ISA Transactions 51 (2012) 30–41

Contents lists available at SciVerse ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Simple robust autotuning rules for 2-DoF PI controllers

R. Vilanova [a,*], V.M. Alfaro [b], O. Arrieta [a,b]

[a] Departament de Telecomunicació i d'Enginyeria de Sistemes, Escola d'Enginyeria, Universitat Autònoma de Barcelona, 08193 Bellaterra, Barcelona, Spain

[b] Departamento de Automática, Escuela de Ingeniería Eléctrica, Universidad de Costa Rica, 11501-2060 San José, Costa Rica

ARTICLE INFO

Article history:
Received 27 October 2009
Received in revised form
21 May 2011
Accepted 7 September 2011
Available online 6 October 2011

Keywords:
PI control
Two-degrees-of-freedom
Robust control

ABSTRACT

This paper addresses the problem of providing simple tuning rules for a Two-Degree-of-Freedom (2-DoF) PI controller (PI_2) with robustness considerations. The introduction of robustness as a matter of primary concern is by now well established among the control community. Among the different ways of introducing a robustness constraint into the design stage, the purpose of this paper is to use the maximum sensitivity value as the design parameter. In order to deal with the well known performance/robustness tradeoff, an analysis is conducted first that allows the determination of the lowest closed-loop time constant that guarantees a desired robustness. From that point, an analytical design is conducted for the assignment of the load-disturbance dynamics followed by the tuning of the set-point weight factor in order to match, as much as possible, the set-point-to-output dynamics according to a first-order-plus-dead-time dynamics. Simple tuning rules are generated by considering specific values for the maximum sensitivity value. These tuning rules, provide all the controller parameters parameterized in terms of the open-loop normalized dead-time allowing the user to select a high/medium/low robust closed-loop control system. The proposed autotuning expressions are therefore compared with other well known tuning rules also conceived by using the same robustness measure, showing that the proposed approach is able to guarantee the same robustness level and improve the system time performance.

1. Introduction

Most of the single-loop controllers used in practice are found under the form of a PI/PID controller. Effectively, since their introduction in 1940 [1,2], commercial *Proportional – Integrative – Derivative* (PID) controllers have been, with no doubt, the most extensive option found on industrial control applications. Their success is mainly due to its simple structure and meaning of the corresponding three parameters. This fact makes PID control easier to understand by the control engineers than other most advanced control techniques. This fact has motivated a continuous research effort to find alternative tuning and design approaches to improve PI/PID based control system's performance.

With regard to the design and tuning of PID controllers, there are many methods that can be found in the literature over the last sixty years. Special attention is paid to the *IFAC workshop PID'00 Past, Present and Future of PID Control* held in Terrassa, Spain, in April 2000, where a glimpse of the state-of-the-art on PID control was provided. It can be seen that most of them are concerned with feedback controllers which are tuned either with a view to the rejection of disturbances [3–5] or for a well-damped fast response to a step change in the controller set-point [6–8]. O'Dwyer [9] presents a collection of tuning rules for PI and PID controllers, which show their abundance.

Recently, tuning methods based on optimization approaches with the aim of ensuring good stability robustness have received attention in the literature [10,11]. Also, great advances on optimal methods based on stabilizing PID solutions have been achieved [12,13]. However these methods, although effective, rely on somewhat complex numerical optimization procedures and do not provide tuning rules. Instead, the tuning of the controller is defined as the solution of the optimization problem.

Among the different approaches, the direct or analytical synthesis constitutes a quite straightforward approach to PID controller tuning. The controller synthesis presented by Martin [6] made use of zero-pole cancelation techniques. Similar relations were obtained by Rivera et al. [7,14], by applying the IMC concepts of Garcia and Morari [15] for tuning PID controllers for low-order process models. A combination of analytical procedures and the IMC tuning can be found in [16–18]. With this respect, the usual approach is to specify the desired closed-loop transfer function and to solve analytically for the feedback controller. In cases where the process model is of simple structure, the resulting controller has the PI/PID structure. Most of the analytically developed tuning rules are related with the servo-control problem while the

* Corresponding author. Tel.: +34 935812197; fax: +34 935814031.
E-mail addresses: Ramon.Vilanova@uab.cat (R. Vilanova), Victor.Alfaro@ucr.ac.cr (V.M. Alfaro), Orlando.Arrieta@uab.cat, Orlando.Arrieta@ucr.ac.cr (O. Arrieta).

0019-0578/$ – see front matter
doi:10.1016/j.isatra.2011.09.001

consideration of the load-disturbance specifications has received not so much attention. However it is well known that if we optimize the closed-loop transfer function for a step-response specification, the performance with respect to load-disturbance attenuation can be very poor [19]. This is indeed the situation, for example, for IMC controllers that are designed in order to attain a desired set-point to output transfer function presenting a sluggish response to the disturbance [18].

From the observation of the poor load-disturbance characteristics of analytically obtained controllers, is of remarkable interest the work of Chen and Seborg [20], where the importance of emphasizing disturbance rejection, as the starting point for design, is discussed. A similar direct synthesis approach posed in [20], was used for disturbance rejection design for First-Order-Plus-Dead-Time (FOPDT) models. Once a process model is assumed, the controller equations are got on a direct way. One of the interesting features of the provided tuning rules in [20] is that all of them are parameterized in terms of the desired closed-loop time constant. The main drawback behind that formulation is that it was constrained to One-Degree-Of-Freedom (1-DOF) PI, PID controllers, where the tuning is performed on the basis of a load-disturbance specification and the performance of set-point was not taking into account (just some ad-hoc values for the set-point weighting factor are used in the examples). Moreover, that tuning does not include any consideration about robustness level, therefore, the resulting closed-loop Performance/Robustness *tradeoff* was not addressed.

The need to deal with both kind of properties and the recognition that a control system is, inherently, a system with Two Degrees-of-Freedom (2-DoF) – two closed-loop transfer functions can be adjusted independently –, motivated the introduction of 2-DoF PI/PID controllers [21]. The 2-DoF formulation is aimed at trying to met both objectives, say good regulation and tracking properties. This second degree of freedom is aimed at providing additional flexibility to the control system design. See for example [22–24] and its characteristics revised and summarized in [25–27], as well as different tuning methods that have been formulated over the last years [25,28–33]. There have also been some particular applications of the 2-DoF formulation based on advanced optimization algorithms (see for example [34–37]). The point is that, with a few exceptions such as the AMIGO [33] and Kappa–Tau; κ–τ; [38] methods, no analytical expressions are provided for all controller parameters (feedback and reference part) and, at the same time, ensure a certain robustness degree for the resulting closed-loop. To provide simple tuning expressions and, at the same time, guarantee some degree of robustness are the main contributions of the paper. This second degree of freedom is found on the presented literature as well as in commercial PID controllers under the form of the well known set-point weighting factor (usually called β) that ranges within $0 \leq \beta \leq 1.0$, being the main purpose of this parameter to avoid excessive proportional control action when a reference change takes place. Therefore the use of *just a fraction* of the reference.

However, performance with respect to load-disturbance attenuation is just one of the drawbacks of the analytical approaches to PI/PID controller design. In fact, the known analytical approaches do not include any consideration on the control system robustness. The usual approach is to measure the robustness of the resulting design (usually in terms of the peak value of the sensitivity function M_s) instead of specifying a desired robustness level from the very beginning. Industrial practice needs to cope with different conditions of process operation, generated by either changes (even slight) in equipment or contour constraints on the process itself. Therefore, there is the need to account for some robustness that prevents the gains of the controller to become excessively high and generate a need for detuning. In addition to these considerations, robustness is an important attribute for control systems, because the design procedures are usually based on the use of low-order linear models identified at the closed-loop control system operation point. Due to the interactions and non-linearities found in most of the industrial process, it is necessary to consider the expected changes in the process characteristics assuming certain relative stability margins, or robustness requirements, for the control system. Therefore, the design of the closed-loop control system must take into account the system *performance* and its *robustness* to the variation of the controlled process characteristics, preserving the well-known *trade-off* between all these variables.

