Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Model-based PID tuning for high-order processes: when to approximate

Antonio Visioli

Abstract— **In this paper different methods for the (model-based) tuning of Proportional-Integral-Derivative (PID) controllers for high-order processes are analyzed and compared. In particular, two approaches in the internal model control framework are addressed and discussed: (i) the (high-order) controller that results from considering the high-order process model is reduced through a Maclaurin series expansion in order to obtain a PID controller; (ii) the process model is first reduced in order to obtain naturally a PID controller (different techniques are considered for this purpose). Simulation results regarding different process dynamics are evaluated in order to draw general conclusions.**

I. INTRODUCTION

Proportional-Integral-Derivative (PID) controllers are the most commonly adopted controllers in industrial settings due to the cost/benefit ratio they are able to achieve. Because of their simple structure, they are particularly suited to control processes whose dominant dynamics is of first or second order [1]; however they are employed also for high-order processes, because of the economic advantages provided by their standardization. However, despite many tuning rules having been devised in the past assuming a first-order plus dead time (FOPDT) or a second-order plus dead time (SOPDT) process model [2], the case where a high-order process dynamics (which is assumed to be known) is explicitly addressed has received a significant attention only recently [3]-[6]. Actually, it has to be taken into account that in many cases, an apparent time delay is indeed due to the presence of a high-order dynamics [7].

It is realized that, because of the relative low-order of the controller, a model reduction has necessarily to be performed. In this context, two approaches can be followed: (i) design a model-based high-order controller by considering the (full) high-order dynamics of the process and then reduce the controller to a PID form; (ii) reduce first the process model to an appropriate low-order form so that a model-based controller results directly to be in PID form. Actually, despite the fact that it is obvious that the procedure for the determination of the process model plays a key role in the controller tuning and therefore in the control system performance, this aspect has been often overlooked in the literature [7].

In this paper the two previously-mentioned approaches are thoroughly analyzed and compared in the internal model control (IMC) framework [8], which has been extensively adopted for the purpose of PID controller tuning, in order to assess their advantages and disadvantages from the point of view of the achievable performance and of the ease of use. The paper is organized as follows. In Section 2 the different considered approaches are reviewed and their characteristics are compared. Section 3 is devoted to the presentation of simulation results regarding processes with different dynamics. These results are discussed in Section 4 and conclusions are drawn in Section 5.

This work was partially supported by MIUR scientific research funds

A. Visioli is with Dipartimento di Elettronica per l'Automazione, University of Brescia, Italy, antonio.visioli@ing.unibs.it

II. METHODOLOGIES

A. IMC design: generalities

The internal model control methodology [8] has been widely adopted for the purpose of PID controller tuning (though, being based on a pole-zero cancellation approach it is not suitable for lag-dominant processes subject to load disturbances [9], [10]). Indeed, it provides the user with a desirable feature as a tuning parameter that handles the trade-off between robustness and aggressiveness of the controller. In a general form, the IMC control design can be described as follows. Consider a standard unity feedback control system (see Figure 1) in which the (stable) process to be controlled is described by the model:

$$G(s) = p_m(s)p_a(s) \tag{1}$$

where $p_a(s)$ is the all-pass portion of the transfer function containing all the nonminimmum phase dynamics ($p_a(0) = 1$). The controller transfer function is chosen as

$$C(s) = \frac{f(s)p_m^{-1}(s)}{1 - f(s)p_a(s)} \tag{2}$$

in which

$$f(s) = \frac{1}{(\lambda s + 1)^r} \tag{3}$$

is the IMC filter where λ is the adjustable time constant and r is an appropriate order so that the controller is realizable. It has to be noted that the nominal closed-loop transfer function, i.e. the transfer function from the set-point signal y_{sp} and the process output y, is

$$T(s) = \frac{p_a(s)}{(\lambda s + 1)^r}. \tag{4}$$

This makes clear the role of the free design parameter λ in selecting the desired closed-loop dynamics (and therefore in handling the trade-off between robustness and aggressiveness, as unavoidable mismatches between the true process dynamics and its model have to be taken into account).

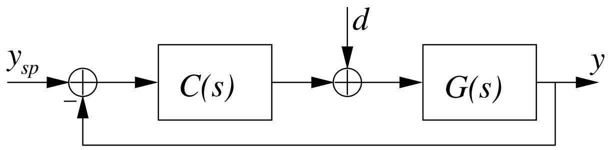

Fig. 1. The control scheme.

