

Publication [P6]

L. M. Eriksson, M. Johansson, "PID Controller Tuning Rules for Varying Time-Delay Systems", in *Proc. 2007 American Control Conference (ACC2007)*, July 11-13, 2007, New York, USA, 7 p.

© 2007 IEEE. Reprinted, with permission.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of Helsinki University of Technology's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

PID Controller Tuning Rules for Varying Time-Delay Systems

Lasse M. Eriksson and Mikael Johansson

Abstract— This paper considers the design of PID controllers for systems with varying time-delays. Using the concept of jitter margin combined with the AMIGO tuning rule methodology, novel tuning rules that are robust to varying time-delays are derived. In addition, we give an expression for the expected lower bound of the jitter margin as these tuning rules are applied. Extensive numerical evaluations demonstrate that, for wide range of processes, the new tuning rules achieve significant improvements in jitter margin at the expense of only slight decreases in other performance criteria.

I. INTRODUCTION

ANY practical control system suffers from delays. These can stem from process dynamics, actuators, sampling or communication delays. The delays are often either assumed negligible or constant, but in some cases the variance in delay times (the so-called jitter) plays a significant role. There exists a variety of methods for control of time-delay systems with constant delays, but the toolset for dealing with varying time-delays is much more limited. In this paper, PID controller tuning for systems with varying time-delays is discussed and new jitter-robust tuning rules are developed. The tuning rules are based on a KLT-approximation (first-order lag with delay) of the process, possibly obtained via a simple open-loop step response test, and guarantee the closed-loop stability under time-varying delays.

While time-delay systems are abundant in practice, our work is partly motivated by the emerging technology of networked control. There is strong current desire to develop technologies that allow a transition from today's wired infrastructures to more flexible and cost-efficient wireless automation systems. This "wireless migration" requires significant advances in a wide range on technologies, including wireless communication, networking, software engineering, and control. Our focus is on control, and on the design of practical controllers that are robust to the varying time-delays incurred by unreliable communications.

The PID controller is the most common controller in industrial applications today, and it is likely to be the most important controller also in wireless automation solutions. There are many different architectural options for networked PID controllers, including the use of "network observers" [1] to compensate for delay, jitter, losses and other network deficiencies, or simply keeping the architecture of the wired

controller and modifying the controller parameters. This paper takes the second path, and develops new tuning rules that aim at providing good time-domain performance while being robust to varying time-delays. Recently, the problem of PID controller tuning for integrating processes with varying time-delays has been considered in [2], but here the focus is on non-integrating processes.

The paper is organized as follows: Section II discusses the preliminaries required to understand the problem and presents the methods used. In Section III, the problem is formulated and solved, and the results are used in Section IV to derive the tuning rules. In Section V the advantages of using the proposed rules are shown via simulations, and Section VI states the conclusions.

II. PRELIMINARIES

A. PID controller

The PID (proportional-integral-derivative) controller is the most common controller used in industry. There are several versions of the basic algorithm

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

$$e(t) = y_r(t) - y(t),$$

where u is the control signal, e the error signal, y_r the reference signal, y the process variable, K the gain, T_i the integration time, and T_d is the derivative time. In this paper we consider the following version of the PID controller, since the "text book" version (1) is very sensitive to noise.

$$u(t) = k \left(b y_r(t) - y_f(t) \right) + k_i \int_0^t \left(y_r(\tau) - y_f(\tau) \right) d\tau + k_d \left(c \frac{dy_r(t)}{dt} - \frac{dy_f(t)}{dt} \right) \quad (2)$$

Here k , k_i , and k_d are the controller gains, b and c the set-point weights, and y_f is the filtered process variable such that

$$Y_f(s) = G_f(s)Y(s) = \frac{1}{(1+sT_f)^2} Y(s), \quad (3)$$

where $Y(s)$ is the Laplace transform of the process variable $y(t)$. T_f is the filter time-constant.

B. Jitter margin

The jitter margin is an upper bound for additional delay that can be added to a closed-loop control system while maintaining stability. The delay can be of any type (constant, time-dependent, random), but the jitter margin determines the upper bound for the delay. The formal definition of the

L. M. Eriksson is a Ph.D. student with the Control Engineering Laboratory, Helsinki University of Technology, P.O.B. 5500, Espoo, FI-02015 Finland (e-mail: lasse.eriksson@tkk.fi).

