Using robust control methods for industrial PID autotuning

ALBERTO LEVA, ANTONIO M. COLOMBO
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Via Ponzio, 34/5 – 20133 Milano
ITALY

Abstract: - This paper describes a possible way for using some results of the robust control theory for the automatic tuning of classical industrial regulators, namely of the PID type. After a brief explanation of the proposed approach, the paper concentrates on describing a PID autotuning tool built in accordance to it. In particular, attention is focused on the autotuner's implementation and operation, while the underlying theoretical background is simply sketched out to permit the necessary degree of comprehension and for its complete illustration the interested reader is conveniently referred to more specific works. A brief example of the autotuner's operation is also reported.

Key-words: - Autotuning; PID control; Industrial control; Process control; Robust control.

1. Introduction

The research interest on PID autotuning has been increasing significantly over the last years, both under the methodological and application-oriented points of view [2, 3, 6]. As a result, several industrial products now encompass autotuning capabilities and their utilization is becoming more and more common and accepted.

In this research, robust control techniques are nowadays beginning to play an important role, due to their capability of coping with modeling uncertainties and mismatches [5]. This makes them interesting because autotuners normally rely on very simple models of the process under control [1, 11, 12], which are easy to parametrize from I/O data [17], lead to simple tuning rules [9, 10] but also likely to produce a remarkable mismatch [4, 5]. As a consequence, either the tuning rules must be quite conservative (thus sacrificing performance) or the mismatch must somehow be considered, so as to reduce conservatism to a minimum. Robust control techniques can help to this end because, if properly used, they can provide a measure of "how much conservatism" is required: simply speaking, if an autotuner is fed with a simplified process model but also with some information on how big the model error is, it is easier to ensure e.g. the required stability properties with the minimum performance loss [13].

However, the application of robust control methods in industrial autotuners is still far from maturity [7]. The reasons for this are complex and involve several phenomena, some methodological and some related to the problem of making innovative approaches acceptable for the industrial community. Sticking for brevity to the first ones, it is apparent that several results providing insight into the behaviour of PID regulators (e.g. the arising and effects of cancellation errors, the dependency of stability and performance indexes on the design parameters and so forth) are seldom taken into account in the industrial domain, while the development of autotuning methods in the academic literature often neglects some very important aspects of industrial control devices (e.g. the set point weighting and the additional pole required for obtaining a proper controller). Neglecting all these facts, i.e. tuning an industrial PID with rules conceived mainly for its "textbook" transfer function, is a feasible approach in many cases but cannot be done if robust control techniques are to be used, as will be sketched out in the following. In one word, anyway, there are some inherent difficulties in transmitting the results of the robust control theory to the application domain.

The aim of this work is to propose a framework for setting up a PID autotuner making effective use of some of these results, and to back up this proposal by describing an industrial autotuner constructed along this reasoning path. The paper is organized as follows. Section 2 describes the proposed approach in general terms, while in section 3 this is illustrated by applying it to the design of a PID autotuner. Carrying on, section 4 reports some practical considerations concerning the implementation and use of the autotuner, thus the implications they have on the usefulness and limitations of the proposed approach, whose possible generalization is dealt with briefly in section 5. Sections 6 and 7 report respectively some simulations and an application example, namely the complete sequence of operations required for tuning a PID loop. Finally, in section 7 some simple concluding remarks are given.

