International Journal of Innovative
Computing, Information and Control
Volume **9**, Number **10**, October **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 4043–4055

AUTOTUNING METHOD FOR $PI^{\lambda}D^{\mu}$ CONTROLLERS DESIGN

Riccardo Caponetto, Giovanni Dongola, Fulvio Pappalardo
and Vincenzo Tomasello

Department of Electrical, Electronic and Computer Engineering
Engineering Faculty
University of Catania
Viale Andrea Doria 6, 95125 Catania, Italy
{riccardo.caponetto; vtomasello}@dieei.unict.it; {gdongola; gpappalardo}@diees.unict.it

Received July 2012; revised November 2012

Abstract. *The paper concerns the design of an autotuning procedure for a fractional order $PI^{\lambda}D^{\mu}$ controller. Given the specification on crossover frequency and phase margin the proposed procedure can be applied to the systems with the order greater than two. The autotuning procedure is divided into two phases: the first one is devoted to the identification of the process at the desired crossover frequency while the second one sets all the parameters of the $PI^{\lambda}D^{\mu}$ controller. The obtained controller ensures an iso-damping response of the closed loop system. The simulation results are given comparing the $PI^{\lambda}D^{\mu}$ controller with the traditional one, confirming the effectiveness of the approach proposed.*
Keywords: Fractional order systems, Auto-tuning, PID control, Iso-damping control

1. **Introduction.** Starting from their introduction a lot of interest is dedicated to fractional order PID controller that assumes the form given in (1), involving an integrator of order λ and a differentiator of order μ, with λ, $\mu \in \mathbb{C}$.

$$PI^{\lambda}D^{\mu} = K_p + K_i\frac{1}{s^{\lambda}} + K_d s^{\mu} \tag{1}$$

$PI^{\lambda}D^{\mu}$ has been introduced in [1] and in the same paper it has been demonstrated that, compared with a classical PID, $\lambda = \mu = 1$, it has a better response if it is used in the control of fractional order systems.

At the moment the research activities are essentially based on the definition of new effective tuning and auto-tuning techniques for $PI^{\lambda}D^{\mu}$ controllers based also on the extension of the classical ones.

In [2, 3], the extension of the derivation and integration orders from the integer to complex numbers provides a more flexible tuning strategy and, as a consequence, an easier way to achieve control requirements with respect to classical PID controllers.

In [4], it has been proposed an optimal $PI^{\lambda}D^{\mu}$ controller based on specified gain and phase margins which guarantees a minimum integral squared error (ISE) criterion.

Another approach, described in [5, 6], is characterized by the use of a new strategy to control first-order systems having a long time delay. In this last work it has been applied a robustness constraint which allows to force the phase of the open-loop system to be flat at the gain crossover frequency.

Further effective robust tuning and auto-tuning strategies have been proposed in [7-11].

The main drawback of $PI^{\lambda}D^{\mu}$ auto-tuning algorithm, see [10], lies in the following: during the phase of plant identification at a frequency with a phase lower than $-180°$ (plant of order greater than two), it is required to look for a negative value for the delay

in the auto-tuning equation. In this paper, it has been proposed an improved relay test routine that allows to overcome this limitation.

From a designer point of view, given the desired crossover frequency ω_{cg} and phase margin φ_m, the prosed procedure allows to design a $PI^{\lambda}D^{\mu}$ able to ensure a closed loop system which is both robust versus gain variations and with an iso-damping step response.

The paper is divided into three sections: in the first one the autotuning procedure for the non-integer $PI^{\lambda}D^{\mu}$ controller is explained; in the second one illustrative examples are given and finally some conclusions are reported.

2. **Plant Identification and Controller Auto-Tuning.** The proposed auto-tuning procedure has been divided into two steps. The first one, the *Relay-test phase*, is used to identify the process at the desired crossover frequency, while the second, the *Autotuning*, allows to determine the controller parameters in order to ensure a robust and iso-damping system response.

After the phases of identification and auto-tuning, Relay and Delay blocks in Figure 1, the switch is commuted on the designed $PI^{\lambda}D^{\mu}$ controller, and the system works in closed loop form.

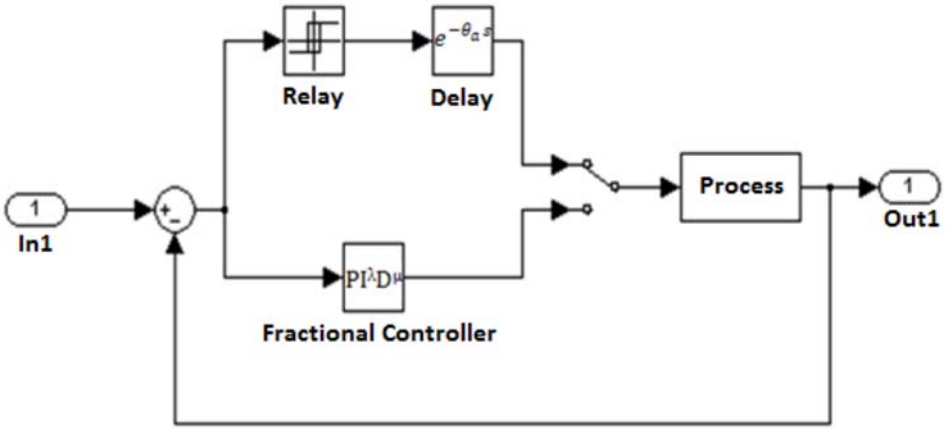

FIGURE 1. Block diagram of the closed loop system

2.1. **Relay-test phase.** The relay auto-tuning process has been widely used for industrial applications, see [12, 13]. The choice of relay feedback is motivated by the possible integration of the system identification and control both in the algorithm and into the control device.

During the relay test phase the ideal relay is substituted with its describing function $N(A) = 4d/A\pi$, where A and d are respectively the input and the output of the relay itself.

By imposing a null reference signal to the closed loop system, see Figure 2, the system goes in free evolution, so the *pseudo-characteristic equation* holds:

$$N(A) * G(j\omega) * e^{-j\omega\theta_a} = -1 \tag{2}$$

In Equation (2), $G(j\omega)$ represents the plant frequency response and θ_a takes into account the presence of the delay.

Equation (2) allows to determine the condition for which the process goes to the limit of stability, approaching the limit cycle oscillations.

