Publication [P8]

L. M. Eriksson, H. N. Koivo, “Comparison of low-complexity controllers in varying time-delay systems”, *SICE Journal of Control, Measurement, and System Integration*, Vol. 1, No. 2, 2008, pp. 111-119.

SICE Journal of Control, Measurement, and System Integration, Vol. 1, No. 2, pp. 111–119, March 2008

Comparison of Low-complexity Controllers in Varying Time-delay Systems

Lasse M. ERIKSSON * and Heikki N. KOIVO *

Abstract: Motivated by the recent developments in networked control systems and control over wireless, this paper presents a comparison of five control algorithms that are based on PID, IMC and fuzzy gain scheduling techniques and discusses their performance in varying time-delay systems. The low complexity of the proposed algorithms makes their use attractive in resource-constrained environments such as wireless sensor and actuator networks. The control system consists of a controller, a simple process and an output delay in the feedback loop. Three different delay models are considered in this framework; constant, random, and correlated random delay. In addition to presenting modifications to the control algorithms to better fit the varying time-delay systems a delay-robust tuning method is proposed, and the performance of various controllers is evaluated using simulation. The results show the benefits of adapting the controller parameters based on delay measurement if its amplitude is significant with respect to the time-constant of the process. Nevertheless, the PID algorithm used in the study also performs well in all scenarios, and this is achieved by its careful tuning.

Key Words: varying time-delay systems, networked control systems, PID control, IMC control, fuzzy control.

1. Introduction

Any practical control system suffers from delays. These can stem from process dynamics, actuators or sampling. The delays are often assumed either negligible or constant, but in some cases the variance in delay times (jitter) plays a significant role. If control is considered, a varying time-delay is a great challenge. Control of varying time-delay systems has been under extensive research during the last decade because of interest of using networks in control systems. In networked control systems (NCS) the varying delays occur due to e.g. medium access protocols with random backoff times and dynamic routing in the case of wireless networks. In addition, especially in wireless networks time synchronization of sensors is not guaranteed, and from the controller point of view the effects of asynchronism can be seen as varying sensing delays.

The control design of varying time-delay or networked control systems has been investigated e.g. in [1]–[5]. In general, the NCS research has focused on complicated control architectures that require significant memory and computational resources, which may be unavailable in wireless systems. The proposed designs include LQG controllers for NCS with delays less than sample time [4], delays longer than sample time [1], and packet-dropping links (e.g.[6]). Stability considerations of NCS are often formulated such that they can be solved with linear matrix inequalities (LMI) as in [7]. The simplicity of control design has rarely been addressed, although it is very relevant in wireless networked control systems.

A simple solution to varying time-delay systems is to use an event-based version of the standard proportional-integral-derivative (PID) controller [8], though the stability analysis of the closed-loop system becomes hard due to the time-variant controller. Another alternative is to rely on the time-based PID controller, but detune it with appropriate methods that provide robustness to the delays [9],[10]. In this case, the stability can be guaranteed by the use of simple stability criteria such as [11].

The PID controller is the most common controller in industrial applications today. It is likely to be the most useful controller also in wireless automation solutions. The aim of this study is to compare the performance of a standard PID controller in varying time-delay systems with four other low-complexity controllers based on PID, internal model control (IMC, [12]) and fuzzy gain scheduling (e.g.[13]) techniques. The idea is to investigate in which conditions the delay-adaptive controllers would improve the performance or robustness in varying time-delay systems. This is important since adapting to the delay requires online delay measurement, which is based on time synchronization. We will also propose a simple controller tuning method for varying time-delay systems and apply it in the comparison.

After describing the control system and delay models in Section 2, the control algorithms and tuning method are presented in Section 3. The simulation results are presented in Section 4 and conclusions are offered in Section 5.

2. Control System

2.1 System Model

The following scenario of a varying time-delay system is considered. A continuous-time process G is controlled with a discrete-time controller C that receives delayed measurements from the process output y. The system layout is presented in Fig. 1. This corresponds to a NCS setting, where a process variable is observed with (wireless) sensors that transmit their data to the controller over a network, and the controller is attached to the actuator. The transmission delays may vary due to several reasons. Because of the network, use of sampling and discrete-time controllers is well motivated. It is assumed that the process output is sampled at a constant rate, the sample time being

* Department of Automation and Systems Technology, Helsinki University of Technology, P.O.Box 5500, FI-02015 TKK, Finland
E-mail: {lasse.eriksson, heikki.koivo}@tkk.fi
(Received November 30, 2007)

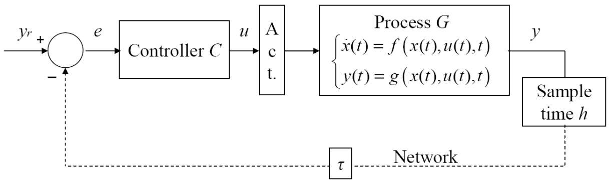

Fig. 1 The control system, where y_r is the reference signal, e is the error between the reference and the sampled and delayed process output y and u is the control signal.

h. The measurements are delayed by $\tau(t)|_{t=kh}$ before they are delivered from sensor(s) to the controller. The scenario would be somewhat different if the delay was a part of the process. Then the delayed output would be sampled, but in both cases the controller receives sampled and delayed measurements. For the process, a first-order lag plus delay (FOLPD) model is assumed, because often such a simple model is used to characterize a more complex system e.g. in the controller tuning phase (e.g. in [14]). The process model is

$$G(s) = \frac{g}{Ts+1}e^{-\tau s}, \tag{1}$$

where T is the process time-constant, τ the constant process delay and g the process gain. Although the process model is simple, it would be straightforward to apply the controller tuning method presented in Section 3.5 for more complicated process models.

For network delays, several delay models have been proposed in the literature. For example, in [3], IP network delays are modeled using a generalized exponential distribution

$$P(\tau) = \begin{cases} \frac{1}{\phi}e^{-\frac{\tau-\eta}{\phi}}, & \text{if } \tau \geq \eta \\ 0, & \text{if } \tau < \eta \end{cases}, \tag{2}$$

where η is the minimum (positive) delay and ϕ^2 is the variance. In [4], network delays are assumed to be either independent from sample to sample (random) or that their probability distribution functions are governed by an underlying Markov chain (state-dependent, but random). This study investigates properties of different control algorithms in varying time-delay systems. Two very general varying time-delay models are chosen for the comparison: 1) Gaussian-distributed random delay and 2) correlated random delay. A constant delay is used as a point of comparison. To be more precise, the delay models are

$$\begin{aligned} &\tau_1(t) = \tau_c \\ &\tau_2(t) = x_1(t),\ X_1 \sim N(\mu, \sigma) \\ &\tau_3(t) = \int_0^t e^{-q\alpha}x_2(t-\alpha)d\alpha,\ X_2 \sim U(a_{min}, a_{max}) \\ &0 \leq \tau_1(t), \tau_2(t), \tau_3(t) \leq \tau_{max}, \end{aligned} \tag{3}$$

where τ_c is a constant delay, x_1 is a Gaussian random variable $x_1 \in N(\mu, \sigma)$ with mean μ and variance σ^2, x_2 is a uniformly distributed random variable on $[a_{min}, a_{max}]$, and q is a filtering coefficient. The values of all delays are bounded between zero and a maximum value, τ_{max}. $\tau_3(t)$ describes a delay, which is partially random, but depends on its previous values. For example, Internet delays have been shown to be correlated due to queues in routers [15].