2. The proposed approach

Typically, the synthesis of PID regulators via robust control methods relies on the H_∞ approach, i.e. the regulator is parametrized on a simplified model of the process and subsequently detuned on the basis of a conservative estimate of the model error magnitude to fulfil a frequency domain ∞-norm criterion.
The first key point of the tuning approach proposed herein is to couple this idea to the exploitation of the 2-dof structure that industrial PIDs exhibit due to the set point weights. This suggests a two-step tuning approach particularly suited for the use of robust control methods: the PID is split into a feedback block, devoted to ensuring stability and disturbance rejection, and a feedforward one, devoted to set point tracking, which are tuned in sequence.
The second point of the proposed approach is how it addresses the problem of gathering the required process information. The basic idea is to employ a parametric process model but a nonparametric description of the model error, which by the way is clearly the maximum information available from experimental data: adopting this policy allows to separate the three problems of parametrizing a process model, estimating the model error and synthesizing the regulator completely. This is very important for constructing autotuners in a *modular* way, i.e. so that a standardized synthesis policy can be easily coupled to different model identification methods for fitting the needs of different application domains. However, this separation is not possible if a parametric description of the model error is adopted.
A third point, and maybe the most important, is that the proposed synthesis process is fed with model error information *explicitly*. This is not so common in autotuners, even in the literature, due to the problems evidenced e.g. in [8] or [19] but is crucial for obtaining a really "robust" autotuner. Though this matter is very important, space limitations will oblige here to omit theoretical aspects almost completely: however the reader can refer to [15] and [16], where they are explained in detail.
Finally, in developing the proposed approach another useful result has been found. The two-step tuning procedure naturally suggested by the industrial PID structure, once exploited (simply speaking for the moment) so as to tune its feedback part for stability and the feedforward one for setpoint tracking, has led to the individuation of a couple of design parameters which allow good control on the autotuner's operation, while their meaning is quite easy to understand. This is a very interesting fact, because the majority of autotuners either have no design parameters at all (which is a safe choice but allows no tailoring of their operation) or allow the user to set some parameters which normally are considered cumbersome to interpret. As a result most users leave those parameters to their default values, which have been decided with no reference to any real-world operating condition. Conversely, as will be shown in the following, the presented autotuner does not suffer of this limitation.

3. The autotuner

In order to illustrate the proposed approach to the use of robust control in PID autotuning, this section presents an autotuner built in accordance to it. This autotuner synthesizes an industrial PID in the ISA form for stable processes on the basis of a single open-loop step test, and the user is requested to specify two parameters: one related to the required degree of robustness and the other to the closed-loop settling time. The ISA-PID control law is expressed as

$$u = K\left[(br-y) + \frac{1}{sT_i}(r-y) + \frac{sT_d}{1+sT_d/N_d}(cr-y)\right] \qquad (1)$$

where r, y and u are the set-point, the process output and the control variable, K is the controller gain, T_i and T_d are the integral and derivative time and N_d is the ratio between T_d and the time constant of an additional pole required for obtaining a proper controller. Due to the set-point weights b and c in the proportional and derivative actions, the control law (1) corresponds to a 2 d.o.f. controller, as shown in Fig. 1

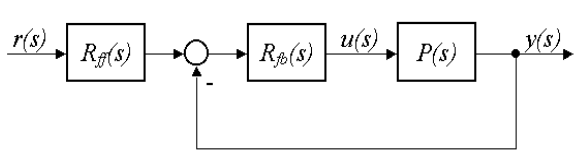

Fig. 1 : Control system with an ISA-PID.

where the feedforward and feedback blocks (R_{ff} and R_{fb} respectively) have the following transfer functions:

$$R_{ff}(s) = \frac{R_{ffN}(s)}{R_{ffD}(s)} = \frac{1+s(bT_i+T_d/N_d)+s^2T_iT_d(c+b/N_d)}{1+s(T_i+T_d/N_d)+s^2T_iT_d(1+1/N_d)}$$
$$R_{fb}(s) = \frac{R_{fbN}(s)}{R_{fbD}(s)} = \frac{1+s(T_i+T_d/N_d)+s^2T_iT_d(1+1/N_d)}{sT_i+s^2T_iT_d/N_d} \qquad (2)$$