M. Johansson is an associate professor with the Automatic Control, Royal Institute of Technology, Stockholm, SE-10044 Sweden (e-mail: mikaelj@s3.kth.se).

jitter margin is given in [3], where three different controller/plant–uncertainty combinations are investigated. The first one is shown in Fig. 1, left, where a continuous-time plant and a continuous-time controller with controller output uncertainty are shown. This continuous-time SISO system is stable for any time-varying delays defined by

$$\Delta(v) = v(t - \delta(t)), \quad 0 \leq \delta(t) \leq \delta_{\max}, \quad (4)$$

if

$$\left| \frac{P(j\omega)C(j\omega)}{1 + P(j\omega)C(j\omega)} \right| < \frac{1}{\delta_{\max} \omega}, \quad \forall \omega \in [0, \infty[. \quad (5)$$

δ_{\max} is the jitter margin. The proof of the result is based on presenting the uncertainty (varying delay) with an operator $\Delta_F := (\Delta - 1) \circ 1/s$ (s being the Laplace operator) and on the small gain theorem.

However, in this paper the jitter is assumed to be after the plant (e.g. sampling jitter) as depicted in Fig. 1, right. Since the signals in the control loop are all continuous, and only the plant and controller switch their positions, the small gain theorem-based stability proof still holds for the control system of Fig. 1, right.

C. AMIGO tuning rules

The objective of this work was to develop tuning rules for the PID controller in varying time-delay systems. The AMIGO tuning rules [4] were selected as the point of comparison, since they provide controllers with good performance and robustness properties. The AMIGO tuning rules are based on the KLT-process model obtained with a step response experiment. The most well-known step response-based tuning rules were presented by Ziegler and Nichols in 1942 [5]. The KLT-process model is given

$$P_{KLT}(s) = \frac{K_p}{1 + sT} e^{-sL}, \quad (6)$$

where K_p is the static gain, T the time-constant, and L the time-delay. The parameters can be estimated from a single step experiment by drawing a tangent to the inflection point of the response. The delay is estimated from the intersection of the tangent and the response initial value, the time-constant from the intersection of the tangent and the response final value (from which the delay is subtracted), and the static gain from the ratio of response and step final values (see [4] for details). The AMIGO rules were developed by analyzing different properties (performance, robustness etc.) of a process test batch.

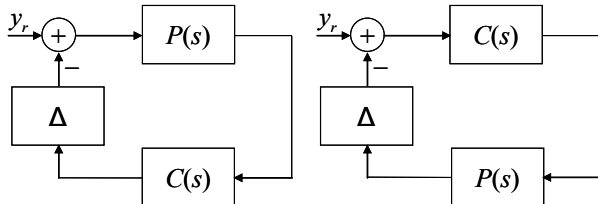


Fig. 1. Continuous-time controller (C) and plant (P) with an uncertain time-varying delay (Δ) in the feedback loop. On the left, Δ is the controller output uncertainty. On the right, Δ is the process output uncertainty.

The AMIGO tuning rules are

$$K = \frac{1}{K_p} \left(0.2 + 0.45 \frac{T}{L} \right), \quad T_i = \frac{0.4L + 0.8T}{L + 0.1T} L, \quad (7)$$

$$T_d = \frac{0.5LT}{0.3L + T}.$$

In order to use the PID controller with filtering (2), the rules are extended as follows [4]:

$$\begin{cases} k = K \\ k_i = \frac{K}{T_i} \\ k_d = K \cdot T_d \end{cases} \quad \begin{cases} b = \begin{cases} 0, & \text{if } \tau \leq 0.5 \\ 1, & \text{if } \tau > 0.5 \end{cases} \\ c = 0 \\ T_f = \begin{cases} 0.05 / \omega_{gc}, & \text{if } \tau \leq 0.2 \\ 0.1 \cdot L, & \text{if } \tau > 0.2. \end{cases} \end{cases} \quad (8)$$

where ω_{gc} is the gain crossover frequency and

$$\tau = \frac{L}{L + T} \quad (9)$$

is the relative dead-time of the process, which has turned out to be an important process parameter for controller tuning.

The development of the AMIGO rules was based on the following robustness criterion: if the Nyquist curve of the loop transfer function does not intersect a circle with center c_R and radius r_R defined as

$$c_R = -\frac{2M^2 - 2M + 1}{2M(M - 1)}, \quad r_R = \frac{2M - 1}{2M(M - 1)}, \quad (10)$$

then the sensitivity function and the complementary sensitivity function are less than M [6]. The robustness is thus captured by one parameter only, M . The value $M = 1.4$ was used in the AMIGO rule development, although finally the rules did not quite satisfy the constraint. For the process test batch a 15 % increase of M was reported.

In order to get some insight of the jitter margin properties of the KLT-processes with AMIGO tuning, the jitter margin was calculated for processes with $K_p = 1$ and different values of T and L ranging from 0.1 to 10. The numerical analysis revealed that the jitter margin of the control loops with AMIGO tuned controllers and a KLT-process depends mainly on process delay as

$$\delta_{\max, AMIGO} \approx 0.71 \cdot L. \quad (11)$$

It was seen that the jitter margin does not depend significantly on the time-constant, but it is rather a function of delay. For delay-dominant ($L \gg T$) processes this is favorable, since the jitter margin is approximately 70 % of the nominal delay. But for lag dominated ($T \gg L$) processes this indicates that the AMIGO rules might give rather poor jitter margins, since the jitter margin does not depend on T .

