

# Development and Evaluation of a PID Auto-Tuning Controller

Ioan Naşcu<sup>1</sup>, Robin De Keyser<sup>2</sup>, Silviu Folea<sup>1</sup>, Tudor Buzdugan<sup>1</sup>

<sup>1</sup> Technical University of Cluj Napoca, Department of Automation, [Ioan.Nascu@aut.utcluj.ro](mailto:Ioan.Nascu@aut.utcluj.ro), [Tudor.Buzdugan@aut.utcluj.ro](mailto:Tudor.Buzdugan@aut.utcluj.ro), [Silviu.Folea@aut.utcluj.ro](mailto:Silviu.Folea@aut.utcluj.ro)

<sup>2</sup> Ghent University, EeSA-department of Electrical Energy, Systems & Automation, [rdk@autoctrl.UGent.be](mailto:rdk@autoctrl.UGent.be)

**Abstract**-This paper presents comparative studies regarding a recently developed extension of the widely used relay-feedback PID auto-tuner. The proposed method consists of two steps: process identification and controller design. First, a non-iterative procedure is suggested for identification of two points on the process Nyquist curve. A second-order-plus-dead-time model is obtained and then used for PID controller design based on the internal model principle (IMC). For the identification of the two points on the Nyquist curve a pure relay in the feedback loop (as used in standard auto-tuning) and a relay which operates on the integral of the error are used. The paper compares the performance of autotuning methods with experimental studies on a commercial auto-tuning PID controller - a Shimaden PID auto-tuner.

**Keywords:** auto-tuning, PID controller, relay experiment, Nyquist plot, commercial PID controllers.

## I. INTRODUCTION

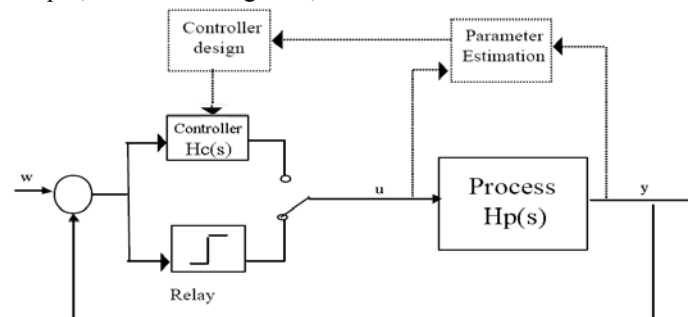
At the beginning of the new millennium the PID controller continues to be a key component of industrial control. Their popularity is justified by the following advantages: they have a simple structure, their principle is well understood by instrumentation engineers and their control capabilities have proven to be adequate for most control loops. Moreover, due to process uncertainties, a more sophisticated control scheme is not necessarily more efficient in real-life applications than a well-tuned PID controller. However, it is common that PID controllers are poorly tuned in practice because the choice of controller parameters requires professional – specialized knowledge by the user. To simplify this task and to reduce the tuning-time (process starting up period), PID controllers can incorporate *auto-tuning* capabilities. The auto-tuners are equipped with a mechanism capable of computing the *correct* parameters automatically when the regulator is connected to the field [Åström, *et al.*, 1984; Åström, *et al.*, 1992; Åström, *et al.*, 1995; Leva, *et al.*, 2002]. Auto-tuning is a very desirable feature and almost every industrial PID controller provides it nowadays. These features provide easy-to-use controller tuning and have proven to be well accepted among process engineers. For the automatic tuning of the PID controllers, several methods have been proposed. Some of these methods are based on identification of one point of the process frequency response, while the others are based on the knowledge of some characteristic parameters of the open-loop process step

response. The identification of a point of the process frequency response can be performed either using a proportional regulator, which brings the closed-loop system to the stability boundary, or by a relay forcing the process output to oscillate. Åström and Hägglund [Åström, *et al.*, 1984] report an important and interesting approach. Their method is based on the Ziegler-Nichols frequency domain design formula. A relay connected in a feedback loop with the process is used in order to determine the critical point.