2.2 Performance Measures

In order to evaluate and compare the different control strategies, the performance criterion must be set. This and the next subsection discuss briefly the chosen criteria. The integral error functions are often used as measures of performance. The most common ones are ITAE (Integral of Time-weighted Absolute Error), IAE (Integral of Absolute Error), ISE (Integral of Square Error) and ITSE (Integral of Time-weighted Square Error). They are given, respectively, in (4) - (7) [16], where the error signal $e(t) = y_r(t) - y(t), t \geq 0$.

$$J_{ITAE} = \int_0^\infty t|e(t)|dt \tag{4}$$

$$J_{IAE} = \int_0^\infty |e(t)|dt \tag{5}$$

$$J_{ISE} = \int_0^\infty e^2(t)dt \tag{6}$$

$$J_{ITSE} = \int_0^\infty te^2(t)dt \tag{7}$$

The cost criteria are well known and it is easy to modify them to derive more suitable versions for evaluating the performance. It should be noted that the cost criteria above do not depend on the controller output, i.e., the control signal, at all. Thus a novel optimal control-like cost function is introduced that also applies for trajectory control. This function is presented in (8), and it will be referred as IERC (Integral of Weighted Sum of Square Error and Required Control Signal Error) criterion.

$$J_{IERC} = \int_0^\infty \left[w_1e^2(t) + w_2\big(y_r(t) - gu(t)\big)^2\right]dt \tag{8}$$

The weights w_1 and w_2, for which $w_1 + w_2 = 1$, define how much the error and the control signal use are considered in the criterion. If $w_1 > w_2$, the control signal is considered less than error and vice versa. The static gain g of the process scales the control and the reference signal to the same level. Thus the difference $y_r(t) - gu(t)$ equals zero, when the control signal is on the level that is required to have the process output on the reference signal level.

2.3 Robustness Measures

In the controller tuning phase robustness to varying time-delays may be addressed by evaluating the stability of the closed-loop system under delay uncertainty. It is possible to tune the controllers such that certain delay-based stability criteria are fulfilled. One such measure of robustness is the jitter margin [11]. The jitter margin is the maximum amplitude of any additional delay that the closed-loop system can tolerate and for which stability is still guaranteed. The delay type or variation is not constrained in any way in the criterion, which together with a relatively simple form makes it very useful in the control design.

The jitter margin is defined as follows. The continuous-time SISO system in Fig. 2, left, is stable for any time-varying delay defined by

$$\Delta(v) = v\big(t - \delta(t)\big), \quad 0 \leq \delta(t) \leq \delta_{max} \tag{9}$$

if

$$\left|\frac{G(j\omega)C(j\omega)}{1 + G(j\omega)C(j\omega)}\right| < \frac{1}{\delta_{max}\omega}, \quad \forall\omega \in [0, \infty[. \tag{10}$$

Here δ_{max} is the jitter margin, i.e., the maximum value of the delay $\delta(t)$, and Δ represents the delay uncertainty [11]. Note that

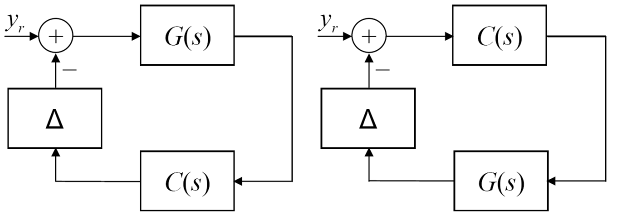

Fig. 2 Controller (C) and plant (G) with an uncertain time-varying delay (Δ) in the feedback loop. On the left, Δ is the controller output uncertainty. On the right, Δ is the process output uncertainty.

the plant itself may contain some constant delays, which should be included in $G(s)$ in order not to make the stability criterion very conservative. Hence Δ should only cover the varying part of the delay as indicated in (9). Since the proof of the criterion is based on the small gain theorem, the same criterion holds in continuous-time also if the plant and controller switch their positions, see Fig. 2, right. The latter setting corresponds to the case considered in this paper and in Fig. 1.

The jitter margin will be applied in Section 3.5 in determining suitable tuning parameters for the controllers. It should be mentioned here that the jitter margin criteria have been derived for pure continuous-time and pure discrete-time cases, but also a combination criterion (continuous-time plant and discrete-time controller) can be considered [11]. Nevertheless, in this study, only the continuous-time criterion is used, because then the controller sample time needs not be fixed in the tuning phase. This is, for sure, an approximation, but using small enough sample time in the controllers makes the error insignificant.

3. Control Methods

3.1 The Compared Control Algorithms

If the control design of varying time-delay systems is considered, it is important to know whether the delay is measurable online or not. If the delay at each measurement time instant is known, it is possible to compensate for the delay, i.e., to adapt the controller parameters on the basis of the delay value. If the exact delays are not known, there may be some knowledge of the delay distribution or the range where the delay varies. This can be used in the control design.

The control methods can also be classified depending on whether the controller is time-based or event-based. The time-based controller has a predefined constant control interval, but the event-based controller acts only upon receiving new information. Event-based controllers have been proposed especially for networked control systems [4],[8]. In NCS the sampling can be done at a constant rate, but because of the networks the measurement packets arrive asynchronously to the controller. Event-based controller acts immediately and only upon receiving new packets, whereas the time-based controller always uses the last received information and calculates the new control signal at each sample time.

The properties of five different control algorithms in varying time-delay systems will be compared. In the comparison, there are two variations of the PID algorithm and two variations of the IMC controller. The last controller is a combination of fuzzy gain scheduling and PID controllers. The basic properties of the compared control algorithms are shown in Table 1, where D-PID is a time-based discrete-time PID controller and D-PID$_{var}$ is an event-based version of D-PID. D-IMC is a time-based discrete-time IMC controller. The other IMC controller (D-IMC$_{var}$) is also time-based, but it uses the measured delay to adapt the controller parameters at each sample. In addition, the fuzzy gain scheduler determines the control signal on the basis of the delay measurement. The rest of this section discusses the different algorithms and controller tuning in detail.

