74	4.35
1.6	0.211	0.616	0.342	0.820	0.179	1.78	4.34
1.7	0.227	0.681	0.334	0.799	0.209	1.81	4.33
1.8	0.241	0.740	0.326	0.781	0.238	1.84	4.32
1.9	0.254	0.793	0.320	0.764	0.264	1.87	4.31
2.0	0.264	0.841	0.314	0.751	0.288	1.89	4.30

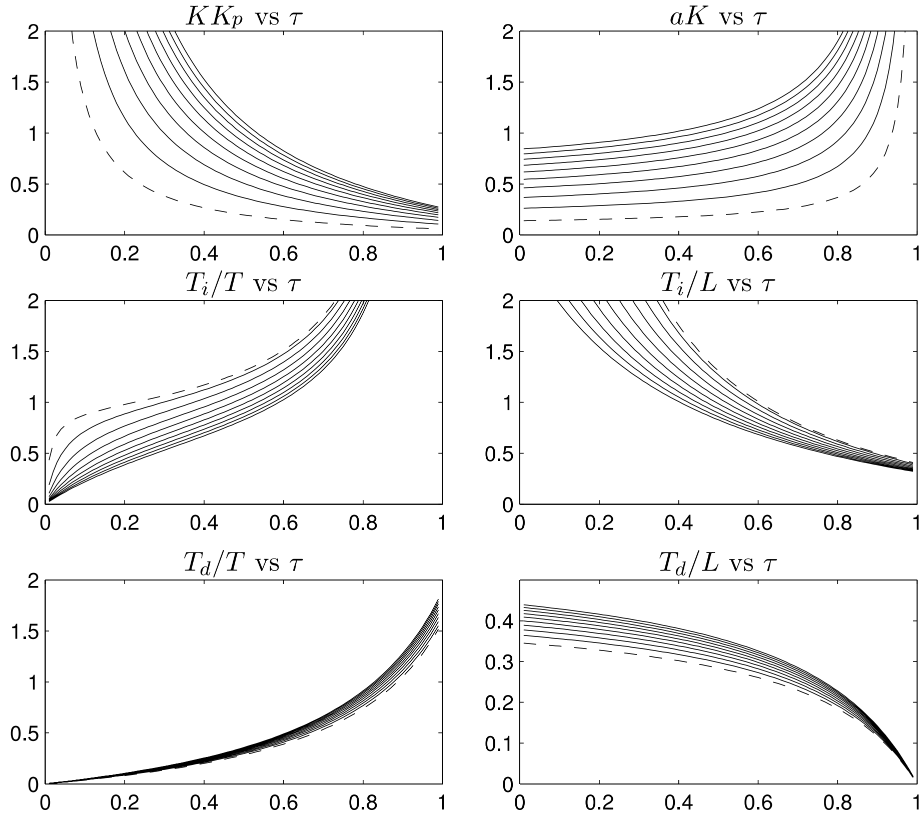

Fig. 12. Controller parameters for the process $P_1(s)$ as a function of relative time delay τ for the tuning parameters $M = 1.1, 1.2, 1.3, \ldots, 2.0$. The curves for $M = 1.1$ are dashed.

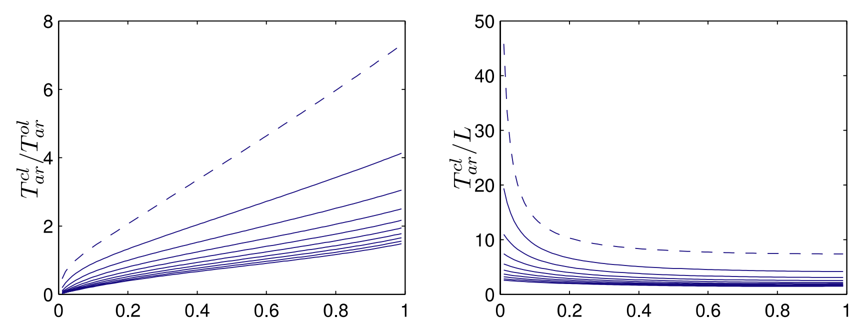

Fig. 13. The ratios $T_{\mathrm{ar}}^{\mathrm{cl}}/T_{\mathrm{ar}}^{\mathrm{ol}}$ and $T_{\mathrm{ar}}^{\mathrm{cl}}/L$ for PID control of the process P_1 with different design parameters $M = 1.1$, (dashed) $1.2, 1.3, \ldots, 2.0$.