This contribution briefly describes the development of two auto-tuning methods based on the identification of *two* points on the process Nyquist curve [Nascu and De Keyser, 2003]. The performance of both autotuning methods are compared with those of Åström-Hägglund method and with experimental studies on a commercial auto-tuning PID controller - a Shimaden PID auto-tuner.

## II. THE EXPERIMENTAL SET-UP

A standard closed loop system with single input single output, as shown in figure 1, is considered.



To perform relay feedback experiments, the process is first brought to steady-state conditions in manual control or with any preliminary tuned PID controller.

The process  $H_p$  is assumed to be linear, stable and proper.

The PID controller has a non-interacting structure cascaded with a first order filter:

$$H_c(s) = K_c \left( 1 + \frac{1}{sT_i} + sT_d \right) \frac{1}{T_f s + 1} \quad (1)$$

In this paper it is assumed that the process dynamics can be described with reasonable accuracy by a second-order-plus-dead-time (SOPDT) model as:

$$H_p(s) = \frac{k_p e^{-\theta s}}{\tau^2 s^2 + 2\zeta\tau s + 1}, \theta \geq 0 \text{ and } \tau, \zeta > 0 \quad (2)$$

In order to tune the PID controller we will approximate  $e^{-\theta s}$  by  $(1-\theta s)$  and using the well-known IMC-PID design method we then obtain:

$$K_c = \frac{2\zeta\tau}{k_p(2T_c + \theta)} \quad (3)$$

$$T_i = 2\zeta\tau, T_d = \frac{\tau^2}{2\zeta\tau}, T_f = \frac{T_c^2}{2T_c + \theta}$$

thus resulting in the closed-loop transfer function:

$$H_0(s) = \frac{H_p(s) \cdot H_c(s)}{1 + H_p(s) \cdot H_c(s)} = \frac{e^{-\theta s}}{(T_c s + 1)^2} \quad (4)$$

The parameter  $T_c$  is the closed-loop time constant and is a design parameter which can be selected by the user in order to tune the controller 'aggressiveness'.

The parameters of the process transfer function (2) are assumed to be unknown and - in order to identify them - 2 points of the frequency response have to be estimated using two relay feedback experiments: the first experiment uses a relay with hysteresis or an integrating relay (= a relay without hysteresis which has as input the integral of the error) and the second experiment uses a pure relay (without hysteresis). After performing these two experiments the values of the process frequency response at two different frequencies  $\omega_1$  and  $\omega_2$  are obtained:

$$H_p(j\omega_1) = a_1 + jb_1 \text{ and } H_p(j\omega_2) = a_2 + jb_2 \quad (5)$$

The procedure how to perform these experiments and how to obtain the values for  $(a_i, b_i; i=1,2)$  is further explained in section III.

### III. THE TUNING METHOD

Note from (2) that  $H_p(j\omega)$  can be written as follows:

$$H_p(j\omega) = H'_p(j\omega)H''_p(j\omega), \quad (6)$$

with

$$H'_p(j\omega) = \frac{k_p}{1 - \tau^2 \omega^2 + 2\zeta\tau j\omega}, \quad H''_p(j\omega) = (1 - \theta j\omega)$$

If we define - for a certain point on the Nyquist plot - the magnitude, the phase angle and the real and imaginary

components of  $H_p(j\omega)$ ,  $H'_p(j\omega)$  and  $H''_p(j\omega)$  to be  $M, \alpha, a, b$ , respectively  $M', \alpha', a', b'$  and  $M'', \alpha'', a'', b''$ , then:

$$M^2 = a^2 + b^2 = M'^2 M''^2 = (a'^2 + b'^2)(a''^2 + b''^2) \quad (7)$$

$$\alpha = \alpha' + \alpha''$$

$$\begin{aligned} \operatorname{tg}(\alpha') &= \frac{b'}{a'} = \operatorname{tg}(\alpha - \alpha'') = \frac{\operatorname{tg}(\alpha) - \operatorname{tg}(\alpha'')}{1 + \operatorname{tg}(\alpha)\operatorname{tg}(\alpha'')} = \\ &= \frac{\frac{b}{a} - \frac{b''}{a''}}{1 + \frac{b}{a}\frac{b''}{a''}} = \frac{a''b - ab''}{aa'' + bb''} \end{aligned} \quad (8)$$

$$b' = a' \frac{a''b - ab''}{aa'' + bb''} \quad a' = b' \frac{aa'' + bb''}{a''b - ab''} \quad (9)$$