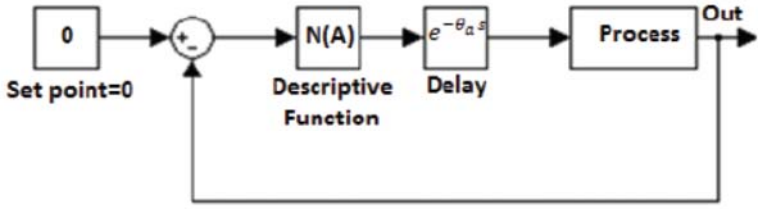

FIGURE 2. Relay test block diagram

When the limit cycle oscillation is held, the plant output signal is a permanent oscillation with fixed amplitude A_c and frequency $\omega_c = \frac{2\pi}{T_c}$.

The magnitude and the phase are therefore given by:

$$|G(j\omega_c)| = \frac{1}{|N(A_c)|} = \frac{A_c\pi}{4d} \tag{3}$$

$$\varphi(j\omega_c) = -\pi + \theta_a\omega_c \tag{4}$$

To identify the process at different frequencies, Equation (4) permits to easily change the oscillation frequency ω_c by acting on the delay θ_a. Varying appropriately θ_a it is therefore possible to identify the system at the desired crossover frequency ω_{cg}.

In order to determine the correct θ_a value, so that $\omega_c = \omega_{cg}$, the following iterative procedure has been implemented:

Relay-test routine

1) Two delays θ_{-1}, θ_0 and the corresponding ω_{-1}, ω_0 are fixed as the initial condition of the algorithm.
2) The iterative relation:

$$\theta_n = \frac{\omega_{cg} - \omega_{n-1}}{\omega_{n-1} - \omega_{n-2}}(\theta_{n-1} - \theta_{n-2}) + \theta_{n-1} \tag{5}$$

   is applied for n steps.
3) If the current value of delay θ_n is negative, a zero is added into the forward chain and the procedure restarts from point 1).
4) If the comparison between ω_c and ω_{cg} is close to zero (≈ 0.01) the procedure is stopped.

As previously introduced, one of the improvements presented in this paper is related to point 3). The possibility to identify the process with an order greater than two has been in fact ensured by adding a zero into the forward chain.

2.2. ***PI$^{\lambda}$D$^{\mu}$* tuning.** From now on the $PI^{\lambda}D^{\mu}$ transfer function assumes the following form:

$$PI^{\lambda}D^{\mu}(s) = \left(\frac{\lambda_1 s + 1}{s}\right)^{\lambda}\left(\frac{\lambda_2 s + 1}{cs + 1}\right)^{\mu} \tag{6}$$

In (6), it is possible to distinguish two different control actions; the *Proportional-integrative action* and the *Proportional-derivative action*.

2.2.1. *Proportional-integrative action.* The proportional-integrative action is obtained through the term:

$$PI^{\lambda}(s) = \left(\frac{\lambda_1 s + 1}{s}\right)^{\lambda} \tag{7}$$

This term is intended to flatten the phase of the system around ω_{cg} in order to obtain a system more robust to the gain variations. With this in view the phase slope ν close to ω_{cg} is computed by using the following relation:

$$\nu = \frac{\varphi_{n+1} - \varphi_{n-1}}{\omega_{n+1} - \omega_{n-1}} \tag{8}$$

where ω_{n-1} and φ_{n-1} are respectively the frequency and the phase at the $n-1$ iteration of the relay-test. The delay θ_{n+1} at the $n+1$ iteration is evaluated via the following relation:

$$\theta_{n+1} = \theta_a + |\theta_a - \theta_{n-1}| \tag{9}$$

The phase of the $PI^{\lambda}(j\omega)$ block is given by:

$$\varphi\left(PI^{\lambda}(j\omega)\right) = \lambda\left(-\arctan\left(\frac{1}{\lambda_1\omega}\right)\right) = \lambda\left(-\frac{\pi}{4} + \arctan(\lambda_1\omega)\right) \tag{10}$$

and its derivate assumes the form:

$$\frac{d\left(\varphi\left(PI^{\lambda}(j\omega)\right)\right)}{d\omega} = \lambda\left(\frac{\lambda_1}{1+(\lambda_1\omega)^2}\right) \tag{11}$$

To obtain a flat phase slope, (11) must assume, at $\omega = \omega_{cg}$, the opposite value of the slope given in (8), so it holds:

$$\lambda\left(\frac{\lambda_1}{1+(\lambda_1\omega_{cg})^2}\right) = -\nu \tag{12}$$

This relation depends both from λ and λ_1. In order to find both values, the first step is to derivate (10) with respect to λ_1 as follows:

$$\frac{d\left(\varphi\left(PI^{\lambda}(j\omega)\right)\right)}{d\lambda_1} = \lambda\left(\frac{1-(\lambda_1\omega_{cg})^2}{\left(1+(\lambda_1\omega_{cg})^2\right)^2}\right) \tag{13}$$

and fixing the derivate equal to zero, the following condition

$$1-(\lambda_1\omega_{cg})^2 = 0 \tag{14}$$

allows to determine λ_1 as:

$$\lambda_1 = \frac{1}{\omega_{cg}} \tag{15}$$

Successively, λ is calculated from (12) and assumes the form:

$$\lambda = -\nu\left(\frac{1+(\lambda_1\omega_{cg})^2}{\lambda_1}\right) = -\nu\left(\frac{2}{\lambda_1}\right) \tag{16}$$

By fixing the previous obtained values of λ and λ_1 the proportional-integrative block has been designed and the following open loop transfer function

$$G_{flat}(s) = PI^{\lambda}(s) * G(s) \tag{17}$$

ensures a flat phase around the crossover frequency ω_{cg}.

2.2.2. *Proportional-derivative action.* The term

$$PD^{\mu}(s) = \left(\frac{\lambda_2 s + 1}{cs + 1}\right)^{\mu} \tag{18}$$

is used to satisfy the phase margin φ_m and the crossover frequency ω_{cg} specifications. The open loop transfer function is now:

$$F(s) = PD^{\mu}(s) * G_{flat}(s) \tag{19}$$

with $s = jw_{cg}$, the previous relation assumes the form:

$$F(j\omega_{cg}) = e^{j(\varphi_m - \pi)} = \cos(\varphi_m - \pi) + j\sin(\varphi_m - \pi) \tag{20}$$

while for Equation (17) it holds:

$$\begin{aligned} G_{flat}(j\omega_{cg}) &= |G_{flat}(j\omega_{cg})| e^{j\varphi\left(G_{flat}(j\omega_{cg})\right)} \\ &= |G_{flat}(j\omega_{cg})| * \left(\cos\left(\varphi\left(G_{flat}(j\omega_{cg})\right)\right) + j\sin\left(\varphi\left(G_{flat}(j\omega_{cg})\right)\right)\right) \end{aligned} \tag{21}$$

Substituting (20) and (21) in (19), the following transfer function for $PD^{\mu}(j\omega_{cg})$ is obtained:

$$PD^{\mu}(j\omega_{cg}) = \left(\frac{\lambda_2 j\omega_{cg} + 1}{x\lambda_2 j\omega_{cg} + 1}\right)^{\mu} = \frac{F(j\omega_{cg})}{G_{flat}(j\omega_{cg})} = a_1 + jb_1 \tag{22}$$

where x takes into account of the high frequency pole added to guarantee the implementation of the controller.