Like any other 2 d.o.f. controller, the ISA-PID can be tuned by a two step procedure: first R_{fb} is synthesized so as to ensure asymptotic stability and good disturbance rejection, and then R_{ff} can be designed with the sole aim of improving the set-point response. It is worth noting that in the block diagram representation of Fig. 1 R_{fb} has the form of a standard, 1 d.o.f. PID controller, thus it can be tuned by using any of the methods available in the literature. The autotuner presented herein adopts this approach, and the tuning procedure can be summarized as follows. First a record of the step response of the process under control is obtained and a model in the form

$$Pn(s) = \mu \frac{e^{-s\tau}}{1+sT} \qquad (3)$$

is fit to this response by the method of areas. On the basis of the response and of the model, an overestimate of the additive model error is computed by using the technique illustrated in [16]. Subsequently, an IMC-based procedure based on that of [18] is used for tuning R_{fb}, determining parameters K, T_i, T_d and N_d as

$$T_i = T + \frac{\tau^2}{2(T_f+\tau)}; \quad K_p = \frac{T_i}{\mu(T_f+\tau)}$$
$$N_d = \frac{T(T_f+\tau)}{T_iT_f} - 1; \quad T_d = \frac{N_dT_f\tau}{2(T_f+\tau)} \qquad (4)$$

This procedure involves a design parameter, called T_f, which is related to the required closed-loop bandwidth and influences the degree of stability. A feature of the presented method is that the model error overestimate yields a lower bound T_{finf} for T_f which surely preserve the stability of the loop including R_{fb} (tuned using the simplified model) with respect to the model error correspondingly introduced. In de, it can be shown that the IMC-PID tuned with formulae (4) guarantees the robust stability of the control loop if

$$|W(j\omega)| < \left|\mu \frac{1+j\omega T_f}{1+j\omega T}\right| \quad \forall \omega \qquad (5)$$

hence, given the model error magnitude overbound, T_{finf} can be determined as illustrated in Fig. 2.

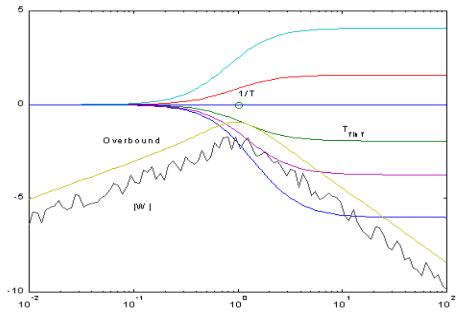

Fig. 2 : Obtaining the value of T_{finf}.

On the basis of the measured response and of the synthesized R_{fb} it is now possible to determine a nonparametric model of the closed-loop part of the control system, so that its step response can be computed as a function of parameter T_f. Finally, the set-point weights b and c in R_{ff} are tuned so that the overall system step response be as close as possible to that of a first order model with unit gain, delay equal to that measured for the process and time constant T_m, which becomes a second design parameter related to performance. This is done by minimizing with respect to b and c the ISE functional

$$J(b,c) = \int_0^{t_{max}} (y(t,b,c) - y_m(t))^2 \, dt$$

between the response of the first order model, i.e.

$$y_m(t) = \begin{cases} 0 & t \le \tau \\ 1-e^{-(t-\tau)/T_m} & t > \tau \end{cases}$$

and that of the actual system, computed using the nonparametric model of the loop. This optimization can be done with respect to both b and c or to the sole b, thus obtaining an output derivation PID.

4. Practical considerations

Under the operator's point of view, the main characteristic of the presented autotuner is to replace the common stability and performance specifications (the closed-loop phase margin φ_m and cutoff frequency ω_c) with parameters T_f and T_m. At a first glance this might not appear a big advantage but, apart from the robustness considerations which make this a very safe approach, the adoption of these two design parameters makes the understanding of the autotuner's operation much easier. In fact T_f can be easily thought as the "degree of robustness desired over a minimum imposed by the autotuner" while T_m can be interpreted as the "desired closed-loop dominant time constant which can be chosen in a range decided by the system".

Experience has shown that for operators it is *far easier* to choose a value for T_f and T_m than for φ_m and ω_c. Thus, if the proposed design parameters are adopted, there is no need to force for them any "universal" default like most autotuners do for their ones, and the system is more likely to really achieve the best possible performance in every specific case. The presented design approach has been tested by asking several people, with very different degrees of theoretical knowledge and of familiarity with process control and instrumentation, to use the autotuner, and the results have been satisfactory. The interface appears to be simple enough and the tuning results are quite good.