From (9) - and afterwards (7) - it can be written:

$$\begin{aligned} (a'^2 + b'^2) &= a'^2 \left[ 1 + \left( \frac{a''b - ab''}{aa'' + bb''} \right)^2 \right] = \\ &= a'^2 \frac{(a''^2 + b''^2)(a^2 + b^2)}{(aa'' + bb'')^2} = a'^2 \frac{(a''^2 + b''^2)^2 (a'^2 + b'^2)}{(aa'' + bb'')^2} \end{aligned}$$

thus leading to:

$$aa'' + bb'' = a'(a''^2 + b''^2) = a'M''^2$$

and using (9) again also to:

$$a''b - ab'' = b'M''^2$$

Thus for the two points obtained from the relay feedback experiments we have:

$$a_1 a_1'' + b_1 b_1'' = a_1' M_1''^2 \quad (10a)$$

$$a_1'' b_1 - a_1 b_1'' = b_1' M_1''^2$$

$$a_2 a_2'' + b_2 b_2'' = a_2' M_2''^2 \quad (10b)$$

$$a_2'' b_2 - a_2 b_2'' = b_2' M_2''^2$$

The parameters  $a_i$  and  $b_i$  ( $i=1, 2$ ) can be obtained using the two relay feedback experiments. In order to identify a point on the process Nyquist curve, a relay connected in a feedback loop with the process is used, forcing the process output to oscillate. Therefore assume that  $d_i$  is the relay amplitude and  $\varepsilon_i$  is the relay hysteresis width. For the given values  $d_i$  and  $\varepsilon_i$  we will obtain oscillation with amplitude  $h_i$  and period  $T_i$  in the process output.

In the first experiment there are two possibilities: using a relay with hysteresis, a point in the third quadrant on the

process Nyquist curve,  $P_h(a_1, j b_1)$ , is identified [Aström, *et al.*, 1995]:

$$\omega_1 = \frac{2\pi}{T_1}, \quad a_1 = \frac{-\pi\sqrt{h_1^2 - \varepsilon_1^2}}{4d_1}, \quad (11a)$$

$$b_1 = -\frac{\pi\varepsilon_1}{4d_1}, \quad M_1^2 = a_1^2 + b_1^2$$

or, using an integrating relay ( $\varepsilon_1=0$ ) the point given by the intersection of the process Nyquist curve and the negative imaginary axis,  $P_i(a_1, j b_1)$ , is identified:

$$\omega_1 = \frac{2\pi}{T_1}, \quad a_1 = 0, \quad b_1 = -\frac{\pi h_1}{4d_1}, \quad M_1^2 = b_1^2 \quad (11b)$$

In the second experiment, using a pure relay ( $\varepsilon_2=0$ ), the point given by the intersection of the process Nyquist curve and the negative real axis,  $P_p(a_2, j b_2)$ , is identified:

$$\omega_2 = \frac{2\pi}{T_2}, \quad a_2 = \frac{-\pi h_2}{4d_2}, \quad b_2 = 0, \quad M_2^2 = a_2^2 \quad (12)$$

The real and imaginary components of  $H'_p(j\omega_i)$  and  $H''_p(j\omega_i)$  can be written as:

$$a'_i = \text{Re}(H'_p(j\omega_i)) = \frac{k_p(1-\tau^2\omega_i^2)}{(1-\tau^2\omega_i^2)^2 + (2\zeta\tau\omega_i)^2} = \frac{(1-\tau^2\omega_i^2) M_i^2}{k_p} \quad (13)$$

$$b'_i = \text{Im}(H'_p(j\omega_i)) = \frac{k_p(-2\zeta\tau\omega_i)}{(1-\tau^2\omega_i^2)^2 + (2\zeta\tau\omega_i)^2} = \frac{-2\zeta\tau\omega_i M_i^2}{k_p}$$

$$a''_i = \text{Re}(H''_p(j\omega_i)) = 1 \quad b''_i = \text{Im}(H''_p(j\omega_i)) = -\theta\omega_i$$