5.1. The average residence time

The response time T_{63} is well approximated by the average response time for systems with essentially monotone step responses. The average residence time for a closed loop system under PID control is given by Eq. (10).

Fig. 13 shows the ratio $T_{\mathrm{ar}}^{\mathrm{cl}}/T_{\mathrm{ar}}^{\mathrm{ol}}$ and $T_{\mathrm{ar}}^{\mathrm{cl}}/L$ for PID control of the process.

6. Examples

This section presents a few examples that illustrate the conservative AMIGO method and compares it with the MIGO designs for PI and PID controllers. Three examples are given, one lag-dominant process, one delay-dominant process, and one process with balanced lag and delay.

Example 2 (*Lag dominated dynamics*). Consider a process with the transfer function

$$P(s) = \frac{1}{(1+s)(1+0.1s)(1+0.01s)(1+0.001s)}$$

Fitting the model (6) to the process we find that the apparent time delay and time constants are $L = 0.073$ and $T = 1.03$, which gives $\tau = 0.066$. The dynamics is

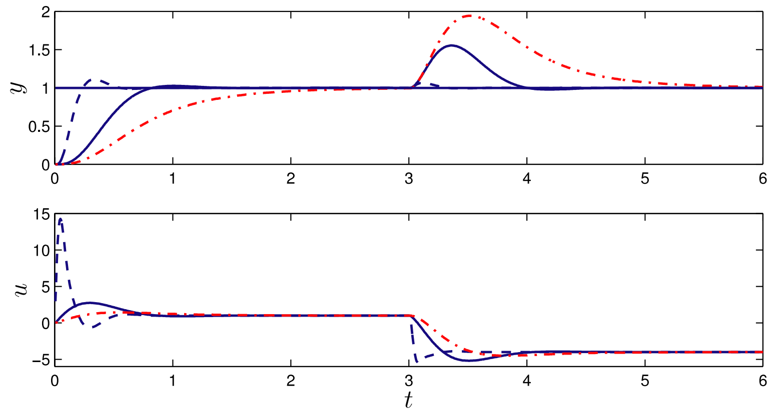

Fig. 14. Responses to a unit step change at time 0 in set point and a load step with amplitude 5 at time 3 for PID controllers designed by AMIGO (full line) and MIGO for PID (dashed line) and PI (dash-dotted line) for a process with the transfer function $G(s) = 1/((1+s)(1+0.1s)(1+0.01s)(1+0.001s))$.

thus lag dominated. Since τ is so small we can expect significant differences between PID and PI control and we can also expect that the conservative AMIGO method is much inferior to the MIGO method.

The MIGO controller parameters are $k_i = 496$, $K = 56.9$, $T_i = 0.115$, and $T_d = 0.0605$ for PID and $k_i = 5.4$, $K = 3.56$, $T_i = 0.660$ for PI. The AMIGO tuning rules, (11) and (13), give the controller parameters $k_i = 18.5$, $K = 6.55$, $T_i = 0.354$, and $T_d = 0.0357$. The set-point weight is $b = 0$ in all cases.

Fig. 14 shows the responses of the system to changes in set point and load disturbances. The figure shows that AMIGO design gives reasonable responses, but that both load disturbance and set-point response are very much inferior compared with the MIGO design. This is expected, since it is a lag-dominant process. The relations between the integral gains are

$$\frac{k_i(\mathrm{MIGO-PID})}{k_i(\mathrm{AMIGO})} = \frac{497}{18.5} \approx 27,$$
$$\frac{k_i(\mathrm{MIGO-PID})}{k_i(\mathrm{MIGO-PI})} = \frac{497}{5.4} \approx 92$$

The response time T_{63} and the average response time T_{ar} for the closed loop systems are 0.16 (0.12), 0.48 (0.41) and 0.89 (0.84) for PID–AMIGO, PID–MIGO and PI respectively. The values of T_{ar} are given in brackets. The average response time is a shorter because the response has an overshoot. This is particularly noticeable for PID–AMIGO.

