International Journal of Innovative Computing, Information and Control
Volume **8**, Number **7(A)**, July **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **4579–4590**

INTERACTIVE TOOLS FOR DESIGNING FRACTIONAL-ORDER PID CONTROLLERS

Sebastián Dormido[1], Enrico Pisoni[2] and Antonio Visioli[2]

[1]Departamento de Informática y Automática
Universidad Nacional de Educación a Distancia (UNED)
Juan del Rosal 16, Madrid 28040, Spain
sdormido@dia.uned.es.

[2]Dipartimento di Ingegneria dell'Informazione
University of Brescia
Viale Branze 38, 25123, Brescia, Italy
{ enrico.pisoni; antonio.visioli }@ing.unibs.it

Received April 2011; revised August 2011

Abstract. *In this paper, we present two Sysquake interactive tools for the design of fractional-order proportional-integral-derivative (FOPID) controllers. The first tool deals with the time and frequency domain design of FOPID controllers, allowing the user to analyze the effects of changing user-chosen parameters. In the time domain, both set-point and load disturbance step responses of the control system are shown, as well as the effect of measurement noise. In the frequency domain, the Bode diagrams of all the most important closed-loop transfer functions are plotted. The second tool allows the user to determine automatically the controller parameters by applying a loop shaping technique, namely, by mapping a point of the process Nyquist plot to a target point of the loop transfer function Nyquist plot with a predefined value of its derivative. In this context, constraints on the gain or phase margin or on the maximum sensitivity can be effectively considered. It is believed that this kind of Computer Aided Control System Design tools are very useful from an educational viewpoint and in allowing a widespread use of FOPID controllers in industry.*

Keywords: Fractional-order PID controllers, CACSD, Tuning, Loop-shaping design, Education

1. **Introduction.** Fractional-Order Proportional-Integral-Derivative (FOPID) controllers have received significant attention in the last decades both from an academic and industrial point of view (see, for example, [1-9] and also [10-12] for more comprehensive reviews). In particular, different tuning methodologies and different tuning rules have been proposed in the literature.

However, in spite of this research effort, the use of FOPID controllers in industry is still quite limited. This can be due to many reasons such as

- the performance improvement that can be obtained by using a FOPID controller has not been fully characterized yet, especially if all the control specifications (set-point following, load disturbance rejection, noise rejection, control effort) are considered;
- simple, effective and robust tuning rules are still not available (especially addressing different control tasks);
- additional functionalities that are well established for standard (integer-order) PID controllers [13] have not been fully developed yet for FOPID controllers.

It is a matter of fact, in any case, that the design of FOPID controllers is a more complex task with respect to the setting of standard integer-order PID controllers. Indeed,

there are five parameters to tune instead of three, if the basic controller expressions are considered, and the physical meaning of the parameters of a FOPID controller is not very intuitive. This is a significant obstacle for the use of this kind of controllers by the process operators.

In order to provide an effective help in this context, two interactive tools implemented in Sysquake [14], which is a software with excellent interactive capabilities, are presented in this paper (note that such a kind of tool is already available for standard PID controllers [15]). Actually, interactive tools have shown to be very useful in the control education field [16-18] because they provide a real-time connection between decisions taken during the design phase and the consequent result. This characteristic is a clear advantage with respect to noninteractive tools, for which an excellent one related to FOPID controller is available [19].

The first interactive tool is related to the basic analysis and design of FOPID controllers [20]. The effects of changing user-chosen parameters are shown both in the time and frequency domain. In the time domain, both set-point and load disturbance step responses of the control system are shown, as well as the effect of measurement noise. In the frequency domain, the Bode diagrams of all the most important closed-loop transfer functions are plotted.

The second interactive tool allows the user to tune the FOPID controller by applying a loop shaping design approach [21]. In particular, the tool determines automatically the controller parameters by mapping a point of the process Nyquist plot to a point of the loop transfer function Nyquist plot with a predefined value of its derivative. In this context, constraints on the gain or phase margin or on the maximum sensitivity can be effectively considered. Indeed, suitable formulae have been devised so that the parameters of a fractional-order PI or PD controller can be determined. Then, the user can change the controller parameters by interactively verifying their effects on the control performance both in the time and frequency domain. The tool is therefore useful for the design of FOPID controllers and, in general, for learning their properties and for understanding their use.

The paper is organized as follows. FOPID controllers are briefly reviewed in Section 2. The basic design interactive tool is then presented in Section 3, while that related to the loop shaping design is described in Section 4. Conclusions are in Section 5.

2. **FOPID Controllers.** The control scheme considered is shown in Figure 1 where C and P are the controller and the process transfer functions respectively, x is the process output, y is the measured output, u is the control variable, r is the reference signal and $e = r - y$ is the control error. Then, d denotes the load disturbance signal and n denotes the measurement noise signal.

The FOPID controller transfer function is based on the definition of the generalized operator ${}_aD_t^\alpha$ (where a and t are the limits and α is the order of the operation), for which the Riemann-Liouville (RL) and the Grünwald-Letnikov (GL) definitions are generally applied. The RL definition is given by ($\alpha > 0$):

$$ {}_aD_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau, \qquad n-1 < \alpha < n \tag{1} $$

where $\Gamma(x)$ is the Gamma function of x.

The GL definition is ($\alpha \in \mathbb{R}$):

$$ {}_aD_t^\alpha f(t) = \lim_{h \to 0} \frac{1}{h^\alpha} \sum_{k=0}^{\left[\frac{t-a}{h}\right]} (-1)^\alpha \binom{\alpha}{k} f(t-kh) \tag{2} $$

where

$$\binom{\alpha}{k} = \frac{\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)} \tag{3}$$

and $[x]$ represents the integer part of x.