Inserting in (10) the relations (13) for  $a'_i, b'_i, a''_i, b''_i; i=1,2$  gives 4 equations with 4 unknown parameters:  $k_p, \theta, \tau^2$  and  $2\zeta\tau$ :

$$a_1 - b_1\theta\omega_1 = \frac{(1-\tau^2\omega_1^2)M_1^2}{k_p} \quad (14.a)$$

$$b_1 + a_1\theta\omega_1 = \frac{-2\zeta\tau\omega_1 M_1^2}{k_p} \quad (14.b)$$

$$a_2 - b_2\theta\omega_2 = \frac{(1-\tau^2\omega_2^2)M_2^2}{k_p} \quad (14.c)$$

$$b_2 + a_2\theta\omega_2 = \frac{-2\zeta\tau\omega_2 M_2^2}{k_p} \quad (14.d)$$

Using in the first experiment a relay with hysteresis and in the second experiment a pure relay (the 'HR' case) then the first identified point on the process Nyquist curve is in the third quadrant and the second one is the intersection with the negative real axis (thus  $b_2=0$ ). From (14.b) and (14.d) results:

$$\theta = \frac{a_2 b_1}{\omega_1 (M_1^2 - a_2 a_1)} \quad (15.a)$$

From (14.c) and (14.a) results:

$$k_p = \frac{M_1^2 M_2^2 (\omega_2^2 - \omega_1^2)}{\omega_2^2 M_2^2 (a_1 - b_1\theta\omega_1) - a_2 \omega_1^2 M_1^2} \quad (15.b)$$

Now, (14.a) and (14.b) give:

$$\tau^2 = \frac{1}{\omega_1^2} + \frac{k_p (b_1\theta\omega_1 - a_1)}{\omega_1^2 M_1^2} \quad (15.c)$$

$$2\zeta\tau = -\frac{k_p (b_1 + a_1\theta\omega_1)}{\omega_1 M_1^2} \quad (15.d)$$

These values can then be used in (3) to calculate the PID-parameters.

Using in the first experiment an integrating relay and in the second experiment a pure relay (the 'IR' case) then the first identified point on the process Nyquist curve is the intersection with the negative imaginary axis (thus  $a_1=0$ ) and the second one is the intersection with the negative real axis (thus  $b_2=0$ ). In this case we obtain the following parameters which can then be used in (3) to tune the PID:

$$\theta = \frac{a_2}{\omega_1 b_1}$$

$$k_p = \frac{a_2 b_1^2 (\omega_1^2 - \omega_2^2)}{b_1 \omega_1 (a_2 \theta \omega_2^2 + b_1 \omega_1)} \quad (16)$$

$$\tau^2 = \frac{1}{\omega_1^2} \left( 1 + \frac{k_p a_2}{b_1^2} \right)$$

$$2\zeta\tau = -\frac{k_p}{\omega_1 b_1}$$

To perform relay feedback experiments, the process is first brought to steady-state conditions in manual control or with any preliminary tuned PI controller. For processes of first or second order without time delay, the value of  $\theta$  in the model (2) will be zero. Strictly applying theory, these processes cannot be forced to oscillate by a pure relay. A relay without hysteresis can be used only if the process Nyquist curve crosses the negative real axis. In fact, considering that in any digital controller implementation the sampling process itself introduces a phase lag and that in real situations the process output is filtered, it can be assumed that all the processes in practical cases will oscillate when a relay controller is connected. Unfortunately, in absence of hysteresis, the relay experiment gives small values for the amplitude of oscillation, difficult to measure in the presence of measurement noise. Thus, in case of small amplitude oscillations, in the pure relay experiment  $a_2$  and  $\theta$  are fixed to zero,  $k_p$  is computed from steady-state conditions and  $\tau^2$  and  $2\zeta\tau$  are computed using measurements only from the first experiment ( $a_1, b_1, \omega_1$ ). If there are no others restrictions, the amplitude of the relay will be set to  $0.1u_0$  and the hysteresis to  $0.01y_0$ . Better results can be obtained using IR method.