It is with this respect that this paper provides its main contribution: a load-disturbance based analytical design being the *only* design parameter the desired robustness level of the resulting control system. At this point, the performance–robustness tradeoff arises and has to be introduced into the design procedure. As for set-point performance the desired closed-loop time constant is to be chosen as fast as possible (robustness permitting) the presented procedure characterizes, for each possible peak value of the sensitivity function (within its usual [1.2–2.0] range), the lowest allowable time constant. This first analysis conducts to a design approach that is divided in two steps: first of all, an equation is provided that generates the desired closed-loop time constant from the specified robustness; on a second step this time-constant is introduced on the parameterized controller parameters relations. It is worth to stress that at this point the approach is presented here just for PI controller design, being the full PID case more involved and its full derivation is to be presented separately.

Even the presented procedure can be applied with any desired robustness level, maybe in practice the designer would like to use the robustness parameter on a more qualitative way, having, for example, three choices depending on the desired degree of robustness: (low, medium, high). This is to say the use of a controller with a minimum acceptable robustness level (that would be represented by $M_s = 2.0$), a robust controller (that would be represented by $M_s = 1.6$) or a highly robust controller (that would be represented by $M_s = 1.4$). With this consideration on hand, the previous corresponding values of M_s are introduced into the previously got general expressions and the resulting relations further simplified in order to get simple robust autotuning rules according to the specified robustness degree.

The organization of the paper is as follows. Next section introduces the framework and notation related to 2-DoF PID controllers as well as how the analytical load-disturbance based design problem is formulated. Section 3 presents the development of the robust approach to PI design. Section 4 is devoted to the obtention of simple direct tuning rules for the most usual robustness levels. Section 5 presents comparative simulation examples and, finally, on Section 6 conclusions are conducted as well as an outline of continuing research.

2. Problem formulation

Considerer the *Two-Degree-of-Freedom* (2-DoF) feedback control system of Fig. 1 where $P(s)$ is the controlled process transfer function, $C_r(s)$ the *set-point controller* transfer function, $C_y(s)$ the *feedback controller* transfer function, and $r(s)$ the set-point, $d(s)$ the load-disturbance, and $y(s)$ the controlled variable. The output of the 2-DoF PI, PI_2, controller is given by

$$u(s) = \underbrace{K_c \left(\beta + \frac{1}{T_i s}\right)}_{C_r(s)} r(s) - \underbrace{K_c \left(1 + \frac{1}{T_i s}\right)}_{C_y(s)} y(s) \tag{1}$$

where K_c is the controller gain, T_i the integral time constant, and β the set-point weighting factor ($0 \leq \beta \leq 1$).

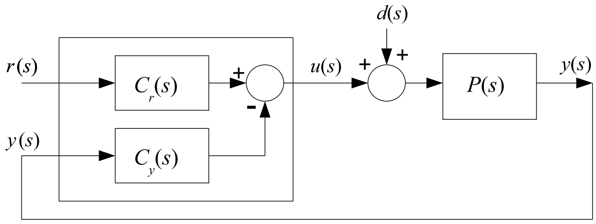

Fig. 1. 2-DoF control system.

The closed-loop control system response to a change in any of its inputs, will be given by

$$y(s) = \underbrace{\frac{C_r(s)P(s)}{1 + C_y(s)P(s)}}_{M_{yr}(s)} r(s) + \underbrace{\frac{P(s)}{1 + C_y(s)P(s)}}_{M_{yd}(s)} d(s) \tag{2}$$

where $M_{yr}(s)$ is the transfer function from set-point to process variable: the *servo-control* closed-loop transfer function or complementary sensitivity function $T(s)$; and $M_{yd}(s)$ is the one from load-disturbance to process variable: the *regulatory control* closed-loop transfer function or disturbance sensitivity function $S_d(s)$.

If $\beta = 1$, all parameters of $C_r(s)$ are identical to the ones of $C_y(s)$. In such situation, it is impossible to specify the dynamic performance of the control system to set-point changes, independently of the performance to load-disturbances changes. Otherwise, if the contrary, $\beta < 1$, given a controlled process $P(s)$, the feedback controller $C_y(s)$ can be selected to achieve a target performance for the regulatory control $M_{yd}(s)$, and then use the set-point weighting factor in the set-point controller $C_r(s)$, to modify the servo-control performance $M_{yr}(s)$.

The proposed *Analytic Robust Tuning of two-degree-of-freedom PI controllers* (ART_2) [28,39], is aimed at producing a control system that responds fast and without oscillations to a step load-disturbance, with a maximum sensitivity lower than a specified value; in order to assure robustness; and which will also show a fast non-oscillating response to a set-point step change, not requiring strong or excessive control effort variations (*smooth* control).

2.1. Outline of controller design procedure

The first step in the two-degree-of-freedom controller synthesis consists of obtaining the feedback controller $C_y(s)$, required to achieve a target $M_{yd}^t(s)$ regulatory closed-loop transfer function. From (2) once the controlled process is given and the target regulatory transfer function, $M_{yd}^t(s)$, specified the required feedback controller can be synthesized. The resulting feedback controller design equation is

$$C_y(s) = \frac{P(s) - M_{yd}^t(s)}{P(s)M_{yd}^t(s)} = \frac{1}{M_{yd}^t(s)} - \frac{1}{P(s)}. \tag{3}$$

Once, as a first step, the feedback controller $C_y(s)$, is obtained from (3), on a second step, the set-point controller $C_r(s)$ free parameter (β) can be used in order to modify the servo control closed-loop transfer function $M_{yr}(s)$.

The outlined design approach is in fact like the direct design as proposed within the IMC framework [7]. In IMC however, the designer has to choose the well known IMC design parameter in order to satisfy the performance/robustness tradeoff. What will be proposed in the formulation presented here is to avoid such step, by an automatic selection of the controller parameters in terms of the desired robustness. The selection of the control system bandwidth is done in such a way the closed-loop bandwidth is as large as possible while meeting the robustness constraint. It could therefore be interpreted as an IMC controller with robustness considerations explicitly incorporated.

3. Tuning rules for 2-DoF PI control

Consider the First-Order-Plus-Dead-Time (FOPDT) controlled process given by

$$P(s) = \frac{K_p e^{-Ls}}{Ts + 1} \tag{4}$$

where K_p is the process gain, T the time-constant, and L its dead-time. From here and after, $\tau_o = L/T$ will be referred as the controlled process *normalized dead-time*. In this work process models with normalized dead-time $\tau_o \leq 2$ are considered. Processes with long dead time will need some kind of dead-time compensation scheme (a Smith predictor, for example).

For the FOPDT process the specified regulatory and closed-loop control target transfer functions are chosen as

$$M_{yd}^t(s) = \frac{Kse^{-Ls}}{(\tau_c Ts + 1)^2} \qquad M_{yr}^t(s) = \frac{e^{-Ls}}{\tau_c Ts + 1} \tag{5}$$

where τ_c will be the *dimensionless design parameter*. It is the ratio of the closed-loop control system time constant to the controlled process time constant. The specified target closed-loop transfer functions (5) will provide non-oscillating responses to step changes in both, the set-point and the load-disturbance, with an adjustable speed.

3.1. Controller parameters

In order to synthesize the 2-DoF PI controller for the FOPDT process it is necessary to use a rational function in s as an approximation of the controlled process dead-time. This approximation will affect the closed-loop response characteristics. Using the Maclaurin first order series for the dead-time: $e^{-Ls} \approx 1 - Ls$ and (4) and (5) in (3), the PI_2 controller tuning equations are obtained as

$$\kappa_c = K_c K_p = \frac{2\tau_c - \tau_c^2 + \tau_o}{(\tau_c + \tau_o)^2} \tag{6}$$

$$\tau_i = \frac{T_i}{T} = \frac{2\tau_c - \tau_c^2 + \tau_o}{1 + \tau_o} \tag{7}$$

where κ_c and τ_i are the controller *normalized parameters*.