D. Multi-objective optimization

The new tuning rules for systems with varying time-delays are developed in this paper by optimizing the PID parameters with respect to several objectives. In order to solve the optimization problem we use a multi-objective constrained optimization method. A general multi-objective optimization (here minimization) problem is given as

$$\begin{aligned} \text{Minimize } F(x) &= \{f_1(x) \cdots f_l(x) \cdots f_k(x)\} \quad (12) \\ \text{s.t. } x &= [x_1 \cdots x_n]^T \in \Omega \\ \Omega &= \left\{ x \left| \begin{array}{l} g_i(x) \leq 0 \quad i = 1, \dots, m_1 \\ h_j(x) = 0 \quad j = 1, \dots, m_2 \end{array} \right. \right\}, \end{aligned}$$

where $f_l(x)$ are nonlinear objective functions that are to be minimized simultaneously, x_i the decision variables and $g_i(x)$ and $h_j(x)$ the nonlinear inequality and equality constraints, respectively (e.g. [7]). There are numerous algorithms for solving the problem (12), of which the goal attainment method will later be used in deriving the tuning rules. The goal attainment problem is defined as

$$\begin{aligned} \text{Minimize } \gamma \quad (13) \\ \text{s.t. } F(x) - \alpha \cdot \gamma \leq F_g \\ x \in \Omega, \end{aligned}$$

where γ is an auxiliary variable, α is a vector of weights and F_g is a vector of goals, i.e. the objective function values that should be attained.

III. JITTER-AWARE TUNING FOR THE KLT-PROCESS

This paper aims at producing PID tuning rules that ensure robustness against time-delay variations in the control loop. The rules can be applied based on the KLT-process model of the plant that can be easily obtained with a single step response experiment. The experiment should be recorded such that the varying time-delay does not affect the response. The KLT-process model should only capture the lag and the dead time of the process. Here, we restrict to stable processes.

A. Problem formulation

In order to derive the new tuning rules for the varying time-delay systems, it is first analyzed how the PID controller should be optimally tuned such that the performance and the jitter margin of the closed-loop system would be maximized simultaneously, while the robustness of the system would remain at the same level as with the AMIGO rules. For this investigation the following objective functions are set:

$$f_1(x) = \int_0^{\infty} t |e(t)| d\tau = \int_0^{\infty} t |y_r(t) - y(t)| d\tau, \quad (14)$$

$$f_2(x) = \frac{1}{\delta_{\max}} = \frac{1}{\min_{\omega \in [0, \infty[} |1 + H(j\omega)|}}, \quad (15)$$

where

$$H(j\omega) = C(j\omega)P(j\omega). \quad (16)$$

For the optimization problem the following vectors, variables and constraint functions are defined.

$$F(x) = [f_1(x) \quad f_2(x)], \quad x = [k \quad k_i \quad k_d]^T > 0, \quad (17)$$

$$g(x) = \min_{\omega} \left(\sqrt{(\operatorname{Re}(H) - c_R)^2 + (\operatorname{Im}(H))^2} - r_R \right) > 0, \quad (18)$$

$$h(x) = \emptyset.$$

The objective function $f_1(x)$ is the ITAE criterion and $f_2(x)$ is the inverse of the jitter margin. The functions are written in a form where their values should be minimized to get the best results. The total objective function $F(x)$ and the optimization variable x are defined in (17). The optimized parameters x are the PID controller gains k , k_i and k_d . The other controller parameters (T_f , b , and c) are set by the AMIGO rules, (8). If the constraint function $g(x)$ in (18) is positive as required, $g(x)$ determines the distance between the Nyquist curve of the loop transfer function and the “robustness circle” (10) in the complex plane. Negative values of $g(x)$ would indicate Nyquist curves intersecting the robustness circle. The robustness requirement (10) is thus treated as a hard constraint.

B. Solving the optimization problem

To solve the problem presented in (14) - (18), we use the goal attainment method. It provides means to define goal values for the objective functions. The goals are chosen as

$$F_g = \left[J_{AMIGO} \quad \frac{1}{T + L} \right], \quad (19)$$

where J_{AMIGO} is the ITAE cost criterion value if AMIGO tuning rules were used for the controller. For the jitter margin, the goal is $T + L$, since this gives relatively large jitter margins for both delay-dominant and lag dominated processes. The weighting of the goals is another way to affect the optimization results. If equal under- or overattainment of objectives is desired, the weights are chosen $\alpha = |F_g|$. The smaller the weight, the more the respective objective is considered. Based on the discussion in Section II.C, the jitter margin goal is relatively large compared to the jitter margin provided by the AMIGO rules for lag dominated processes. Because of the trade-off between the objectives, choosing the weights equal to the goal values would presumably result in poor performance for these processes. To avoid significant decrease in performance when $T \gg L$, the weights are set otherwise equal to the goal values, but the weight of the second objective is multiplied by $2T$. Hence, the weights are $\alpha = [J_{AMIGO} \quad 2T / (T + L)]$. For the robustness constraint we set $M = 1.5$, since the AMIGO rules give $M \approx 1.4 - 1.6$.