Obviously, in general, the resulting controller is not in PID form, i.e.:

$$C(s) = K_p \left(\frac{T_i s + 1}{T_i s} \right) \frac{T_d s + 1}{T_f s + 1}, \tag{5}$$

if the series ("interacting") form is considered, or

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \frac{1}{T_f s + 1}, \tag{6}$$

if the ideal ("non-interacting") form is implemented; K_p is the proportional gain, T_i and T_d are the integral and derivative time constants respectively, and T_f is the filter time constant. A PID controller results if the process model has one positive zero and two poles (note that this results if a FOPDT transfer function is considered and a first order Padè approximation is adopted for the delay term [11]), whilst a PI controller results if the plant has a simple first order dynamics. Thus, if a high order process model is considered, this must be reduced to this suitable form before applying the IMC design or, alternatively, the resulting high-order controller has to be subsequently reduced to a PID form.

B. PID tuning with the Skogestad's half rule

The method proposed by Skogestad in [6] considers a process model reduction based on the so-called "half rule", which states that the largest neglected (denominator) time constant is distributed evenly to the effective dead time and the smallest retained time constant. In practice, given a high order transfer function, each numerator term $(T_0 s + 1)$ with $T_0 > 0$ is first simplified with a denominator term $(\tau_0 s + 1)$, $\tau_0 > 0$ using the following rules:

$$\frac{T_0 s + 1}{\tau_0 s + 1} \approx \begin{cases} T_0/\tau_0 & \text{for} \quad T_0 \geq \tau_0 \geq \theta \\ T_0/\theta & \text{for} \quad T_0 \geq \theta \geq \tau_0 \\ 1 & \text{for} \quad \theta \geq T_0 \geq \tau_0 \\ T_0/\tau_0 & \text{for} \quad \tau_0 \geq T_0 \geq 5\theta \\ \frac{(\tilde{\tau}_0/\tau_0)}{(\tilde{\tau}_0 - T_0)s+1} & \text{for} \quad \tilde{\tau}_0 \stackrel{\text{def}}{=} \min(\tau_0, 5\theta) \geq T_0 \end{cases} \tag{7}$$

where θ is the final effective delay (to be determined subsequently). It has to be noted that τ_0 is normally chosen as the closest larger denominator time constant ($\tau_0 > T_0$), except when a larger denominator time constant does not exist or there is a smaller denominator time constant closer to T_0; this is true if the ratio between T_0 and the smaller denominator time constant is less than the ratio between the larger denominator time constant and T_0 and less than 1.6 at the same time.

Once this procedure has been terminated for all the positive numerator time constants, the process transfer function is in the following form:

$$\tilde{G}(s) = \frac{\prod_j (-T'_{j0} s + 1)}{\prod_i (\tau_{i0} + 1)} e^{-\theta_0 s} \tag{8}$$

where $T'_{j0} > 0$ and the time constants are ordered according to their magnitude. Then, a SOPDT transfer function

$$G(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-\theta s} \tag{9}$$

is obtained by applying the half rule, i.e. by setting

$$\tau_1 = \tau_{10}, \qquad \tau_2 = \tau_{20} + \frac{\tau_{30}}{2}, \tag{10}$$

$$\theta = \theta_0 + \frac{\tau_{30}}{2} + \sum_{i \geq 4} \tau_{i0} + \sum_j T'_{j0}. \tag{11}$$

It appears that, being the rules (7) based on the final apparent time delay θ, in the first part of the algorithm there is the need to guess this final value and to iterate in case at the end the result is incorrect.

Once the SOPDT process model is obtained, the PID parameters are determined by applying the IMC design procedure (and by approximating the delay term as $e^{-\theta s} = 1 - \theta s$) and by possibly modifying the value of T_i in order to address the case of lag-dominant processes [6] (note that this fact in not of concern in the examples presented in Section III). It results that the PID parameters in (5) are selected as

$$K_p = \frac{\tau_1}{k(\lambda + \theta)}, \quad T_i = \tau_1, \quad T_d = \tau_2, \quad T_f = 0.01 T_d. \tag{12}$$

Note that the conversion of the tuning rule (12) for the PID controller in the ideal form is straightforward and a recommended choice for the desired closed-loop time constant is $\lambda = \theta$ [6].