In order to assure that the controller parameters (6) and (7) have positive values, the design parameter τ_c must be selected within the range

$$0 < \tau_c \leq 1 + \sqrt{1 + \tau_o}. \tag{8}$$

The resulting regulatory control closed-loop transfer function is

$$M_{yd}(s) = \frac{T_i s e^{-Ls}}{K_c(\tau_c Ts + 1)^2}. \tag{9}$$

3.2. Set-point weighting factor

As the closed-loop transfer functions are related by $M_{yr}(s) = C_{yr}(s)M_{yr}(s)$, by using controller $C_r(s)$, $M_{yr}(s)$ can be written as

$$M_{yr}(s) = \frac{K_c\,(\beta T_i s + 1)}{T_i s} M_{yd}(s). \tag{10}$$

Introducing in (10) the regulatory control closed-loop transfer function (9) and also the controller parameters (6) and (7), the servo-control transfer function then becomes

$$M_{yr}(s) = \frac{(\beta T_i s + 1)\,e^{-Ls}}{(\tau_c Ts + 1)^2}. \tag{11}$$

As the servo-control target transfer function was specified in (5), from (5), (10) and (11) in order to obtain a non-oscillatory response, an adequate selection of the set-point weighting factor

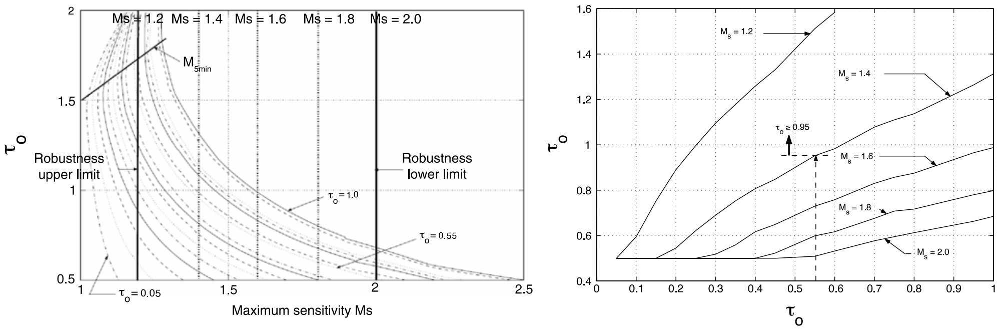

Fig. 2. Control system robustness inverse M_s and lower limits for τ_c.

would be $\beta = \tau_c T/T_i$, and then

$$\beta = \frac{\tau_c T}{T_i}, \quad 0 < \tau_c \leq 1 \tag{12}$$

outside this range

$$\beta = 1, \quad 1 < \tau_c < 1 + \sqrt{1 + \tau_o}. \tag{13}$$

Effectively, it can be verified that $\tau_i \leq 1$. Therefore, if $\tau_c > 1$, as $\beta = \tau_c(T/T_i)$ we will have $\beta = \tau_c/\tau_i > 1$. In addition if $\tau_c \leq 1\tau_i$ is always larger than τ_c therefore assuring $\beta = \tau_c/\tau_i \leq 1$. The constraint $\beta \leq 1$ is introduced because in commercial controllers the set-point weighting factor (when available) is restricted to have a value lower than one. This selection for the $0 < \tau_c \leq 1$ range, will made the set-point controller zero to cancel one of the closed-loop poles. This weighting factor also has influence in the controller output when the set-point changes. Effectively, the instantaneous change on the control signal caused by a sudden change in the reference signal of magnitude Δr is given by $\Delta u_r = K_c\beta\Delta e = K_c\beta\Delta r$ therefore, when very fast regulatory control responses are desired, high controller gain values are required, and the controller instantaneous output change when the set-point changes may be high. Then the controller output will be limited to be not greater than the total change on the set-point and then the set-point weighting factor selection criteria becomes

$$\beta = \min\left\{\frac{1}{K_c}, \frac{\tau_c T}{T_i}, 1\right\}. \tag{14}$$

3.3. Control system robustness

The maximum sensitivity

$$M_s = \max_{\omega} |S(j\omega)| = \max_{\omega} \left|\frac{1}{1 + C_y(j\omega)P(j\omega)}\right| \tag{15}$$

will be used as an indication of the closed-loop control system robustness.

The use of the maximum sensitivity as a robustness measure, has the advantage that lower bounds to the gain and phase margins [38] can be assured according to

$$A_m > \frac{M_s}{M_s - 1} \tag{16}$$

$$\phi_m > 2\sin^{-1}\left(\frac{1}{2M_s}\right). \tag{17}$$

A robustness analysis has been performed and shown in Fig. 2. This analysis shows that the control system maximum sensitivity M_s depends of the model normalized dead-time τ_o and the design parameter τ_c.

In order to avoid the loss of robustness when a very low τ_c is used, it is necessary to establish a lower limit to this design parameter. This relative loss of stability is greater when the normalized model dead-time τ_o is high. The lower limits to the design parameter for a specific robustness level can be obtained. These limits are shown in Fig. 2. From this figure the design parameter lower limit for a given robustness level can be expressed in parameterized form as

Table 1
Eq. (18) constants.

M_s	1.2	1.4	1.6	1.8	2.0
k_1	0.4836	0.4152	0.3441	0.3254	0.3042
k_2	1.8982	0.9198	0.6659	0.4853	0.3822

$$\tau_{c\min} = k_1(M_s) + k_2(M_s)\tau_o \tag{18}$$

where the k_1 and k_2 are show in Table 1.

The design parameter equation (18) can be expressed as a single equation as

$$\tau_{c\min} = k_{11}(M_s) + \left[\frac{k_{21}(M_s)}{k_{22}(M_s)}\right]\tau_o \tag{19}$$

$$k_{11}(M_s) = 1.384 - 1.063M_s + 0.262M_s^2$$
$$k_{21}(M_s) = -1.915 + 1.415M_s - 0.077M_s^2$$
$$k_{22}(M_s) = 4.382 - 7.396M_s + 3.0M_s^2.$$

Also from Fig. 2 it can be seen that; as usual; as the system becomes slower its robustness increases but if very slow responses are specified the system robustness starts to decrease, therefore the upper limit of the design parameters τ_c also needs to be constrained by combining the design parameter performance and robustness constraints it may be selected within the range

$$\max(0.50, \tau_{c\min}) \leq \tau_c \leq 1.50 + 0.3\tau_o \tag{20}$$

where $\tau_{c\min}$ is given by (19).

4. Simplified autotuning rules for 2-DoF PI control

To provide the possibility of specify any possible desired robustness level within the range $M_s \in [1.2 - 2.0]$ is of great interest as this provides a complete view of the robustness–performance *tradeoff* as well as a quantified measure of how restrictive a robustness level can be depending on the process normalized dead-time. However, from a more practical point of view, the following question arises: When a desired $M_s = 1.57$ will be specified? With this respect, as the M_s value is being recognized as a *de facto* standard measure of robustness, an M_s value of 2.0 is recognized as the minimum acceptable robustness level. This corresponds, by using (16) and (17) to the classical $A_m \geq 2$ and $\phi_m \geq 30^o$. This could be considered a *low* degree of robustness. According to a similar measure, and in order to make the analysis simpler, a *medium* degree of robustness is associated here with $M_s = 1.6$ while a *high* degree of

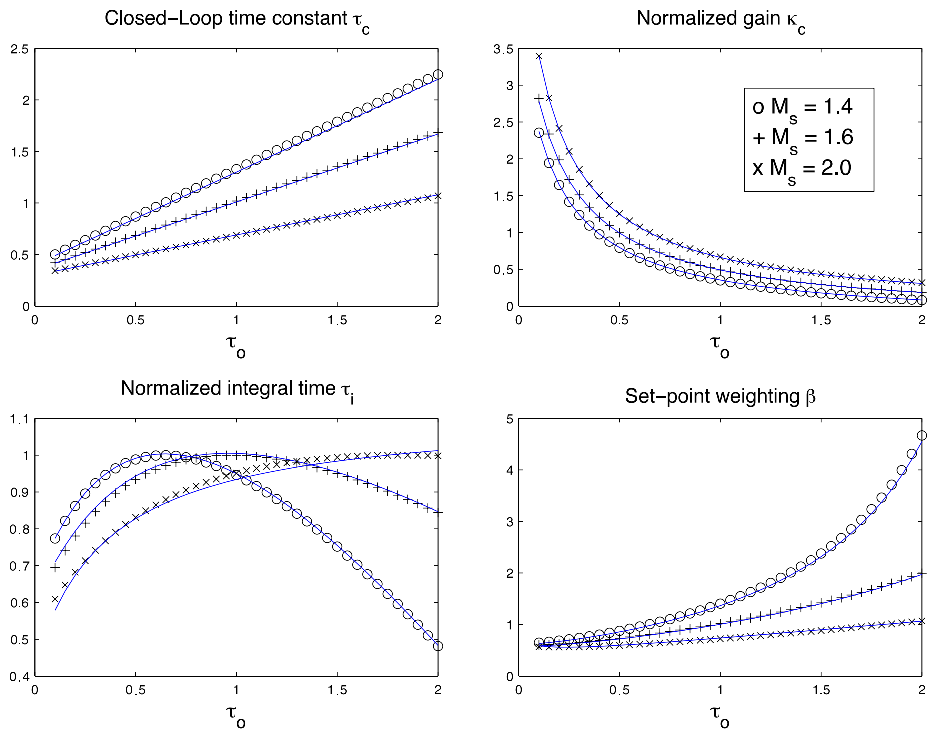

Fig. 3. PI normalized parameters for low, medium and high robustness.

robustness will correspond to $M_s = 1.4$. This broad classification allows a qualitative specification of the control system robustness.

