DESIGN AND VALIDATION OF A PID AUTO-TUNING ALGORITHM

Diego F. Sendoya-Losada and Jesús D. Quintero-Polanco
Department of Electronic Engineering, Faculty of Engineering, Surcolombiana University, Neiva, Huila, Colombia
E-Mail: diego.sendoya@usco.edu.co

ABSTRACT

The purpose of this paper is to present the development of a PID auto-tuner algorithm based on specifications, which is generally valid for several processes. Based on prior art results, the rationale follows the principles of approximating the closed loop response to a second order transfer function. However, it is shown that the derived algorithm is generally valid and can work well on several examples which are much more complex than second order systems.

Keywords: auto-tuning, closed loop control, frequency response, PID.

1. INTRODUCTION

Tuning controllers for optimal closed loop performance depends heavily on the process to be controlled and identification is still a burden for the control engineer and remains a significant time-consuming task. Auto-tuning is a very desirable feature and almost every industrial PID controller provides it nowadays [1,2]. These features provide easy-to-use controller tuning and have proven to be well accepted among process engineers [3].

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 others are based on the knowledge of some characteristic parameters of the openloop process step response. The identification of a point of the process frequency response can be performed either using a proportional regulator, which brings the closedloop system to the stability boundary, or by a relay forcing the process output to oscillate [4-7]. Usually these preliminary tests are used to determine a model for the process, along with the tuning of controller parameters [8, 9].

This paper presents the design and validation of a generally valid algorithm for a specifications based PID auto-tuner. The next section provides the underlying principles of the proposed algorithm. The third section is a validation of the auto-tuner on two complex systems: i) a second order plus integrator (double integrator in the closed loop) and ii) a first order with integrator, time delay and non-minimum phase dynamics. Furthermore, the conclusions summarize the main outcome of this work.

2. MATERIALS AND METHODS

The development of this auto-tuning algorithm is based on the prior art where two relay-based PID auto-tuners have been presented: the Kaiser-Chiara auto-tuner and the Kaiser-Rajka auto-tuner [10]. Hence the proposed algorithm is an extended combination of the two: the Kaiser-Chiara-Rajka auto-tuner algorithm (KCR).

The approximation of a closed loop response by a dominant second order transfer function $T(s)$ with gain one:

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

with ω_n the natural frequency and ζ the damping factor, gives the relationship between the closed loop percentovershoot ($\%OS$) and the peak magnitude M_p in frequency domain [11], schematically depicted in Figure-1:

$$\%OS = 100e^{-\pi/\sqrt{1-\zeta^2}}$$

$$M_p = \frac{1}{2\zeta\sqrt{1-\zeta^2}}$$

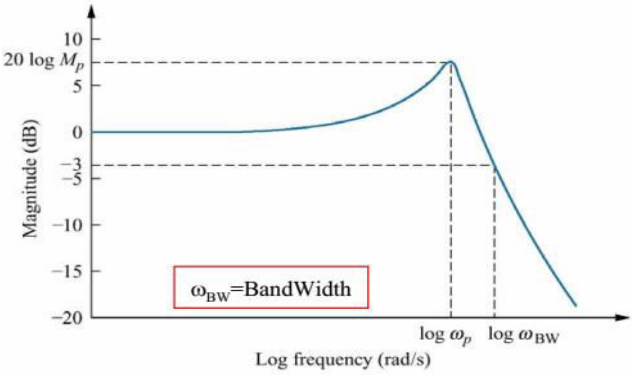


Figure-1. Schematic representation of the magnitude of a closed loop approximation with a second order system.

By specifying the allowed overshoot in the closed loop, it follows that the closed loop transfer function must fulfill the condition:

$$|T(j\omega)| = \left|\frac{G(j\omega)}{1 + G(j\omega)}\right| = M_p$$

with $G(j\omega)$ the open loop transfer function of both the process and controller. Re-writing this equation in its complex form:

$$T(j\omega) = \frac{R(\omega) + jI(\omega)}{[1 + R(\omega)] + jI(\omega)}$$

with R the real part and I the imaginary part, and taking $|T(j\omega)|^2$ results that:

$$(R + c)^2 + I^2 = r^2$$

where $c = M_p^2/(M_p^2 - 1)$ and $r = M_P/(M_p^2 - 1)$, which is nothing else than the equation of a (Hall-)circle with radius r and center in $\{-c, 0\}$[11]. In order to have a peak magnitude, only those circles with $M > 1$ are of interest. At this point, the reader should be reminded that in the Nyquist plane, specifying$|T(j\omega)|$ means that the Nyquist curve of the process and controller is tangent to the M-circle, schematically depicted in Figure-2.