Assuming null initial conditions, the Laplace transform $\mathcal{L}$ of a fractional derivative of a signal $f(t)$ (being $f(t)$ a causal function of t and omitting the subscripts for simplicity) is simply given by

$$\mathcal{L}\{D^{\alpha} f(t)\} = s^{\alpha} F(s), \qquad \alpha \in \mathbb{R}. \tag{4}$$

Thus, the classical PID controller can be generalized into a FOPID controller, the so called $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller, whose integro-differential equation can be expressed as:

$$u(t) = K\left(e(t) + \frac{1}{T_i} D^{-\lambda} e(t) + T_d D^{\mu} e(t)\right) \tag{5}$$

where K is the proportional gain, T_i is the integral time constant, T_d is the derivative time constant, λ is the (non-integer) order of the integrator and μ is the (non-integer) order of the derivative action. The corresponding transfer function is expressed as

$$C(s) = \frac{U(s)}{E(s)} = K\left(1 + \frac{1}{T_i s^{\lambda}} + T_d s^{\mu}\right). \tag{6}$$

Note that this controller is in ideal form, but expressing it in parallel form is trivial [13]; by considering $k = K$, $k_i = K/T_i$ and $k_d = K \cdot T_d$ we obtain

$$C(s) = k + \frac{k_i}{s^{\lambda}} + k_d s^{\mu}. \tag{7}$$

It turns out in any case that in the $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller there are five parameters to tune, with respect to the three parameters of the standard PID controller (for which λ and μ are fixed to one). Actually, as typically done for standard PID controllers [13], it is worth filtering the derivative action by means of a first-order filter in order to avoid an excessive amplification of the measurement noise. The value of the filter time constant is chosen as a fraction of the derivative time constant; therefore, the controller transfer function can be written as

$$C(s) = K\left(1 + \frac{1}{T_i s^{\lambda}} + \frac{T_d s^{\mu}}{1 + \frac{T_d}{N} s + 1}\right). \tag{8}$$

This obviously introduces an additional parameter. From one point of view, having more parameters to tune provides more flexibility in the design, but from another point of view they make the design more complex, especially because the physical meaning of the new parameters is not very clear. Indeed, while their interpretation in the frequency domain has been well investigated (see, for example, [22]), the interpretation in the time domain is not very clear, especially if all the different typical control tasks (set-point following, load disturbance rejection, measurement noise rejection) are considered.

Furthermore, in order to implement it, the fractional $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller has to be approximated by an integer-order controller. There are many methods for doing it effectively [23, 24]. In the interactive tools we describe in this paper we consider the Oustaloup's method [25] which consists in approximating the transfer function

$$H(s) = s^{\nu} \ \nu \in \mathbb{R} \tag{9}$$

by a rational function with a recursive distribution of n zeros and poles

$$\tilde{H}(s) = k' \prod_{i=1}^{n} \frac{1 + \frac{s}{\omega_{zi}}}{1 + \frac{s}{\omega_{pi}}} \tag{10}$$

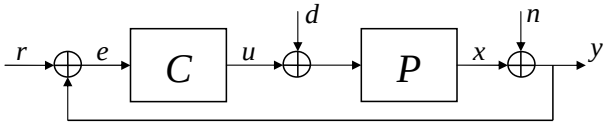

FIGURE 1. The considered control scheme

where k' is an adjusted gain so that if $k = 1$ then the gain is 0 dB for a 1 rad/s frequency. Zeros and poles are determined, by considering a frequency range $[\omega_l, \omega_h]$ for which the approximation is accurate (this is not guaranteed outside this interval), according to the following formulae for $\nu > 0$:

$$\begin{aligned} \xi &= \left(\frac{\omega_h}{\omega_l}\right)^{\frac{\nu}{n}} \\ \eta &= \left(\frac{\omega_h}{\omega_l}\right)^{\frac{1-\nu}{n}} \\ \omega_{z1} &= \omega_l\sqrt{\eta} \\ \omega_{pi} &= \omega_{z,i-1}\xi \qquad i = 1, \ldots, n \\ \omega_{zi} &= \omega_{p,i-1}\eta \qquad i = 1, \ldots, n-1 \end{aligned} \tag{11}$$

For $\nu < 0$ the role of poles and zeros is interchanged.

It appears that the user has also to select the order n of the approximation and the lower and upper frequency ω_l and ω_h of the range for which the approximation is valid. The influence of these parameters on the achieved performance might not be very clear also in this case and therefore there is the need to include them in the interactive tool.

3. **The Interactive Tool for Basic FOPID Control Design.** The tool is similar to that proposed in [15] for standard PID controllers. The main screen of the tool is shown in Figure 2 where it can be seen that on the left-hand side the parameters of the controller can be selected (and performance and robustness indexes can be seen), while the right-hand side is reserved for the plots. Details are given in the next subsections.

3.1. **Process selection.** The kind of process can be selected, among a wide variety (including processes with dead time), by means of a menu (see Figure 3). The process transfer function is shown symbolically in the top left-hand side of the screen, with interactive elements for changing its representative parameters. When the user modifies any of these parameters, the symbolic representation of the process transfer function is immediately updated, as well as all the corresponding graphic elements. It is worth noting that, in addition to those already provided, the user can insert any kind of (single-input-single-output) transfer function by using a simple Matlab-like command.