Table 1 The properties of the control algorithms.

Controller name	Control interval	Delay	Control method
D-PID	Constant	Unmeasured	PID
D-IMC	Constant	Unmeasured	IMC
D-PID$_{var}$	Varying	Unmeasured[1]	PID
D-IMC$_{var}$	Constant	Measured[2]	IMC
Fuzzy	Constant	Measured[2]	Fuzzy + PID

[1] Timestamps required, [2] synchronization and timestamps required.

3.2 PID Control

The PID controller is widely used in industry. In the mid 1990's the PID controller was used in over 95 % of the control loops in process control [16]. The good properties of the controller can only be achieved if the controller is well tuned. The tuning of PID controllers has been discussed in numerous papers and books, but nearly always in systems with constant delays. Varying delays have not been addressed very often. Some results exist, though, e.g. in [17] the tuning of the continuous-time PID controller in state-dependent delay systems is discussed. A discrete-time PID controller tuning method that optimizes the closed-loop performance and improves robustness to varying delays is presented in [18]. PID tuning rules for varying time-delay systems have been proposed in [9] and [10].

Generally, the continuous-time PID controller algorithm is given in time domain as

$$u(t) = K\Big(e(t) + \frac{1}{T_i}\int_0^t e(\alpha)d\alpha + T_d\frac{de(t)}{dt}\Big) \quad (11)$$

with tuning parameters K (gain), T_i (integration time) and T_d (derivative time) [19]. If measurement noise is present in the control system, the controller needs to filter the measurements. Often the filter only acts on the derivative part resulting in the following approximation in Laplace domain:

$$D(s) = KT_d sE(s) \approx \frac{K_d s}{1 + K_d s/N}E(s). \quad (12)$$

By varying the value of the filter constant N the high-frequency gain of the derivative part can be limited, which decreases the effect of the measurement noise. N is typically chosen from the range of 3 to 20 [19]. However, if varying time-delays are considered, first-order filtering may result in small jitter margins as indicated in [9], and second-order filters may become more useful. Hence we choose to use the following PID algorithm

$$u(t) = K_p\Big(by_r(t) - y_f(t)\Big) + K_i\int_0^t \Big(y_r(\alpha) - y_f(\alpha)\Big)d\alpha + K_d\Big(c\frac{dy_r(t)}{dt} - \frac{dy_f(t)}{dt}\Big), \quad (13)$$

where the controller parameters K_p, K_i and K_d are related to those in (11) by

$$K_p = K, \quad K_i = \frac{K}{T_i}, \quad K_d = KT_d. \quad (14)$$

Here b and c are the set-point weights and $y_f(t)$ is the filtered process variable for which

$$Y_f(s) = G_f(s)Y(s) = \frac{1}{(T_f s + 1)^2} Y(s). \tag{15}$$

T_f is the filter time-constant.

To be used in NCS, discrete-time approximations of the controllers are needed. The proportional part of the PID controller is static and requires no approximation, only sampling. The backward difference method can be used in the approximation of the integral and derivative parts. The discrete-time PID controller algorithm approximating (13) is given in (16) - (19). In addition, the discrete-time approximation of the filter (15) is given in (20).

$$u(k) = p(k) + i(k) + d(k) \tag{16}$$

$$p(k) = K_p\Big(by_r(k) - y_f(k)\Big) \tag{17}$$

$$i(k) = i(k-1) + K_i h\Big(y_r(k) - y_f(k)\Big) \tag{18}$$

$$d(k) = \frac{K_d}{h}\Big(cy_r(k) - y_f(k) - cy_r(k-1) + y_f(k-1)\Big) \tag{19}$$

$$y_f(k) = \frac{h^2 y(k) - T_f^2 y_f(k-2)}{\left(h + T_f\right)^2} + \frac{2T_f y_f(k-1)}{h + T_f} \tag{20}$$

The time-based (D-PID) and event-based (D-PID$_{var}$) PID controllers are both implemented on the basis of the same equations (16) - (20), but in the latter case the sample time h is varying. The sample time is determined from the timestamp difference of the two most recent measurements when new measurement arrives at the controller. The new control signal is applied immediately on the process.

Figure 3 shows how the measurements are delivered from sensors to the controller over the network. Sampling is done at a constant rate, but because of the varying delay between the sensor and controller the controller receives the packets irregularly. The time-delay of a measurement made at time k is denoted by $\tau(k)$.

The D-PID$_{var}$ controller calculates the new control signal value only when new information (new measurement) becomes available. Otherwise the controller keeps the last value of the control signal and applies that to the process input. The controller rejects old measurements if these are received. Consider Fig. 3 and the time between $k+1$ and $k+2$. The measurement made at k is delivered to the controller at time $k + \tau(k) > k + 1 + \tau(k+1)$. The measurement transmitted at k is rejected by the controller, since a newer measurement (made at $k+1$) reaches the controller before the one made at k. In order to do this the measurement packets need to provide information on when the measurement was made, i.e., timestamps are needed.

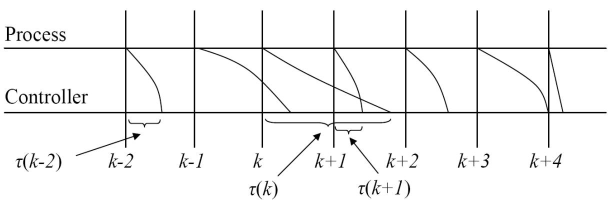

Fig. 3 Measurements in varying time-delay channel.

3.3 Internal Model Control

The IMC [12] method can be used if the process model is known. Generally, the method is applicable only if the delays of the system are constant. In this study, the IMC method is also applied for varying time-delay systems.

Figure 4 presents the IMC principle. In the diagram, the process G is controlled with an IMC controller Q, $\tilde{G}$ is the process model, and disturbances are denoted with d. The model output error $y - \tilde{y}$ is subtracted from the reference signal and fed into the IMC controller which calculates the control signal.

The controller Q is calculated so that the process model is first divided into two parts

$$\tilde{G} = \tilde{G}_+ \tilde{G}_-, \tag{21}$$

where $\tilde{G}_+$ includes all unstable zeros and delays of $\tilde{G}$. The rest of the model is included in $\tilde{G}_-$. The controller becomes

$$Q = \tilde{G}_-^{-1} f, \tag{22}$$

where the low-pass filter f is

$$f(s) = \frac{1}{\left(\lambda s + 1\right)^n}. \tag{23}$$

The low-pass filter is required in order to have a causal controller, and $\lambda > 0$ is the tuning parameter of the IMC method. The value of λ has a significant effect on the performance and robustness of the controlled system. There is a trade-off; a very fast and simultaneously very robust tuning is generally difficult to achieve. Especially in varying time-delay systems, where robustness to delay plays an important role, tuning of λ turns out to be crucial.