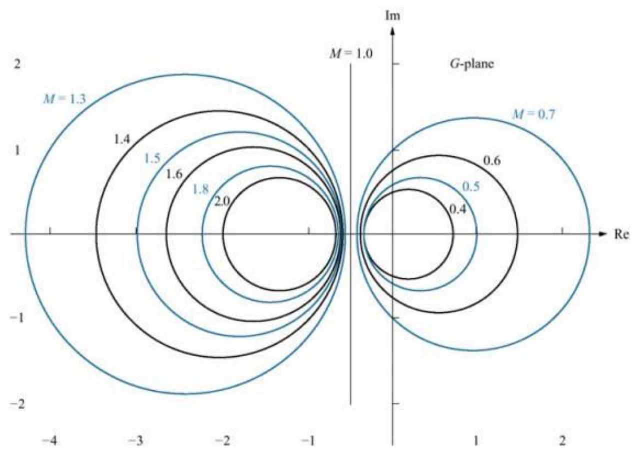


Figure-2. Hall circles in the Nyquist plane.

The equivalent of $|T(j\omega)|$ in the Nichols chart is a curve in the N-grid to which the process and controller curve should be tangent (see Figure-3).

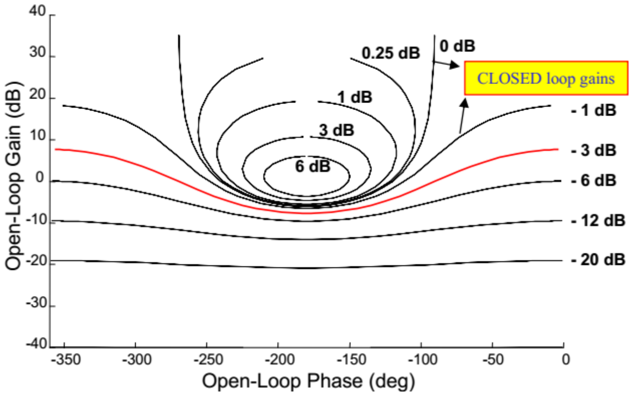


Figure-3. N-grid in the Nichols plane.

Practice has learned that a good response is achieved when a fluent curve is obtained, going smoothly around the specifications. Consequently, the $P * C$ curve will be tangent to the $\%OS$ curve at the intersection with 0 dB line (magnitude of the open loop, i.e. $P * C$). The 0 dB line represents the unit circle in the Nyquist plot, hence the phase margin (PM) can be calculated.

Intersection with the unit circle is achieved by adding the condition$R^2 + I^2 = 1$. Solving for R and I yields:

$$R = 0.5\frac{1 - 2M_p^2}{M_p^2}$$

$$I = -\frac{\sqrt{M_p^2 - 0.25}}{M_p^2}$$

The phase margin $PM$is given by $\tan(PM) = |I|/|R|$.Thus:

$$PM = \tan^{-1}\left(\frac{\sqrt{M_p^2 - 0.25}}{M_p^2 - 0.5}\right)$$

It was earlier stated that specifying PM does not suffice to guarantee a good closed loop performance for all cases. Therefore, the next step is to determine the crossover frequency; i.e. the frequency where the process and controller cross the 0 dB line (open loop).

If the settling time of the closed loop is specified, using the previousM_p definition, it can be obtained:

$$\omega_p = \omega_n\sqrt{1 - 2\zeta^2}$$

from where the natural frequency ω_n can be extracted and used in

$$\omega_{BW} = \omega_n\sqrt{(1 - \zeta^2) + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}$$

to calculate the bandwidth frequency (i.e. the frequency where $P * C$ intersect the -3 dB closed loop magnitude line in Figure-4).