3.2. **Controller selection.** Five buttons are available for selecting the desired fractional controller type. The buttons correspond to proportional (P), integral (I), proportional-integral (PI), proportional-derivative (PD) and proportional-integral-derivative (PID). Below the buttons, there are sliders which allows to modify the appropriate controller parameters. In particular, there is a slider for the proportional gain, for the integral time constant, for the derivative time constant, for the filter N of the derivative action, for the non-integer order λ of the integrator, for the non-integer order μ of the derivative action, for the number of poles n of the approximating integer controller (see (10)), and for the endpoints ω_l and ω_h of the bandwidth for which the approximation (10) is valid. Note that the number of available sliders changes according to the chosen controller. For example, if a PI controller is selected, then the sliders related to T_d, N and μ disappear.

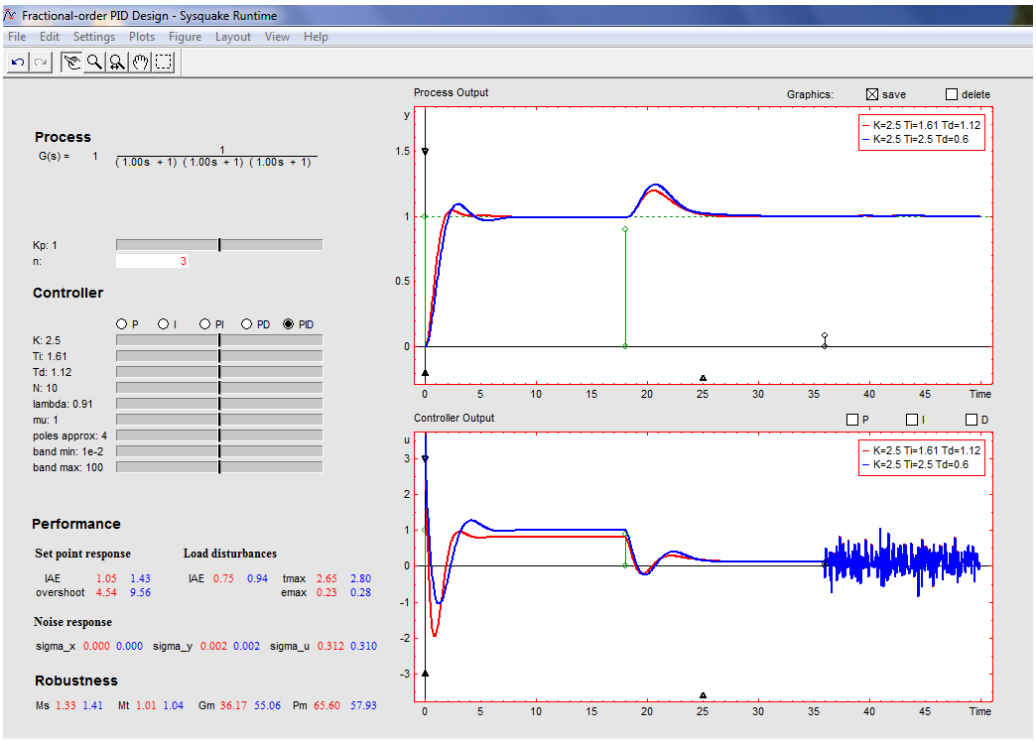

FIGURE 2. The main screen of the first interactive tool

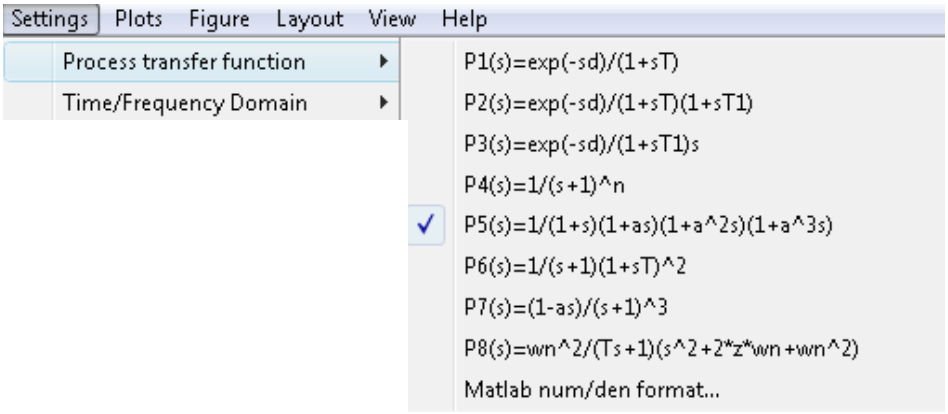

FIGURE 3. Selection of process transfer function

3.3. **Performance and robustness indexes.** Parameters that characterize the performance and the robustness of the control system are also displayed on the screen and updated when the user modifies the controller parameters. The performance criteria are related to the set-point response, the load disturbance response, and the measurement noise response. Regarding the set-point response, the integrated absolute error (IAE) and the overshoot are shown [26]. The integrated absolute error is employed also for the performance of load step disturbance response, together the maximal error (emax), and the time to reach the maximum (tmax). Note that the integrated absolute errors and the maximal error values are normalized to unit step changes in set-point and load disturbances. Finally, the performance indexes related to response to measurement noise are the standard deviations of the process variable (sigma_x), measured output (sigma_y), and control signal (sigma_u).

