On Fractional-Order PID Design

Mohammad Reza Faieghi and Abbas Nemati
Department of Electrical Engineering,
Miyaneh Branch, Islamic Azad University,
Miyaneh,
Iran

1. Introduction

Fractional-order calculus is an area of mathematics that deals with derivatives and integrals from non-integer orders. In other words, it is a generalization of the traditional calculus that leads to similar concepts and tools, but with a much wider applicability. In the last two decades, fractional calculus has been rediscovered by scientists and engineers and applied in an increasing number of fields, namely in the area of control theory. The success of fractional-order controllers is unquestionable with a lot of success due to emerging of effective methods in differentiation and integration of non-integer order equations.

Fractional-order proportional-integral-derivative (FOPID) controllers have received a considerable attention in the last years both from academic and industrial point of view. In fact, in principle, they provide more flexibility in the controller design, with respect to the standard PID controllers,because they have five parameters to select (instead of three). However, this also implies that the tuning of the controller can be much more complex. In order to address this problem, different methods for the design of a FOPID controller have been proposed in the literature.

The concept of FOPID controllers was proposed by Podlubny in 1997 (Podlubny et al., 1997; Podlubny, 1999a). He also demonstrated the better response of this type of controller, in comparison with the classical PID controller, when used for the control of fractional order systems. A frequency domain approach by using FOPID controllers is also studied in (Vinagre et al., 2000). In (Monje et al., 2004), an optimization method is presented where the parameters of the FOPID are tuned such that predefined design specifications are satisfied. Ziegler-Nichols tuning rules for FOPID are reported in (Valerio & Costa, 2006). Further research activities are runnig in order to develop new tuning methods and investigate the applications of FOPIDs. In (Jesus & Machado, 2008) control of heat diffusion system via FOPID controllers are studied and different tuning methods are applied. Control of an irrigation canal using rule-based FOPID is given in (Domingues, 2010). In (Karimi et al., 2009) the authors applied an optimal FOPID tuned by Particle Swarm Optimzation (PSO) algorithm to control the Automatic Voltage Regulator (AVR) system. There are other papers published in the recent years where the tuning of FOPID controller via PSO such as (Maiti et al., 2008) was investigated.

More recently, new tuning methods are proposed in (Padula & Visioli, 2010a). Robust FOPID design for First-Order Plus Dead-Time (FOPDT) models are reported in (Yeroglu et al., 2010). In (Charef & Fergani, 2010) a design method is reoported, using the impulse response. Set point weighting of FOPIDs are given in (Padula & Visioli, et al., 2010b). Besides, FOPIDs for integral processes in (Padula & Visioli, et al., 2010c), adaptive design for robot manipulators in (Delavari et al., 2010) and loop shaping design in (Tabatabaei & Haeri, 2010) are studied.

The aim of this chapter is to study some of the well-known tuning methods of FOPIDs proposed in the recent literature. In this chapter, design of FOPID controllers is presented via different approaches include optimization methods, Ziegler-Nichols tuning rules, and the Padula & Visioli method. In addition, several interesting illustrative examples are presented. Simulations have been carried out using MATLAB via Ninteger toolbox (Valerio & Costa, 2004). Thus, a brief introduction about the toolbox is given.

The rest of this chapter is organized as follows: In section 2, basic definitions of fractional calculus and its frequency domain approximation is presented. Section 3 introduces the Ninteger toolbox. Section 4 includes the basic concepts of FOPID controllers. Several design methods are presented in sections 5 to 8 and finally, concluding remarks are given in section 9.

2. Fractional calculus

In this section, basic definitions of fractional calculus as well as its approximation method is given.