Equation (22) can be rewritten as:

$$\left(\frac{\lambda_2 j\omega_{cg} + 1}{x\lambda_2 j\omega_{cg} + 1}\right) = (a_1 + jb_1)^{\frac{1}{\mu}} = a + jb \tag{23}$$

$$(a_1 + jb_1)^{\frac{1}{\mu}} = \rho^{\frac{1}{\mu}} * e^{j\frac{\Phi}{\mu}} = \rho^{\frac{1}{\mu}}\left(\cos\left(\frac{\Phi}{\mu}\right) + j\sin\left(\frac{\Phi}{\mu}\right)\right)$$

and then:

$$a = \rho^{\frac{1}{\mu}} * \cos\left(\frac{\Phi}{\mu}\right) \tag{24}$$

$$b = \rho^{\frac{1}{\mu}} * \sin\left(\frac{\Phi}{\mu}\right) \tag{25}$$

From Equation (23), the following conditions hold:

$$a = \left(\frac{x(\lambda_2\omega_{cg})^2 + 1}{(x\lambda_2\omega_{cg})^2 + 1}\right) \tag{26}$$

$$b = \left(\frac{\lambda_2\omega_{cg} - x\lambda_2\omega_{cg}}{(x\lambda_2\omega_{cg})^2 + 1}\right) \tag{27}$$

$$\lambda_2 = \left(\frac{b^2 + a(a-1)}{b\omega_{cg}}\right) \tag{28}$$

$$x = \left(\frac{a-1}{b^2 + a(a-1)}\right) \tag{29}$$

By applying the below reported iterative algorithm it is possible to determine x, λ_2 and μ that represent the design parameters of (22).

1) μ is first fixed to a small value (ex. $\mu = 0.48$)
2) from Equations (24) and (25), a and b are calculated
3) from Equations (28) and (29), λ_2 and x are estimated
4) until $x > 0$, $a > 1$ and $b > 0$, μ is iteratively incremented if $x \leq 0$ or $a \leq 1$ or $b \leq 0$ to obtain the minimum value $\mu_{\min}$

5) x and λ_2 are respectively estimated at $\mu_{\min}$

By adding the conditions $a > 1$ and $b > 1$, to the traditional one $x > 0$, and starting with $\mu > 0.48$ the algorithm converges recursively, being more efficient than the one proposed in [11].

3. **Numerical Examples.** In order to validate the procedures introduced in the previous sections, the following system has been considered:

$$G(s) = \left(\frac{1}{s^2(2s+1)}\right) \tag{30}$$

The system $G(s)$ to be identified and controlled is characterized by the following elements: a unity gain, two poles in $s = 0$ and one pole in $s = 0.5$.

In this example the design specifications have been fixed as:

- $\omega_{cg} = 1.97\frac{rad}{sec}$
- $\varphi_m = 60°$
- Gain variations robustness

To start the relay-test routine, the initial delay is fixed to $\theta_{-1} = 0.05s$ obtaining the first oscillating output signal y_{-1} for the closed loop system.

The amplitude, the period and the frequency of this output signal are computed obtaining $A_{-1} = 0.5753v$, $T_{-1} = 2.156s$ and $\omega_{-1} = 2.9143\frac{rad}{s}$.

The same computation is done fixing a new $\theta_0 = 0.1s$ and obtaining $A_0 = 1.1322v$, $T_0 = 3.042s$ and $\omega_0 = 2.0655\frac{rad}{s}$.

Starting from the evaluated couples $(\omega_{-1}, \theta_{-1})$ and (ω_0, θ_0) and by applying Equation (5), the new delay value $\theta_1 = 0.1056s$ is obtained.

The relay-test iterative procedure has been stopped when $|\omega_{cg} - \omega_n| \approx 0.01$.

The iteration results are shown in Table 1 where it is possible to note that the desired cross-over frequency $\omega_u = 1.9727 \cong \omega_{cg}$ is reached at the fourth iteration.

By substituting the value of the amplitude and the oscillation period, obtained at the final iteration, in Equations (3) and (4) it is finally possible to determine the amplitude and the phase of the $G(s)$ at w_{cg}, that take respectively the values $|G(j\omega_{cg})| = -24.2173dB$ and $\varphi\left(G(j\omega_{cg})\right) = -257.6122°$.

As it can be noted, see Figure 3, the real values of the magnitude and phase of $G(s)$ are very close to the estimated ones.

Once the module and the phase of $G(s)$ are determined at the cross-over frequency, the tuning phase of the $PI^{\lambda}D^{\mu}$ controller starts.

The first control block that will be considered is the integral one.

To estimate the phase slope around ω_{cg}, a value of $\theta_{cg-\delta}$ equal to $0.08s$ is applied in Equation (9) obtaining $\theta_{cg+\delta} = 0.1392s$ with the corresponding frequencies: $\omega_{cg-\delta} = 2.3074\frac{rad}{s}$, $\omega_{cg+\delta} = 1.7512\frac{rad}{s}$.

Then from Equation (4) it is possible to determine the phases: $\varphi\left(G(j\omega_{u-\delta})\right) = -2.957\ rad$ and $\varphi\left(G(j\omega_{u+\delta})\right) = -2.8978rad$.