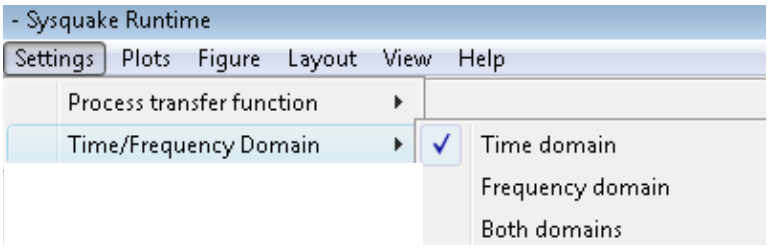

FIGURE 4. Selection of graphic mode

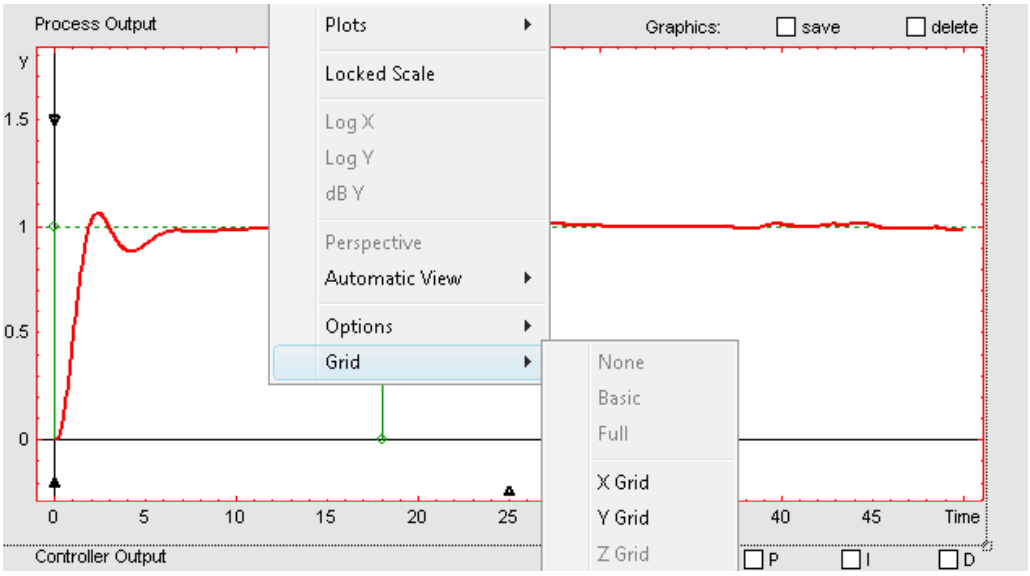

FIGURE 5. Graphical options available

Typical indexes, such as the maximum sensitivity (Ms), the maximum complementary sensitivity (Mt), and the gain (Gm) and phase (Pm) margins are shown in order to give information about the robustness of the control system [26].

3.4. **Graphics.** Two graphics are shown on the right-hand side of the screen. Three representation modes can be selected from the Settings menu (see Figure 4). These modes are time domain, frequency domain, and time/frequency domain. In the first case the time response of the process output is shown in the top part, while the corresponding control variable is shown in the bottom part. In this latter case the contribution of the proportional, integral and derivative actions can be plotted separately. In the first part of the plot, the set-point step response is plotted, while in the second and third part the load step disturbance response and the measurement noise responses are shown respectively. The initial time and the amplitude of the input signals can be easily adjusted by clicking and dragging on the endpoint of the vertical green lines. The vertical and horizontal scales can be changed using the black triangles available in the graphics. By placing the mouse cursor on the plot, the related coordinates are shown automatically. By clicking with the right button of the mouse on a graphic, other options are available (see Figure 5).

Among them, a grid can be inserted to better evaluate the result from a quantitative point of view. In the control variable plot, the three control actions of the fractional $PI^{\lambda}D^{\mu}$ controller can be plotted separately by using the buttons above the plot.

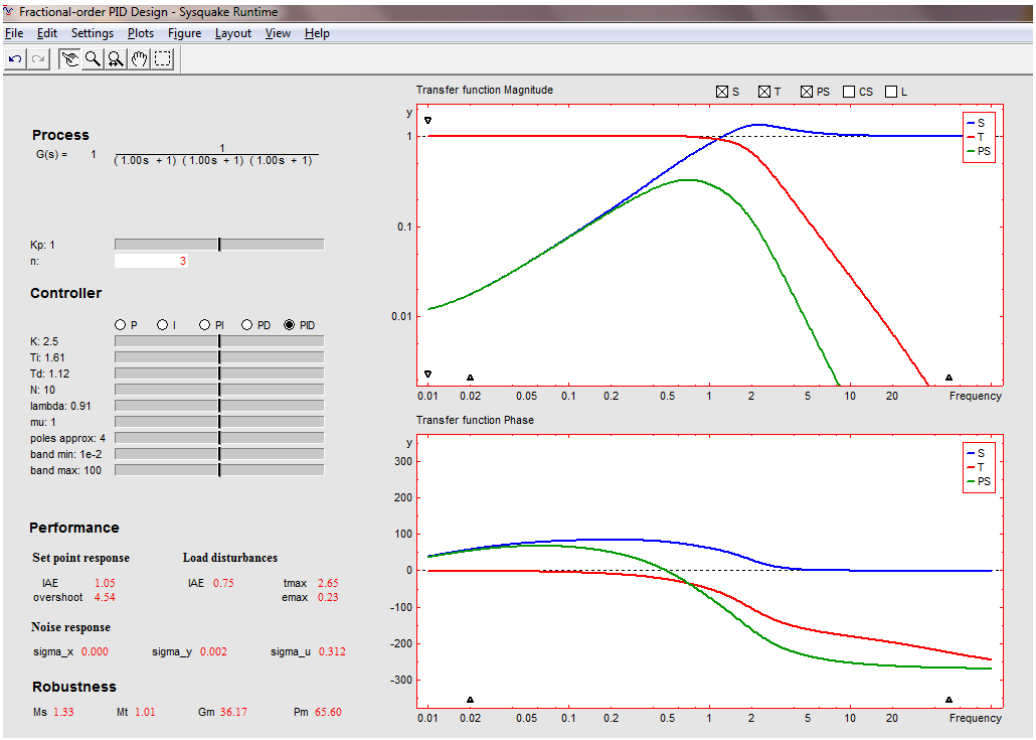

FIGURE 6. An example of the frequency mode

The checkboxes 'save' and 'delete' above the process output graphic make possible to store a simulation for comparison. When the save button is selected, the current design is frozen and displayed in blue, and a new design in red appears, so that the two designs can be compared (see Figure 2). The performance and robustness measures are also shown in the two cases so that a quantitative comparison can be done.