Notice that the magnitudes of the control signals are about the same at load disturbances, but that there is a major difference in the response time. The differences in the responses clearly illustrates the importance of reacting quickly.

The example shows that derivative action can give drastic improvements in performance for lag dominated processes. It also demonstrates that the control performance can be increased considerably by obtaining better process models than (6).

Next we will consider a process where the lag and the delay are balanced.

Example 3 (*Balanced lag and delay*). Consider a process with the transfer function

$$G(s) = \frac{1}{(s+1)^4}$$

Fitting the model (6) to the process we find that the apparent time delay and time constants are $L = 1.42$ and $T = 2.9$. Hence $L/T = 0.5$ and $\tau = 0.33$. The MIGO controller parameters become $k_i = 0.54$, $K = 1.19$, $T_i = 2.22$, $T_d = 1.20$, and $b = 0$. Since τ is in the mid range we can expect moderate differences between the conservative AMIGO design and the MIGO designs for PID control. We can also expect that the load rejection for the PID controller is at least twice as good as for PI control.

The AMIGO tuning rules (11) give the controller parameters $k_i = 0.47$, $K = 1.12$, $T_i = 2.40$, and $T_d = 0.71$, and from (13) we get $b = 0$. The values of the gain and the integral time are close to those obtained from the MIGO design. The MIGO design gives the following parameters for PI control $k_i = 0.18$, $K = 0.43$, $T_i = 2.43$.

Fig. 15 shows the responses of the system to changes in set point and load disturbances. The figure shows that the responses obtained by MIGO and AMIGO are quite similar, which can be expected because of the similarity of the controller parameters. The integral gains for the PID controllers are also similar, $k_i(\mathrm{MIGO}) = 0.54$ and $k_i(\mathrm{AMIGO}) = 0.47$.

The response time T_{63} and the average response time T_{ar} for the closed loop systems are 5.34 (4.84), 5.22 (4.08) and 5.82 (5.62) for PID–AMIGO, PID–MIGO and PI

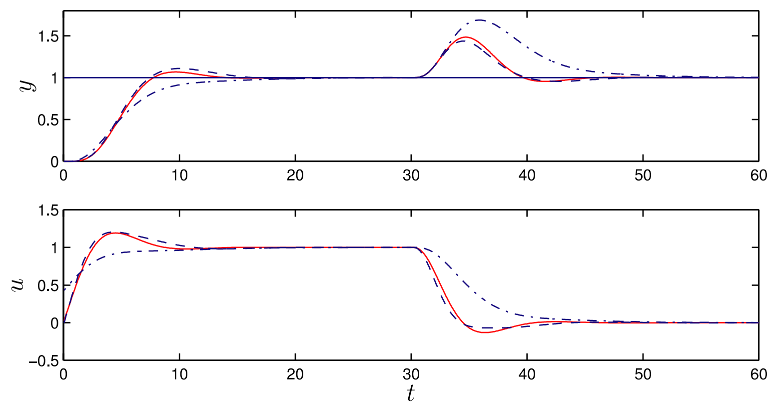

Fig. 15. Responses to a unit step change at time 0 in set point and a unit load step at time 30 for PID controllers designed by AMIGO (full line) and MIGO for PID (dashed line) and PI (dash-dotted line) for a process with the transfer function $G(s) = 1/(1+s)^4$.

respectively. The values of T_{ar} are given in brackets. The average response time is a little shorter because the response has an overshoot.

Finally we will consider an example where the dynamics is dominated by the time delay.

Example 4 (*Delay dominated dynamics*). Consider a process with the transfer function

$$G(s) = \frac{1}{(1+0.05s)^2} e^{-s}$$

Approximating the process with the model (6) gives the process parameters $L = 1.0$, $T = 0.093$ and $\tau = 0.93$. The large value of τ shows that the process is delay dominated. We can thus expect that there are small differences between PI and PID control, and that MIGO and AMIGO give similar performances.