TABLE 1. Delay, amplitude, period and frequency results during the three iterations

n	$\theta_n\ (s)$	$A_n\ (v)$	$T_n\ (s)$	$\omega_n\ (\frac{rad}{s})$
-1	0.05	0.5753	2.156	2.9143
0	0.1	1.1322	3.042	2.0655
1	0.1056	1.1923	3.126	2.01
2	0.1096	1.2365	3.185	1.9727

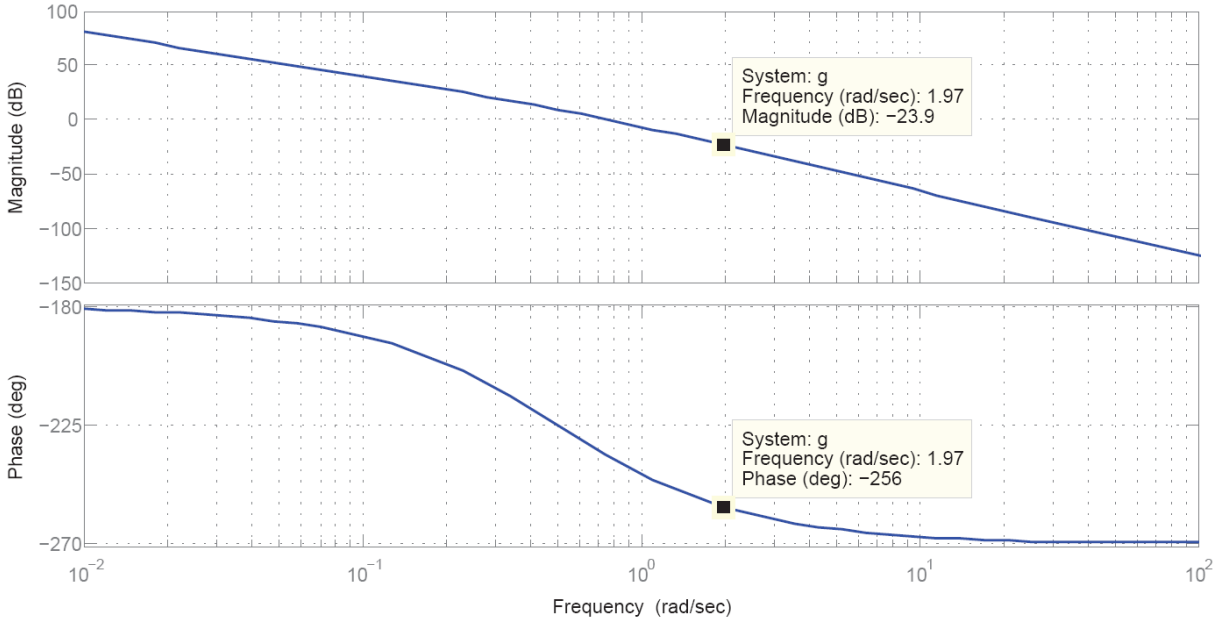

FIGURE 3. $G(j\omega)$ bode diagrams

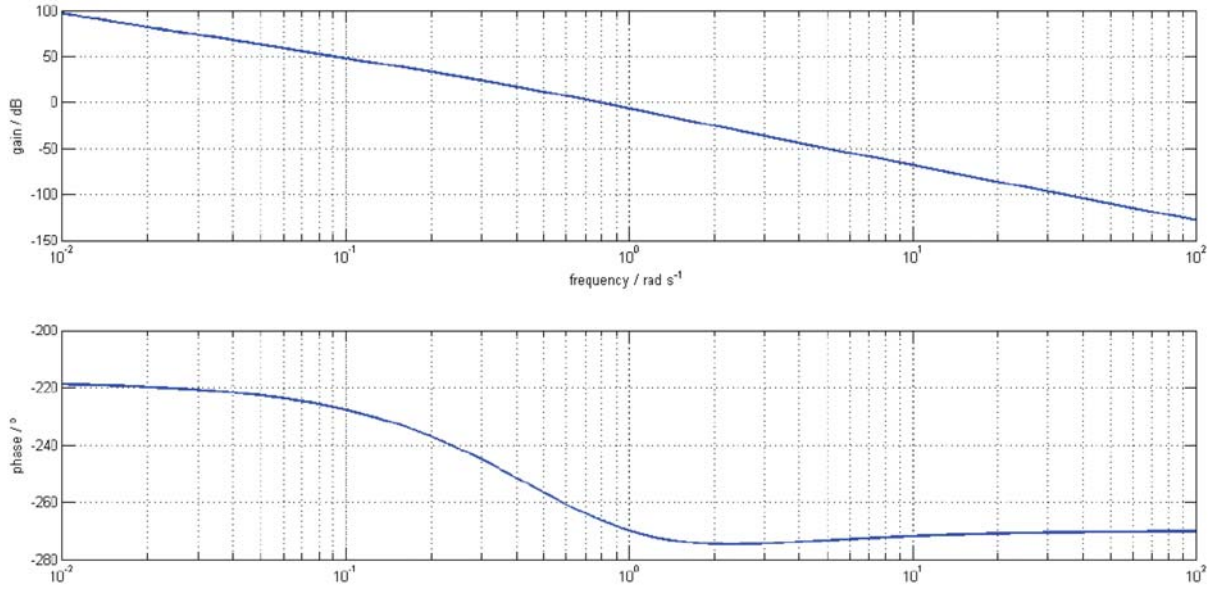

FIGURE 4. $G_{1flat}(j\omega)$ bode diagram

The phase slope $\nu = -0.1064$ is therefore obtained from Equation (8).

Finally from Equations (15) and (16), $\lambda_1 = 0.5069$ and $\lambda = 0.4198$ are determined so that the integral part of the fractional order controller is fixed:

$$PI^{\lambda}(s) = \left(\frac{0.5069s + 1}{s}\right)^{0.4198}$$

From Equation (17) it is possible to compute $|G_{flat}(j\omega_{cg})| = -25.4309dB$ and $\varphi(G_{flat}(j\omega_{cg})) = -276.5°$, showed also in Figure 4.