#### IV. EXPERIMENTS

The paper compares the three auto-tuning methods based on taking the process to its stability limit with the auto-tuning method implemented in the numerical controller SR73A. A MATLAB test-bed for simulating the auto-tuning algorithms and for studying the behavior of the SR73A controller with simulated processes has been implemented. The performances of the methods have been compared on three types of processes and the advantages of using one method or another for each situation has been highlighted.

Sluggish processes have been selected because this numerical controller has a minimum sampling time of 1 second.

For testing the performances of the auto-tuning mechanism implemented in the SR73A controller a test-bed has been set up, and consists of the PC running the performance evaluation mechanism, an RS232-RS485 converter for updating the parameters of the controller from the PC and a NI data acquisition board for providing the industrial controller with simulated plant analog signals and reading its commands, as presented in Figure 2.

For using the National Instruments PC-1200 data acquisition board with MATLAB, a series of MEX-files have been developed in C.

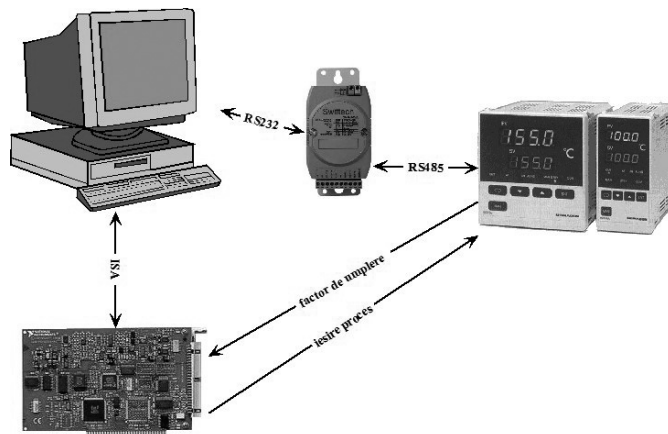
MEX stands for MATLAB Executable. MEX-files are dynamically linked subroutines produced from C or Fortran source code that, when compiled, can be run from within MATLAB in the same way as MATLAB M-files or built-in functions. The external interface functions provide functionality to transfer data between MEX-files and MATLAB, and the ability to call MATLAB functions from C or Fortran code.

The source code for a MEX-file consists of two distinct parts:

A computational routine that contains the code for performing the computations that you want implemented in the

MEX-file. Computations can be numerical computations as well as inputting and outputting data.

A gateway routine that interfaces the computational routine with MATLAB by the entry point mexFunction and its parameters prhs, nrhs, plhs, nlhs, where prhs is an array of right-hand input arguments, nrhs is the number of right-hand input arguments, plhs is an array of left-hand output arguments, and nlhs is the number of left-hand output arguments. The



gateway calls the computational routine as a subroutine.

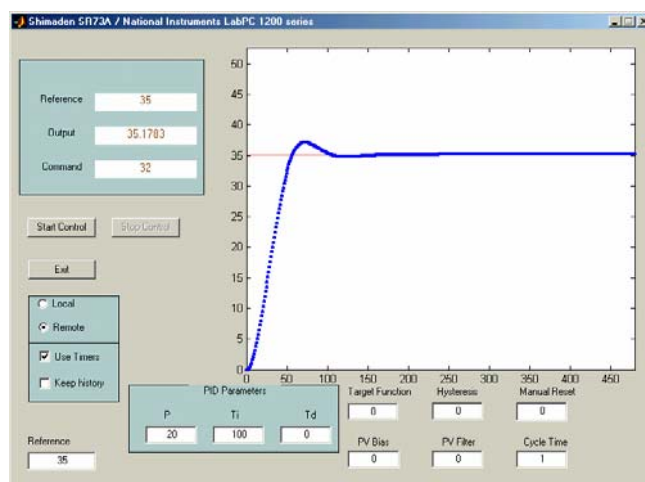


Figure 2. The setup of the test-bed and the graphical interface of the application testing the SR73A controller.