When implementing the IMC controller, it is useful to recognize the dependency between the IMC controller (Q in Fig. 4) and the controller in the classical feedback loop (C in Fig. 1), which is

$$C = \frac{Q}{1 - \tilde{G}Q}. \tag{24}$$

If the controller (24) is used, the process delay must be approximated with a linear transfer function in order to be able to calculate the controller. For instance, the first order Padé approximation is applicable

$$e^{-\tau s} \approx \frac{-\frac{\tau}{2}s + 1}{\frac{\tau}{2}s + 1}. \tag{25}$$

If the delay is known, as can be assumed if it is constant, the use of the above approximation is straightforward. If the delay varies, there are two possibilities: 1) if the exact delay of each measurement is known, one can re-compute the controller at

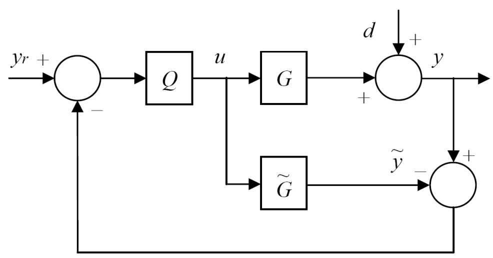

Fig. 4 The IMC structure (modified from [12]).

each sample time in the case of discrete-time controller, or 2) if the delay distribution is known rather than exact delay times, the expectation value of the delay can be used in (25).

Using (21) - (23) and the process model (1) with $g = 1$, the IMC controller becomes

$$Q(s) = \tilde{G}_-^{-1}(s)f(s) = \frac{Ts+1}{\left(\lambda s+1\right)^n}. \tag{26}$$

Using (24), the IMC controller can be given in the form of the classical feedback loop as in Fig. 1, which results in

$$C(s) = \frac{Ts+1}{\left(\lambda s+1\right)^n - e^{-\tau s}}. \tag{27}$$

The controller (27) is not realizable, and hence the delay needs to be approximated. Using (25) for the delay and $n = 1$ for the filter order, we obtain

$$C(s) = \frac{T\tau s^2 + (\tau+2T)s + 2}{\tau\lambda s^2 + 2(\tau+\lambda)s}. \tag{28}$$

This is a continuous-time IMC controller. Its discrete-time equivalent, D-IMC, is derived using the Tustin approximation. This results in the following control law:

$$\begin{aligned} u(k) &= \frac{2\lambda\tau}{(h+\lambda)\tau+h\lambda}u(k-1) + \frac{(h-\lambda)\tau+h\lambda}{(h+\lambda)\tau+h\lambda}u(k-2) \\ &+\frac{(T+h/2)\tau+hT+h^2/2}{(h+\lambda)\tau+h\lambda}e(k) + \frac{h^2-2T\tau}{(h+\lambda)\tau+h\lambda}e(k-1) \\ &+\frac{(T-h/2)\tau-hT+h^2/2}{(h+\lambda)\tau+h\lambda}e(k-2). \end{aligned} \tag{29}$$

The D-IMC and D-IMC$_{var}$ controllers, used in the comparison, calculate the control signals according to (29). The difference is that D-IMC has constant parameters and it uses the expectation value of the delay as the parameter τ while D-IMC$_{var}$ updates the delay parameter τ at each sample based on the delay measurement.

3.4 Fuzzy Gain Scheduling

Gain scheduling can be used to adapt the controller gain(s) on the basis of e.g. the process state or other scheduling variable(s). In gain scheduling the controller parameters are changed during control in a predefined way. The technique falls into the category of adaptive control, and it can enlarge the operation area of linear controllers into nonlinear systems [13]. Fuzzy logic can be used for deciding the controller gain(s) based on measurements.

Fuzzy gain scheduling is used in this study for calculating the weighting of five PID controllers on the basis of the delay value. The PID controllers are tuned for different constant values of the delay. The delay is measured online, and fuzzy logic is then used for giving weights (between 0 and 1) on the outputs of each controller. The total control signal is actually a weighted sum of the outputs of the PID controllers.

All five input membership functions (MFs) of the fuzzy gain scheduler are Gaussian. It is assumed that the delay has some minimum (zero) and maximum values (τ_{max}). The MFs are distributed evenly in the delay range. One of the MFs has its maximum value at zero and another MF has its maximum at τ_{max}. The other MFs lie between these two. The Sugeno method of fuzzy inference is applied in the scheduler and the output MFs are constant (on/off-type). The rule base is simple: if input belongs to the range of MF 1, then output 1 is on and others are off. There are in total five similar rules, one for each input-output pair.

Each PID controller in the fuzzy gain scheduler calculates the control signal at every sample time. The controllers are implemented otherwise as in (16) - (20), but the output of the controllers is slightly adjusted. If u_i^{out} is the output of the i^{th} PID controller and $w_{f,i}$ the corresponding weight calculated by fuzzy logic, the output of the total controller u^{total} is calculated as

$$u^{total}(k) = \sum_{i=1}^{5} w_{f,i}(k)u_i^{out}(k), \tag{30}$$

$$\begin{aligned} u_i^{out}(k) &= u_i(k) - u_i(k-1) + u^{total}(k-1) \\ &= \Delta u_i(k) + u^{total}(k-1), \end{aligned} \tag{31}$$

where u_i is the control signal calculated according to the PID algorithm in i^{th} controller. Each PID controller calculates how much the control signal should change from the value they calculated at the previous sample time, and that increment is added to the value of the control signal that was previously applied to the process. The total control signal is a weighted sum of the control signals calculated by PIDs, and the weights are determined by the fuzzy gain scheduler.

3.5 Tuning of the Controllers

This section discusses the controller tuning in detail and proposes a tuning method for varying time-delay systems. In the tuning phase we aim at providing the controllers adequate jitter margins so that the closed-loop system is maintained stable for all delays, constant or time-varying, of allowed amplitudes. We will consider the first-order process controlled with the proposed low-complexity controllers discussed in the previous sections. The performance comparison is done for the FOLPD process in (1), whose time-constants vary from 0.2 to 6. The delay is bounded into the range $0 \le |\tau(t)| \le 2$ so that the study considers both delay dominant ($\tau > T$), lag-dominated ($\tau < T$) and balanced lag and delay ($\tau \approx T$) processes.