The graphic features for the frequency domain mode are very similar to the previous case. Here one or more Bode plots are shown. Five buttons allow the user to plot the sensitivity function S, the complementary sensitivity function T, the transfer function PS between the load disturbance and the process output, the control sensitivity transfer function CS (namely, the transfer function between the reference signal r and the controller output u), and the open-loop transfer function L. A screenshot in the frequency domain mode is shown in Figure 6.

When both modes are selected, the graphic at the top of the page is devoted to the time domain (the process output or the control variable can be selected alternatively), while that at the bottom is devoted to the frequency domain (the magnitude or the phase of the Bode plot can be selected alternatively).

3.5. **Illustrative example.** As an illustrative example, the process

$$P(s) = \frac{1}{(s+1)^3} \tag{12}$$

has been considered. At the beginning, a standard PID controller (namely, with $\lambda = \mu = 1$) tuned according to the SIMC tuning rules [27] is selected by using the appropriate sliders (it is $K = 2.5$, $T_i = 2.5$, $T_d = 0.6$, $N = 10$). Then, the 'save' checkbox has been selected, so that the design has been saved. Finally, by easily modifying the controller parameters, the values for the FOPID controller have been selected as $K = 2.5$, $T_i = 1.61$, $T_d = 1.12$, $\lambda = 0.91$, $\mu = 1$, $N = 10$. The FOPID controller is approximated by using the

CRONE method with $n = 4$ zeros and poles. The resulting time-domain screenshot in that shown in Figure 2, while the frequency response of the control system is that shown in Figure 6. The performance improvement can be noted by looking at the performance indexes provided by the tool. It is worth noting that the improvement has been obtained easily by exploing the user-friendly interface.

4. **The Interactive Tool for Loop Shaping FOPID Control Design.**

4.1. **Methodology.** The loop shaping design described in [26] is here extended to FOPID controllers. The general idea is to map a point of the process Nyquist plot to a point of the loop transfer function Nyquist plot (the so-called target point). The first derivative of the loop transfer function Nyquist plot at the target point is also imposed.

For this purpose, it is convenient to consider the FOPID controller transfer function (7). Then, given a design frequency ω, the process frequency response can be expressed as

$$P(j\omega) = a + jb = re^{j\phi}. \tag{13}$$

If we consider a FOPD controller (a similar reasoning can be applied to the FOPI controller), the point on the Nyquist plot of the loop transfer function corresponding to the frequency ω can be expressed as

$$L(j\omega) = kP(j\omega) + k_d(j\omega)^\mu P(j\omega) = x + jy. \tag{14}$$

By writing

$$j^\mu = e^{j\frac{\pi}{2}\mu} = \cos\left(\frac{\pi}{2}\mu\right) + j\sin\left(\frac{\pi}{2}\mu\right), \tag{15}$$

we obtain

$$\frac{L(j\omega)}{P(j\omega)} = \left(k + k_d\omega^\mu\cos\left(\frac{\pi}{2}\mu\right)\right) + j\left(k_d\omega^\mu\sin\left(\frac{\pi}{2}\mu\right)\right), \tag{16}$$

that is, we obtain the following two equations:

$$k + k_d\omega^\mu\cos\left(\frac{\pi}{2}\mu\right) = \mathrm{Re}\left(\frac{L(j\omega)}{P(j\omega)}\right) =: A(\omega) \tag{17}$$

$$k_d\omega^\mu\sin\left(\frac{\pi}{2}\mu\right) = \mathrm{Im}\left(\frac{L(j\omega)}{P(j\omega)}\right) =: B(\omega) \tag{18}$$

where it is worth noting that $A(\omega)$ and $B(\omega)$ are known because the original and the target point are known. Thus, (17) and (18) is a system of two equations with three unknowns, namely, k, k_d and μ. A third equation can be obtained by specifying the angle α of the loop transfer function at the target point. In fact, by noting that the derivative of $P(j\omega)$ can be expressed as (see (13))

$$P'(j\omega) = r'e^{j\phi} + jr\phi'e^{j\phi}, \tag{19}$$

we have

$$\begin{aligned}
\tfrac{dL(j\omega)}{d\omega} &= j\tfrac{dC(j\omega)}{dj\omega}P(j\omega) + \tfrac{L(j\omega)}{P(j\omega)}P'(j\omega) \\
&= jk_d\mu j^{\mu-1}\omega^{\mu-1}(a+jb) + (x+jy)\left(\tfrac{r'}{r} + j\phi'\right) \\
&= k_d\mu\omega^{\mu-1}\left[(a+jb)\left(\cos\tfrac{\pi}{2}\mu + j\sin\tfrac{\pi}{2}\mu\right)\right] + (x+jy)\left(\tfrac{r'}{r} + j\phi'\right) \\
&= k_d\mu\omega^{\mu-1}\left[\left(a\cos\tfrac{\pi}{2}\mu - b\sin\tfrac{\pi}{2}\mu\right) + j\left(a\sin\tfrac{\pi}{2}\mu + b\cos\tfrac{\pi}{2}\mu\right)\right] \\
&\quad + \left[\left(x\tfrac{r'}{r} - y\phi'\right) + j\left(x\phi' + y\tfrac{r'}{r}\right)\right]
\end{aligned} \tag{20}$$