5. Generalizing the approach

In this work it has been chosen to show the proposed approach "in action", i.e. to present its application to the construction of an autotuner. This has been done for emphasizing its applicability, relying on the fact that the previous discussion on the key points of the approach should allow to appreciate its generality.

Anyway, some examples of how the presented tuning method could be extended might be worth noting. For example, one could synthesize $R_{fb}(s)$ by means of the IMC or a similar method, but with a nominal model structure different from (3). This would allow to tailor the achievable stability and disturbance rejection properties to a desired class of process models, thus obtaining specialized versions of the autotuner for different application domains. If the ∞-norm of the model error can be expressed as a function of a very small set of design parameters, ideally of a single one as here was done with T_f - see **(5)** - the synthesis procedure remains substantially similar.

Along the same reasoning path, the synthesis of $R_{ff}(s)$ could also be tailored to a specific class of processes by choosing a different desired response $y_m(t)$. As long as the response of the closed-loop part of the control system can be computed as a function of the design parameter used in the synthesis of $R_{fb}(s)$, the ISE minimization procedure too remains the same.

Furthermore, the same technique could be applied to a regulator structure different from the ISA-PID, possibly chosen on the basis of the process dynamics' structure so that the synthesis of $R_{fb}(s)$ be particularly straightforward, e.g. as indicated in [14].

6. Two simulation examples

section reports two simulation examples: one refers to a process described by the transfer function

$$P_1(s) = \frac{1}{(1+5s)^4}$$

while in the other case the process is considerably less suited to PID control, having the transfer function

$$P_2(s) = \frac{1}{1+s+1.5625s^2}$$

which has two complex poles with natural frequency 0.8 and damping 0.4. The tuning results in these two cases are reported in the following table (output derivation was requested, so parameter c is zero in both cases), while Fig. 3 and Fig. 4 report the corresponding closed-loop step responses.

	K_p	T_i	T_d	N_d	b
P_1	0.81	13.34	2.17	1.17	0.79
P_2	0.11	1.04	0.03	0.04	7.29

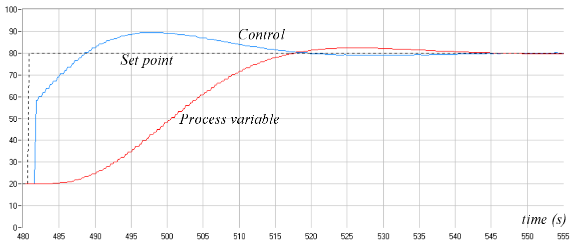

Fig. 3: simulation results of example 1.

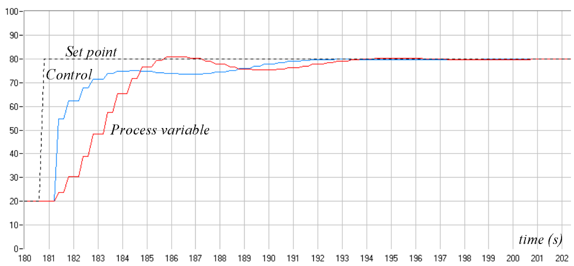

Fig. 4: simulation results of example 2.

It can be appreciated that the tuning results are satisfactory, but what is maybe more important is the ease in achieving them, i.e. the simplicity of the interface provided by the autotuner, which is treated in the following section.

7. Using the presented autotuner

This section illustrates the presented autotuner under the user's point of view by describing the tuning of a PID loop. The first step is obviously to invoke the autotuner, which brings up the dialog of Fig. 5. In the meantime, the autotuner scans the loops in the control system and allows to select which one has to be tuned. There is a tuning permission logic (not described here for brevity) and, if required, this logic can be customized when assembling and configuring the control system.

Fig. 5: Selecting the loop to be tuned.

Once the loop to be tuned has been chosen, a step test is performed on the process. The test evolution is shown in a convenient window and lasts until the user clicks the OK button. A Cancel button is always available for resetting the procedure completely. After the step test, the system continues recording the process variable for a while in order to determine the amount of noise present. This requires no user interaction.