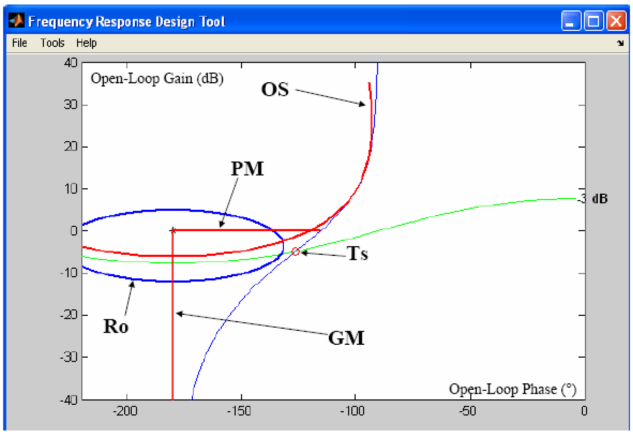


Figure-4. Illustrative example of the Nichols plot FRtool [12] with graphical specifications: OS–overshoot; PM–phase margin; T_s–settling time; R_O–robustness; GM–gain margin; the thin blue line denotes the frequency response of the process and controller $(P * C)$.

Based on the second order closed loop approximation, the phase is given by the integrator from the controller (-90°) and a first order system (-45°). For the phase margin of $PM = 45°$, it follows that ω_c must correspond to the time constant of the first order system. Following the same rationale, the magnitude of the second order approximation starts at -20 dB/dec and around ω_c varies towards -40 dB/dec. This means a decrease of about 7-8 dB between ω_c and ω_{BW} (see open loop magnitude in the Nichols plot from Figure-4). It follows that $\omega_{BW} \approx 1.5\omega_c$ and generalization to higher order systems gives $\omega_c \leq \omega_{BW} \leq 2\omega_c$. From ω_c a sinusoid with period $T_c = \omega_c/2\pi$ is applied to the process and obtain the output:

$$G(j\omega_c) = Me^{j\varphi} = M(\cos\varphi + j\sin\varphi)$$

for this ω_c can be found using the transfer function analyzer algorithm [11,13]. The task is now to find the controller parameters such that the specification for phase margin is ful-filled, given $\%OS$, T_s, M and φ. A typical choice of the PM lies between 40° and 70°: the larger the PM, the more robustness in the loop, less overshoot but larger settling times. Notice that the process is unknown apriori. The controller is derived in its textbook form:

$$R(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

which for the critical frequency becomes:

$$R(j\omega_c) = K_p\left[1 + j\left(T_d\frac{2\pi}{T_c} - \frac{1}{T_i\frac{2\pi}{T_c}}\right)\right]$$

Starting from the controller frequency response, the loop frequency response is given by:

$$\begin{aligned} R(j\omega_c)G(j\omega_c) &= 1e^{j(-180°+PM)} \\ &= \cos(-180° + PM) \\ &\quad + j\sin(-180° + PM) \\ &= -a - jb \end{aligned}$$

with $a = \cos(PM)$ and $b = \sin(PM)$, schematically shown in Figure-5.

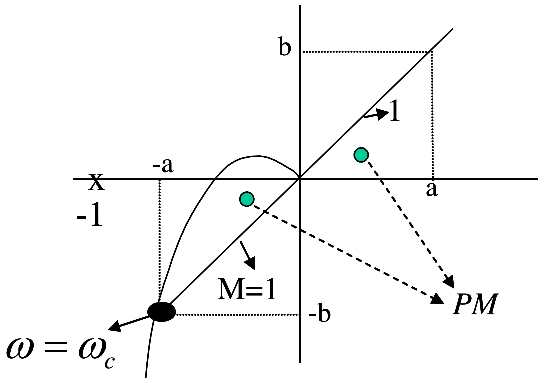


Figure-5. Schematic of the KCR tuning principle.

Based on above equations, the controller is given by:

$$R(j\omega_c) = K_p\left[1 + j\left(T_d\omega_c - \frac{1}{T_i\omega_c}\right)\right] = K_p(1 + j\alpha)$$

where

$$\begin{aligned} K_pM[(\cos\varphi - \alpha\sin\varphi) + j(\sin\varphi + \alpha\cos\varphi)] \\ = -[\cos(PM) + j\sin(PM)] \end{aligned}$$

From the real and imaginary parts:

$$\alpha = \frac{\tan(PM) - \tan\varphi}{1 + \tan(PM)\tan\varphi} = \tan(PM - \varphi) = T_d\omega_c - \frac{1}{T_i\omega_c}$$

and using $T_i = 4T_d$:

$$T_d\omega_c - \frac{1}{4T_d\omega_c} = \tan(PM - \varphi)$$

from where

$$T_d\omega_c = \frac{\sin(PM - \varphi) \pm 1}{2\cos(PM - \varphi)}$$

and the controller parameter

$$T_i = T_c\frac{\sin(PM - \varphi) \pm 1}{\pi\cos(PM - \varphi)}$$

which gives only one positive result. Taking into account that

$$\left(K_pM\right)^2(1 + \alpha) = 1$$

with

$$1 + \alpha^2 = 1 + \tan^2(PM - \varphi) = \frac{1}{\cos^2(PM - \varphi)}$$

which gives the K_p controller parameter:

$$K_p = \pm \frac{\cos(PM - \varphi)}{M}$$

with only one positive result.