The optimization is done using a simulation model with controller (2) and process (6). The objective function values are calculated at each iteration, and the optimization is run until the values of the functions do not decrease further. A unit step at 1 s is used as the reference signal that is applied to the disturbance input (y_r in Fig. 1), and the simulation is run until the response of the system reaches the reference signal after the step is given and the error remains in zero.

C. Optimization results

The PID controller parameters were optimized using MATLAB's *fgoalattain* function that implements the goal attainment method described above. The AMIGO tuning rules were used as initial values for the optimization. Different values of process time-constants and delays were used. The optimization was run using 11 values for both variables,

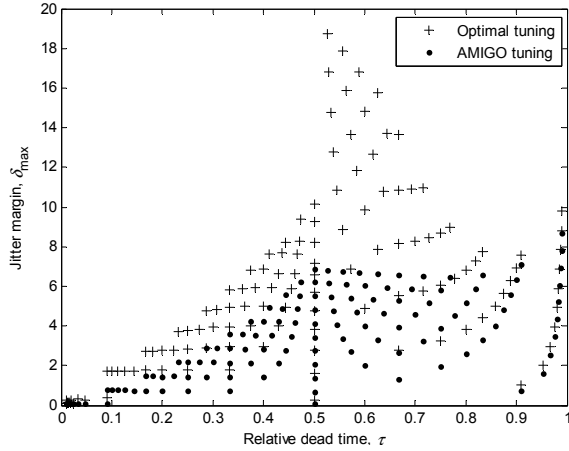


Fig. 2. Jitter margin of the KLT-processes after optimization and with AMIGO tuning rules.

whereas the process gain K_p was kept constant, since it only scales the controller gains (see (7)). The values used in the optimization were

$$K_p = 1, T = [0.1 \ 1 \ \dots \ 10], L = [0.1 \ 1 \ \dots \ 10]. \quad (20)$$

The purpose of the optimization was to produce tuning rules that give the closed-loop system a large jitter margin. Fig. 2 compares the jitter margin of the KLT-processes with parameters (20) between AMIGO tuning rules and the optimized tuning parameters. The jitter margin is plotted with respect to the relative dead time (9) of the process. Investigation of the figure reveals that the jitter margin is improved for all processes in the optimization. The relative improvement of the jitter margin compared to the AMIGO tuning is shown in Fig. 3 (upper graph). The jitter margin is significantly improved for processes with $\tau < 0.8$.

The lower graph in Fig. 3 shows the ITAE criterion ratio between AMIGO and optimal tuning. The performances are equal if the percentage is 100 %, and the lower percentages indicate decrease in performance for the optimized tuning. The performance of the closed-loop system is not necessarily as good as with the AMIGO rules, and in some cases the jitter margin improvement comes at the cost of lower performance. There are, though, quite a few controllers that perform equal to the AMIGO rules, but simultaneously give better jitter margins. To be precise, 60 % of the controllers give performance index over 90 % indicating that the ITAE cost criterion value is at maximum 10 % higher than with the AMIGO rules. Over 90 % of the controllers give performance index over 80 %. Only 1.5 % of the controllers perform poorly (performance between 50 % and 60 %), and these are the ones with $\tau < 0.1$, i.e. if $L \ll T$. Nevertheless, for these processes the jitter margin has improved enormously. This is because of two reasons: first, the AMIGO tuning gives very small jitter margins for small L , and second, the objective value of the jitter margin in the optimization is high relative to the nominal delay. Consider, for example, the process with $L = 0.1$ and $T = 1$. The jitter margin goal is 1.1, i.e. 11 times bigger than the delay, and it is

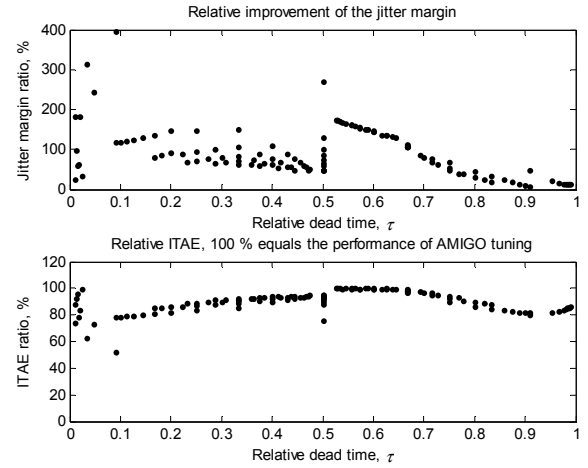


Fig. 3. Improvement of the jitter margin and performance comparison with respect to the AMIGO tuning rules.

almost 16 times bigger than the jitter margin with AMIGO rules. In these cases, it is hard to reach the goal without sacrificing the performance.