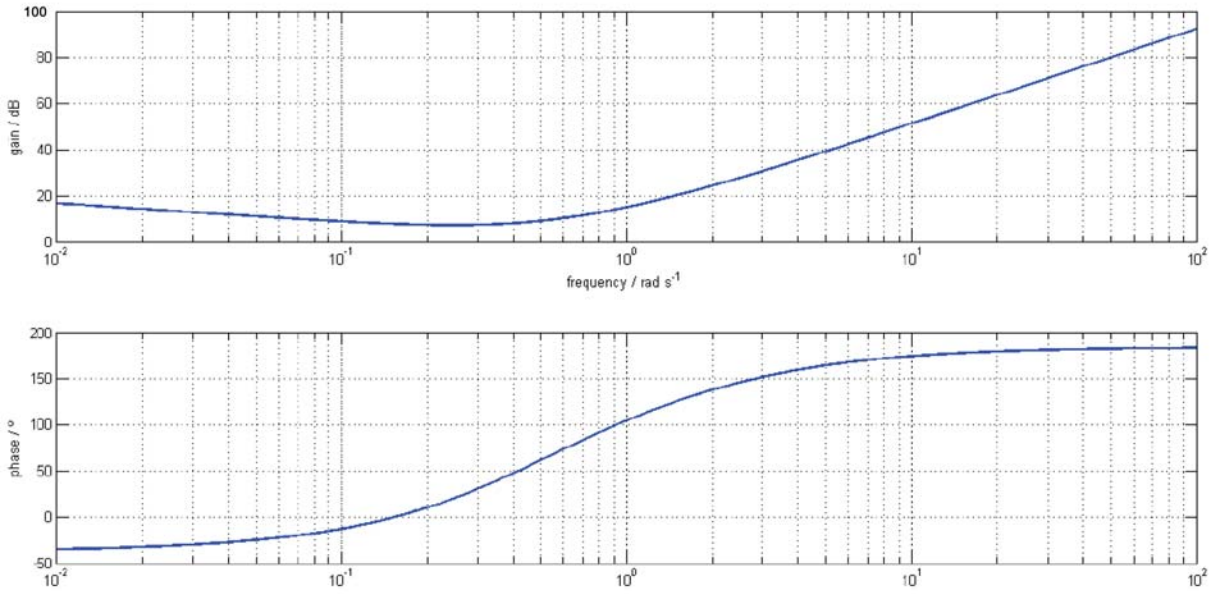

FIGURE 5. $PI^{\lambda}D^{\mu}(j\omega)$ bode diagram

The successive PD^{μ} tuning phase has been performed as described in Section 2.2.2 looking for a $\varphi_m = 60°$. According to the described procedure, the parameters $\mu = 1.826$, $x = 6.45 * 10^{-7}$, $\lambda_2 = 4.4251$ and $c = x * \lambda_2 = 2.8542 * 10^{-6}$ are determined so that the PD^{μ} block assumes the form:

$$PD^{\mu}(s) = \left(\frac{4.4251s + 1}{2.8542 * 10^{-6}s + 1}\right)^{1.826}$$

Respectively in Figures 5 and 6, the bode diagrams of the $PI^{\lambda}D^{\mu}(j\omega)$ controller and the open loop $F(j\omega)$ transfer function are plotted. As it is possible to note the specifications have been respected.

The Simulink model of the $PI^{\lambda}D^{\mu}$ controller is shown in Figure 7, where the integro-differential equations of fractional order have been implemented according to the definition of Grunwald-Letnikov in [14]. By using this definition it is not necessary to approximate the fractional order PID controller with a transfer function of integer and high order so as a consequence the simulation results speed and without approximation.

The step responses of the controlled system (Figure 7) are shown in Figure 8, where it can be observed that the system exhibits robust performances to gain variations, keeping constant the overshoot of the time response.

In the following it is reported the tuning of a standard PID controller on the same plant. The three PID parameters have been fixed applying the closed loop Zielgler-Nochols methods.

In Figure 9, the system output and the values of K_c and T_c used during the tuning procedure are shown.

It is possible to note the small values of the gain K_c and the long T_c period. The controller parameters are reported below, and the step response is depicted in Figure 10. $K_p = 0.6 * 10^{-7}$, $T_i = 1 * 10^4$, $T_d = 2.5 * 10^3$.

Even if the system remains stable the setpoint is reached very slowly.

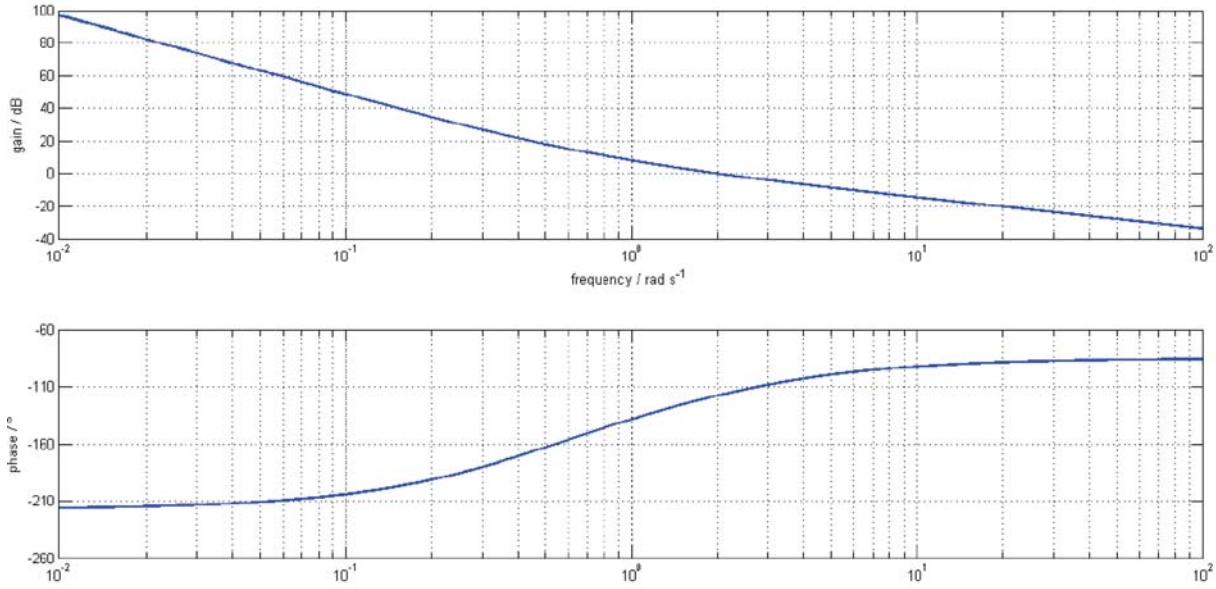

FIGURE 6. $F_1(j\omega)$ bode diagram

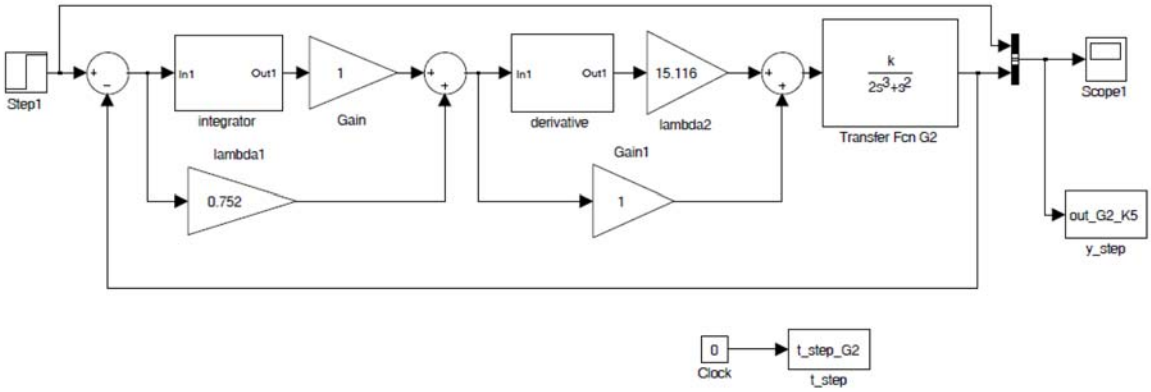

FIGURE 7. Block diagram of the controlled system

As a further example the following fractional order system has been taken into account:

$$G_f(s) = \left(\frac{1}{s(1+s)^{\frac{1}{3}}}\right) \tag{31}$$

In this case the design specifications have been fixed as:

- $\omega_{cg} = 2\frac{rad}{sec}$
- $\varphi_m = 60°$
- Gain variations robustness

In Figure 12, it is reported the bode diagram of the fractional order system.

By applying the proposed auto tuning procedure the following $PI^{\lambda}D^{\mu}$ controller has been determined:

$$PI^{\lambda}D^{\mu}(s) = \left(\frac{0.5s+1}{s}\right)^{0.32}\left(\frac{1061.5s+1}{0.671s+1}\right)^{0.15}$$

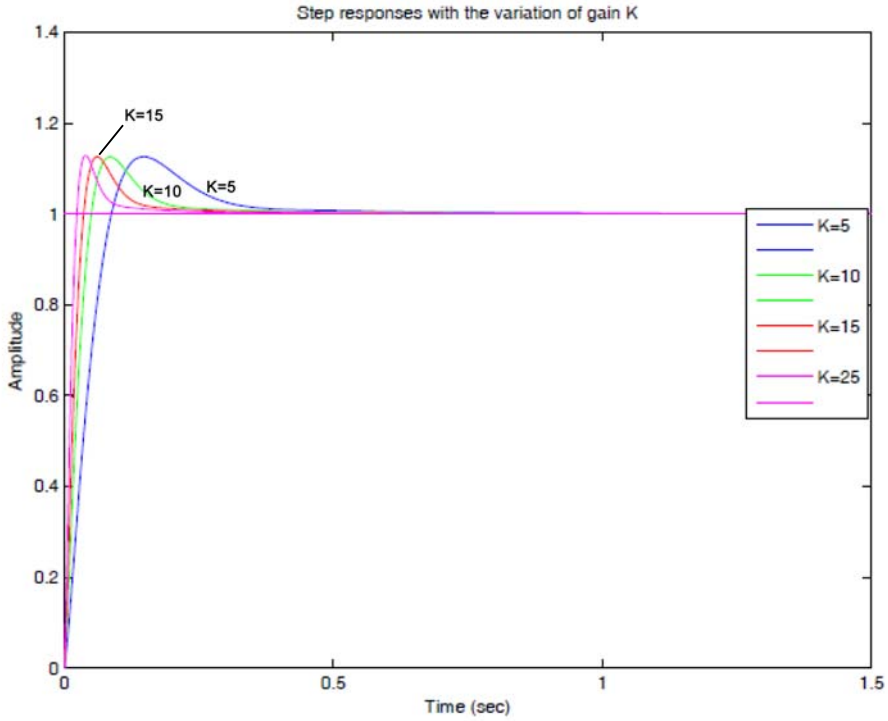

FIGURE 8. Comparison of step responses of the controlled system

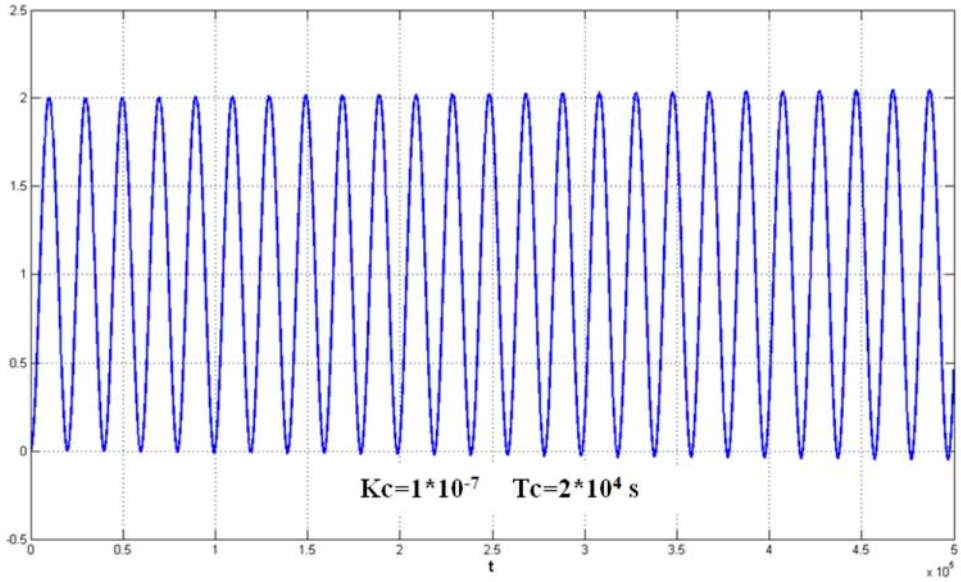

FIGURE 9. Closed loop system output during tuning procedure

Figure 12 shows the Bode diagram of the open loop system:

$$F(s) = PI^{\lambda}D^{\mu}(s) * G_f(s)$$

It is possible to note that the desired specifications are met. Finally, Figures 13 and 14 show respectively the nominal closed loop step response and the step response for different values of the open loop gain.