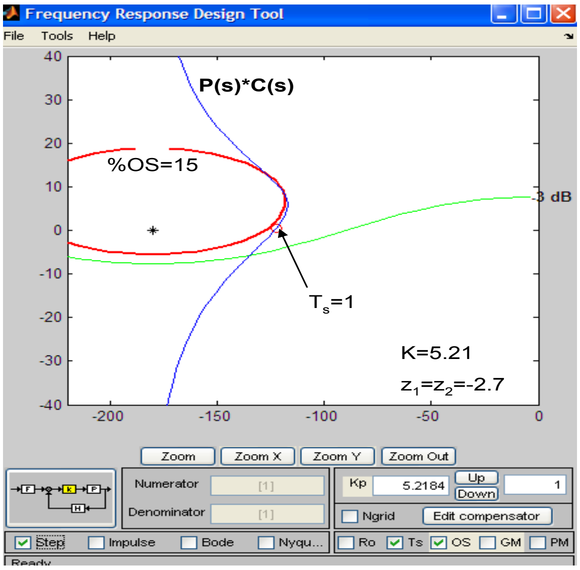
3. RESULTS AND DISCUSSIONS

3.1 Second order system with integrator

Consider a flight control system for position control (it can be as well as servo-control system, antenna, disk drive, DVD, etc.) represented by:

$$P_1(s) = \frac{32}{s(s+4)(s+16)}$$

Notice that the system will introduce an extra integrator in the loop, next to the one of the PID controller. In order to compare the results of the proposed PID auto-tuner, the computer aided design FRtool has been used. A snapshot of the graphical user interface in FRtool and controller parameters are given in Figure-6.



Numerator [1]
Denominator [1]
Kp 5.2184 1
□ Ngrid
☑ Step □ Impulse □ Bode □ Nyqu...
□ Ro ☑ Ts ☑ OS □ GM □ PM

Figure-6.FRtool design for$P_1(s)$.

Both PID controllers have the specifications: $\%OS = 15$ and $T_s = 1$ second. For the KCR auto-tuner, the specification for settling time gives $\omega_c = 6.28$ rad/s it follows $M = 0.03, \varphi = -169°$ and $PM = 51.7°$. The controller parameters are calculated using and given for both controllers in Table-1. Figure-6 shows the design of the PID controller using the knowledge of the transfer function of the process, by means of FRtool.

Table-1. Controller parameters for $P_1(s)$.

PID	K_p	T_i	T_d
FRtool KCR	28.13 19.07	0.74 0.69	0.18 0.17

Figure-7 shows the comparison between the 'best' design possible (FRtool) and the KCR auto-tuner. It is remarkable that the proposed PID auto-tuner can provide similar results to a PID controller designed based on the model of the process.

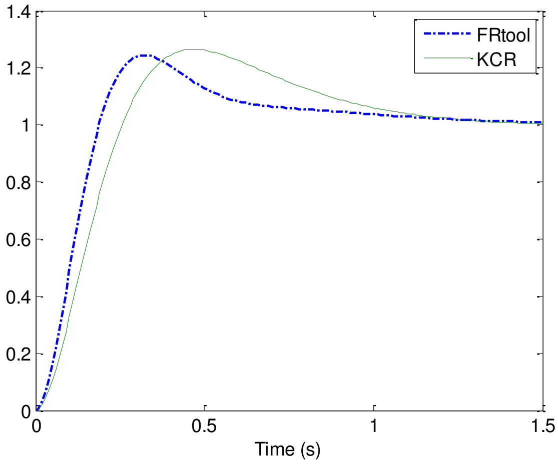


Figure-7. Closed loop step responses for the two PID controllers.

Figure-8 depicts the frequency response plots for the open loop and the closed loop of the system $P_1(s)$ and the KCR controller. The plotted crossover frequency is very close to the value of the calculated crossover frequency: 6 rad/s, corresponding indeed to a phase margin of about 50°. Similarly, the bandwidth frequency is about 10 rad/s, close to the one calculated by the tuning algorithm using. In this case, the second order approximation is close to the actual system dynamics; hence it does not introduce much computational errors in the algorithm.