IV. PROPOSED TUNING RULES

A. Derivation of the tuning rules

The PID controller parameters k , k_i , and k_d obtained from the optimization were plotted with respect to time-constant and delay. The surfaces were smooth overall, but showed somewhat irregular behavior for small values of delay. The tuning rules were designed by approximating the surfaces with explicit functions of T and L .

First, the problem was scaled into 2D. The parameters were plotted against time-constant for fixed values of delay, and against delay for fixed values of time-constant. These figures are shown for k and k_i in Fig. 4 (logarithmic scale for parameters). Using these figures, it was possible to derive the model structures for the tuning rules. For example, in the upper two plots of Fig. 4 it is seen that the parameter k is proportional to the time-constant and inversely proportional to the delay. Thus a natural candidate for the tuning rule structure would be

$$k(T, L) = \frac{1}{K_p} \left(a_k \frac{T}{L} + b_k \right) \quad (21)$$

a_k and b_k being the coefficients to be estimated from the optimized surface of k . In fact, this model structure equals to the AMIGO rules. It turned out that the model structure did not have enough degrees of freedom, since the estimated surface error was large. Thus, a slightly more complicated structure was chosen.

The search for a good model structure candidate was initiated by investigating how the PID parameter k depends on the time-constant at different values of delay. For each value of delay L , a line was fitted

$$k(T, L_j) = \theta_{1,j}^k T + \theta_{2,j}^k, \quad (22)$$

where L_j is a constant value of delay (j^{th} line), and $\theta_{1,j}^k$ and $\theta_{2,j}^k$ are parameters of the j^{th} fitted line. In other words, the

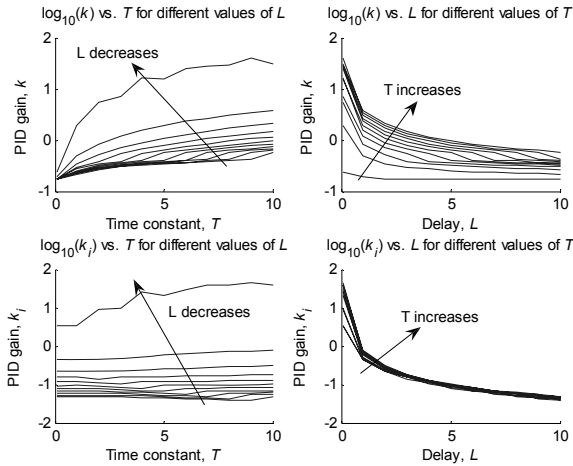


Fig. 4. Gains k and k_i with respect to process time-constant T and delay L .

dependency on parameters was separated such that k is proportional to the time-constant, and the parameters θ^k are functions of the delay, i.e.

$$k(T, L) = \theta_1^k(L) \cdot T + \theta_2^k(L). \quad (23)$$

The parameter θ_1^k is shown as a function of delay in Fig. 5. The parameter is clearly inversely proportional to the delay. Since also θ_2^k was observed to be inversely proportional to delay, the following model structure was chosen for k .

$$k(T, L) = \theta_1^k(L) \cdot T + \theta_2^k(L) = \underbrace{\frac{a_1^k}{L}}_{\theta_1^k(L)} T + \underbrace{\frac{a_2^k}{L} + b_2^k}_{\theta_2^k(L)} \quad (24)$$

a_1^k , a_2^k and b_2^k are the coefficients estimated from the optimized surface of k .

The model structure for k_i was chosen using the same approach as for k , since the dependencies on T and L seemed to have similar shapes (see Fig. 4). Nevertheless, the model needed to have more degrees of freedom. Different from modeling of k , here the search was initiated by first fitting curves for each value of the *time-constant*, not delay. The following model structure was chosen.

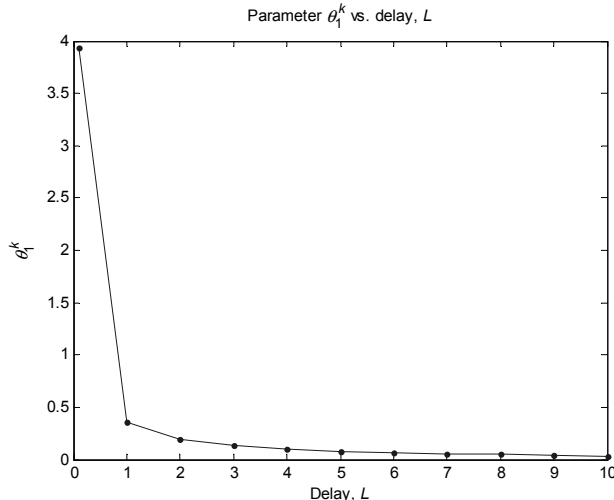


Fig. 5. Parameter θ_1^k with respect to delay.