Summarizing, the method is based on simple, easy to remember, tuning rules. However, the possible iterations in the model reduction algorithm make the overall procedure somewhat difficult to automate.

C. Isaksson and Graebe's analytical PID design

The technique proposed by Isaksson and Graebe in [5] is also based on a suitable process model reduction before applying the IMC design. The model reduction is performed as followa. Let the initial (high-order) process model be described by the transfer function

$$\tilde{G}(s) = \frac{B(s)}{A(s)} \tag{13}$$

Then, the numerator and denominator polynomials are considered separately and the polynomials $B_1(s)$ and $A_1(s)$ that retain only the slowest roots are determined. Subsequently, the polynomials $B_2(s)$ and $A_2(s)$ that retain the low-order coefficients are calculated. Finally, the reduced-order model is obtained as

$$G(s) = \frac{\frac{1}{2}(B_1(s) + B_2(s))}{\frac{1}{2}(A_1(s) + A_2(s))} \tag{14}$$

By choosing $B_1(s)$ and $B_2(s)$ of first order and $A_1(s)$ and $A_2(s)$ of second order and by subsequently applying the IMC design (with a first order filter (3)), a PID controller (6) naturally arises. If there are no zeros, two solutions can be exploited: (i) a second order denominator is calculated in the reduction procedure and a second order filter (3) is applied in the IMC design, yielding to a PID controller; (ii) a first order denominator is calculated in the reduction procedure and a first order filter (3) is applied in the IMC design, yielding to a PI controller.

It has to be noted that, differently from the method described in subsection II-B, the case of complex conjugate roots is also addressed in [5], but it will not be considered hereafter (see Section III).

Summarizing, the Isaksson and Graebe's method can be easily automated, although it is not explicitly based on tuning formulae.

D. Model approximation with step response data

Usually, for the purpose of PID tuning, a FOPDT or SOPDT process model is obtained by means of step response data. In this context, the least-squares based method proposed in [12] is considered in this paper. The nice feature of this technique is that it is capable of providing a SOPDT process model without any iteration and, being based on process output integrals, it is very robust to measurement noise.

Starting from the identified model, the tuning rule (12) has been adopted. However, for this purpose, the obtained SOPDT model must have real poles and no zeros. Thus, if a zero is determined the half rule is then adopted, whilst if complex conjugate poles occur, a FOPDT model (obtained with the same identification method) is actually employed. In this latter case a PI controller results.

It is worth stressing that, given a high-order process model, the reduced order transfer function based on step response data can be obtained without the need of time consuming and costly experimental results, as a simulation can be performed [13]. Doing so, the overall procedure can be easily automated, although a relatively significant computational power is actually necessary.

E. PID tuning by means of a Maclaurin series expansion

The methods described in the previous subsections are based on the reduction of the process model before applying the IMC design. Conversely, it is possible to apply the IMC procedure described in subsection II-A by considering the full process dynamics and then reduce the obtained high-order controller to a PID controller form. For this purpose, a Maclaurin series expansion can be employed. The expression of the resulting controller can be always written as [3], [4]:

$$C(s) = \frac{r(s)}{s} \tag{15}$$

and expanding $C(s)$ in a Maclaurin series in s we obtain:

$$C(s) = \frac{1}{s}\left[r(0) + r'(0)s + \frac{r''(0)}{2}s^2 + \cdots\right] \tag{16}$$

It turns out that the first part of the series expansion contains a proportional term, an integral term and a derivative term and therefore, if the high-order terms are neglected, a PID controller (6) results (a first order filter can be easily added in order to make the controller proper and its time constant can be selected sufficiently small so that its dynamics is not significant).

Hence, the overall procedure can be easily automated, although it is not based on tuning formulae and its computational burden is somewhat considerable. However, it has to be stressed that a wrong choice of the design parameter λ can result in the overall control system being unstable (see Section IV). Although this can be easily checked before applying the controller, it can be considered as a major drawback of the method.