The tuning procedure follows the guidelines presented in [10], where PID tuning for varying time-delay systems is considered. Whereas in [10] the jitter margin of the control loops is maximized while also maximizing the closed-loop performance, here the desired jitter margin is known in advance, and it is treated as a hard constraint in the controller performance optimization. All controllers are tuned using constrained optimization and simulation techniques. The controller tuning problem is posed as

$$\begin{aligned} &Min \quad f(x) = \int_0^{\infty} \left[w_1 e^2(t,x) + w_2\left(y_r(t) - gu(t,x)\right)^2\right]dt \\ &s.t. \quad x > 0 \\ &\qquad\;\; \delta_{max}(x) > \delta_0, \end{aligned} \tag{32}$$

where the cost function $f(x)$ is the IERC performance criterion (8), x is the vector of the controller parameters (K_p, K_i and K_d in PID case and λ in IMC case), and δ_0 is the desired jitter margin.

The PID controller parameters K_p, K_i and K_d and IMC controller parameter λ were found by solving the optimization problem (32) for a unit step reference with weights $w_1 = w_2 = 0.5$. The controllers were tuned for the process model (1) with static gain $g = 1$, which was also used in the comparison. During the tuning procedure of PID and IMC controllers, the delay was kept constant at $\tau = 1$ s, which is the expectation value of each of the delay functions used in the study. The same method was also applied in the fuzzy controller's tuning with certain modifications that are discussed later in this section. Since the process delay was assumed to be $\tau = 1$ s and the maximum delay is $\max(\tau(t)) = 2$ s, the required jitter margin is $\delta_0 = 1$ s.

The measurement filter time-constant T_f of the PID controllers was set to 0.1τ. Note that the choice $T_f = 0.1\tau$ obeys the AMIGO tuning rules [14] that have recently been proposed for process control. The choice of T_f has a significant effect on the jitter margin as shown in [9], but since the AMIGO tuning provides jitter margins $\delta_{max,AMIGO} \approx 0.7\tau$ (see [9]), it is likely that by detuning K_p, K_i and K_d the desired jitter margin $\delta_0 = \tau = 1$ s could be achieved without changing the value of T_f. The reason for not optimizing T_f is that this parameter often depends on the level of measurement noise and it may be set according to different objectives. By limiting the number of variables to three instead of four also makes the optimization procedure less time consuming. Besides, the tuning rule $T_f = 0.1\tau$ is already justified based on the investigation in [14]. For all the controllers sample time $h = 0.05$ s is used, which is still acceptable even for the smallest $T = 0.2$. The relatively large sample time is also justified from the networked control systems' point of view. Faster sampling increases traffic in the network and causes longer delays. In addition, in wireless sensor networks, $h = 0.05$ s is already a relatively short sample time.

The parameter optimization was applied on the D-PID and D-IMC controllers. The D-PID$_{var}$ and D-IMC$_{var}$ controllers did not need to be tuned separately, because the delay was kept constant while tuning. With constant delay these controllers are equal to D-PID and D-IMC, respectively, and the same controller parameters are applicable. On the other hand, the optimal tuning of these delay-based controllers would actually depend on the delay realization, and thus optimization may not be a suitable method in their tuning.

The five PID controllers of the fuzzy gain scheduler were tuned for different values of constant delay and jitter margins. The idea was to tune each PID controller for certain minimum delay, which roughly corresponds to the membership function's minimum value, and for a desired jitter margin which is determined by the width of the membership function. These settings for the optimization problem are presented in Table 2.

The PID controller tuning parameters K_p, K_i and K_d obtained by solving the optimization problem (32) for process time-constant values $T \in [0.2\ 6]$ are shown in Fig. 5 (K_p), Fig. 6 (K_i) and Fig. 7 (K_d).

Solving (32) in the case of IMC controller (29) showed that

Table 2 PID controller settings for the fuzzy gain scheduler.

Controller	Minimum delay (s)	Jitter margin (s)
FuzzyPID1	0	0.3
FuzzyPID2	0.2	0.6
FuzzyPID3	0.7	0.6
FuzzyPID4	1.3	0.6
FuzzyPID5	1.7	0.3

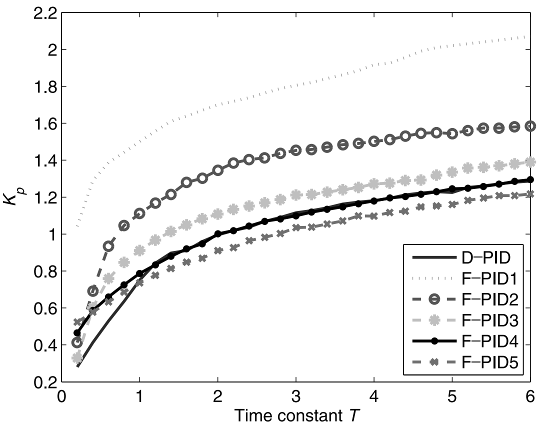

Fig. 5 PID controller gain K_p vs. process time-constant.

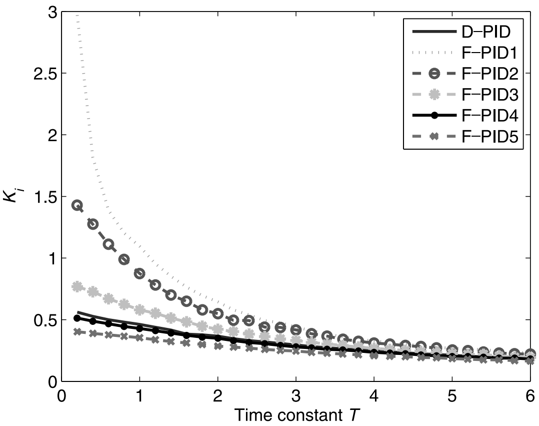

Fig. 6 PID controller gain K_i vs. process time-constant.