$$k_i(T, L) = \frac{\theta_1^i(T)}{L^2} + \frac{\theta_2^i(T)}{L} \quad (25)$$

Furthermore, the following dependencies were identified

$$\begin{aligned} \theta_1^i(T) &= a_1^i T^3 + b_1^i T^2 + c_1^i, \\ \theta_2^i(T) &= a_2^i T^2 + b_2^i T + c_2^i, \end{aligned} \quad (26)$$

which gave the final model structure

$$k_i(T, L) = \frac{a_1^i T^3 + b_1^i T^2 + c_1^i}{L^2} + \frac{a_2^i T^2 + b_2^i T + c_2^i}{L}. \quad (27)$$

Derivative gain k_d was simpler to model, and the resulting model structure is

$$k_d(T, L) = a_1^d T^2 + a_2^d T. \quad (28)$$

B. Tuning rules

The proposed tuning rules are based on the KLT-process model. The coefficients and rule structures were obtained by analyzing 121 KLT-processes with different values for time-constants and delays. It should be noted that the rules are derived for time-constants and delays in the range $[0.1, 10]$, and using the formulas outside of these ranges might give infeasible parameters. However, the method for finding the tuning rules as presented in this paper could be applied for deriving tuning rules for processes that are out of the range of these rules.

The proposed jitter-aware PID controller tuning rules are

$$k = \frac{1}{K_p} \left(\frac{0.4T - 0.04}{L} + 0.16 \right), \quad (29)$$

$$k_i = \frac{1}{100K_p} \left(\frac{-0.11T^3 + 1.5T^2 - 1.5}{L^2} + \frac{0.35T^2 + 4T + 50}{L} \right), \quad (30)$$

$$k_d = \frac{1}{100K_p} (0.4T^2 + 11T). \quad (31)$$

V. PERFORMANCE COMPARISON

This section gives an example of applying the proposed tuning rules and also compares them with AMIGO tuning. First, a general comparison is made by analyzing the performance, robustness and jitter margin criteria for the KLT-process.

A. Comparison of the rules

The tuning rules (29) - (31) are approximations of the optimized tuning parameters, and thus differences in performance, jitter margin and robustness are expected when comparing the rules and the optimized parameters. In the following figures the proposed tuning rules are compared with AMIGO tuning in various aspects. In the upper plot of Fig. 6 the improvement of the jitter margin is presented for the KLT-process with parameters (20) similarly as in Fig. 3. For example, value 100 % indicates that the proposed tuning rules give 100 % better jitter margin than the AMIGO rules. The lower plot of Fig. 6 shows the ITAE ratio of AMIGO and proposed tuning. The proposed tuning rules give better jitter margin for all KLT-processes. In 13 % of the processes also the performance increases which means that there are

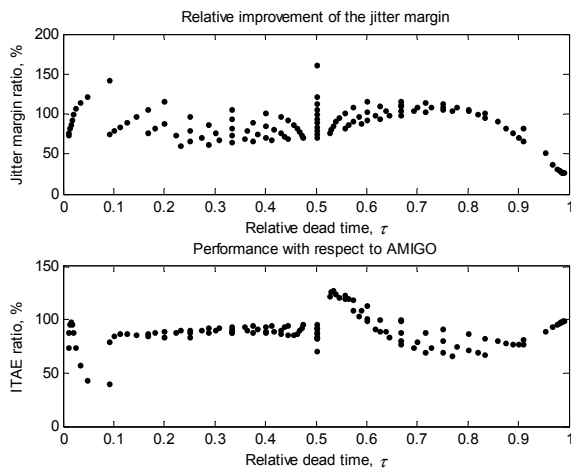


Fig. 6. Improvement of the jitter margin and performance comparison between AMIGO and proposed tuning rules.

processes for which both the jitter margin and performance are improved. It can be seen that the jitter margin increases by over 50 % for all processes where $0 \leq \tau \leq 0.95$. The relative decrease of performance is greatest for certain processes where τ is small, i.e. $L \ll T$, as was the case for the optimized parameters (see Fig. 3). The relative increase of the jitter margin is smallest for large values of τ . This is because, as mentioned before, the AMIGO rules give quite good jitter margins for the delay-dominant processes where $0.9 \leq \tau \leq 1$, i.e. $L \gg T$. In the cases with large L it might pay off to use more sophisticated control algorithms to compensate for the delay (e.g. the Smith predictor, [8]).

The proposed tuning rules were also experimented with the process test batch that was used in the development of AMIGO rules (133 processes, see [4]). For the non-integrating processes, whose KLT-parameters are in the required range, the proposed tuning rules give significant enhancements in jitter margin and performance.