According to this principle, the above mentioned three values of M_s are used here to generate the corresponding estimate for the lowest allowable closed-loop time-constant with (18) and introduce such time-constant value into the PI parameter Eqs. (6), (7) and (12). The resulting controller parameters will be, in this case, expressed just in terms of the process normalized dead-time τ_o as:

- High-Robustness Tuning ($M_s = 1.4$)

$$\kappa_c = \frac{-0.23\tau_o + 0.64}{\tau_o + 0.16};$$
$$\tau_i = \frac{-0.85\tau_o^2 + 2.1\tau_o + 0.65}{\tau_o + 1}; \qquad \beta = \frac{0.9\tau_o + 0.4}{\tau_i}. \tag{21}$$

- Medium-Robustness Tuning ($M_s = 1.6$)

$$\kappa_c = \frac{-0.17\tau_o + 0.74}{\tau_o + 0.16};$$
$$\tau_i = \frac{-0.44\tau_o^2 + 1.85\tau_o + 0.6}{\tau_o + 1}; \qquad \beta = \frac{0.66\tau_o + 0.35}{\tau_i}. \tag{22}$$

- Low-Robustness Tuning ($M_s = 2.0$)

$$\kappa_c = \frac{-0.1\tau_o + 0.86}{\tau_o + 0.15}; \qquad \tau_i = \frac{1.12\tau_o + 0.16}{\tau_o + 0.37};$$
$$\beta = \frac{0.39\tau_o + 0.3}{\tau_i}. \tag{23}$$

Fig. 3 shows the generated values for a grid of $\tau_o \in [0.1 - 2.0]$ as well as the regression curves that gives rise to the above formulas for the normalized gain (κ_c), integral time (τ_i) and set-point weighting factor β.

It is interesting to note that as the robustness degree is increased, the fastest allowable closed-loop time constant, τ_c, increases generating a slower controlled system. Accordingly the controller gain decreases and the set-point weighting increases in order to compensate such loose of gain. However, the behavior of the integral time is a little bit more complex. Whereas the general tendency is to decrease as robustness is increased, this is not completely true for all plants. For plants with a normalized dead-time in the middle region ($\tau_o \approx 1$), it is not true that a more robust tuning implies a smaller τ_i. An M_s value of 1.6 generates higher values for τ_i than $M_s = 1.4$ and $M_s = 2.0$. In addition, whereas for $M_s = 1.4$ and $M_s = 1.6$ the normalized integral time decreases with τ_o, it increases for $M_s = 2.0$. What robustness degree imposes on integral time is the rate of change with respect to τ_o. As more robustness degree is desired, the derivative $\partial\tau_i/\partial\tau_o$ takes higher values.

In order to evaluate the obtained autotuning expressions a performance evaluation is conducted for the two aspects considered when generating the complete full expressions: achieved robustness and deviation of the closed-loop response with respect the one obtained for the original tuning. As the desired step response is specified as of first order with a time constant equal to the fastest one allowable, the deviation with respect to this target is also evaluated. Fig. 4 shows the achieved robustness for the three considered cases. It is seen that the Low-Robustness case is easily achieved and even with large margins for small and large values of the normalized dead-time. In all cases the achieved robustness level can be considered accordingly to the specified one.

The simplified tuning rules are generated from the full procedure presented in the previous section. It is therefore a must to check how the closed-loop responses generated by using the simplified tuning rules deviate from the ones corresponding with the full one obtained in the previous section. As the approximation may have different repercussion on the step response and disturbance attenuation both performance degradations are measured independently. As a measure of closeness to the original full design, the following *IAE* index is evaluated for the set of plants within the working interval of the normalized dead-time τ_o

$$\mathrm{IAE}(y, y_{\mathrm{app}}) = \int_0^\infty |y(\tau) - y_{\mathrm{app}}(\tau)|d\tau \tag{24}$$

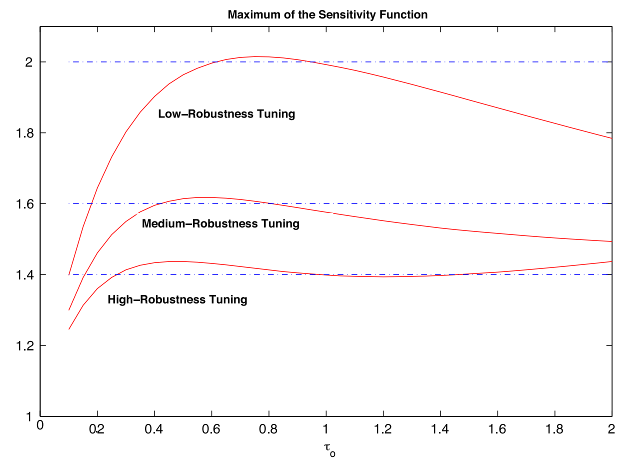

Fig. 4. Achieved M_s values with respect normalized dead-time for low, medium and high robustness.

being $y(t)$ the closed-loop output generated by applying the controller obtained by the full design and $y_{app}(t)$ the closed-loop response generated by the simple approximation. It is distinguished if the deviation is caused when the system operates as a servo system or because of the presence of a load-disturbance. Fig. 5 shows the deviation for the load-disturbance attenuation time response is quite small for all the τ_o range. However, for the set-point step-response case an important degradation is observed for values $\tau_o > 0.8$, specially for the High-Robustness case ($M_s = 1.4$).

In order to go further on this mismatch, another deviation has been computed in terms of the same functional. As the original design problem was formulated in order to achieve a first-order dynamics, the effect of the delay approximation starts to manifest. This can be seen if we evaluate how close are the closed-loop responses generated by the original full design $y(t)$ and the simple approximated one $y_{app}(t)$ to that of first order specified by the target $M^t_{yr}(s)$, $y^t(t)$. We therefore compute

$$\mathrm{IAE}(y^t, y) = \int_0^{\infty} |y^t(\tau) - y(\tau)| d\tau \tag{25}$$

$$\mathrm{IAE}(y^t, y_{app}) = \int_0^{\infty} |y^t(\tau) - y_{app}(\tau)| d\tau. \tag{26}$$

The result is also shown in Fig. 5 where it can be confirmed that the previously detected mismatch among y and y_{app} is mostly due because of the deviation, on the full tuning case, from the desired first-order dynamics.

Table 2
Example 1—PI parameters; complete and autotuning.

Complete tuning					Autotuning		
M_s^d	$\tau_{c\min}$	K_c	T_i	β	K_c	T_i	β
1.4	0.875	0.7914	0.9789	0.8688	0.7955	0.9917	0.8571
1.6	0.677	0.9958	0.7346	0.6864	0.9924	0.9333	0.7286
2.0	0.500	1.2547	0.8312	0.5978	1.2462	0.8276	0.5981

5. Examples

Several examples are presented in order to show the efficiency of the proposed simple tuning rules. A simple example is proposed first where the performance of the simple PI tuning is compared against the full design for the three defined robustness levels. On a second example, a comparison is performed with several well known approaches comparing performance and achieved robustness.

In all the examples it is supposed that all variables can vary in the 0%–100% normalized range and that in the normal operation point, the controlled variable, the set-point and the control signal, have all values close to 70%. The corresponding system and controller outputs to a 20% set-point change followed by a 10% load-disturbance change are shown.

5.1. Example 1

Consider the FOPDT controlled process

$$P_1(s) = \frac{e^{-0.5s}}{s+1}. \tag{27}$$

By using the full design equations, the controller parameters can be obtained by varying the tuning parameter τ_c. Using the process normalized dead-time ($\tau_o = 0.5$ for this example) and (18) and (20) the recommended lower limit for the design parameter to obtain a specified minimum robustness are estimated and listed in Table 2.