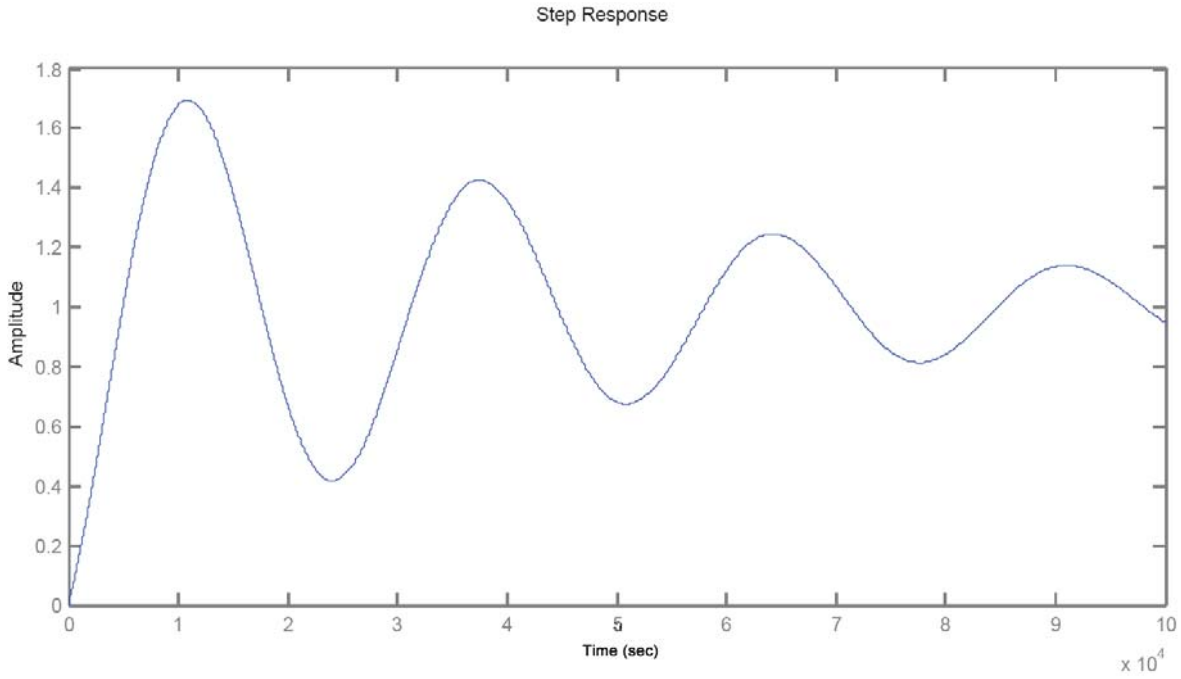

FIGURE 10. Step response of the system with standard PID controller

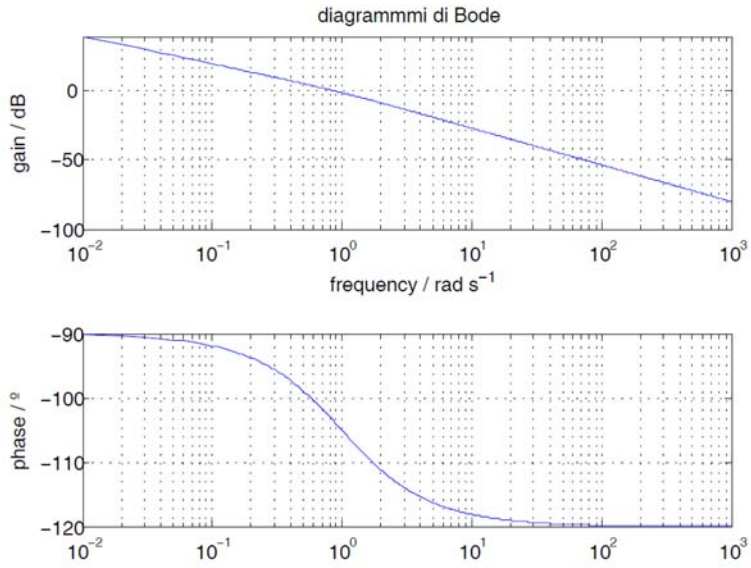

FIGURE 11. $G_f(j\omega)$ bode diagrams

As it happens in the case of fractional order system, the proposed autotuning procedure allows to design effective and robust controller. It is relevant to outline that the proposed designed strategy has been implemented on an HIL (Hardware In the Loop) system, a Dspace board. Both the relay test and the $PI^{\lambda}D^{\mu}$ tuning have been in fact designed to be applied to real world system using the HIL approach. In the previous examples the systems to be controlled have been simulated via their transfer function, the fact remains that can be replaced with the real system interfaced with the HIL device.

4. **Conclusions.** An auto-tuning method for fractional order $PI^{\lambda}D^{\mu}$ controller has been proposed. The method permits a flexible and direct selection of the parameters of the