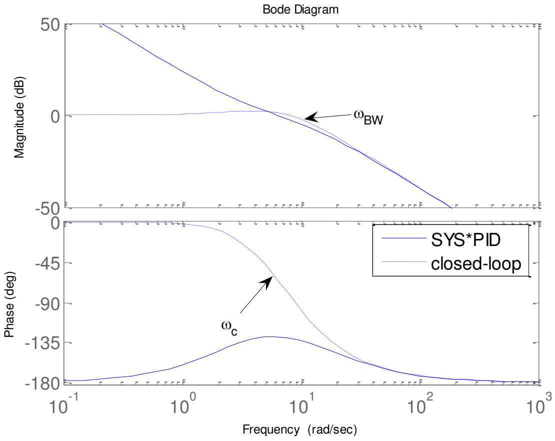


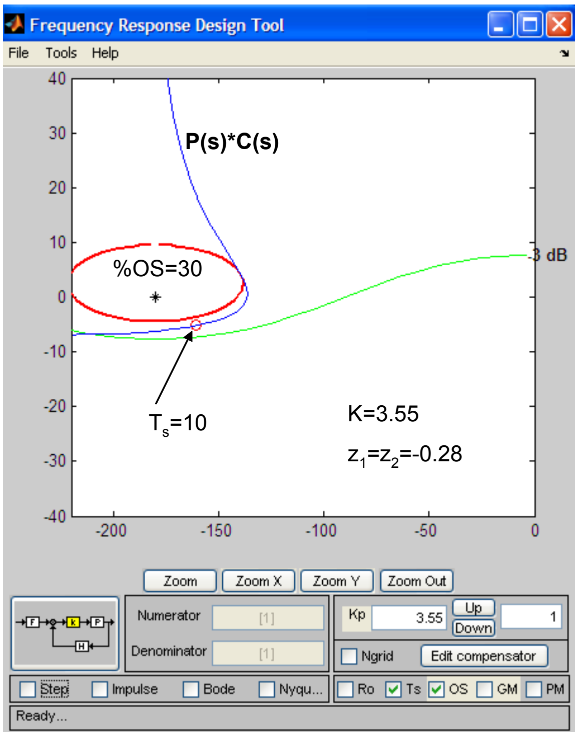
Figure-8. Validation of controller in open loop (SYS*PID) and closed loop for the specified crossover frequency (phase margin) and bandwidth frequency (settling time) for $P_1(s)$.

3.2 Non-minimum phase system with integrator and delay

Consider now a very challenging process, namely a non-minimum phase, with time delay and integrator represented by:

$$P_2(s) = \frac{0.25(1-s)}{s(s+2)} e^{-0.2s}$$

A snapshot of the graphical user interface in FRtool and controller parameters are given in Figure-9.



Numerator [1]
Denominator [1]
Kp 3.55
1
□ Ngrid
□ Step □ Impulse □ Bode □ Nyqu... □ Ro ☑ Ts ☑ OS □ GM □ PM
Ready...

Figure-9. FRtool design for $P_2(s)$.

Both PID controllers have the specifications: $\%OS = 30$ and $T_s = 20$ second. For the KCR auto-tuner, the specification for settling time gives $\omega_c = 0.62$ rad/s it follows$M = 0.29, \varphi = 179°$ and $PM = 35°$. The controller parameters are given in Table-2 for both controllers. Figure-9 shows the design of the PID controller using the knowledge of the transfer function of the process, by means of FRtool.

Table-2. Controller parameters for $P_2(s)$.

PID	K_p	T_i	T_d
FRtool	1.98	7.14	1.78
KCR	2.74	6.33	1.58

Figure 10 shows the comparison between the 'best' design possible (FRtool) and the KCR auto-tuner. Once again the proposed PID auto-tuner can provide similar results to a PID controller designed based on the model of the process.

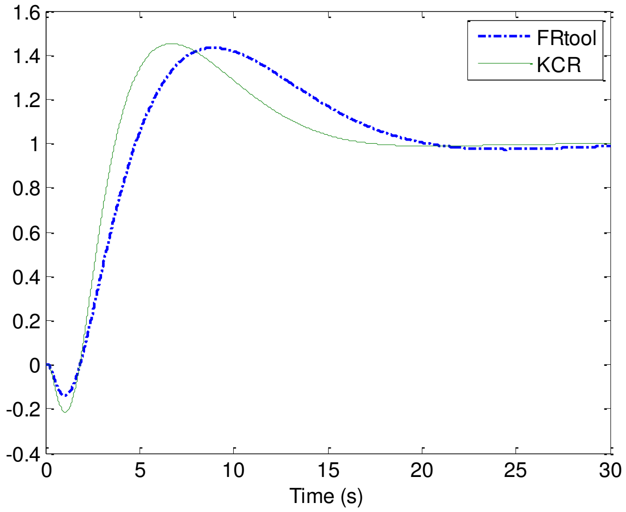


Figure-10. Closed loop step responses for the two PID controllers.