III. SIMULATION RESULTS

In order to analyze and compare the different methodologies, the following processes with high order dynamics have been considered:

$$G_1(s) = \frac{(15s+1)^2(4s+1)(2s+1)}{(20s+1)^3(10s+1)^3(5s+1)^3(0.5s+1)^3}, \tag{17}$$

$$G_2(s) = \frac{(-0.3s+1)(0.08s+1)}{(2s+1)(s+1)(0.4s+1)(0.2s+1)(0.05s+1)^3}, \tag{18}$$

$$G_3(s) = \frac{(-45s+1)}{(20s+1)^3(18s+1)^3(5s+1)^3}\cdot \cdot\frac{(4s+1)}{(10s+1)^2(16s+1)(14s+1)(12s+1)}, \tag{19}$$

$$G_4(s) = \frac{1}{(s+1)^4}, \tag{20}$$

$$G_5(s) = \frac{1}{(s+1)^8}, \tag{21}$$

$$G_6(s) = \frac{1}{(s+1)^{20}}. \tag{22}$$

The main characteristics of the processes are summarized in Table I. It has to be noted that transfer functions $G_1(s)$ and $G_3(s)$ have been taken from [14], $G_2(s)$ from [6] and $G_4(s) - G_6(s)$ are representative of typical industrial processes [15], [16].

The reduced order models that have been adopted for the PI(D) tuning are reported in Table II. Note that two transfer functions might occur for the Isaksson and Graebe's technique, whereas the process dynamics has no zeros, as explained in subsection II-C. Indeed, the first one yields to a PID controller (with a second order IMC filter), whilst the second one yields to a PI controller (with a first order IMC filter). Besides, whereas a FOPDT transfer function is reported for the step response based method, this means that the resulting SOPDT model has complex conjugate poles and has not therefore been employed (see subsection II-D).

In order to make a fair comparison, for each method and for

$G_1(s)$	Minimum phase dynamics
$G_2(s)$	Presence of a nondominant positive zero
$G_3(s)$	Presence of a dominant positive zero
$G_4(s)$	Minimum phase dynamics with a small number of coincident poles
$G_5(s)$	Minimum phase dynamics with a medium number of coincident poles
$G_6(s)$	Minimum phase dynamics with a high number of coincident poles

TABLE I

MAIN CHARACTERISTICS OF THE CONSIDERED PROCESSES.

each process, the value of λ that minimizes the integrated absolute error, defined as

$$IAE = \int_0^\infty |e(t)|dt, \tag{23}$$

have been selected for both the set-point and the load disturbance step responses (i.e. a unit step has been applied on signals y_{sp} and d separately, see Figure 1). For those methods that do not provide the value of the filter time constant T_f explicitly, this has been selected in such a way its dynamics is negligible.
The resulting values of the integrated absolute error and the corresponding optimal values of λ are reported in Table III. Note again that for the Isaksson and Graebe's method two cases (PID and PI control) can emerge, depending on the fact that a second order or first order IMC filter respectively has been adopted (begin no zeros in the process to be controlled). Analogously, a PID or a PI controller results from the technique based on the step response, depending on the use of a SOPDT model or a FOPDT model (the latter in case the identified SOPDT model has complex conjugate poles). In order to evaluate better the results, the phase margin PM and the open loop cutoff frequency ω_c for each considered control system are reported in Table IV. Finally, the resulting (set-point and load) unit step responses are plotted in Figures 2-7. For the sake of clarity, the process responses obtained with a PI controller resulting from the Isaksson and Graebe's method are not reported.

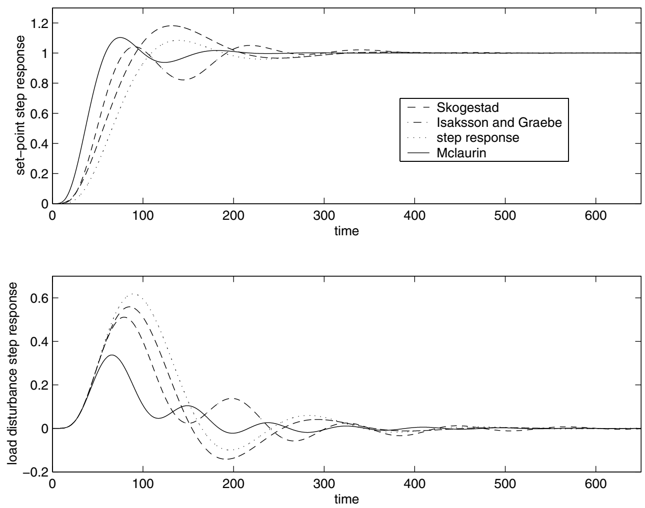

Fig. 2. Optimal responses for $G_1(s)$.