The final step, illustrated in

Fig. 6, is to choose the desired response shape. The user moves the cursor in the XY graph at the lower right corner of the window depicted below, and the corresponding closed-loop response is continuously updated and shown in the upper half of the window. For reference and convenience, also the open-loop process response (normalized to unit gain so as to be usefully comparable to the closed-loop one) is shown.

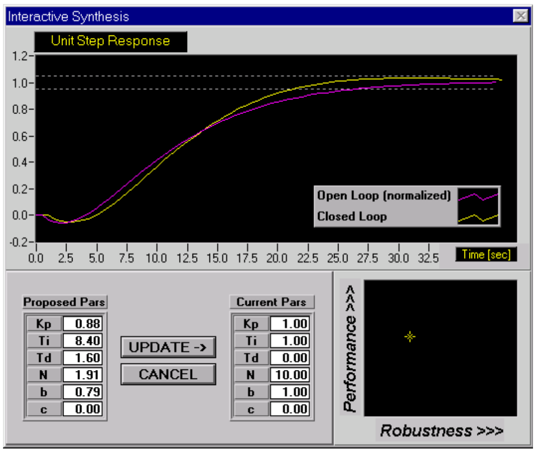

Fig. 6: Choosing the desired response shape.

The two axes of the XY graph control parameters T_f and T_m respectively, but provide also useful visual cues for the synthesis: if the user wants a faster response the cursor must go up, while if greater conservatism (or robustness) is desired the cursor must go right. Of course the best is the upper right corner, but in any real case a tradeoff must be accepted and this method allows to choose the balance by seeing the system response directly. Note that the process model used for computing the tuning results on-line is the convolution model computed by the autotuner on the basis of the measured open-loop response: as such, the forecast transients shown to the user are very close to the real ones. Once the result is satisfactory, clicking on the UPDATE button writes the new parameters to the regulator. Just for reference, Fig. 7 reports the actually obtained response: it can be seen that the accordance with the forecast one is excellent.

Finally remark that, as previously noted, this is the *operator* interface. Along the tuning, a lot of information concerning the process (transfer function estimates, noise characteristics and so on) is produced by the system; this information has not been shown here for brevity and is concealed to the plant operator for apparent reasons, but is available if required and can be very useful for the engineer.

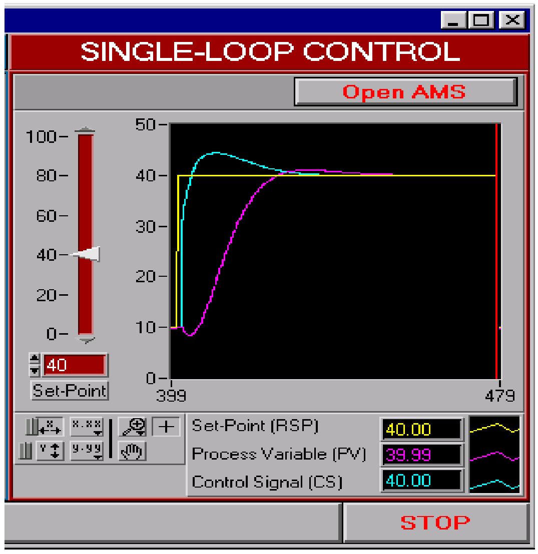

Fig. 7: The actually obtained response.

8. Conclusions

An approach to PID autotuning based on some results of the robust control theory has been briefly illustrated, and as an example of its application a PID autotuner built in accordance to it has been presented. It is worth noting that the approach – thus the autotuner - is aimed at the synthesis of *industrial* PID regulators, hence it takes into account also those parameters (namely, the additional pole and the set-point weights) which are normally neglected in “textbook” autotuners.

The key features of the presented approach are basically two: it is based on a nominal process model but takes into account also a measurement of the model error, and it exploits the 2 d.o.f. structure of industrial PIDs in the ISA form so as to achieve good robustness with the minimum performance loss.