Hence, by equating the real and imaginary parts, we obtain:

$$\begin{aligned}\mathrm{Re}\left(\tfrac{dL(j\omega)}{d\omega}\right) &= k_d\mu\omega^{\mu-1}\left(a\cos\tfrac{\pi}{2}\mu - b\sin\tfrac{\pi}{2}\mu\right) + \left(x\tfrac{r'}{r} - y\phi'\right)\\ \mathrm{Im}\left(\tfrac{dL(j\omega)}{d\omega}\right) &= k_d\mu\omega^{\mu-1}\left(a\sin\tfrac{\pi}{2}\mu + b\cos\tfrac{\pi}{2}\mu\right) + \left(x\phi' + y\tfrac{r'}{r}\right)\end{aligned} \tag{21}$$

Then, the following equation

$$\frac{\mathrm{Re}\left(\tfrac{dL(j\omega)}{d\omega}\right)}{\mathrm{Im}\left(\tfrac{dL(j\omega)}{d\omega}\right)} = \tan(\alpha) \tag{22}$$

can be rewritten as

$$\frac{\sin\alpha}{\cos\alpha} = \frac{k_d\mu\omega^{\mu-1}\left(a\sin\tfrac{\pi}{2}\mu + b\cos\tfrac{\pi}{2}\mu\right) + \left(x\phi' + y\tfrac{r'}{r}\right)}{k_d\mu\omega^{\mu-1}\left(a\cos\tfrac{\pi}{2}\mu - b\sin\tfrac{\pi}{2}\mu\right) + \left(x\tfrac{r'}{r} - y\phi'\right)} \tag{23}$$

that is,

$$\begin{aligned}&k_d\mu\omega^{\mu-1}\left[\left(a\sin\tfrac{\pi}{2}\mu + b\cos\tfrac{\pi}{2}\mu\right)\cos\alpha - \left(a\cos\tfrac{\pi}{2}\mu - b\sin\tfrac{\pi}{2}\mu\right)\sin\alpha\right]\\ &= \left(x\phi' + y\tfrac{r'}{r}\right)\cos\alpha + \left(x\tfrac{r'}{r} - y\phi'\right)\sin\alpha.\end{aligned} \tag{24}$$

Eventually, we obtain

$$k_d\mu\omega^{\mu-1}\left[(-a\sin\alpha + b\cos\alpha)\cos\frac{\pi}{2}\mu + (a\cos\alpha + b\sin\alpha)\sin\frac{\pi}{2}\mu\right] = C(\omega) \tag{25}$$

where

$$C(\omega) = \left(x\phi' + y\frac{r'}{r}\right)\cos\alpha - \left(x\frac{r'}{r} - y\phi'\right)\sin\alpha \tag{26}$$

is known. Thus, (17), (18) and (26) represent a system of three equations with three unknowns. By considering that $a = r\cos\phi$ and $b = r\sin\phi$, (25) can be rewritten as

$$-k_d\mu\omega^{\mu-1}r\left[\sin(\phi-\alpha)\cos\frac{\pi}{2}\mu + \cos(\phi-\alpha)\sin\frac{\pi}{2}\mu\right] = C(\omega) \tag{27}$$

and, by diving (27) by (18), we have

$$-\frac{\mu r}{\omega}\left[\sin(\phi-\alpha)\cot\frac{\pi}{2}\mu + \cos(\phi-\alpha)\right] = \frac{C(\omega)}{B(\omega)}. \tag{28}$$

Equation (28) can be solved numerically for μ (it can be easily proven that the solution is unique) and then from (18) we have

$$k_d = \frac{B}{\sin\tfrac{\pi}{2}\mu}\omega^{-\mu} \tag{29}$$

and finally from (17):

$$k = A - k_d\omega^{\mu}\cos\frac{\pi}{2}\mu. \tag{30}$$

As already mentioned, a similar reasoning can be applied to a FOPI controller, where the parameters to be determined are k, k_i and λ.

It is worth stressing at this point that specifying the angle α of the loop transfer function at the target point can be effectively used in order to satisfy stability margin constraints. Indeed, constraints on the gain and phase margins can be addressed by a suitable selection of the target point. Further, by specifying also the angle of the loop transfer function at the target point, a constraint on the maximum sensitivity M_s (which can be represented as a circle centered in the critical point $(-1, 0)$ with radius $1/M_s$), or on the maximum complementary sensitivity M_t (which can be represented as a circle centered in the point $-M_t^2/(M_t^2-1)$ with radius $M_t/(M_t^2-1)$) can be also addressed explicitly. A constraint that considers both the maximum sensitivity M_s and

the maximum complementary sensitivity M_t can be selected by specifying the M circle that encloses both the M_s and M_t circles. The M circle has the centre in the point $(x_1 + x_2)/2$ and radius $(x_1 - x_2)/2$ where

$$x_1 = \max\left\{\frac{M_s + 1}{M_s}, \frac{M_t}{M_t - 1}\right\} \tag{31}$$

and

$$x_2 = \max\left\{\frac{M_s - 1}{M_s}, \frac{M_t}{M_t + 1}\right\}. \tag{32}$$

For further details, see [26].