In order to evaluate the performance of the simple tuning rules, the corresponding values of M_s^d are taken. The controller parameters for the complete and autotuning relations are shown in Table 2.

Fig. 6 shows the closed-loop time responses for the different controller values. As it can be seen, output responses and control values for the tuning got using the complete expressions and those got from the simple autotuning ones cannot be distinguished.

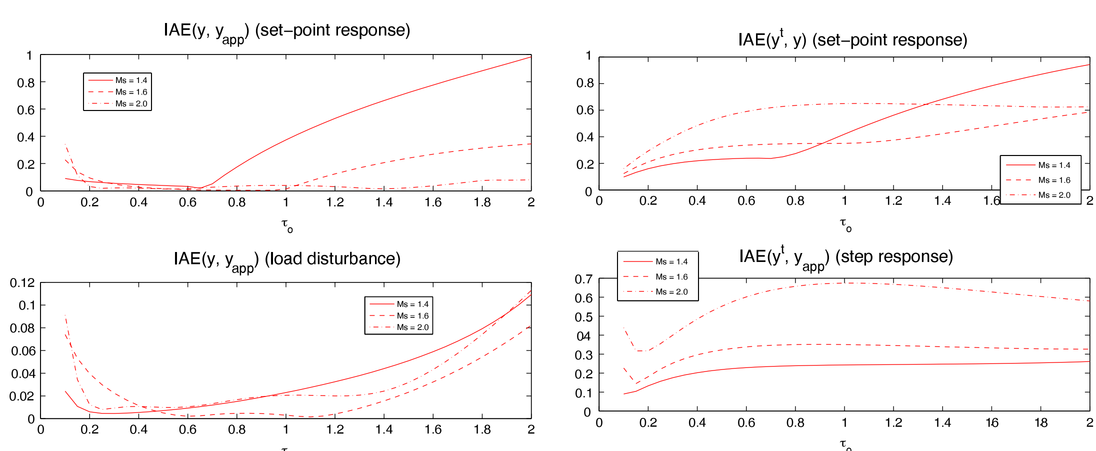

Fig. 5. IAE error deviations. Simplified tuning vs. full tuning according to index (24) and full and simplified designs with respect to the target specifications computed according to index (25) and (26).

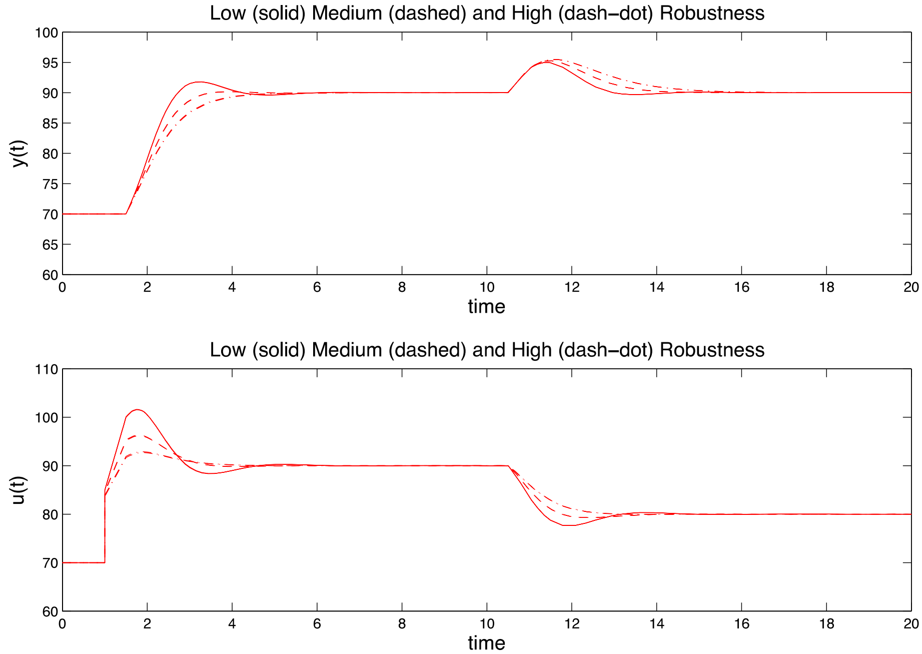

Fig. 6. Example 1—system responses for the three robustness levels and comparing the complete and simple autotuning rules.

Table 3
Process model parameters.

α	K_p	T	L	τ_o
0.25	1.0	1.049	0.298	0.284
0.50	1.0	1.247	0.691	0.554
1.0	1.0	2.343	1.860	0.794

Therefore the performance obtained is completely equivalent to that of the full tuning rules.

5.2. Example 2

In this example the proposed method is compared with other well known and recognized PI tuning methods that can be found in the literature: The AMIGO [33] and Kappa–Tau; κ–τ; [38] methods. The methods are chosen because incorporate the M_s value as an explicit design specification; provide a guide on the selection of the set-point weight β, and because of its simplicity therefore providing the controller parameters in terms of simple relations that involve the process characteristics. These methods could therefore be considered quite similar to the one proposed in this paper.

The following fourth order controlled process will be considered in the study

$$P_2(s) = \frac{1}{\prod_{n=0}^{3} (\alpha^n s + 1)} \tag{28}$$

with $\alpha = \{0.25, 0.50, 1.0\}$. Using a two-point identification procedure [40] FOPDT models were obtained whose parameters are show in Table 3. These parameters will be the ones used for tuning the PI controllers.

1. *Proposed simple robust autotuning*—The controller parameters are obtained with the proposed simplified tuning rules. It can be verified that the achieved robustness, M_s^r, accomplishes with the desired level with the only exception of the cases $\alpha = 1.0$ where the achieved M_s^r is slightly higher but, in any case, within reasonable margins according to the specified level.
2. *AMIGO Tuning*—We use the revised version of the method in [41] for 2-DoF PI controllers. All the obtained system robustness are higher ($M_s^r \approx 1.2$) than the one used in the method specification ($M_s = 1.4$) resulting in slow responses. This method will be therefore associated to a high robustness specification.
3. κ–τ *Tuning*—This method, proposed in [38], also provides the parameters for a 2-DoF PI/PID controller on the basis of a FOPDT specification and a desired M_s robustness level: $M_s = 1.4$ (High-Robustness) or 2.0 (Low-Robustness). The desired M_s values are obtained, on a global sense, with less margin than the proposed method. With the exception of the High-Robustness case for $\alpha = 1$, all the obtained values are lower than the ones provided by the proposed method. This will have a clear repercussion on the time performance.

In order to compare how the presented methods perform, tunings of similar robustness level will be evaluated. Therefore two cases will be distinguished: the proposed, AMIGO and κ–τ for the High-Robustness tuning and the proposed and κ–τ for the Low-Robustness. In addition, the proposed tuning for the Medium-Robustness case will be considered in both cases. It will be shown that as a compromise solution, the $M_s = 1.6$ specification can be considered as a good candidate for a reasonable robustness level with not so much performance deterioration.

The evaluation and corresponding comparison will be done according to criteria aimed to represent both robustness and performance. The following measures will be used:

- *Robustness*: As a rather usual measure for robustness will use the Sensitivity and Complementary Sensitivity peaks, M_s and M_t respectively, providing M_t a measure of the allowable multiplicative uncertainty bound. This measure, as it has been mentioned above, is considered as an explicit design specification for the considered methods.
- *Output performance*: The Integrated Absolute Error (IAE) of the error $e = r - y$ will be computed. This value should be as small as possible

$$\mathrm{IAE} = \int_0^\infty |e(t)|dt.$$

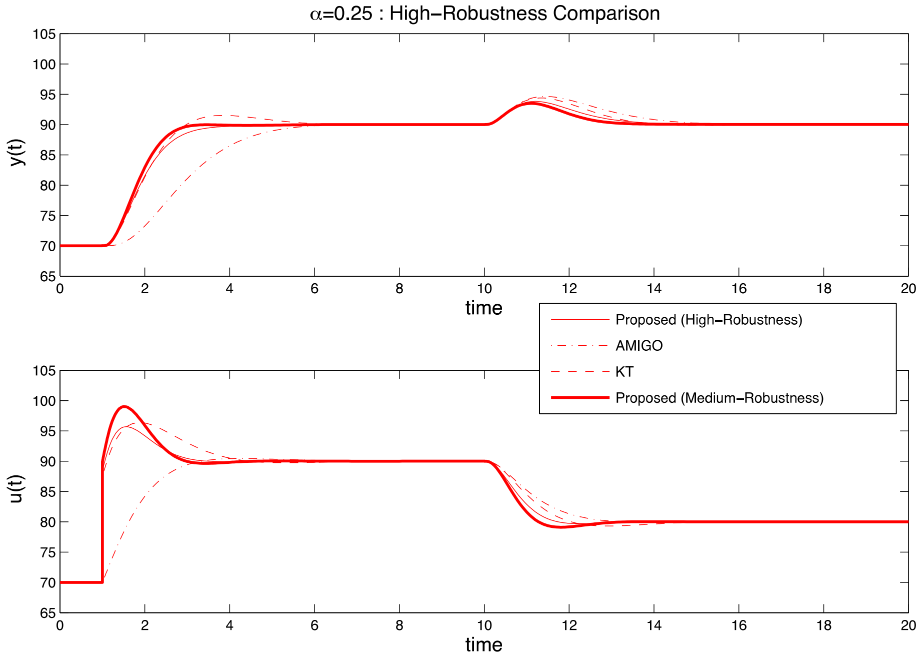

Fig. 7. Example 2—High-Robustness tuning comparison for example 2 ($\alpha = 0.25$).

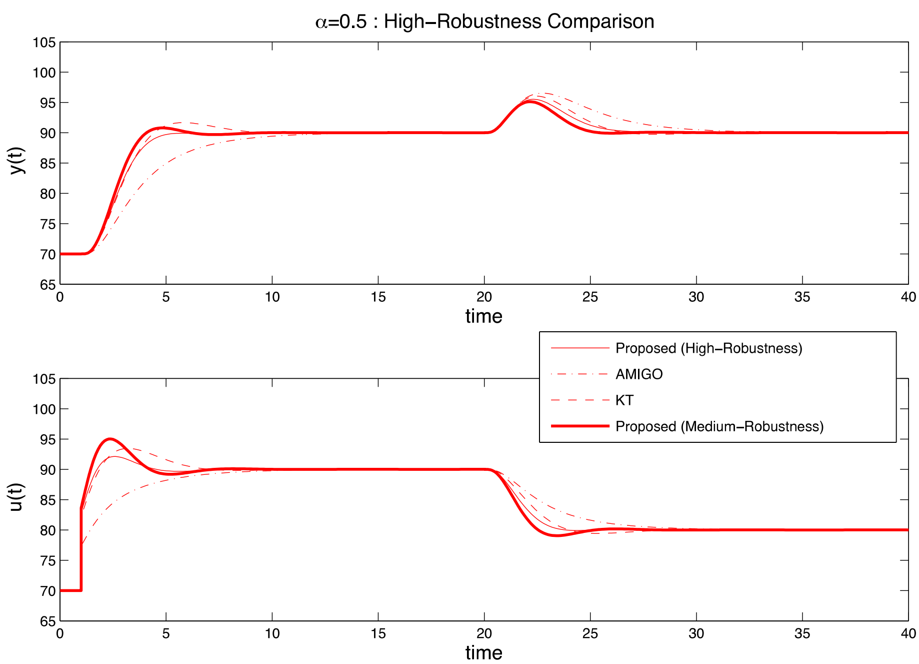

Fig. 8. Example 2—High-Robustness tuning comparison for example 2 ($\alpha = 0.5$).

- *Input performance*: To evaluate the manipulated input usage, the total variation (TV) of the control signal, $u(t)$, is computed. This value is defined, for a discrete signal as the sum of the size of its increments

$$TV = \sum_{i=1}^{\infty} |u_{i+1} - u_i|.$$

This quantity should be as small as possible and provides a measure of the smoothness of the control signal. In order to define it properly for a continuous signal (that is the case in our examples) a sampled version of the control signal has to be used.

This will provide a more global and complete comparison framework. The figures provide the output responses to both a step reference change and a load disturbance, as well as the generated control actions. As it has been mentioned above, in order to be more realistic it is considered that the controllers operate at 70% of their operating regime. Figs. 7–9 show the resulting outputs for the High-Robustness tunings whereas in Figs. 10–12 the Low-Robustness case is considered.

Straight conclusions could be drawn from the figures, showing the different time responses. It is clear that the proposed autotuning provides a more homogeneous response for the different cases. The AMIGO approach results excessively conservative and should be used just in case really High-Robustness levels are required and

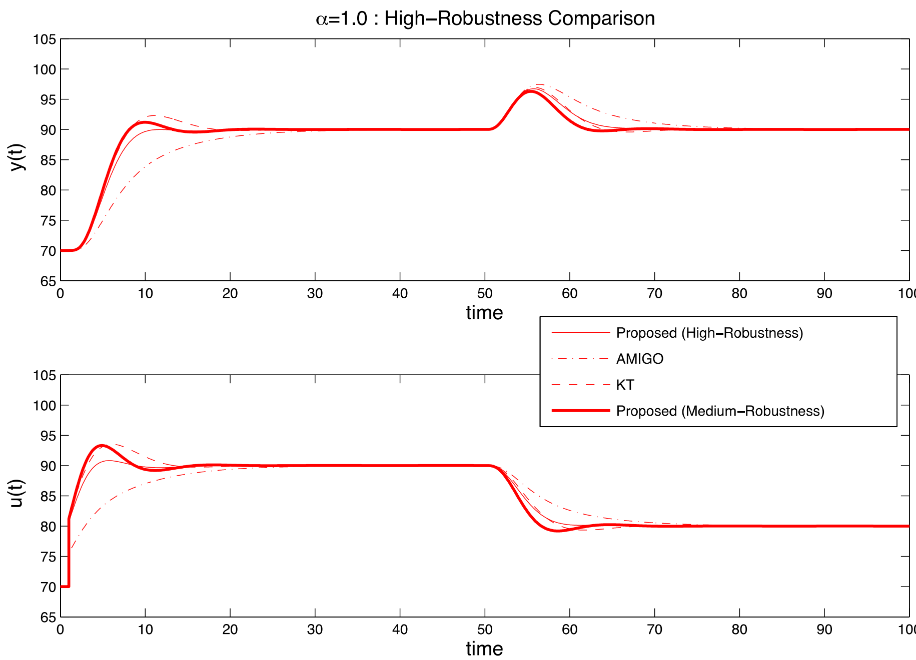

Fig. 9. Example 2—High-Robustness tuning comparison for example 2 ($\alpha = 1.0$).

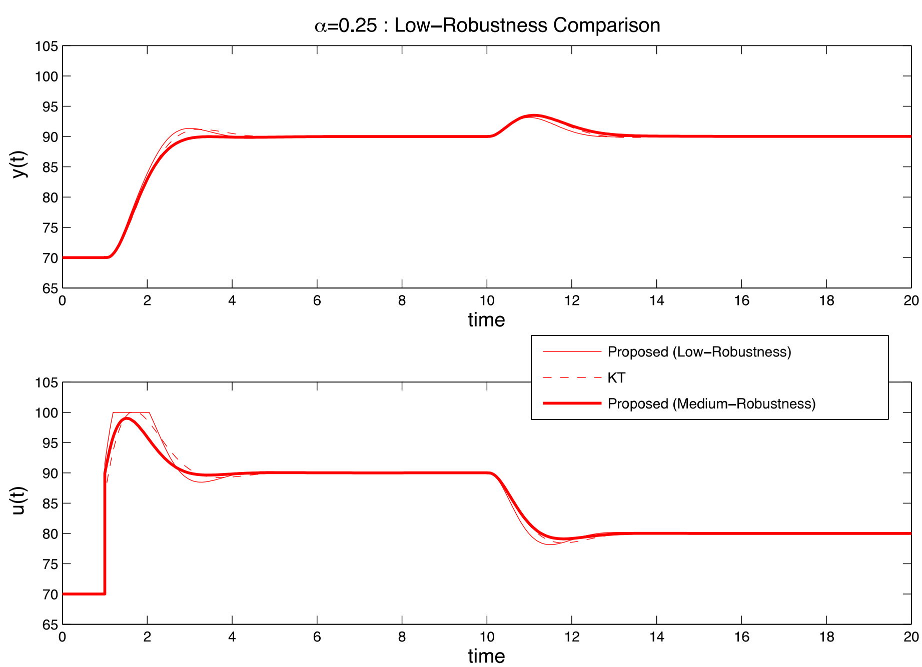

Fig. 10. Example 2—Low-Robustness tuning comparison for example 2 ($\alpha = 0.25$).

performance remains as a secondary objective. This can be verified by the data supplied in Table 4 where the IAE values corresponding to the AMIGO tuning are extremely higher than the ones provided by either the κ–τ or the proposed method. If we concentrate on the tunings conceived to provide High-Robustness level, the proposed method clearly provides better performance values for both set-point tracking and load-disturbance attenuation. It is also worth to note that the input performance values are also smaller, (with the exception of those for the AMIGO tuning that are the smallest ones but paying an excessive performance degradation).