2.1 Definitions

The differintegral operator, denoted by ${}_aD_t^q$, is a combined differentiation-integration operator commonly used in fractional calculus. This operator is a notation for taking both the fractional derivative and the fractional integral in a single expression and is defined by

$${}_aD_t^q = \begin{cases} \dfrac{d^q}{dt^q} & q > 0 \\ 1 & q = 0 \\ \displaystyle\int_a^t (d\tau)^{-q} & q < 0 \end{cases} \tag{1}$$

Where *q* is the fractional order which can be a complex number and *a* and *t* are the limits of the operation. There are some definitions for fractional derivatives. The commonly used definitions are Grunwald–Letnikov, Riemann–Liouville, and Caputo definitions (Podlubny, 1999b). The Grunwald–Letnikov definition is given by

$${}_aD_t^q f(t) = \frac{d^q f(t)}{d(t-a)^q} = \lim_{N\to\infty}\left[\frac{t-a}{N}\right]^{-q} \sum_{j=0}^{N-1} (-1)^j \binom{q}{j} f(t - j\left[\frac{t-a}{N}\right]) \tag{2}$$

The Riemann–Liouville definition is the simplest and easiest definition to use. This definition is given by

$$ {}_aD_t^q f(t) = \frac{d^q f(t)}{d(t-a)^q} = \frac{1}{\Gamma(n-q)} \frac{d^n}{dt^n} \int_0^t (t-\tau)^{n-q-1} f(\tau) d\tau \tag{3} $$

where *n* is the first integer which is not less than *q* i.e. $n-1 \le q < n$ and Γ is the Gamma function.

$$ \Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt \tag{4} $$

For functions *f(t)* having *n* continuous derivatives for $t \ge 0$ where $n-1 \le q < n$, the Grunwald–Letnikov and the Riemann–Liouville definitions are equivalent. The Laplace transforms of the Riemann–Liouville fractional integral and derivative are given as follows:

$$ L\left\{ {}_0D_t^q f(t) \right\} = s^q F(s) - \sum_{k=0}^{n-1} s^k \, {}_0D_t^{q-k-1} f(0) \qquad n-1 < q \le n \in N \tag{5} $$

Unfortunately, the Riemann–Liouville fractional derivative appears unsuitable to be treated by the Laplace transform technique because it requires the knowledge of the non-integer order derivatives of the function at $t=0$. This problem does not exist in the Caputo definition that is sometimes referred as smooth fractional derivative in literature. This definition of derivative is defined by

$$ {}_aD_t^q f(t) = \begin{cases} \dfrac{1}{\Gamma(m-q)} \displaystyle\int_0^t \frac{f^{(m)}(\tau)}{(t-\tau)^{q+1-m}} d\tau & m-1 < q < m \\ \dfrac{d^m}{dt^m} f(t) & q = m \end{cases} \tag{6} $$

where *m* is the first integer larger than *q*. It is found that the equations with Riemann–Liouville operators are equivalent to those with Caputo operators by homogeneous initial conditions assumption. The Laplace transform of the Caputo fractional derivative is

$$ L\left\{ {}_0D_t^q f(t) \right\} = s^q F(s) - \sum_{k=0}^{n-1} s^{q-k-1} f^{(k)}(0) \qquad n-1 < q \le n \in N \tag{7} $$

Contrary to the Laplace transform of the Riemann–Liouville fractional derivative, only integer order derivatives of function *f* are appeared in the Laplace transform of the Caputo fractional derivative. For zero initial conditions, Eq. (7) reduces to

$$ L\left\{ {}_0D_t^q f(t) \right\} = s^q F(s) \tag{8} $$

In the rest of this paper, the notation D^q, indicates the Caputo fractional derivative.

2.2 Approximation methods

The numerical simulation of a fractional differential equation is not simple as that of an ordinary differential equation. Since most of the fractional-order differential equations do not have exact analytic solutions, so approximation and numerical techniques must be used. Several analytical and numerical methods have been proposed to solve the fractional-order differential equations. The method which is considered in this chapter is based on the approximation of the fractional-order system behavior in the frequency domain. To simulate a fractional-order system by using the frequency domain approximations, the fractional order equations of the system is first considered in the frequency domain and then Laplace form of the fractional integral operator is replaced by its integer order approximation. Then the approximated equations in frequency domain are transformed back into the time domain. The resulted ordinary differential equations can be numerically solved by applying the well-known numerical methods.