The plant is simulated in MATLAB but it is controlled by the industrial controller through the NI PC-1200 DAQ. At each sampling time the data acquisition board measures the duty factor of the PWM control signal issued by the controller and via the MEX routines is fed into the model. The output of the simulated plant is evaluated and sent back to the controller as an analog signal.

The proposed auto-tuning algorithms have been tested by simulated examples. Three different models of the process transfer function has been considered:

$$H_{P1}(s) = \frac{1}{(40s+1)(20s+1)} \quad (17)$$

$$H_{P2}(s) = \frac{1}{(20s+1)^3} \quad (18)$$

$$H_{P3}(s) = \frac{(20s+1)}{(40s+1)(40s+1)} \quad (19)$$

Setpoint changes have been analyzed. For simulations, the amplitude of the setpoint is 1. When using the industrial controller, the output of the plant has been normalized and plotted. The closed loop step responses using IR (curve 1) and HR (curve 2) methods, Ziegler-Nichols method (curve 3) and SR73 (curve 4) digital controller are presented. Open loop process step response (curve 5) has been also presented. The design parameter  $T_c$  can be specified by the user or can be chosen automatically as a percent of  $2\zeta\tau$  or  $2\zeta\tau+0$ . Here  $T_c$  is fixed at  $(2\zeta\tau)/2$  for the upper plots and respectively  $(2\zeta\tau)/5$  for lower plots.

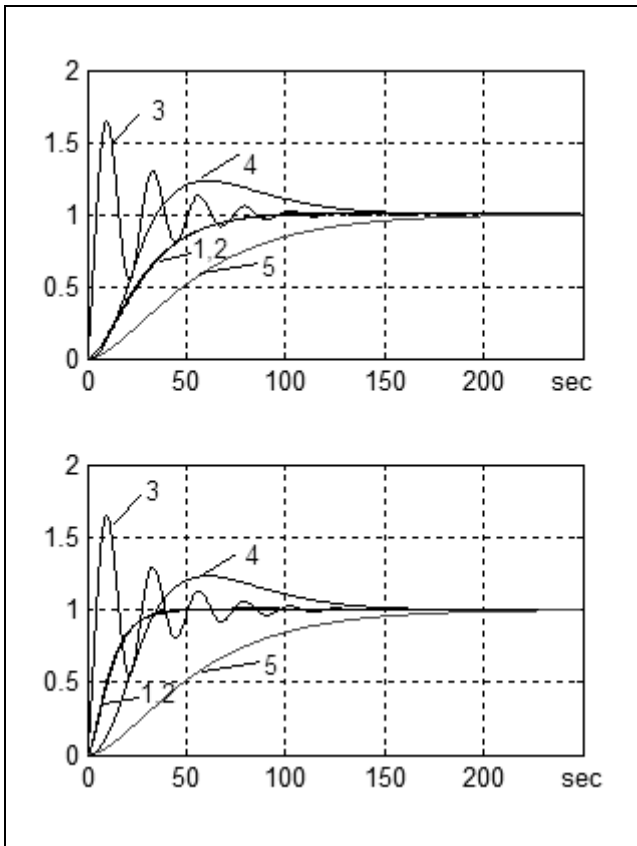


Figure 3. Performance comparison between: open loop (5), IR method (1), HR method (2), Ziegler-Nichols method (3) and SR73 (4) digital controller. Step response, process given by (17),  $T_c$  fixed at  $(2\zeta\tau)/2$  for the upper plots and  $(2\zeta\tau)/5$  for lower plots.