If we pay attention to the achieved robustness levels, it is seen that for the proposed method, M_s values are slightly better than those achieved by applying the κ–τ method. Moreover, from the preceding observations this robustness is achieved with also an increase in time domain performance.

The situation for the Low-Robustness tuning is quite similar (time domain performance is better for the proposed method) with the only point that the M_s values are not smaller for the κ–τ method. This also traduces into a slightly more aggressive control action for the proposed method. However, it should be kept in mind that the initial proposal was a Low-Robustness method, specified by a threshold of $M_s = 2.0$. Achieved values for M_s are therefore not to expected to be so smaller (see Table 5).

The situation depicted from the comparison of the High and Low Robustness tunings clearly shows the compromise between the robustness level and achieved time domain performance.

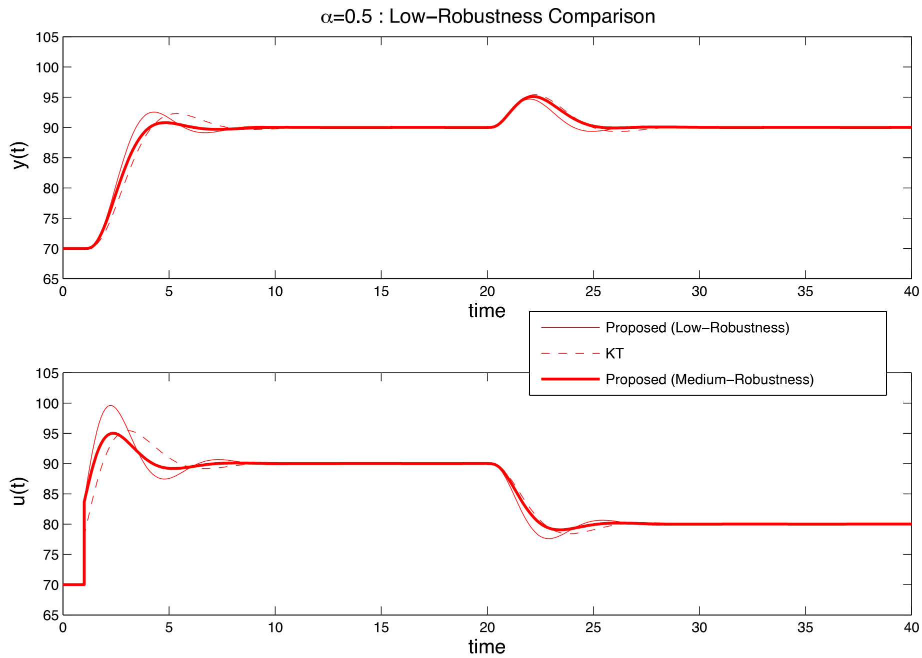

Fig. 11. Example 2—Low-Robustness tuning comparison for example 2 ($\alpha = 0.5$).

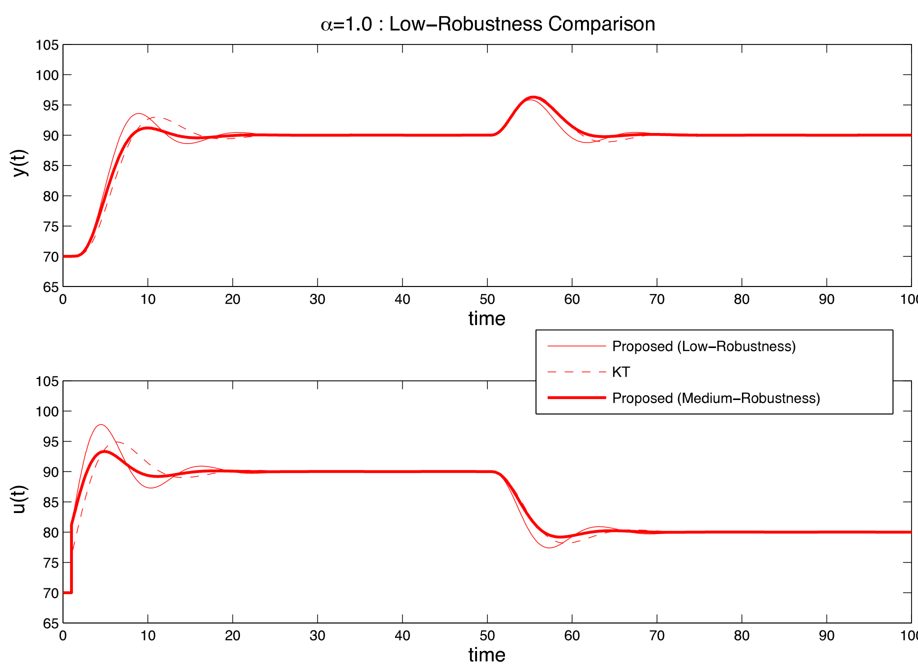

Fig. 12. Example 2—Low-Robustness tuning comparison for example 2 ($\alpha = 1.0$).

This fact suggests the intermediate (Medium-Robustness) tuning that takes the reasonable value of 1.6 as the specification for the desired M_s value and it is seen to provide a time domain performance considerably better than the High-Robustness case and along the same lines than the Low-Robustness case (and in some particular cases even better).

The Medium-Robustness tuning (22) is therefore postulated as a reasonably simple autotuning rule that provides good robustness levels and a time domain performance according to methods that are less robust.

6. Conclusions

An approach for automatic tuning of robust PI 2-DoF controller has been proposed. The method is analytically based; therefore called Analytical Robust Tuning (ART_2); and starts from a first-order-plus-dead-time controlled process model to obtain a control system that responds fast and without oscillations to a step load-disturbance, with a maximum sensitivity lower than a specified value; in order to assure robustness; and which will also show a fast non-oscillating response to a set-point step change, not requiring strong or excessive control effort variations (*smooth* control).

Given a prescribed robustness level expressed in terms of the Maximum Sensitivity value (M_s), the lowest allowable closed-loop time constant is determined. On that basis, the disturbance to output transfer function is matched and, on a second step, the control system performance to a set-point modified by an adequate selection of the two-degree-of-freedom controller

Table 4
Performance evaluation for the High-Robustness tunings.

Method	α	M_s^r	Set-point		Load-disturbance	
			IAE_r	TV_r	IAE_d	TV_d
Proposed (H)	0.25	1.311	20.363	3.152	7.436	1.062
AMIGO	0.25	1.21	40.566	2.088	11.417	1.023
$\kappa-\tau$	0.25	1.291	21.816	3.328	9.052	1.146
Proposed (M)	0.25	1.398	17.673	3.882	5.659	1.185
Proposed (H)	0.5	1.405	37.219	2.490	17.358	1.034
AMIGO	0.5	1.21	62.935	2.000	31.467	1.000
$\kappa-\tau$	0.5	1.404	40.886	2.729	20.219	1.122
Proposed (M)	0.5	1.562	34.075	3.191	13.248	1.230
Proposed (H)	1.0	1.407	91.352	2.232	48.520	1.005
AMIGO	1.0	1.24	162.143	2.000	81.071	1.000
$\kappa-\tau$	1.0	1.509	96.170	2.763	49.850	1.144
Proposed (M)	1.0	1.629	86.448	2.852	37.991	1.216

Table 5
Performance evaluation for the Low-Robustness tunings.