The jitter margin of the proposed tuning rules seems to have a smoothly behaving lower bound as can be seen in Fig. 6. It is possible to estimate the lower bound of the jitter margin based on τ . Let $f(\tau)$ be this lower bound and $\delta_{\max,NEW}(\tau)$ the jitter margin of the proposed tuning rules. The lower bound is smaller than the relative growth (percentage) of the jitter margin (Fig. 6) for all τ , i.e.

$$f(\tau) \leq \left(\frac{\delta_{\max,NEW}(\tau) - \delta_{\max,AMIGO}(\tau)}{\delta_{\max,AMIGO}(\tau)} \right) \cdot 100\%, \quad 0 \leq \tau \leq 1. \quad (32)$$

Based on (11) and (32), the jitter margin for the KLT-process when applying the proposed tuning rules is approximately

$$\delta_{\max,NEW} \geq 0.71 \cdot L \cdot (f(\tau) + 1). \quad (33)$$

The function $f(\tau)$ was estimated from the lowest jitter margin values in Fig. 6. The following function was obtained.

$$f(\tau) = -12.3\tau^4 + 17.1\tau^3 - 5.5\tau^2 + 0.72. \quad (34)$$

These equations give an approximation of the jitter margin based on the KLT-parameters without a need for more complicated calculations such as (5).

In Fig. 7 the robustness of the proposed tuning rules is evaluated by drawing the loop transfer function Nyquist curves and the robustness circle (10) with $M = 1.5$. The figure shows that most of the curves do not intersect the robustness circle, but there are few curves that go inside the circle indicating that the robustness constraint is not completely fulfilled. This also happens with the AMIGO tuning rules. A profound comparison reveals that the robustness is on the average improved with the new rules.

B. Simulation example

There are nine different types of processes in the process test batch with different values of parameters (typically delay and time-constant). For example, process type P_1 is the KLT-process with process gain $K_p = 1$, delay $L = 1$ and time-constant ranging from 0.02 to 1000. To demonstrate the superiority of the proposed tuning, we select for a simulation example the process type P_9 , for which the AMIGO tuning gives a relatively small jitter margin. The processes P_9 are given

$$P_9(s) = \frac{1}{(s+1)((sT_p)^2 + 1.4sT_p + 1)}, \quad T_p = 0.1, 0.2, \dots, 1.0. \quad (35)$$

To obtain better jitter margins for the processes (35) the proposed tuning rules are applied. First, KLT-approximations of the processes are required. Regardless of the value of parameter T_p , the process gain of (35) is $K_p = 1$. The upper plot of Fig. 8 shows the KLT-parameter values T and L for processes P_9 . The lower plot shows the jitter margins when AMIGO tuning or the proposed tuning are applied based on the KLT-models. It can be seen that the KLT-parameters fit in the range of the new tuning rules, and they can be applied. The proposed tuning gives considerably larger jitter margins for processes (35).

The nominal (without additional delay) step responses of the P_9 -processes with AMIGO tuning and proposed tuning are shown in Fig. 9. AMIGO gives responses with overshoot whereas the proposed tuning gives better damped responses. The settling times are approximately the same for both tuning rules. But when an additional time-delay is added into

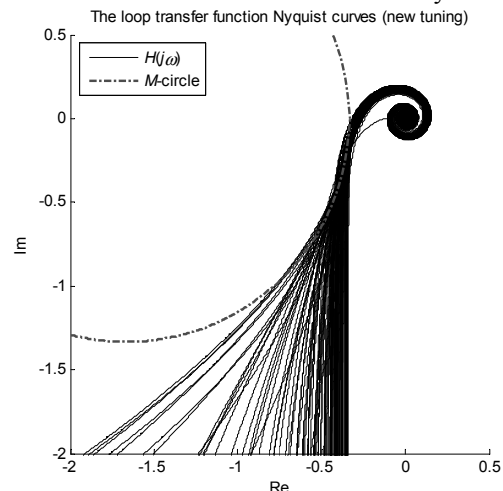


Fig. 7. The loop transfer function Nyquist curves.

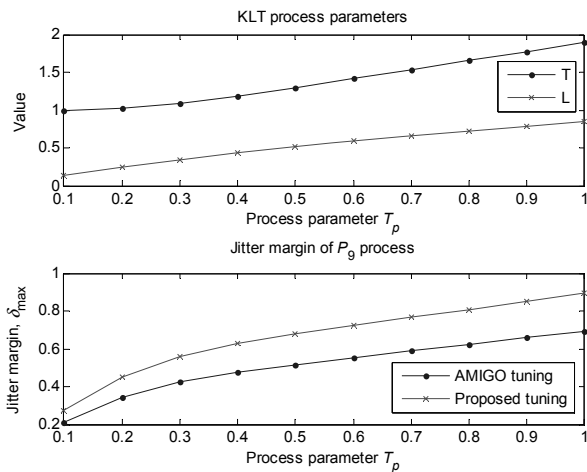


Fig. 8. KLT-process parameters (upper) and jitter margins (lower) with AMIGO tuning and the proposed tuning for P_9 .

the control loop, the responses of the AMIGO tuned processes go unstable if the delay amplitude is large enough. The control loops that are tuned with the proposed tuning are less sensitive to additional delay. Fig. 10 shows the step responses of process $P_9(T_p = 0.1)$, when the additional delay has a maximum value of 0.26 s. The figure shows how the AMIGO tuned process goes unstable while the proposed tuning rules still give stable response with little oscillation.