In figure 3 a second order process given by (17) is considered. It can be observed that the IR (1) and HR (2) auto-tuning methods give similar results and better performance than the classical Ziegler-Nichols (fast response but big overshoot) auto-tuning method or SR73 (4) digital controller (slow response and overshoot). The change of the user-defined parameter  $T_c$  in IR and HR methods is influencing the closed-loop behavior (upper plots slower response lower plots faster response). The open loop behavior (5) is significantly improved for the IR (1) and HR (2) auto-tuning methods by reducing the settling time.

The difference between the classical Ziegler-Nichols (3) and the proposed IR (1) and HR (2) methods is visible. However, by choosing the right value for the design-parameter  $T_c$  in IR and HR methods it is possible to obtain similar performance as the ZN auto-tuner (that has no design parameter).

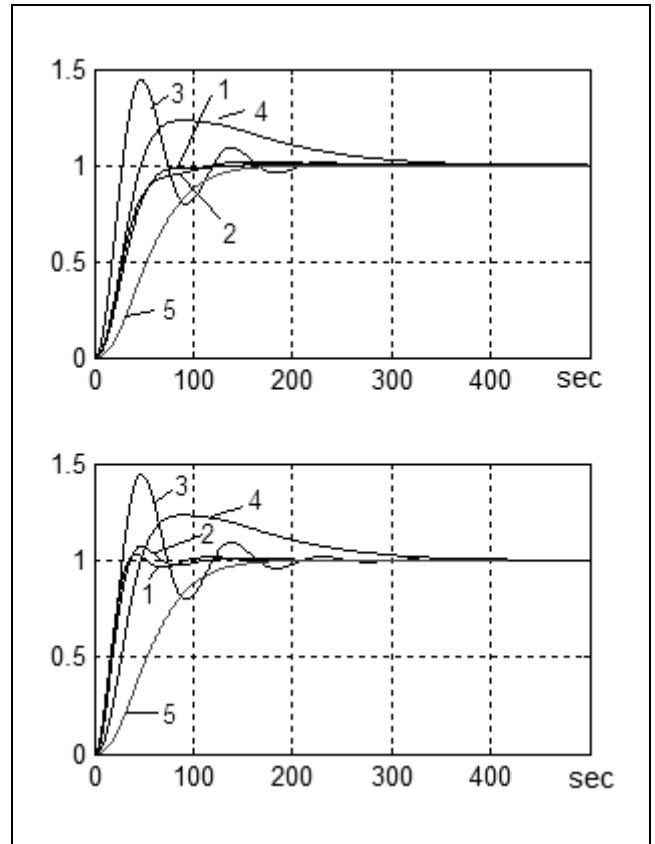


Figure 4. Performance comparison between: open loop (5), IR method (1), HR method (2), Ziegler-Nichols method (3) and SR73 (4) digital controller. Step response, process given by (18),  $T_c$  fixed at  $(2\zeta\tau)/2$  for the upper plots and  $(2\zeta\tau)/5$  for lower plots.

Similar results (figure 4) are obtained considering a third order process given by (18).

Considering the process given by (19) the results (figure 5) are different for IR method (1) and HR method (2) (better results for IR). Also the results with SR73 (4) digital controller are better than for previous cases.