One of the best-known approximations is due to Oustaloup and is given by (Oustaloup, 1991)

$$s^q = k\prod_{n=1}^{N}\frac{1+\frac{s}{\omega_{zn}}}{1+\frac{s}{\omega_{pn}}} \quad q>0 \tag{9}$$

The approximation is valid in the frequency range $[\omega_l, \omega_h]$; gain k is adjusted so that the approximation shall have unit gain at 1 rad/sec; the number of poles and zeros N is chosen beforehand (low values resulting in simpler approximations but also causing the appearance of a ripple in both gain and phase behaviours); frequencies of poles and zeros are given by

$$\alpha = (\frac{\omega_h}{\omega_l})^{\frac{q}{N}} \tag{10}$$

$$\eta = (\frac{\omega_h}{\omega_l})^{\frac{1-q}{N}} \tag{11}$$

$$\omega_{z1} = \omega_l\sqrt{\eta} \tag{12}$$

$$\omega_{zn} = \omega_{p,n-1}\eta, \quad n = 2,...,N \tag{13}$$

$$\omega_{pn} = \omega_{z,n-1}\alpha, \quad n = 1,...,N \tag{14}$$

The case $q<0$ may be dealt with inverting (9).

In Table 1, approximations of $1/s^q$ have been given for $q \in \{0.1, 0.2, ..., 0.9\}$ with maximum discrepancy of 2 dB within (0.01, 100) rad/sec frequency range (Ahmad & Sprott, 2003).

q	Approximated transfer function
0.1	$\frac{1584.8932(s+0.1668)(s+27.83)}{(s+0.1)(s+16.68)(s+2783)}$
0.2	$\frac{79.4328(s+0.05623)(s+1)(s+17.78)}{(s+0.03162)(s+0.5623)(s+10)(s+177.8)}$
0.3	$\frac{39.8107(s+0.0416)(s+0.3728)(s+3.34)(s+29.94)}{(s+0.02154)(s+0.1931)(s+1.73)(s+15.51)(s+138.9)}$
0.4	$\frac{35.4813(s+0.03831)(s+0.261)(s+1.778)(s+12.12)(s+82.54)}{(s+0.01778)(s+0.1212)(s+0.8254)(s+5.623)(s+38.31)(s+261)}$
0.5	$\frac{15.8489(s+0.03981)(s+0.2512)(s+1.585)(s+10)(s+63.1)}{(s+0.01585)(s+0.1)(s+0.631)(s+3.981)(s+3.981)(s+25.12)(s+158.5)}$
0.6	$\frac{10.7978(s+0.04642)(s+0.3162)(s+2.154)(s+14.68)(s+100)}{(s+0.01468)(s+0.1)(s+0.631)(s+4.642)(s+31.62)(s+215.4)}$
0.7	$\frac{9.3633(s+0.06449)(s+0.578)(s+5.179)(s+46.42)(s+416)}{(s+0.01389)(s+0.1245)(s+1.116)(s+10)(s+89.62)(s+803.1)}$
0.8	$\frac{5.3088(s+0.1334)(s+2.371)(s+42.17)(s+749.9)}{(s+0.01334)(s+0.2371)(s+4.217)(s+74.99)(s+1334)}$
0.9	$\frac{2.2675(s+1.292)(s+215.4)}{(s+0.01292)(s+2.154)(s+359.4)}$

Table 1. Approximation of $1/s^q$ for different q values

3. The Ninteger toolbox

Ninteger is a toolbox for MATLAB intended to help developing fractional-order controllers and assess their performance. It is freely downloadable from the internet and implements fractional-order controllers both in the frequency and the discrete time domains. This toolbox includes about thirty methods for implementing approximations of fractional-order and three identification methods. The Ninteger toolbox allow us to implement, simulate and analyze FOPID controllers easily via its functions. In the rest of this chapter, all the simulation studies have been carried out using the Ninteger toolbox.