Figure-11 depicts the Bode plot, validating the specification crossover frequency around 0.7 rad/s. Due to the fact that this is an exotic system, the bandwidth frequency has to be seen from the phase plot, whereas the definition of PM is used to find the phase margin -110°, corresponding to +250° in the Bode plot. From here it follows that the bandwidth frequency is about 1.1 rad/s.

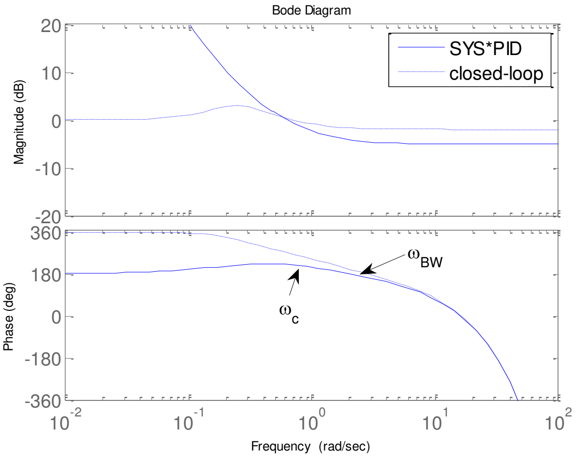


Figure-11. Validation of controller in open loop (SYS*PID) and closed loop for the specified crossover frequency (phase margin) and bandwidth frequency (settling time) for $P_2(s)$

From these examples, which are by far much more difficult than systems closer to a second order approximation, it can be concluded that the PID auto-tuner performs comparable with a PID controller tuned based on the knowledge of the transfer function.

3.3 Conclusions

The development and validation of a generally valid, specifications-based, PID auto-tuner have been presented in this paper. The PID controller has been tested successfully on two illustrative examples: i) a double integrator in the loop and ii) a double integrator, with time delay and non-minimum phase dynamics. The simulation results were promising and ongoing research is focused to apply this PID auto-tuner on several real-life applications.

REFERENCES

[1] D. Bernstein & L. Bushnell. 2002. The history of control: from idea to technology. IEEE Control Systems Magazine. 22(2): 21-23.

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

[3] A., Leva, C. Cox & A. Ruano. 2002. Hands-on PID auto-tuning: a guide to better utilization. IFAC Professional Brief.

[4] K.J., Åström & T. Hägglund. 1984. Automatic tuning of simple regulators with specifications on phase and amplitude margins. Automatica. 20: 645-651.

[5] C.C., Hang, K.J. Åström& W.K. Ho. 1991. Refinements of the Ziegler-Nichols Tuning Formula. In: IEE Proc. Design, Control Theory and Appl. 138: pp. 111-118.

[6] K.J., Åström, C.C. Hang, P. Persson & W.K. Ho. 1992. Towards intelligent PID control. Automatica. 28: 1-9.

[7] J. G. Ziegler & N.B. Nichols. 1942. Optimum settings for automatic controllers. Transactions of ASME. 64: 759-768.

[8] T. S. Schei. 1994. Automatic tuning of PID controllers based on transfer function estimation. Automatica. 30: 1983-1986.

[9] C. Scali, G. Marchetii & D. Semino. 1999. Relay with additional delay for identification and auto-tuning of completely unknown processes. Ind. Eng. Chem. Res. 38: 1987-1997.

[10] R. De Keyser, C. Ionescu. 2010. A Comparative Study of Three Relay-based PID Auto-tuners. Accepted for presentation at the IASTED International Conference on Control and Applications, which will be held July 15, 2010 to July 17, 2010, in Banff, Alberta, Canada.

[11] N.S. Nise. 2007. Control systems engineering, 4th ed., Wiley India Pvt. Ltd.

[12] R. De Keyser & C. Ionescu. 2006. FRtool: a frequency response tool for CACSD in Matlab. In: Proc. of the IEEE Conf. on Computer Aided Control Systems Design, Munich, Germany. 2275-2280.