VI. CONCLUSIONS

This paper considered the problem of designing tuning rules for PID controllers in systems with varying time-delays. Inspired by the AMIGO tuning rules for the classical PID control set-up, we have exploited on the concept of jitter margin to develop tuning rules that are robust to varying time-delays. The optimal controller parameters were first solved by simultaneous performance and jitter margin

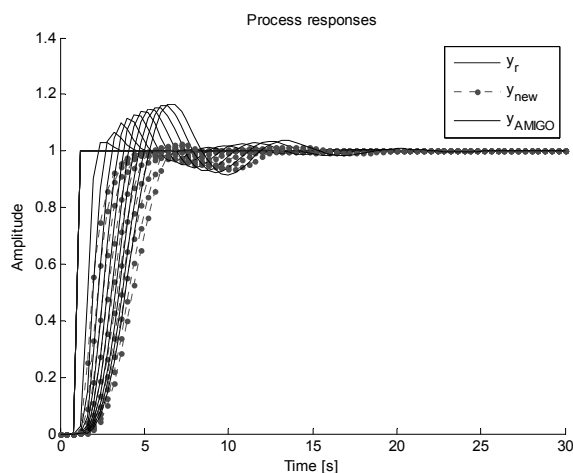


Fig. 9. P_9 step responses with AMIGO and proposed tuning.

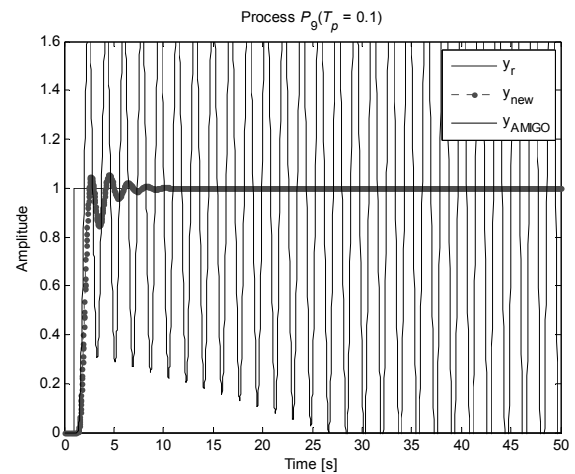


Fig. 10. Process $P_9(T_p = 0.1)$ step response when an additional random delay of maximum amplitude 0.26 s is added into the control loop.

maximization for a KLT-process, and based on the obtained surfaces the new tuning rules were identified. Numerical examples have demonstrated that it is possible to achieve significant improvements in jitter margin at the expense of only slight decrease in other performance criteria.

This paper deals only with continuous-time PID controller and plant model, but the methodology could be extended to the discrete-time case. For networked control systems, for example, the discrete-time controller case appears very relevant. Furthermore, tuning rules have only been developed for the parameters k , k_i and k_d . It may be useful to develop design rules also for the other controller parameters, in particular for the filtering time-constant T_f , which has a direct influence on the jitter margin.

REFERENCES

- [1] P. Naghshtabrizi and J. P. Hespanha, "Designing an observer-based controller for a networked control system," In *Proc. 44th Conf. on Decision and Control*, Dec. 2005.
- [2] L. Eriksson and T. Oksanen, "PID Controller Tuning for Integrating Processes: Analysis and New Design Approach," In *Proc. Fourth International Symposium on Mechatronics and its Applications*, Sharjah, UAE, Mar. 2007.
- [3] C.-Y. Kao and B. Lincoln, "Simple stability criteria for systems with time-varying delays," *Automatica*, vol. 40, pp. 1429-1434, Aug. 2004.
- [4] K. J. Åström and T. Hägglund, "Revisiting the Ziegler-Nichols step response method for PID control," *Journal of Process Control*, vol. 14, pp. 635-650, Sep. 2004.
- [5] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, pp. 759-768, 1942.
- [6] K. J. Åström, H. Panagopoulos, and T. Hägglund, "Design of PI controllers based on non-convex optimization," *Automatica*, vol. 34, pp. 585-601, May 1998.
- [7] G. P. Liu, J. B. Yang, and J. F. Whidborne, *Multiobjective optimisation and control*, Baldock: Research Studies Press Ltd., 2003, ch. 4.
- [8] O. J. M. Smith, "A controller to overcome dead time", *ISA Journal*, vol. 6, no. 2, pp. 28-33, 1959.