In order to use this toolbox in our simulation studies, the function *nipid* is suitable for implementing FOPID controllers. The toolbox allow us to implement this function either from *command window* or *SIMULINK*. In order to use *SIMULINK*, a library is provided called *Nintblocks*. In this library, one can find the *Fractional PID* block which implements FOPID controllers. We can specify the following parameters of a FOPID via *nipid* function or *Fractional PID* block:

- proportional gain
- derivative gain
- fractional derivative order
- integral gain
- fractional integral order

- bandwidth of frequency domian approximation
- number of zeros and poles of the approximation
- the approximating formula

It was pointed out in (Oustaloup et al., 2000) that a band-limit implementation of fractional order controller is important in practice, and the finite dimensional approximation of the fractional order controller should be done in a proper range of frequencies of practical interest. This is true since the fractional order controller in theory has an infinite memory and some sort of approximation using finite memory must be done.

In the simulation studies of this chapter, we will use the *Crone* method within the frequency range (0.01, 100) rad/s and the number of zeros and poles are set to 10.

4. Fractional-order Proportional-Integral-Derivative controller

The most common form of a fractional order PID controller is the $PI^{\lambda}D^{\mu}$ controller (Podlubny, 1999a), involving an integrator of order *λ* and a differentiator of order *μ* where *λ* and *μ* can be any real numbers. The transfer function of such a controller has the form

$$G_c(s) = \frac{U(s)}{E(s)} = k_P + k_I \frac{1}{s^{\lambda}} + k_D s^{\mu}, \quad (\lambda, \mu > 0) \tag{15}$$

where *$G_c(s)$* is the transfer function of the controller, *E(s)* is an error, and *U(s)* is controller's output. The integrator term is $1/s^{\lambda}$, that is to say, on a semi-logarithmic plane, there is a line having slope *-20λ* dB/decade. The control signal *u(t)* can then be expressed in the time domain as

$$u(t) = k_P e(t) + k_I D^{-\lambda} e(t) + k_D D^{\mu} e(t) \tag{16}$$

Fig. 1 is a block-diagram configuration of FOPID. Clearly, selecting *λ = 1* and *μ = 1*, a classical PID controller can be recovered. The selections of *λ = 1*, *μ = 0*, and *λ = 0*, *μ = 1* respectively corresponds conventional PI & PD controllers. All these classical types of PID controllers are the special cases of the fractional $PI^{\lambda}D^{\mu}$ controller given by (15).

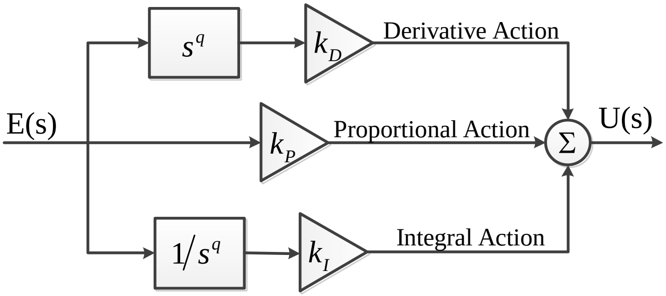

Fig. 1. Block-diagram of FOPID

It can be expected that the $PI^{\lambda}D^{\mu}$ controller may enhance the systems control performance. One of the most important advantages of the $PI^{\lambda}D^{\mu}$ controller is the better control of dynamical systems, which are described by fractional order mathematical models. Another advantage lies in the fact that the $PI^{\lambda}D^{\mu}$ controllers are less sensitive to changes of parameters of a controlled system (Xue et al., 2006). This is due to the two extra degrees of freedom to better adjust the dynamical properties of a fractional order control system. However, all these claimed benefits were not systematically demonstrated in the literature. In the next sections, different design methods of FOPID controllers are discussed. In all cases, we considered the unity feedback control scheme depicted in Fig.2.