Method	α	M_s^r	Set-point		Load-disturbance	
			IAE_r	TV_r	IAE_d	TV_d
Proposed (L)	0.25	1.551	17.065	4.351	4.071	1.412
$\kappa-\tau$	0.25	1.465	18.580	4.173	5.221	1.338
Proposed (M)	0.25	1.398	17.673	3.882	5.659	1.185
Proposed (L)	0.5	1.865	34.417	4.614	11.219	1.656
$\kappa-\tau$	0.5	1.615	42.053	3.284	15.578	1.375
Proposed (M)	0.5	1.562	34.075	3.191	13.248	1.230
Proposed (L)	1.0	2.051	92.822	4.366	35.217	1.791
$\kappa-\tau$	1.0	1.763	106.698	3.224	43.682	1.457
Proposed (M)	1.0	1.629	86.448	2.852	37.991	1.216

set-point weighting factor β. The use of $\beta \leq 1$ values allows to decrease the servo-control response maximum overshot when very fast responses have been specified for the regulatory control. However, values larger than 1 may be generated if the system response is too slow. The resulting tuning can take any desired value for M_s as the design parameter and generate, in a parameterized way, the three controller parameters (K_c, T_i and β).

On the basis of the general approach, three different robustness levels are defined corresponding to the maximum sensitivity values of: $M_s = 1.4$, $M_s = 1.6$ and $M_s = 2.0$. Simple tuning rules are generated by considering these M_s values. The resulting autotuning rules provide all the controller parameters parameterized in terms of the model normalized dead-time allowing the user to select for a High/Medium/Low Robust closed-loop system. The proposed autotuning expressions are therefore compared with other well known tuning rules also conceived with the same robustness spirit, showing the proposed approach is able to guarantee the same robustness level with an improvement of the system time performance.

A natural extension of the presented work is to consider 2-DoF PID controllers as well as the use of second order plus time delay process models for design. In addition to the difficulty in getting PID designs with assured robustness levels there is the additional point of more complex controller and process model parameterizations. As the process model has one additional parameter it is much more difficult to find suitable forms for the controller parameters in terms of the process and problem data. This research is being carried and will be reported elsewhere.

Acknowledgments

This work has received financial support from the Spanish CICYT program under grant DPI2010-15230.

Also, the financial support from the University of Costa Rica and from the MICIT and CONICIT of the Government of the Republic of Costa Rica is greatly appreciated.

References

[1] Babb M. Pneumatic instruments gave birth to automatic control. Control Engineering 1990;37(12):20–2.
[2] Bennett S. The past of PID controllers. In: IFAC digital control: past, present and future of PID control. 2000.
[3] Cohen GH, Coon GA. Theoretical considerations of retarded control. ASME Transactions 1953;75:827–34.
[4] López AM, Miller JA, Smith CL, Murrill PW. Tuning controllers with error-integral criteria. Journal of Instrumentation Technology 1967;14:57–62.
[5] Ziegler JG, Nichols NB. Optimum settings for automatic controllers. ASME Transactions 1942;64:759–68.
[6] Martin J, Smith CL, Corripio AB. Controller tuning from simple process models. Journal of Instrumentation Technology 1975;22(12):39–44.
[7] Rivera DE, Morari M, Skogestad S. Internal model control. 4. PID controller desing. Industrial and Engineering Chemistry, Process Design and Development 1986;25:252–65.
[8] Rovira A, Murrill PW, Smith CL. Tuning controllers for setpoint changes. Instrumentation & Control Systems 1969;42:67–9.
[9] O'Dwyer A. Handbook of PI and PID controller tuning rules. London (UK): Imperial College Press; 2003.
[10] Ge M, Chiu M, Wang Q. Robust PID controller design via LMI approach. Journal of Process Control 2002;12:3–13.
[11] Toscano R. A simple PI/PID controller design method via numerical optimization approach. Journal of Process Control 2005;15:81–8.
[12] Silva G, Datta A, Battacharayya S. New results on the synthesis of PID controllers. IEEE Transactions on Automatic Control 2002;47(2):241–52.
[13] Ho M, Lin C. PID controller design for robust performance. IEEE Transactions on Automatic Control 2003;48(8):1404–9.
[14] Rivera DE. Internal model control: a comprehensive view. Tech. rep. Arizona State University: Department of Chemical, Bio and Materials Engineering, College of Engineering and Applied Sciences; 1999.
[15] García CE, Morari M. Internal model control. 1. A unifying review and some new results. Industrial and Engineering Chemistry, Process Design and Development 1982;21:308–23.
[16] Isaksson AJ, Graebe SF. Analytical PID parameter expressions for higher order systems. Automatica 1999;35:1121–30.
[17] Kaya I. Tuning PI controllers for stable process with specifications on gain and phase margings. ISA Transactions 2004;43:297–304.
[18] Skogestad S. Simple analytic rules for model reduction and PID controller tuning. Modeling, Identification and Control 2004;25(2):85–120.
[19] Arrieta O, Vilanova R. PID autotuning settings for balanced servo/regulation operation. In: 15th IEEE mediterranean conference on control and automation. MED07. 2007.
[20] Chen D, Seborg DE. PI/PID controller design based on direct synthesis and disturbance rejection. Industrial and Engineering Chemistry Research 2002; 41:4807–22.
[21] Araki M, Taguchi H. Two-degree-of-freedom PID controllers. International Journal of Control, Automation and Systems 2003;1:401–11.
[22] Araki M. On two-degree-of-freedom PID control system. Tech. rep. SICE Research Commitee on Modeling and Control Design of Real Systems. 1984.
[23] Araki M. PID control systems with reference feedforward (PID-FF control system). In: Proc. of 23rd SICE anual conference. 1984. p. 31–2.
[24] Araki M. Two-degree-of-freedom control system—I. Systems and Control 1985;29:649–56.
[25] Taguchi H, Araki M. Two-degree-of-freedom PID controllers—their functions and optimal tuning. In: IFAC digital control: past, present and future of PID control. 2000.
[26] Taguchi H, Araki M. Survey of researches on two-degree-of-freedom PID controllers. In: The 4th Asian control conference. 2002.
[27] Taguchi H, Kokawa M, Araki M. Optimal tuning of two-degree-of-freedom PD controllers. In: The 4th Asian control conference. 2002.
[28] Alfaro VM, Vilanova R, Arrieta O. Analytical robust tuning of PI controllers for first-order-plus-dead-time processes. In: 13th IEEE international conference on emerging technologies and factory automation. 2008.
[29] Åström KJ, Hang CC, Persson P, Ho WK. Towards intelligent PID control. Automatica 1992;28(1):1–9.
[30] Åström K, Hägglund T. Revisiting the Ziegler–Nichols step respose method for PID control. Journal of Process Control 2004;14:635–50.
[31] Åström KJ, Panagopoulos H, Hägglund T. Design of PI controllers based on non-convex optimization. Automatica 1998;34(5):585–601.
[32] Hang C, Cao L. Improvement of transient response by means of variable set point weighting. IEEE Transaction on Industrial Electronics 1996;4:477–84.
[33] Hägglund T, Åström K. Revisiting the Ziegler–Nichols tuning rules for PI control. Asian Journal of Control 2002;4(4):364–80.
[34] Kim DH. Tuning of 2-DOF PID controller by immune algorithm. In: Congress on evolutionary computation. CEC'02. 2002. p. 675–80.
[35] Kim DH. The comparison of characteristics of 2-DOF PID controllers and intelligent tuning for a gas turbine generating plant. Lecture Notes in Computer Science, Berlin (Heidelberg): Springer; 2004.
[36] Sugiura M, Yamamoto S, Sawaki J, Matsuse K. The basic characteristics of two-degree-of-freedom PID position controller using a simple design method for

linear servo motor drives. In: 4th international workshop on advanced motion control. AMC'96-MIE. 1996. p. 59–64.
[37] Zhang J-G, Liu Z-Y, Pei R. Two degree-of-freedom PID control with fuzzy logic compensation. In: First international conference on machine learning and cybernetics. 2002. p. 1498–501.
[38] Åström K, Hägglund T. PID controllers: theory, design and tuning. Research Triangle Park (NC, USA): Instrument Society of America; 1995.
[39] Alfaro VM. Analytical tuning of optimum and robust PID regulators. Master's thesis. Escuela de Ingeniería eléctrica. Universidad de Costa Rica. 2006 [in Spanish].
[40] Alfaro VM. Low-order models identification from process reaction curve. Ciencia y Tecnología (Costa Rica) 2006;24(2):197–216 [in Spanish].
[41] Åström K, Hägglund T. Advanced PID control. ISA-The Instrumentation Systems, and Automation Society; 2006.