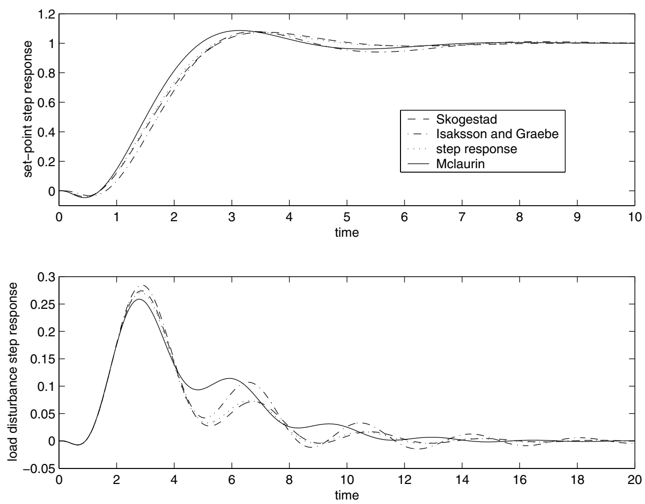

Fig. 3. Optimal responses for $G_2(s)$.

IV. DISCUSSION

From the results obtained it appears that the approach based on the Maclaurin series expansion provides in general the best performance, both for the set-point following and the load disturbance rejection task. This is due to its capability of providing a higher open loop cutoff frequency without decreasing the phase margin with respect to the other methods. From another point of view, this means that in the set-point step responses a low rise time is achieved without impairing the overshoot and in the load disturbance step responses, a low peak error results without the occurrence of significant oscillations.
It turns out that it is better to reduce the model of the controller than that of the plant, as the approximation introduced by adopting only the first three terms of the series expansion is not detrimental in the range of frequencies that is significant for the considered control system. However this is true only if an appropriate value of λ is selected. Indeed, a wrong choice of λ might yield the system to instability. For example, for system $G_3(s)$, if $\lambda \leq 6$ or $\lambda \geq 162$ the resulting closed loop system is unstable and, in any case, if $\lambda \geq 20$ at least one of the PID parameter results to be less than zero. Actually, it might happen that a quite narrow range of values for λ is suitable. Despite the fact that an inappropriate value of λ can be easily recognized during the design phase, this can be considered as a major drawback of the method, which has been overlooked in the literature. Indeed, this makes the overall design more complicated and, most of all, the physical meaning of the filter time constant, which should handle the trade-off between aggressiveness and robustness and control activity of the control system, is somewhat lost.
It has to be also noted that the optimal values of λ are significantly different between the considered methodology, although it appears, as expected, that in general a higher-order filter (i.e. for the Maclaurin series based technique or when a PID controller is adopted instead of a PI controller in the Isaksson and Graebe's method) implies a lower value

Process	Skogestad	Isaksson and Graebe	step response
$G_1(s)$	$\dfrac{1}{(20s+1)(15s+1)}e^{-35.5s}$	$\dfrac{25.5s+1}{2642s^2+73.25s+1}$	$\dfrac{1}{44.46s+1}e^{-27.38s}$
$G_2(s)$	$\dfrac{1}{(2s+1)(1.2s+1)}e^{-0.77s}$	$\dfrac{-0.26s+1}{3.21s^2+3.38s+1}$	$\dfrac{1}{2.48s^2+3.17s+1}e^{-0.79s}$
$G_3(s)$	$\dfrac{1}{(30s+1)(20s+1)}e^{-180s}$	$\dfrac{-42s+1}{8564s^2+115.7s+1}$	$\dfrac{1}{106.6s+1}e^{-127.7s}$
$G_4(s)$	$\dfrac{1}{(1.5s+1)(s+1)}e^{-1.5s}$	$\dfrac{1}{2.25s^2+3s+1}\;\dfrac{1}{2.5s+1}$	$\dfrac{1}{2.71s+1}e^{-1.38s}$
$G_5(s)$	$\dfrac{1}{(1.5s+1)(s+1)}e^{-5.5s}$	$\dfrac{1}{14.52s^2+5.02s+1}\;\dfrac{1}{4.51s+1}$	$\dfrac{1}{4.24s+1}e^{-3.88s}$
$G_6(s)$	$\dfrac{1}{(1.5s+1)(s+1)}e^{-17.5s}$	$\dfrac{1}{95.98s^2+11.40s+1}\;\dfrac{1}{10.70s+1}$	$\dfrac{1}{7.76s+1}e^{-12.72s}$

TABLE II

RESULTING MODEL REDUCTIONS FOR THE DIFFERENT METHODOLOGIES.