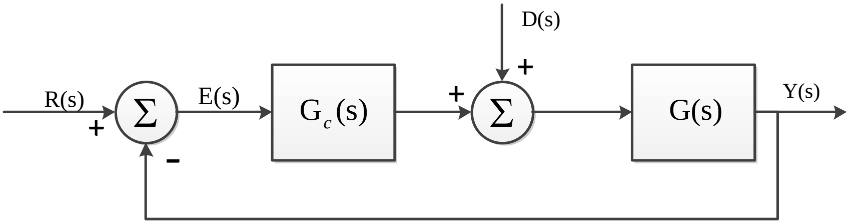

Fig. 2. The considered control scheme; G(s) is the process, $G_c(s)$ is the FOPID controller, R(s) is the reference input, E(s) is the error, D(s) is the disturbance and Y(s) is the output

5. Tuning by minimization

In (Monje et al., 2004) an optimization method is proposed for tuning of FOPID controllers. The analytic method, that lies behind the proposed tuning rules, is based on a specified desirable behavior of the controlled system. We start the section with basic concepts of this design method, and then control pH neutralization process is presented as an illustrative example.

5.1 Basic concepts

In this method, the desirable dynamics is described by the following criteria:

1. No steady-state error:

Properly implemented a fractional integrator of order $k+\lambda$, $k \in N$, $0 < \lambda < 1$, is, for steady-state error cancellation, as efficient as an integer order integrator of order $k + 1$.

2. The gain-crossover frequency ω_{cg} is to have some specified value

$$\left|G_c(j\omega_{cg})G(j\omega_{cg})\right| = 0 \text{ dB} \tag{17}$$

3. The phase margin φ_m is to have some specified value

$$-\pi + \varphi_m = \arg\left(G_c(j\omega_{cg})G(j\omega_{cg})\right) \tag{18}$$

4. So as to reject high-frequency noise, the closed loop transfer function must have a small magnitude at high frequencies; thus it is required that at some specified frequency ω_t its magnitude be less than some specified gain

$$\left|T(j\omega) = \frac{G_c(j\omega)G(j\omega)}{1 + G_c(j\omega)G(j\omega)}\right| < A \text{ dB } \forall\ \omega \geq \omega_t \rightarrow \left|T(j\omega)\right| = A \text{ dB} \tag{19}$$

5. So as to reject output disturbances and closely follow references, the sensitivity function must have a small magnitude at low frequencies; thus it is required that at some specified frequency ω_s its magnitude be less than some specified gain

$$\left| S(j\omega) = \frac{1}{1+G_c(j\omega)G(j\omega)} \right| < B \text{ dB } \forall\ \omega \leq \omega_s \rightarrow \left| S(j\omega) \right| = B \text{ dB} \tag{20}$$

6. So as to be robust in face of gain variations of the plant, the phase of the open-loop transfer function must be (at least roughly) constant around the gain-crossover frequency

$$\frac{d}{d\omega} \arg\left(G_c(j\omega)G(j\omega) \right) |_{\omega=\omega cg} = 0 \tag{21}$$

A set of five of these six specifications can be met by the closed-loop system, since the FOPID has five parameters to tune. The specifications 2-6 yield a robust performance of the controlled system against gain changes and noise and the condition of no steady-state error is fulfilled just with the introduction of the fractional integrator properly implemented, as commented before.

In (Monje et al., 2004), the use of numerical optimization techniques is proposed to satisfy the specifications 2-6. Motivated from the fact that the complexity of a set of five nonlinear equations (17-21) with five unknown parameters (k_P, k_I, k_D, λ and *μ)* is very significant, the optimization toolbox of MATLAB has been used to reach out the better solution with the minimum error. The function used for this purpose is called *fmincon*, which finds the constrained minimum of a function of several variables. In this case, the specification in Eq. (17) is taken as the main function to minimize, and the rest of specifications (18-21) are taken as constrains for the minimization, all of them subjected to the optimization parameters defined within the function *fmincon*.