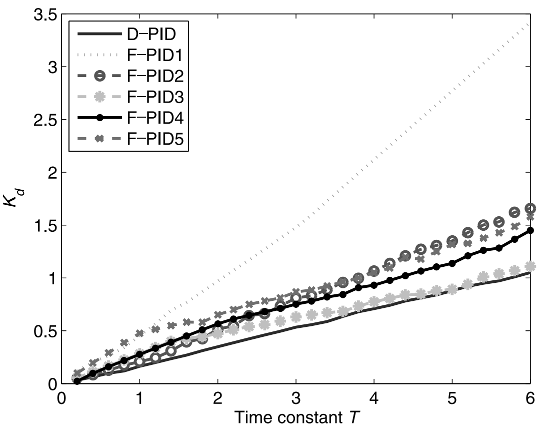

Fig. 7 PID controller gain K_d vs. process time-constant.

the optimal tuning parameter λ^* is linearly proportional to the process time-constant, but it was also found to be bounded by the jitter margin as

$$\lambda^* = \begin{cases} \frac{T}{\sqrt{2}}, & T \geq \sqrt{2} \\ \delta_0 + \epsilon \approx 1, & T < \sqrt{2} \end{cases} \tag{33}$$

where $\epsilon > 0$ is a small number. These results can be shown to hold in the continuous-time case, although the tuning rules were derived by optimization and simulation using a discrete-time controller. Obviously, the latter tuning rule for λ in (33) comes from the constraint regarding the jitter margin. This can be verified by analyzing the jitter margin of an IMC-controlled closed-loop system. To simplify the analysis, consider the process model (1) and the IMC controller (27), i.e., an ideal IMC controller without delay approximation. With $n = 1$ the closed-loop system becomes

$$G_{cl} = \frac{G(s)C(s)}{1 + G(s)C(s)} = \frac{e^{-\tau s}}{\lambda s + 1} \tag{34}$$

which gives a jitter margin

$$\delta_{max} < \left| \frac{1 + G(j\omega)C(j\omega)}{j\omega G(j\omega)C(j\omega)} \right| = \sqrt{\lambda^2 + \frac{1}{\omega^2}}. \tag{35}$$

It is clearly seen that the lower bound of the jitter margin is obtained when $\omega \to \infty$, which gives the tuning rule $\lambda^* > \delta_{max}$ or $\lambda^* = \delta_0 + \epsilon$.

The upper condition in (33) is obtained by minimizing the IERC cost function with respect to λ. First, the output and control signals need to be calculated. Since the reference signal used is a unit step, the output and control signals are given in Laplace domain as

$$Y(s) = \frac{e^{-\tau s}}{s(\lambda s + 1)}, \quad U(s) = \frac{(Ts + 1)e^{-\tau s}}{s(\lambda s + 1)}. \tag{36}$$

Based on the inverse Laplace transforms of (36), the IERC cost criterion becomes

$$\begin{aligned} J_{IERC} &= w_1\Big(\int_0^\tau 1dt + \int_\tau^\infty e^{-\frac{2}{\lambda}(t-\tau)}dt\Big) \\ &+ w_2\Big[\int_0^\tau 1dt + \int_\tau^\infty \Big(1 - \frac{T}{\lambda}\Big)^2 e^{-\frac{2}{\lambda}(t-\tau)}dt\Big]. \end{aligned} \tag{37}$$

Assuming that both weights are equal, i.e., $w = w_1 = w_2$, the cost function becomes

$$J_{IERC} = w\Big(2\tau + \lambda - T + \frac{T^2}{2\lambda}\Big). \tag{38}$$

The optimal value of λ is obtained by minimizing (38) with respect to λ, which results in the tuning rule in (33), i.e. $\lambda^* = T/\sqrt{2}$. Because of the constraint imposed by the jitter margin requirement, λ always needs to be greater than the desired jitter margin, and hence we obtain the boundaries for the two conditions.

4. Simulation Results

All the controllers described in Section 3, the process (1) and the delay models (3) were implemented in Simulink to test the performance of the controllers in varying time-delay systems. In the simulations, the reference signal consisted of 200 unit steps from zero to one or from one to zero each lasting 30 s. The total simulation time was 6000 s for each process time-constant and controller combination. During the simulations the delays varied between 0 and 2 seconds. The simulations were run with Gaussian random delay ($\tau_2(t)$ in (3)) with delay parameters $\mu = 1$, $\sigma^2 = 0.3$ and $\tau_{max} = 2$, and with correlated random delay ($\tau_3(t)$ in (3)) with parameters $q = 0.1$, $a_{min} = -1.3$, $a_{max} = 1.5$ and $\tau_{max} = 2$ resulting in a delay with $E\{\tau_3(t)\} = 1$.

The simulation was first run with a constant time-delay of 1 s to verify that the controllers produce similar results in the nominal case. From the results it could be seen that both PID controllers act exactly in the same way since the algorithms are the same for constant delay. Additionally, both IMC algorithms produce identical results, but the PID, IMC and fuzzy gain scheduler results differ a little. The differences of the controllers in the nominal case are depicted in Figs. 8 and 9, where relative IERC and ISE cost functions are shown, respectively. The cost function values are scaled such that for each time-constant the controller with the lowest performance (i.e. maximum cost) equals one in the graphs, and the other controllers' cost criteria are divided by the maximum cost giving values less than one.

It is seen in Figs. 8 and 9 that the fuzzy gain scheduler gives the largest IERC values for almost all time-constants, but on the other hand, it gives simultaneously clearly the best ISE performance. The good ISE performance is expected, since if the delay is constantly one, the control signal at the fuzzy gain scheduler is almost entirely based on one PID controller (FuzzyPID3 in Table 2). This controller is tuned for a much narrower time-delay range than the other compared control algorithms, so it is expected to be more aggressive than the others. Hence the ISE performance is better, but good tracking performance naturally comes with greater use of control signal. Thus the IERC cost increases as the control signal is also considered in the criterion. Although not shown here, also the relative IAE, ITAE and ITSE criteria were calculated. For the constant delay case IAE resembles the relative ISE graph with some minor differences. The relative IAE and ITAE graphs are exactly the same, and the same applies for ISE and ITSE, because the time-weighting in the criteria is cancelled when the relative values are calculated.

Figures 10 and 11 show the relative IERC and ISE cost function values for the Gaussian random delay case. Also here the

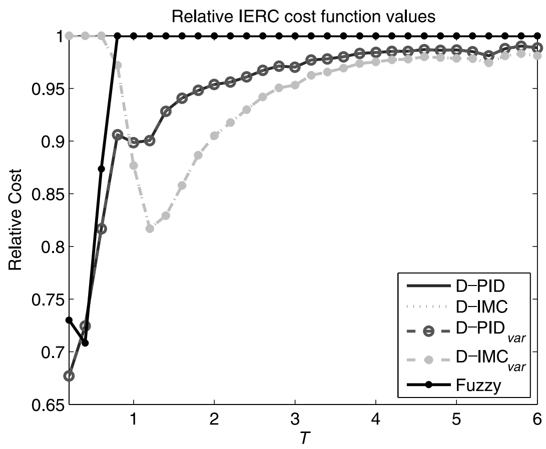

Fig. 8 Relative IERC cost function values in the constant delay case with $\tau_1(t) = 1$ s.