The MIGO controller parameters become $K = 0.216$, $T_i = 0.444$, $T_d = 0.129$, and $b = 1$. The AMIGO tuning rules (11) give the controller parameters $K = 0.242$, $T_i = 0.470$, and $T_d = 0.132$, and from (13) we get $b = 1$.

Fig. 16 shows the responses of the system to changes in set point and load disturbances. The responses of the MIGO and the AMIGO method are similar. The integral gains become $k_i(\text{MIGO}) = 0.49$ and $k_i(\text{AMIGO}) = 0.51$. The response time T_{63} and the average response time T_{ar} for the closed loop systems are 1.95 (2.05), 1.88 (1.94) and 2.34 (2.35) for PID–MIGO, PID–AMIGO and PI respectively. The values of T_{ar} are given in brackets. The estimates of the response times are thus quite good.

This is a process where the benefits of using PID control are small compared to PI control. The MIGO controller parameters for PI control become $K = 0.16$ and $T_i = 0.37$, which gives an integral gain of $k_i = 0.43$. The responses are shown in Fig. 16.

The control signal in Fig. 16 has some irregularities. They can be eliminated by filtering the measured signal by a second order filter. The effective filter time constant is chosen as $T_f = 1/20\omega_{gc} = 0.1L$. The result is shown in Fig. 17.

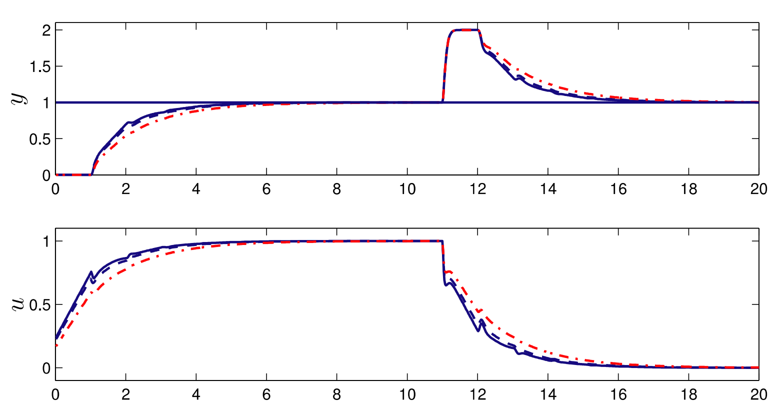

Fig. 16. Responses to step changes in set point and load for PID controllers designed by AMIGO (full line) and MIGO (dashed line) for a process with the transfer function $G(s) = e^{-s}/(1+0.05s)^2$.

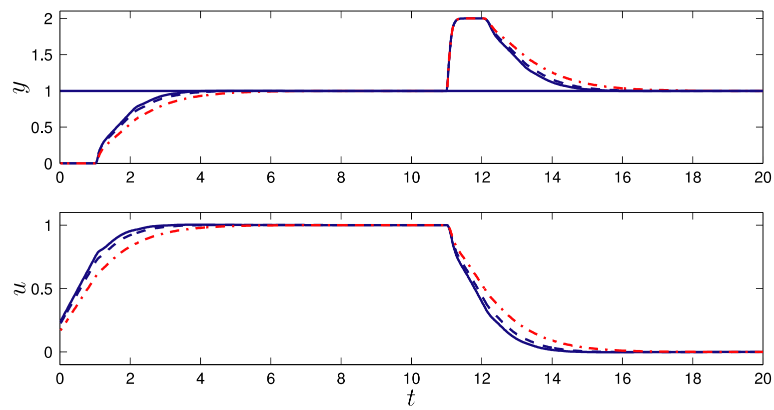

Fig. 17. Responses to step changes in set point and load for PID controllers similar to Fig. 16 but the controller is provided with a filter as described in Section 5.

7. Conclusions

This paper has revisited tuning of PID controllers based on step response experiments in the spirit of Ziegler and Nichols. A large test batch of processes has been used to develop simple tuning rules based on a few features of the step response. The processes are approximated by the *KLT* model representing first order dynamics and a time delay.