Process	Task	Skogestad	Isaksson and Graebe		step response		Maclaurin
			PID	PI	PID	PI	
$G_1(s)$	setpoint	71.38 (9.6)	63.08 (54.4)			73.49 (37.2)	42.07 (4.4)
	load	54.45 (0.01)	48.13 (39.1)			60.16 (20.8)	25.35 (2.8)
$G_2(s)$	setpoint	1.852 (0.83)	1.963 (1.22)		1.819 (0.78)		1.783 (0.18)
	load	0.869 (0.03)	0.985 (0.37)		0.863 (0.01)		0.986 (0.07)
$G_3(s)$	setpoint	376.5 (120.1)	251.7 (122.3)			301.1 (159.0)	231.6 (11.5)
	load	384.3 (115.2)	253.6 (113.0)			307.7 (153.6)	233.5 (10.9)
$G_4(s)$	setpoint	3.133 (0.49)	2.722 (0.87)	4.099 (3.44)		3.998 (2.06)	2.041 (0.41)
	load	1.952 (0.001)	1.073 (0.39)	3.183 (2.33)		3.063 (0.96)	0.876 (0.19)
$G_5(s)$	setpoint	11.58 (3.01)	7.776 (2.78)	9.550 (8.83)		9.662 (4.96)	6.561 (0.74)
	load	10.95 (2.06)	6.401 (2.34)	8.880 (7.88)		9.030 (4.00)	5.394 (0.61)
$G_6(s)$	setpoint	36.86 (11.2)	23.01 (8.83)	25.84 (24.6)		27.36 (12.2)	19.85 (0.90)
	load	36.49 (10.5)	21.88 (8.42)	25.22 (23.8)		26.95 (11.4)	18.81 (0.86)

TABLE III

SUMMARY OF THE OPTIMAL IAE'S (AND CORRESPONDING VALUES OF λ).

Process	Task	Skogestad		Isaksson and Graebe				step response				Maclaurin	
				PID		PI		PID		PI			
		PM	ω_c	PM	ω_c	PM	ω_c	PM	ω_c	PM	ω_c	PM	ω_c
$G_1(s)$	setpoint	50.76	0.020	77.9	0.019					60.9	0.017	61.6	0.031
	load	42.01	0.025	43.3	0.038					47.6	0.227	27.1	0.062
$G_2(s)$	setpoint	58.9	0.658	62.5	0.630			59.1	0.671			58.9	0.733
	load	23.0	1.337	21.4	1.404			23.6	1.323			23.5	1.509
$G_3(s)$	setpoint	55.7	$3.33 \cdot 10^{-3}$	55.8	$4.65 \cdot 10^{-3}$					62.7	$3.72 \cdot 10^{-3}$	61.2	$4.84 \cdot 10^{-3}$
	load	55.1	$3.39 \cdot 10^{-3}$	54.8	$4.86 \cdot 10^{-3}$					62.1	$3.80 \cdot 10^{-3}$	61.3	$5.02 \cdot 10^{-3}$
$G_4(s)$	setpoint	50.7	0.459	53.8	0.544	59.4	0.306			60.4	0.316	58.5	0.707
	load	41.1	0.576	17.3	1.157	41.8	0.447			52.6	0.380	12.8	1.487
$G_5(s)$	setpoint	54.2	0.114	56.5	0.162	63.2	0.122			62.3	0.120	64.5	0.175
	load	50.0	0.127	54.5	0.193	58.9	0.139			58.0	0.136	62.8	0.232
$G_6(s)$	setpoint	55.4	0.034	56.6	0.051	64.8	0.044			60.4	0.041	61.9	0.058
	load	54.6	0.035	56.2	0.053	63.6	0.046			59.3	0.043	62.0	0.061

TABLE IV

SUMMARY OF THE RESULTING PHASE MARGINS PM (IN DEGREES) AND CUTOFF FREQUENCIES ω_c (IN RAD/S).

of λ (and the load disturbance rejection task requires a lower value of λ than the set-point following task).