Remark 4.1. *It is worth noting that, by applying the loop shaping procedure, the stability of the closed-loop system (as well as the positiveness of the controller parameters) is not guaranteed in general for all the selections of the design parameters. However, the user can easily recognize when such a kind of undesirable situations happen.*

Remark 4.2. *The procedure described above involves the solution of the nonlinear equation (28). In order to avoid numerical problems, it is convenient to select a value of μ and then to obtain the values of k_d, k and α directly by using Equations (18), (17) and (22) respectively. In other words, for given values of the design frequency ω and of the target point (x, y), the user can satisfy the value of α by conveniently modifying the value of μ. Note that the value of μ has a very intuitive geometrical interpretation because it represents the angle between the proportional and derivative actions. Indeed, this angle can be changed in an interactive way in the tool.*

4.2. **The interactive tool: overview.** The main screen of the tool is shown in Figure 7 where it can be seen that on the right-hand side both the process and loop transfer function Nyquist plots are shown, while in the left-hand side the parameters of the controller can be selected (and performance and robustness indexes can be seen) and the control system step responses (both for the set-point and load disturbance) are plotted. Plot functionalities (modification of the scale, insertion of a grid, and so on) are again available for improving the presentation of the results.

The process selection is made as in the tool for the basic design (see Section 3.1) and performance and robustness measures are still available.

4.3. **Controller selection.** In the controller section, two kinds of tuning are available. The loop shaping mode automatically selects the controller parameters according to the method described in Section 4.1. Note that when a FOPID controller (where all the three actions are employed) is selected, because a method for the determination of all the five parameters is not available, the loop shaping technique for integer-order PID controllers [26] is actually employed, that is, the values $\lambda = \mu = 1$ are selected. The design constraints (namely, phase margin P_m, gain margin G_m, maximum sensitivity M_s, maximum complementary sensitivity M_t, and M) can be chosen by suitable buttons, while the design frequency and the non-integer parameter λ and μ can be selected by two sliders or by dragging the suitable points on the Nyquist plot.

If the free mode is chosen, the controller parameters can be changed by using the sliders or by dragging the arrows (one for each gain) in the Nyquist plot. In particular, modifying the length of the arrows affects the gains of the controller, while modifying the angles of the integral and derivative action (by clicking on small circles drawn on purpose on the unit circle) affects the values of λ and μ respectively.

It is worth noting that all the controller parameters can also be selected by inserting them by means of an appropriate menu.

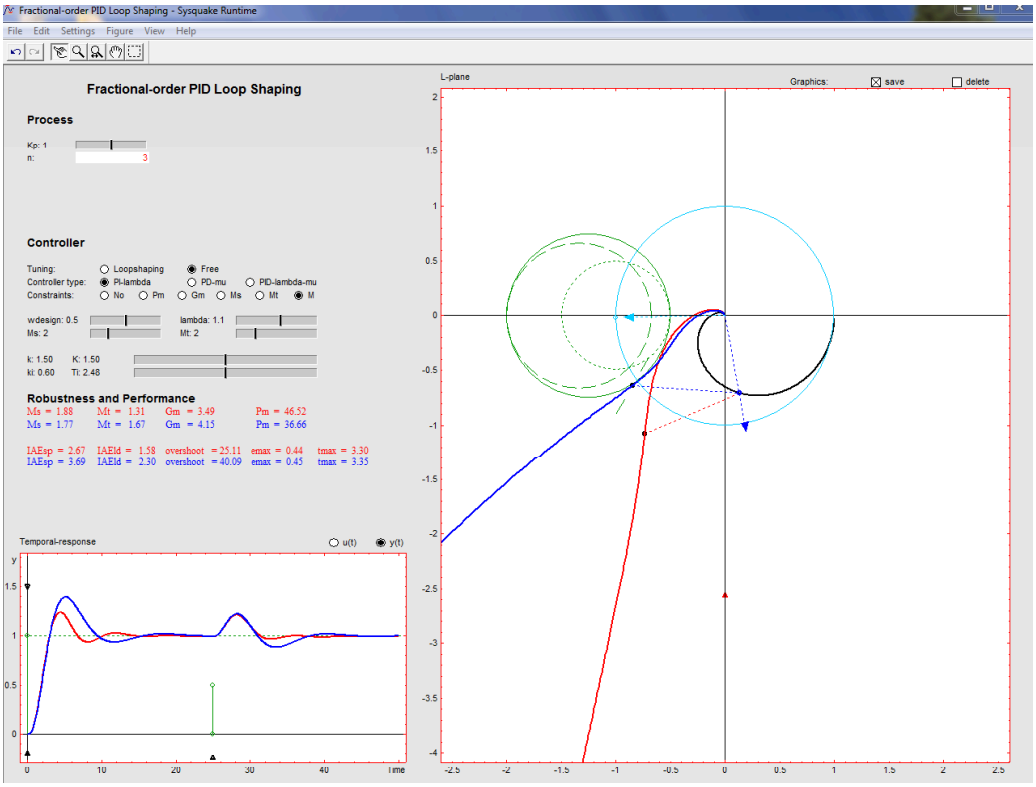

FIGURE 7. The main screen of the loop shaping design interactive tool

4.4. **Example.** As an illustrative example, consider again process (12). After having inserted the process parameters, the PI loop shaping design has been selected with a design frequency $\omega = 0.5$, a constraint on the maximum sensitivity $M_s = 2$ and a constraint on the maximum complementary sensitivity $M_t = 2$. Then, the value of the parameter λ has been modified until the loop transfer function Nyquist plot is tangent to the M circle. It results $\lambda = 1.2$, $k = 1.09$ and $k_i = 0.61$. The obtained result can be evaluated in Figure 7. Then, after having selected the 'save' button, the free mode has been selected and the controller parameters are modified as $\lambda = 1.1$, $k = 1.5$ and $k_i = 0.6$, so that a performance improvement results (see again Figure 7).