All processes in the test batch are tuned using the MIGO design method which maximizes the integral gain k_i subject to robustness constraints. This design method is suitable for control problems where load disturbance rejection is the major concern. The design method does not take set-point changes or noise into account. These aspects should be treated using set-point weighting, set-point filtering, and measurement signal filtering. Guidelines for this have been presented in the paper.

The results show that there are very good correlations between the controller parameters and the process parameters of the *KLT* model for $\tau > 0.5$, where τ is the relative time delay $\tau = L/(L+T)$. For smaller values of τ it is possible to find conservative tuning rules, but in these cases it is possible to find better controller parameters based on improved modeling. The reason is that the simple *KLT* model approximates high order dynamics with a time delay. It is questionable if more accurate models can be obtained based on normal step response measurement.

The conservative AMIGO tuning rules for the design parameter $M = 1.4$ are given by Eq. (11). They are very close to the MIGO parameters obtained for the true *KLT* model. For other processes they may increase the maximum sensitivity up to 15%. The formula works for a full range of process dynamics including processes with integration and pure time delay processes.

The analysis has provided lots of insight, for example that derivative action only gives marginal improvements for τ close to one. For $\tau = 0.5$ the integral gain can be doubled by introducing derivative action. For smaller values of τ the differences can be very significant.

Several rules of thumb are also developed. For example the gain crossover frequency satisfies the inequality $\omega_{gc}L \geqslant 0.5$, which corresponds to the fundamental limitations for a system with time delay L. The inequality is very close to an equality for $\tau > 0.5$, buy very far from equality for small τ.

It is common practice to base tuning rules for PID control on the *KLT* process (P_1). The result of this paper shows that this may be misleading. The results for PI control show that designs based on $P_1(s)$ give too high gain for many of the other processes in the test batch. It is better to base the designs on $P_2(s)$ for PI control. For PID control designs based on $P_1(s)$ seem to work quite well for $\tau > 0.5$. For smaller values of τ designs based on $P_1(s)$ can, however, be extremely conservative.

References

[1] K.J. Åström, T. Hägglund, The future of PID control, Control Engineering Practice 9 (2001) 1163–1175.
[2] M. Morari, E. Zafiriou, Robust Process Control, Prentice-Hall, Englewood Cliffs, NJ, 1989.
[3] E.F. Camacho, C. Bordons, Model Prediction Control in the Process Industry, in: Advances in Industrial Control, Springer-Verlag, Berlin, 1995.
[4] S.J. Qin, T.A. Badgwell, An overview of industrial model predictive control technology, CACHE, AIChE, 1997, pp. 232–256.
[5] L. Desbourough, R. Miller, Increasing customer value of industrial control performance monitoring—Honeywell's experience, in: Sixth International Conference on Chemical Process Control, AIChE Symposium Series Number 326, vol. 98, 2002.
[6] S.G. Akkermans, S.G. Stan, Digital servo IC for optical disc drives, Control Engineering Practice 9 (11) (2002) 1245–1253.
[7] C.L. Smith, Digital Computer Process Control, Intext Educational Publishers, Scranton, PA, 1972.
[8] D.E. Seborg, T.F. Edgar, D.A. Mellichamp, Process Dynamics and Control, Wiley, New York, 1989.
[9] G.K. McMillan, Tuning and Control Loop Performance, Instrument Society of America, Research Triangle Park, NC, 1983.