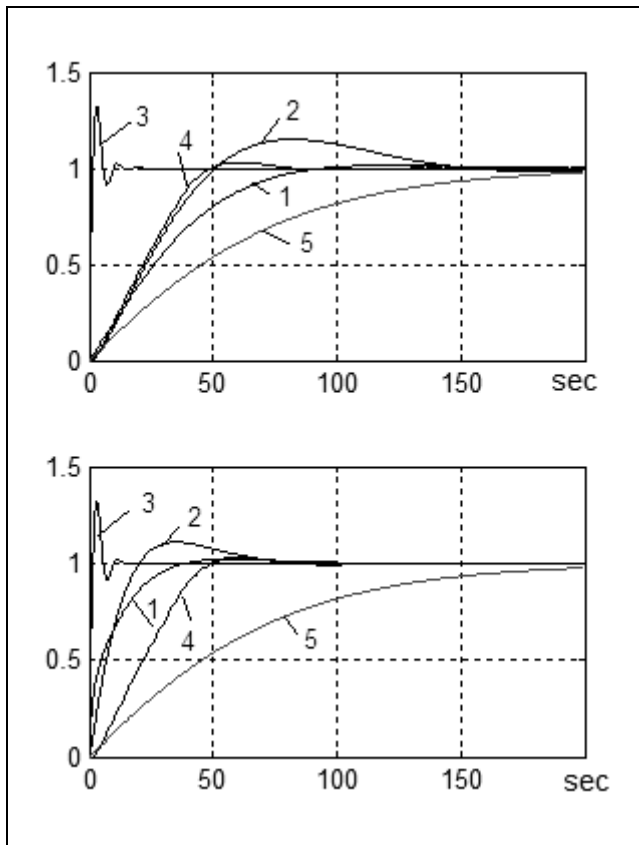


Figure 5. Performance comparison between: open loop (5), IR method (1), HR method (2), Ziegler-Nichols method (3) and SR73 (4) digital controller. Step response, process given by (19),  $T_c$  fixed at  $(2\zeta\tau)/2$  for the upper plots and  $(2\zeta\tau)/5$  for lower plots.

## V. CONCLUSION

Some relay-based algorithms for auto-tuning of PID controllers have been presented, assuming a simple process model structure and achieving the controller tuning by

identifying two points on the process frequency response and using the IMC-PID design method.

The methods are simple and as other auto-tuning techniques, does not require an advanced knowledge of control engineering theory. These auto-tuning methods yield good PID parameters for a specific class of processes (stable processes with self-regulation); it is not a general methodology for arbitrary process models.

The controller developed with the IR and HR auto-tuning method gives good performance for step response (fast dynamics, no overshoot), in most of the cases the results being better than in the case of Ziegler-Nichols method or SR73 digital controller. The change of the user-defined parameter  $T_c$  is influencing the closed-loop behaviour.

## REFERENCES

- [1] Åström, K.J. and T. Hägglund. Automatic tuning of simple regulators with specifications on phase and amplitude margins, *Automatica*, **20**, 1984, pp.645.
- [2] Åström, K.J., C.C. Hang, P. Persson and W.K. Ho. Towards intelligent PID control. *Automatica*, **28**, 1992, pp.1
- [3] Åström, K.J. and T. Hägglund. *PID Controllers: Theory, Design and Tuning*. Instrument Society of America, Research Triangle Park, NC, USA, 1995
- [4] Leva, A., C. Cox and A. Ruano. Hands-on Pid autotuning: a guide to better utilisation. *IFAC Professional Brief*, 2002
- [5] Nasu I., De Keyser R., A Novel Application of Relay-feedback for PID Auto-tuning, *IFAC CSD'03 Conference on Control Systems Design*, 2003, Bratislava, Slovak Republic, CDpaper – id0876172, 6p
- [6] Ogata, K. *Modern Control Engineering*. Prentice-Hall, 1990
- [7] Ho, W.K., C.C. Hang and L.S. Cao. Tuning of PID controllers based on gain and phase margin specifications, *Automatica*, **31**, No.3, 1995, pp.497.
- [8] Poulin, E., A. Pomerleau, A. Desbiens and D. Hodouin. Development and evaluation of an auto-tuning and adaptive PID controller, *Automatica*, **32**, 1996, pp.71.
- [9] Schei, T.S.. Automatic tuning of PID controllers based on transfer function estimation. *Automatica*, **30**, 1994, pp.1983
- [10] SR73A and SR74A Series Digital Controller Instruction Manual
- [11] Lab-PC-1200 / AI User Manual
- [12] Lab-PC-1200 / AI Register-Level Programmer Manual
- [13] NI-DAQ User Manual for PC Compatibles
- [14] NI-DAQ Function Reference Help (february 2001 edition)
- [15] *MATLAB - The Language of Technical Computing*
- [16] *MATLAB - Creating Graphical User Interfaces*