Indeed, a satisfactory response can be achieved very easily.

5. **Conclusions.** In this paper, two interactive software tools for the analysis and design of FOPID controllers have been presented. These tools allow the user to verify interactively the effect of changing the different controller parameters and they represent therefore a contribution for a better understanding of fractional controllers, in particular, of the physical meaning of the various involved parameters, both in the time and frequency domain.

The tools can therefore represent a step toward the full characterization of FOPID controllers from an industrial point of view. In fact, the tools can be employed to understand in which situations FOPID controllers can replace the standard PID controllers effectively and to train process operators to use them.

Also from the academic point of view, it is believed that the software tools can be very useful for the research of new methodologies for the design of $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controllers (new tuning rules, performance assessment techniques, and so on), because they allow the researcher to quickly verify the effectiveness of a devised method.

The software tools are freely available upon request to the authors.

REFERENCES

[1] I. Podlubny, Fractional-order systems and $PI^{\lambda}D^{\mu}$ controllers, *IEEE Transactions on Automatic Control*, vol.44, pp.208-214, 1999.

[2] C. A. Monje, B. M. Vinagre, A. J. Calderon, V. Feliu and Y. Q. Chen, On fractional PI^{λ} controllers: Some tuning rules for robustness to plant uncertainties, *Nonlinear Dynamics*, vol.38, pp.369-381, 2004.

[3] D. Valerio and J. S. da Costa, Tuning of fractional PID controllers with Ziegler-Nichols-type rules, *Signal Processing*, vol.86, pp.2771-2784, 2006.

[4] B. M. Vinagre, C. A. Monje, A. J. Calderon and J. I. Suarez, Fractional PID controllers for industry application: A brief introduction, *Journal of Vibration and Control*, vol.13, pp.1419-1429, 2007.

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

[6] S. R. Barbosa, J. A. T. Machado and S. I. Jesus, On the fractional PID control of a laboratory servo system, *Proc. of the 17th IFAC World Congress*, Seoul, Korea, 2008.

[7] Y. Q. Chen, T. Bhaskaran and D. Xue, Practical tuning rule development for fractional order proportional and integral controllers, *ASME Journal of Computational and Nonlinear Dynamics*, vol.3, pp.0214031-0214037, 2008.

[8] A. Biswas, S. Das, A. Abraham and S. Dasgupta, Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution, *Engineering Applications of Artificial Intelligence*, vol.22, pp.343-350, 2009.

[9] F. Padula and A. Visioli, Tuning rules for optimal PID and fractional-order PID controllers, *Journal of Process Control*, 2010.

[10] D. Valerio, *Fractional Robust System Control*, Ph.D. Thesis, University of Lisbon, 2005.

[11] C. A. Monje, *Design Methods of Fractional Order Controllers for Industrial Applications*, Ph.D. Thesis, University of Extremadura, 2006.

[12] D. Valerio and J. S. da Costa, A review of tuning methods for fractional PIDs, *Preprints IFAC Workshop on Fractional Differentiation and Its Applications*, Badajoz, Spain, 2010.

[13] A. Visioli, *Practical PID Control*, Springer, London, UK, 2006.

[14] Y. Piguet, *Sysquake 4 User Manual*, http://www.calerga.com, 2009.

[15] J. L. Guzman, K. J. Astrom, S. Dormido, T. Hagglund, M. Berenguel and Y. Piguet, Interactive modules for PID control, *IEEE Control Systems Magazine*, vol.28, no.5, pp.118-134, 2008.

[16] S. Dormido, Control learning: Present and future, *Annual Reviews in Control*, vol.28, no.1, pp.115-136, 2004.

[17] S. Dormido, S. Dormido-Canto, R. Dormido, J. Sanchez and N. Duro, The role of interactivity in control learning, *International Journal of Engineering Education*, vol.21, no.6, pp.1122-1133, 2005.

[18] J. L. Guzman, *Interactive Control System Design*, Ph.D. Thesis, University of Almeria, Spain, 2006.

[19] D. Valerio, *Ninteger v. 2.3. Fractional Control Toolbox for Matlab*, 2005.

[20] E. Pisoni, A. Visioli and S. Dormido, An interactive tool for fractional order PID controllers, *Proc. of the 35th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Porto, Argentina, pp.1470-1475, 2009.

[21] S. Dormido, E. Pisoni and A. Visioli, An interactive tool for loop-shaping design of fractional order PID controllers, *Proc. of the 4th IFAC Workshop Fractional Differentiation and Its Applications*, Badajoz, Spain, 2010.

[22] C. A. Monje, *Design Methods of Fractional Order Controllers for Industrial Applications*, Ph.D. Thesis, University of Extremadura, Spain, 2006.

[23] B. M. Vinagre, Some approximations of fractional order operators used in control theory and applications, *Fractional Calculus and Applied Analysis*, vol.3, pp.945-950, 2000.

[24] I. Podlubny, I. Petras, B. M. Vinagre, P. O'Leary and L. Dorcak, Analogue realizations of fractional-order controllers, *Nonlinear Dynamics*, vol.29, no.1, pp.281-296, 2002.

[25] A. Oustaloup, F. Levron, B. Mathieu and F. Nanot, Frequency-band complex noninteger differentiator: Characterization and synthesis, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol.47, no.1, pp.25-39, 2000.

[26] K. J. Åström and T. Hägglund, *Advanced PID Control*, ISA Press, Research Triangle Park, USA, 2006.

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