From the results obtained, it also appears that the Isaksson and Graebe's method provides in general better performance that the Skogestad's one and, as expected, the PID controller is better than the PI controller in the context of the technique described in subsection II-C. Note, however, that the tuning rules (12) have been conceived with the aim of being applicable to a wide range of processes and of being easy to memorize. As for the method based on the step response data, it can be deduced that in general it provides worse performance than the Isaksson and Graebe's one whilst no general conclusions can be drawn with respect to the Skogestad's method.

V. CONCLUSIONS

In order for a method to be adopted in the industrial context, its pros and cons should be clearly pointed out. For this purpose, in this paper different model based PID tuning methods for high-order processes have been analyzed and compared. As a main result, it has been found that the best performances are obtained by reducing the controller

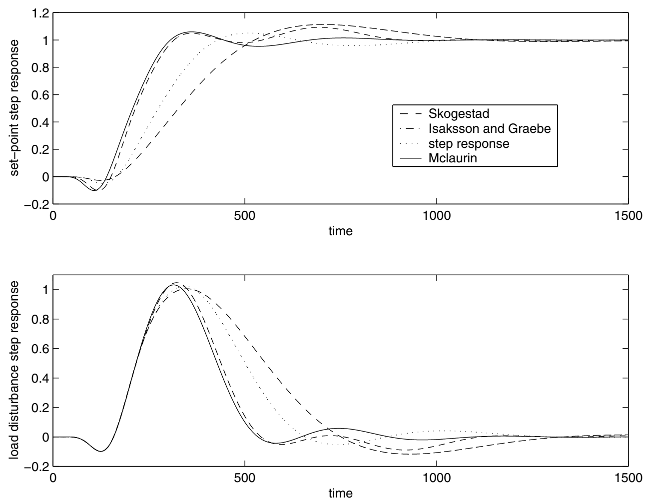

Fig. 4. Optimal responses for $G_3(s)$.

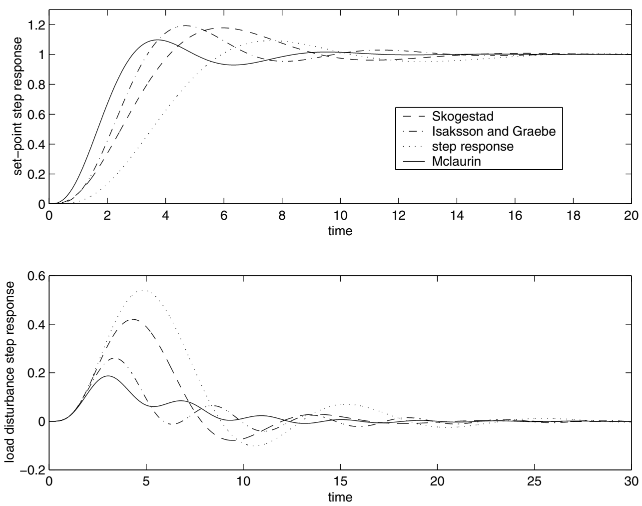

Fig. 5. Optimal responses for $G_4(s)$.

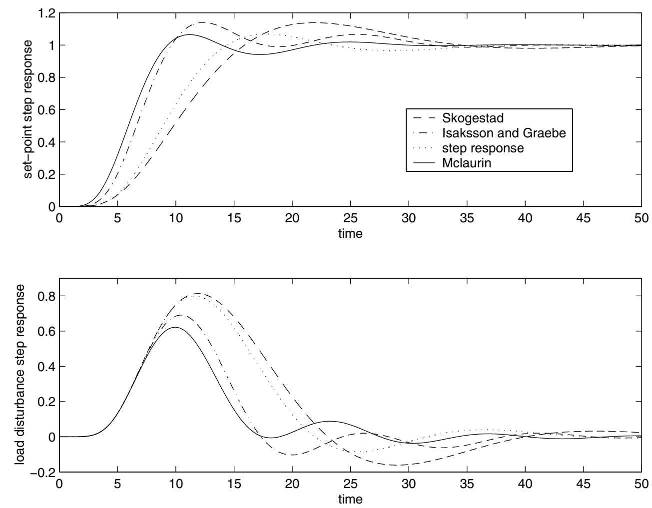

Fig. 6. Optimal responses for $G_5(s)$.