[10] F.G. Shinskey, Process-Control Systems. Application, Design, and Tuning, fourth ed., McGraw-Hill, New York, 1996.
[11] T.E. Marlin, Process Control, McGraw-Hill, 2000.
[12] A.B. Corripio, Tuning of Industrial Control Systems, Instrument Society of America, 1990.
[13] C.A. Smith, A.B. Corripio, Principles and Practice of Automatic Process Control, Wiley, 1997.
[14] D.C. McFarlane, K. Glover, Robust Controller Design Using Normalized Coprime Factor Plant Descriptions, in: Lecture Notes in Control and Information Sciences, vol. 138, Springer-Verlag, 1990.
[15] D. McFarlane, K. Glover, A loop shaping design procedure using H_∞ synthesis, IEEE Transactions on Automatic Control 37 (6) (1992) 759–769.
[16] S. Skogestad, I. Postlethwaite, Multivariable Feedback Control, Wiley, 1996.
[17] D.E. Rivera, M. Morari, S. Skogestad, Internal model control—4. PID controller design, Industrial & Engineering Chemistry Process Design and Development 25 (1986) 252–265.
[18] D.P. Atherton, PID controller tuning, Computing & Control Engineering Journal April (1999) 44–50.
[19] I.-L. Chien, P.S. Fruehauf, Consider IMC tuning to improve controller performance, Chemical Engineering Progress October (1990) 33–41.
[20] J. Gerry, Tuning process controllers start in manual, InTech May (1999) 125–126.
[21] R. Ciancone, T. Marlin, Tune controllers to meet plant objectives, Control May (1992) 50–57.
[22] A. Haalman, Adjusting controllers for a deadtime process, Control Engineering 65 (July) (1965) 71–73.
[23] A.M. Lopez, P.W. Murrill, C.L. Smith, Tuning PI and PID digital controllers, Instruments and Control Systems 42 (February) (1969) 89–95.
[24] B.W. Pessen, How to 'tune in' a three mode controller, Instrumentation Second Quarter (1954) 29–32.
[25] F.G. Shinskey, How good are our controllers in absolute performance and robustness? Measurement and Control 23 (May) (1990) 114–121.
[26] S. Skogestad, Simple analytic rules for model reduction and PID controller tuning, Journal of Process Control 13 (4) (2003) 291–309.
[27] B.D. Tyreus, W.L. Luyben, Tuning PI controllers for integrator/dead time processes, Industrial & Engineering Chemistry Research (1992) 2628–2631.
[28] M. Zhuang, D. Atherton, Tuning PID controllers with integral performance criteria, in: Control'91, Heriot-Watt University, Edinburgh, UK, 1991.
[29] P. Cominos, N. Munro, PID controllers: recent tuning methods and design to specifications, IEE Proceedings—Control Theory and Applications 149 (1) (2002) 46–53.
[30] J.G. Ziegler, N.B. Nichols, Optimum settings for automatic controllers, Trans. ASME 64 (1942) 759–768.
[31] G. Vinnicombe, Uncertainty and Feedback: $\mathscr{H}_\infty$ Loop-Shaping and The μ-Gap Metric, Imperial College Press, London, 2000.
[32] K.J. Åström, H. Panagopoulos, T. Hägglund, Design of PI controllers based on non-convex optimization, Automatica 34 (5) (1998) 585–601.
[33] T. Hägglund, K.J. Åström, Revisiting the Ziegler–Nichols tuning rules for PI control, Asian Journal of Control 4 (4) (2002) 364–380.
[34] H. Panagopoulos, K.J. Åström, T. Hägglund, Design of PID controllers based on constrained optimisation, IEE Proceedings—Control Theory and Applications 149 (1) (2002) 32–40.
[35] I.M. Horowitz, Synthesis of Feedback Systems, Academic Press, New York, 1963.
[36] H. Taguchi, M. Araki, Two-degree-of-freedom PID controllers—their functions and optimal tuning, in: IFAC Workshop on Digital Control—Past, Present, and Future of PID Control, Terrassa, Spain, 2000.
[37] K.J. Åström, T. Hägglund, PID Controllers: Theory, Design, and Tuning, Instrument Society of America, Research Triangle Park, NC, 1995.
[38] G.H. Cohen, G.A. Coon, Theoretical consideration of retarded control, Trans. ASME 75 (1953) 827–834.
[39] P.B. Deshpande, R.H. Ash, Elements of Computer Process Control with Advanced Control Applications, Instrument Society of America, Research Triangle Park, NC, 1981.
[40] K.J. Åström, Limitations on control system performance, European Journal on Control 6 (1) (2000) 2–20.
[41] B. Kristiansson, B. Lennartsson, Robust and optimal tuning of PI and PID controllers, IEE Proceedings—Control Theory and Applications 149 (1) (2002) 17–25.
[42] L.G. Gibilaro, F.P. Lees, The reduction of complex transfer function models to simple models using the method of moments, Chemical Engineering Science 24 (January) (1969) 85–93.