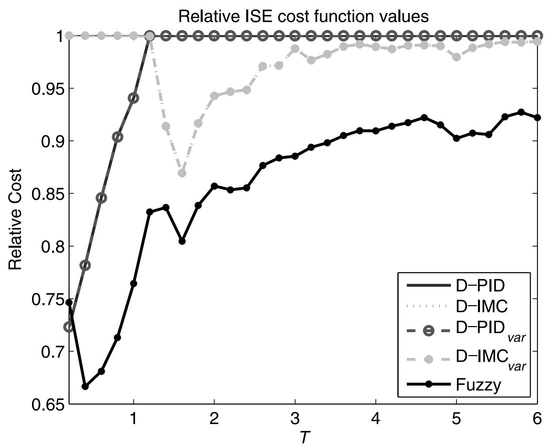

Fig. 9 Relative ISE cost function values in the constant delay case with $\tau_1(t) = 1$ s.

fuzzy gain scheduler performs the worst for most processes, if the IERC criterion is considered. The reasons for this behavior are similar as in the constant delay case. Now all the five PID controllers affect the control signal, and they are all tuned for a much narrower delay band than the other control algorithms. Hence the gains of the controllers are relatively larger, which results in greater use of the control signal. As the delay is random, the switching between the controllers is fast, and this might lead to a noisy control signal and thus the IERC criterion becomes large. The tracking performance (ISE) is comparable with other controllers for $T < 3$ s, but for larger time-constants also this performance measure is modest.

Of the other controllers, the regular time-based PID controller D-PID seems to perform well with respect to both criteria, whereas its event-based version clearly looses in performance. It can be seen that the simplest controllers (D-PID and D-IMC) generally perform best with respect to the IERC criterion, for which the controllers were optimized in the tuning phase. Also the ISE criterion suggests that the standard PID and IMC controllers perform roughly as well as the ones with delay-adaptability, although some gains are achieved especially with the delay-adaptive IMC controller (D-IMC$_{var}$).

The results are quite different for the correlated random delay case. The performance criteria are shown in Figs. 12 and 13. Here the benefits of using slightly more complicated algorithms are seen. As the delay variation is slower than in the purely random delay case, the controllers can better adapt to the delay and this has a positive effect on the performance. In IERC-sense, the delay-adaptive IMC controller is the best for most of the time-constants, whereas the fuzzy gain scheduler gives the best ISE-values. It is also seen that for larger time-constants the advantages gained by adapting to the delay decrease as the performance of controllers converges to roughly the same values. This is expected as the delay amplitude to time-constant ratio becomes small, and the variance of delay becomes less significant.

5. Conclusions

This paper discussed the problem of controlling varying time-delay systems with low-complexity controllers motivated by the increasing interest in networked control, especially control over wireless. Altogether five different controllers were proposed to solve the control problem. The discrete-time PID, IMC and fuzzy gain scheduling controllers were formulated for varying time-delay systems and guidelines were given regarding delay-robust tuning of the controllers. The performance of the controllers was compared with a range of processes with different time-constants. One constant and two general varying time-delay models were used in the comparison.

The simulation results suggest that in the purely random delay case the delay-adaptive controllers might not guarantee better performance for the system, and the conventional PID

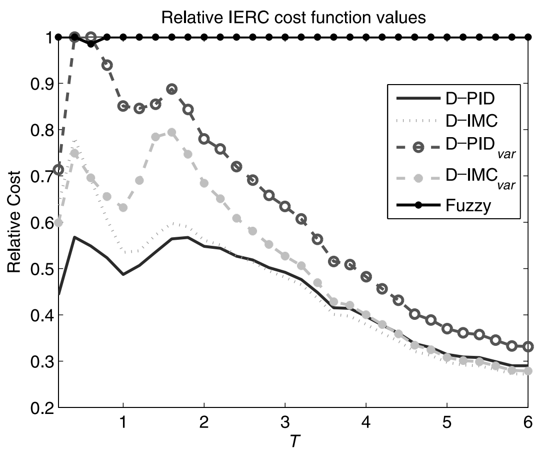

Fig. 10 Relative IERC cost function values for the Gaussian random delay $\tau_2(t)$.

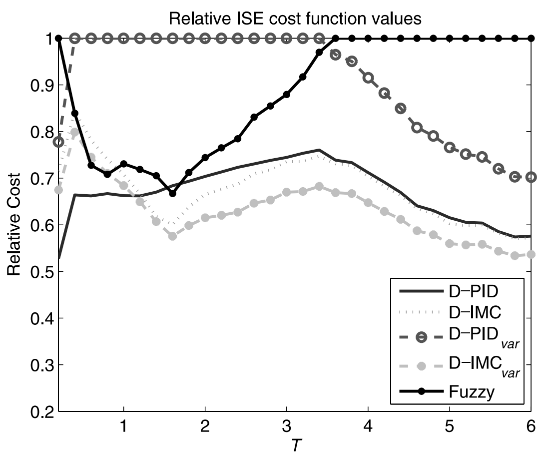

Fig. 11 Relative ISE cost function values for the Gaussian random delay $\tau_2(t)$.

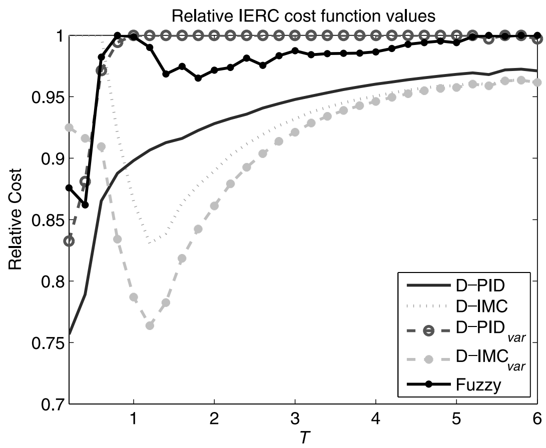

Fig. 12 Relative IERC cost function values for the correlated random delay $\tau_3(t)$.

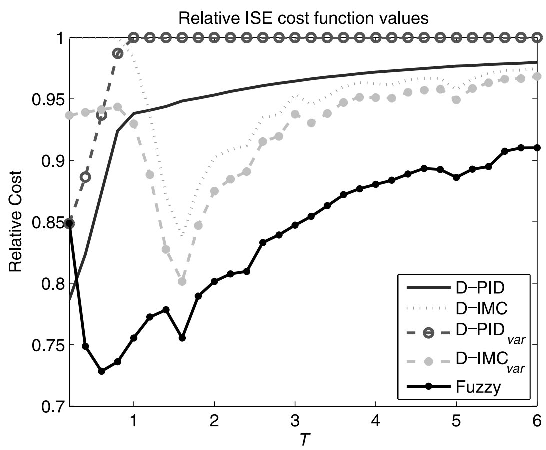

Fig. 13 Relative ISE cost function values for the correlated random delay $\tau_3(t)$.

or IMC controllers are feasible. But if the delay is correlated, which is a relevant case in NCS, improvements in performance can be achieved with slightly more complicated controllers such as the delay-adaptive IMC controller and delay-based fuzzy gain scheduling. The ratio between the delay amplitude and the process time-constant affects how much is gained in performance when using the advanced controllers. If the delay and the process time-constant are of the same size, great improvements can be achieved with the delay-adaptive controllers. If the delay amplitude is insignificantly small with respect to the time-constant, the controllers perform in a similar way and the advantage of taking the varying delay into account in the control algorithm diminishes. The delay-adaptive control algorithms require measuring of the delay, which might sometimes be problematic. Often, the delay can at least be estimated online. Only linear process models were considered in the paper, but most likely fuzzy gain scheduling would perform best for nonlinear processes.