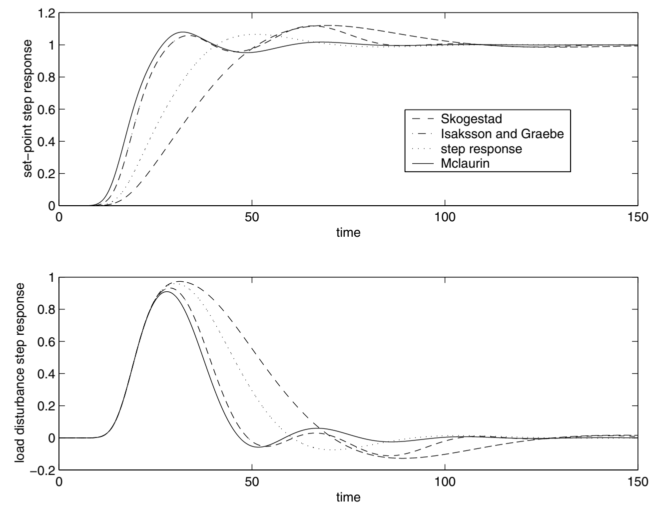

Fig. 7. Optimal responses for $G_6(s)$.

model instead of the process one, although a careful choice of the IMC filter time constant is needed. The computational complexity of the methods considered has also been addressed in order to evaluate the ease of their implementation in Distributed Control Systems.

REFERENCES

[1] K. J. Åström, T. Hägglund, *PID controllers: theory, design and tuning*, ISA Press, Research Triangle Park (NC), 1995.

[2] A. O'Dwyer, *Handbook of PI and PID controller tuning rules*, World Scientific Press, 2003.

[3] Y. Lee, S. Park, M. Lee, C. Brosilow, "PID controller tuning for desired closed-loop responses for SI/SO systems", *AIChE Journal*, Vol. 44, No. 1, pp. 106-115, 1998.

[4] Y. Lee, S. Park, M. Lee, "PID controller tuning to obtain desired closed loop responses for cascade control systems", *Industrial Engineering and Chemistry Research*, Vol. 37, pp. 1859-1865, 1998.

[5] A. J. Isaksson, S. F. Graebe, "Analytical PID parameter expressions for higher order systems", *Automatica*, Vol. 35, pp. 1121-1130, 1999.

[6] S. Skogestad, "Simple analytic rules for model reduction and PID controller tuning", *Journal of Process Control*, Vol. 13, pp. 291-309, 2003.

[7] A. Leva, "Autotuning process controller with improved load disturbance rejection", *Journal of Process Control*, Vol. 15, pp. 223-234, 2005.

[8] M. Morari, E. Zafiriou, *Robust process control*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[9] C. Scali, D. Semino, "Performance of optimal and standard controllers for disturbance rejection in industrial processes", *Proc. IEEE Int. Conf. on Ind. Elec., Cont. and Instr. IECON'91*, pp. 2033-2038, 1991.

[10] F. G. Shinskey, *Process Control Systems - Application, Design, and Tuning*, McGraw-Hill, New York, USA, 1996.

[11] D. E. Rivera, S. Skogestad, M. Morari, "Internal model control. 4. PID controller design", Ind. Eng. Chem. Proc. Des. Dev., Vol. 25, pp. 252-265, 1986.

[12] Q.-G. Wang, X. Guo, Y. Zhang, "Direct identification of continuous time delay systems from step response data", *Journal of Process Control*, Vol. 11, pp. 531-542, 2001.

[13] B. Huang, "A pragmatic approach towards assessment of control loop performance", *International Journal of Adaptive Control and Signal Processing*, Vol. 17, pp. 589-608, 2003.

[14] L. Wang, W. R. Cluett, *From plant data to process control - ideas for process identification and PID design*, Taylor and Francis, London, Uk, 2000.

[15] K. J. Åström, T. Hägglund, "Benchmark systems for PID control", *Preprints IFAC Workshop on Digital Control PID'00*, pp. 181-182, 2000.

[16] F. G. Shinskey, "PID-deadtime control of distributed processes", *Preprints IFAC Workshop on Digital Control PID'00*, pp. 14-18, 2000.