In the implementation of the presented autotuner, the characteristics of the approach have been exploited and the design the user interface has been particularly taken care of, so as to reduce the theoretical knowledge required for using it to a minimum. The results obtained are encouraging, both with respect to the prototype's operation and to the further insight in the problem of autotuning that this research and engineering work has produced.

Acknowledgements

This work has been partially supported by MURST (Ministero dell'Università e della Ricerca Scientifica e Tecnologica) and by CNR (Consiglio Nazionale delle Ricerche).

Demonstrative application

A demonstrative application for the autotuner can be obtained by contacting the corresponding author: it consists of the demo program (a Windows executable) and of a brief manual in PostScript format.

References

[1] Åström K.J. and T. Hägglund, Industrial Adaptive Controllers Based on Frequency Response Techniques, *Automatica*, Vol. 27 , No. 4, 1991, 599-609.

[2] Åström K.J. and T. Hägglund, *PID Controllers: Theory, Design and Tuning – Second Edition*, Instrument Society of America, 1995.

[3] Åström K.J., C.C. Hang, P. Persson and W. K. Ho, Towards Intelligent PID Control, *Automatica*, Vol. 28, 1992, pp.1-9.

[4] Boyd S.P. and C.H. Barratt, *Linear Controller Design: Limits of performance*, Prentice-Hall, 1991.

[5] Doyle J.C., B.A. Francis and A.R. Tannenbaum, *Feedback Control Theory*, MacMillan, 1992.

[6] Gawthrop P.J, Self-Tuning PID Controllers: Algorithms and Implementation, *IEEE Trans. On Automatic Control*, Vol. 31, No. 3, 1986, pp. 201-209.

[7] Gawthrop P.J.and K.W. Lim, Robustness of Self-Tuning Controllers, *IEE Proceedings-D*, Vol. 129, 21-29.

[8] Giarré L. and M. Milanese, SM identification of approximating models for H_∞ Robust Control, *Proc. 35th Conference on Decision and Control*, 1996.

[9] Hägglund T. and K.J. Åström, Automatic Tuning of PID Controllers Based on Dominant Pole Design, *Proc. IFAC Workshop on Adaptive Control of Chemical Processes*, 1985.

[10] Hang C.C., K.J. Åström and W.K. Ho, Refinements of the Ziegler-Nichols Tuning Formula, *IEE Proceedings-D*, Vol. 138, No. 2, 1991, pp. 111-118.

[11] Ho W.K., C.C. Hang and J. H. Zhou, Performance and Gain and Phase Margins of Well-Known PI tuning formulas, *IEEE Trans. on Control Systems Technology*, Vol. 3, 1995, pp. 245-248.

[12] Ho W.K., C.C. Hang and L. S. Cao, Tuning of PID Controllers Based on Gain and Phase Margin Specifications, *Automatica*, Vol. 31, 1995, pp. 497-502.

[13] Isermann R. and K.H. Lachmann, *Adaptive Control Systems*, Prentice-Hall, 1992.

[14] Leva A., How to Construct PID Synthesis Rules for Structurally Complex Process Models, *Proc. SSC '98 Conference*, 1998.

[15] Leva A. and A.M. Colombo, A Method for Optimizing the Set-Point Weights in ISA-PID Autotuners, *Int. Rep. 98-068, 1998, Dipartimento di Elettronica e Informazione, Politecnico di Milano.*

[16] Leva A. and A.M. Colombo, Model Mismatch Estimation for the Robust Autotuning of Industrial Regulators, *Int. Rep. 98-073, 1998, Dipartimento di Elettronica e Informazione, Politecnico di Milano.*

[17] Leva A. and M. Lovera, Estimating the Step Response of an Unknown SISO Process on the Basis of a Relay Experiment, *Proc. MCEA '98 Conference*, 1998.

[18] Morari M. and E. Zafiriou, *Robust Process Control*, Prentice-Hall, 1989.

[19] Wahlberg B. and L.Ljung, Hard Frequency-Domain Error Bounds from Least-Squares Like Identification Techniques, *IEEE Trans. on Automatic Control*, Vol. 37, 1992, pp. 901-912.