The comparison shows that the varying control interval is not a sufficient modification to the PID controller in varying time-delay systems. Despite of the fact that the controller can act immediately upon receiving new information, it is still tuned for certain constant delay (here 1 s). Comparing the measurements' time difference of arrival at the controller only gives the relative delay between the two measurements and not the total delay, which has passed since the measurement was taken. Varying the control interval and using otherwise exactly the same control algorithm does not lead to gains in performance. In addition, such a modification makes the analysis of the controller more difficult than that of the regular PID, and there is no easy way to guarantee the stability of such time-variant controller. On the contrary, delay-robust tuning of a time-based PID controller may be easily solved using the proposed method, which results in good performance in varying time-delay systems.

The effect of the controller sample time was not considered in this study. In the tuning phase it could be omitted by using the continuous-time stability criterion for delay-robustness and a small sample time to make the approximations hold. It is, however, acknowledged that the sample time and the varying time-delay are interconnected and their ratio affects the control performance, but the deeper investigation of this is left for future work.

References

[1] B. Lincoln: *Dynamic Programming and Time-Varying Delay Systems*, Ph.D. thesis, Lund Inst. of Tech., Sweden, 2003.

[2] N.J. Ploplys, P.A. Kawka, and A.G. Alleyne: Closed-Loop Control over Wireless Networks, *IEEE Control Systems Magazine*, Vol. 24, pp. 58–71, 2004.

[3] Y. Tipsuwan and M.-Y. Chow: On the Gain-Scheduling for Networked PI Controller Over IP Network, *IEEE/ASME Transactions on Mechatronics*, Vol. 9, pp. 491–498, 2004.

[4] J. Nilsson: *Real-Time Control Systems with Delays*, Ph.D. thesis, Lund Inst. of Tech., Sweden, 1998.

[5] L. Eriksson and H.N. Koivo: Comparison of PID, IMC and Fuzzy Gain Scheduling Controllers in Varying Time-Delay Systems, *Proc. 5th Vienna Symposium on Mathematical Modelling*, Vienna, Austria, Feb. 8–10, 2006.

[6] O.C. Imer, S. Yuksel, and T. Basar: Optimal Control of LTI Systems over Unreliable Communication Links, *Automatica*, Vol. 42, pp. 1429–1439, 2006.

[7] X. Nian: Stability of Linear Systems with Time-Varying Delays: An Lyapunov Functional Approach, *Proc. American Control Conference*, Denver, USA, Jun. 4–6, 2003.

[8] K.-E. Årzén: A Simple Event-Based PID Controller, *Preprints 14th World Congress of IFAC*, Beijing, China, Jul. 5–9, 1999.

[9] L.M. Eriksson and M. Johansson: Simple PID Tuning Rules for Varying Time-Delay Systems, *Proc. The 46th IEEE Conference on Decision and Control*, New Orleans, USA, Dec. 12–14, 2007.

[10] L.M. Eriksson and M. Johansson: PID Controller Tuning Rules for Varying Time-Delay Systems, *Proc. 2007 American Control Conference*, New York, USA, Jul. 11–13, 2007.

[11] C.-Y. Kao and B. Lincoln: Simple stability criteria for systems with time-varying delays, *Automatica*, Vol. 40, pp. 1429–1434, 2004.

[12] C.E. Garcia and M. Morari: Internal Model Control. 1. A Unifying Review and Some New Results, *Industrial & Engineering Chemistry Process Design and Development*, Vol. 21, pp. 308–323, 1982.

[13] P. Viljamaa: *Fuzzy Gain Scheduling and Tuning of Multivariable Fuzzy Control – Methods of Fuzzy Computing in Control Systems*, Doctoral thesis, Tampere Univ. of Tech., Finland, 2000.

[14] 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, 2004.

[15] J.C. Bolot: Characterizing End-to-End Packet Delay and Loss in the Internet, *Journal of High-Speed Networks*, Vol. 2, pp. 289–298, 1993.

[16] K.J. Åström and T. Hägglund: *PID Controllers: Theory, Design, and Tuning*, 2nd ed., Instrument Society of America, Research Triangle Park, NC, 1995.

[17] H.N. Koivo and A. Reijonen: Tuning of PID Controllers for Varying Time-Delay Systems, *Proc. IEEE International Conference on Mechatronics 2004*, Istanbul, Turkey, Jun. 3–5, 2004.

[18] L. Eriksson and H.N. Koivo: Tuning of Discrete-Time PID Controllers in Sensor Network based Control Systems, *Proc. 6th IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Espoo, Finland, Jun. 27–30, 2005.

[19] K.J. Åström and B. Wittenmark: *Computer-controlled Systems Theory and Design*, 3rd ed., Prentice Hall, Englewood Cliffs, NJ, 1997.

Lasse M. Eriksson

Received his M.S. and Lic.Sc. degrees from Helsinki University of Technology (TKK), Finland, in 2003 and 2007, respectively. In 2001, he joined the Control Engineering Laboratory at TKK, and he is currently a researcher at the Department of Automation and Systems Technology. His research interests include control design, tuning and implementation of (wireless) networked control systems. He is a graduate student member of IEEE.

Heikki N. Koivo

Received his BSEE degree from Purdue University, Indiana, the MS degree in Electrical Engineering and the Ph.D. degree in Control Sciences from University of Minnesota. He is a Professor of Control Engineering at Helsinki University of Technology. Before joining TKK in 1995, he served in various professorial positions at the University of Toronto and at Tampere University of Technology, Finland. His research interests include study of complex systems, adaptive and learning control, mechatronics, microsystems and wireless communication systems. He is a member of the Editorial Board of Journal of Intelligent and Fuzzy Systems, Intelligent Automation and Soft Computing, and Journal of Systems and Control Engineering. He is a Fellow of the Finnish Academy of Technology.