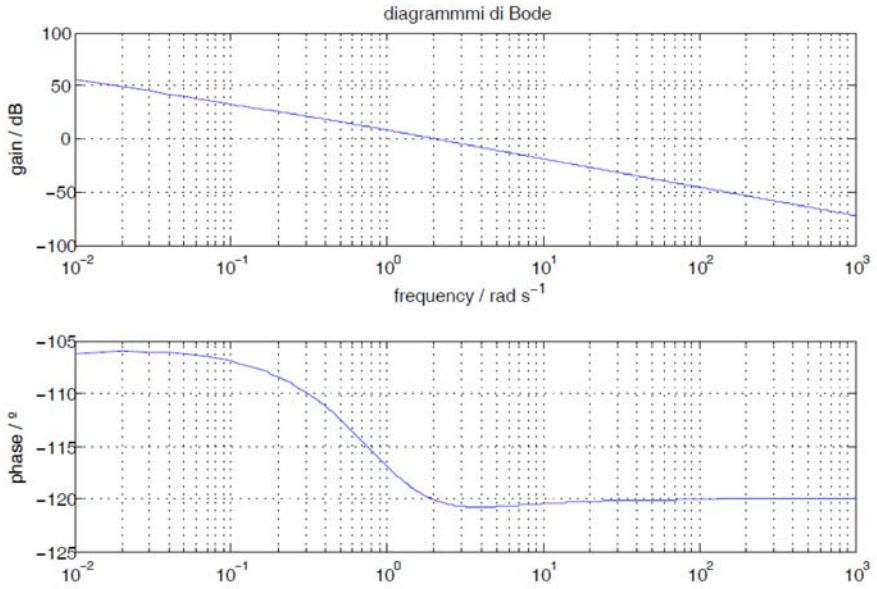

FIGURE 12. $F(j\omega)$ bode diagrams

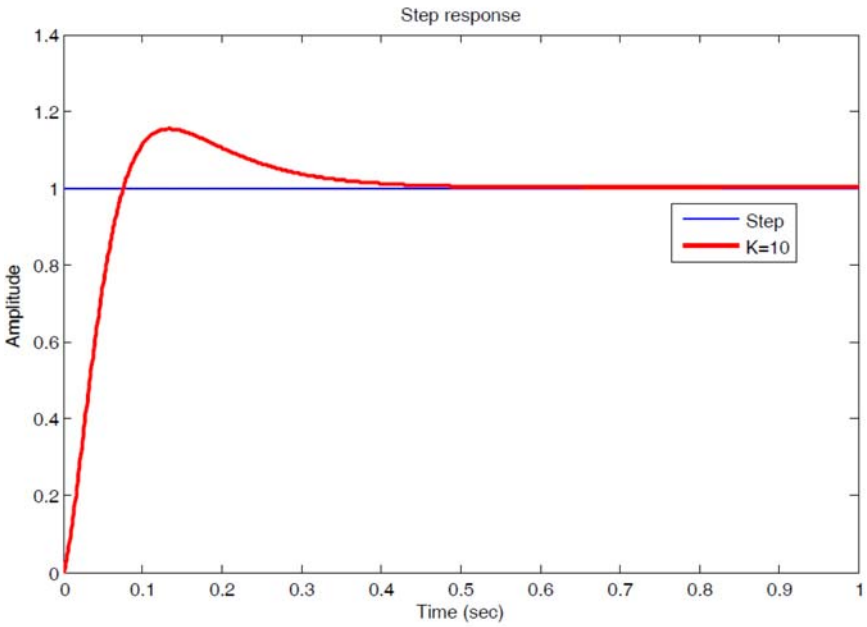

FIGURE 13. Step response

controller through the knowledge of the magnitude and phase of the plant at the frequencies of interest. Specifications on crossover frequency, ω_{cg}, and phase margin, φ_m, can be easily accomplished guaranteeing iso-damping response of the system versus gain variations. Experimental results illustrate the effectiveness of the proposed approach.

Acknowledgment. This work has been supported by the Italian Ministry of University and Research (MIUR) under PRIN projects "Non integer order systems in modeling and control", grant no. 2009F4NZJP.

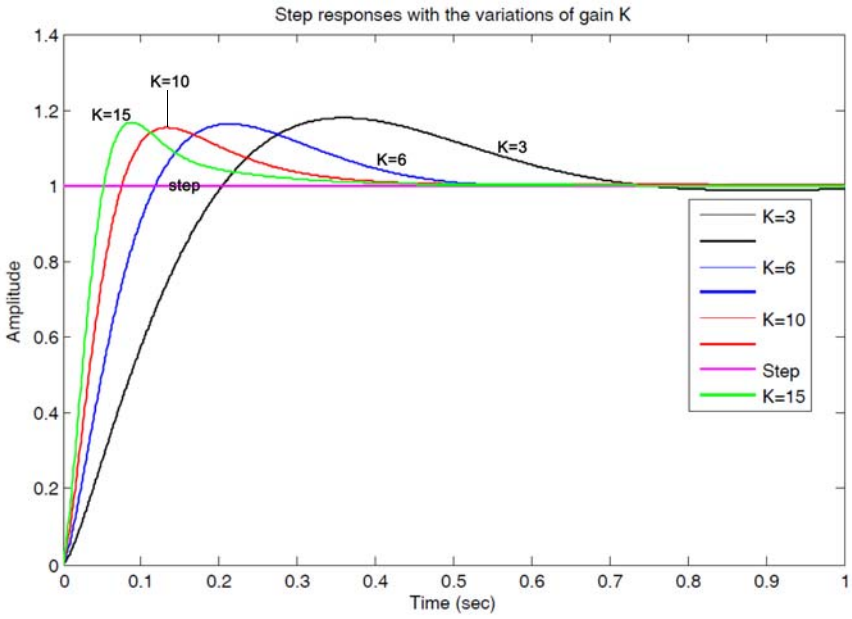

FIGURE 14. Comparison of step responses of the controlled system

REFERENCES

[1] I. Podlubny, Fractional order systems and PID controller, *IEEE Trans. on Automatic Control*, vol.44, no.1, pp.208-214, 1999.

[2] R. Caponetto, L. Fortuna and D. Porto, Parameter tuning of a non-integer order PID controller, *The 15th International Symposium on Mathematical Theory of Networks and Systems*, Notre Dame, Indiana, 2002.

[3] R. Caponetto, L. Fortuna and D. Porto, A new tuning strategy for non integer order PID controller, *The 1st IFAC Workshop on Fractional Differentiation and Its Application*, Bordeaux, France, 2004.

[4] J. F. Leu, S. Y. Tsay and C. Hwang, Design of optimal fractional-order PID controllers, *Journal of the Chinese Institute of Chemical Engineers*, vol.33, no.2, pp.193-202, 2002.

[5] Y. Q. Chen, B. M. Vinagre and C. Monje, A proposition for the implementation of non-integer PI controllers, *Thematic Actionsystems with Non-Integer Derivations*, Bordeaux, France, 2003.

[6] C. Monje, A. Caldern and B. Vinagre, PI vs fractional DI control: First results, *CONTROL 2002: The 5th Portuguese Conference on Automatic Control*, Aveiro, Portugal, pp.359-364, 2002.

[7] D. Valerio and J. S. da Costa, Tuning-rules for fractional PID controllers, *The 2nd IFAC Workshop on Fractional Differentiation and Its Application*, Porto, Portugal, 2006.

[8] Y. Chen, H. Dou, B. M. Vinagre and C. A. Monje, Robust tuning method for fractional order PI controllers, *The 2nd IFAC Workshop on Fractional Differentiation and Its Application*, Porto, Portugal, 2006.

[9] C. A. Monje, B. M. Vinagre, V. Feliu and Y. Chen, On auto-tuning of fractional order $PI\lambda D\mu$ controllers, *The 2nd IFAC Workshop on Fractional Differentiation and Its Application*, Porto, Portugal, pp.19-21, 2006.

[10] A. Monje, B. M. Vinagre, V. Feliu and Y. Chen, Tuning and auto-tuning of fractional order controllers for industry applications, *Control Engineering Practice*, vol.16, pp.798-812, 2008.

[11] G. E. Santamaria, I. Tejado, B. M. Vinagre and A. Monje, Fully automated and tuning and implementation of fractional PID controllers, *Proc. of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, CA, USA, 2009.

[12] C. C. Hang, K. J. Astrom and Q. G. Wang, Relay feedback autotuning of process controllers – A tutorial review, *Journal of Process Control*, vol.12, pp.143-162, 2002.

[13] Y. Q. Chen and K. L. Moore, Relay feedback tuning of robust PID controllers with iso-damping property, *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, vol.35 no.1, pp.23-31, 2005.

[14] K. B. Oldham and J. Spanier, *The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order*, Dover Books on Mathematics, 2006.