5.2 Example: pH neutralization process

The pH dynamic model of a real sugar cane raw juice neutralization process can be modelled by the following FOPDT dynamic:

$$G(s) = \frac{0.55e^{-s}}{62s+1} \tag{22}$$

Assume that the design specifications are as follows:

- Gain crossover frequency $\omega_{cg} = 0.08$
- Phase margin $\varphi_m = 0.44\pi$
- Robustness to variations in the gain of the plant must be fulfilled.
- $\left|T(j\omega)\right| \leq -20 \text{ dB}, \forall\ \omega \geq \omega_t = 10 \text{ rad/sec}$
- $\left|S(j\omega)\right| \leq -20\text{dB}, \forall\ \omega \leq \omega_s = 0.01 \text{ rad/sec}$

Using the function *fmincon*, the FOPID controller to control the plant is

$$G_c(s) = 7.9619 + \frac{0.2299}{s^{0.9646}} + 0.1594\ s^{0.0150} \tag{23}$$

Simulation block-diagram of the system is depicted in Fig. 3 and the step response of the closed-loop system is illustrated in Fig. 4.

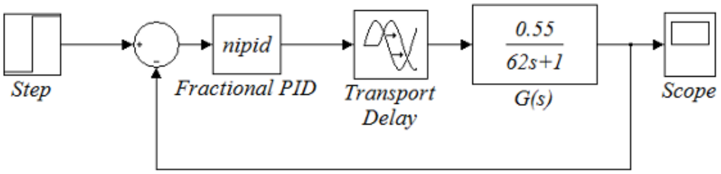

Fig. 3. Simulation block-diagram for control of pH neutralization process

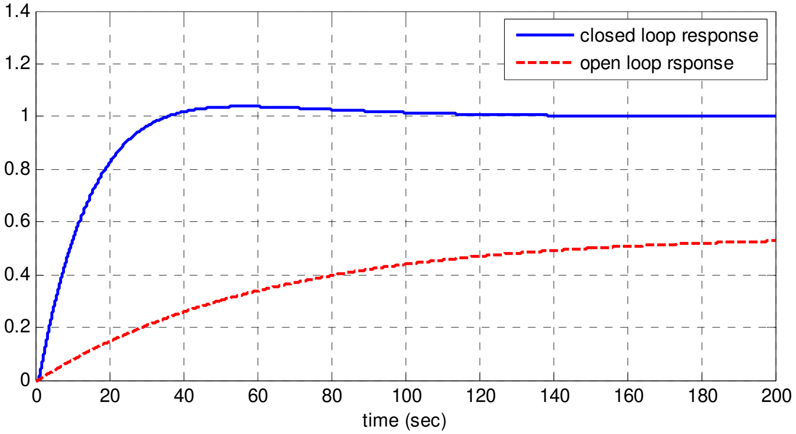

Fig. 4. Step responses of closed loop and open loop pH neutralization process

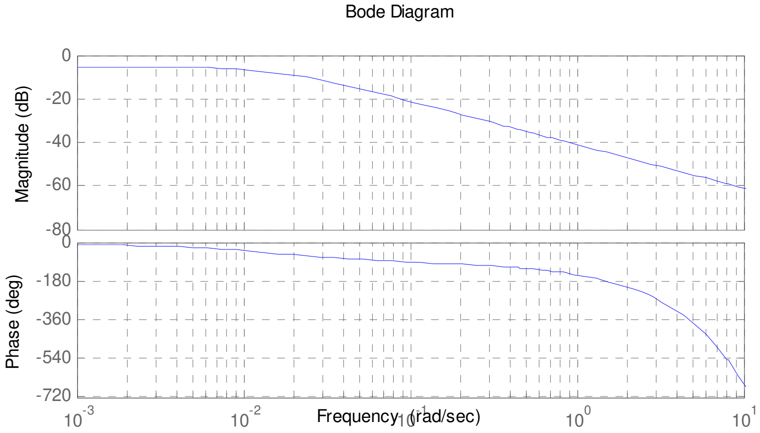

Fig. 5. Bode plot of pH neutralization process

As shown in the Fig.4 the closed loop step response has no steady state error and a fulfilling rise time in the comparison of the open loop response. In order to evaluate the effect of FOPID in frequency response of the process, let us consider Fig.5 as bode plot of the open loop pH neutralization process. The diagram is provided via "Control System Toolbox" of MATLAB. The bode diagram of the FOPID defined in (23) is also depicted in Fig. 6 and finally, the bode plot of $G(s)G_c(s)$ is depicted in Fig. 7.

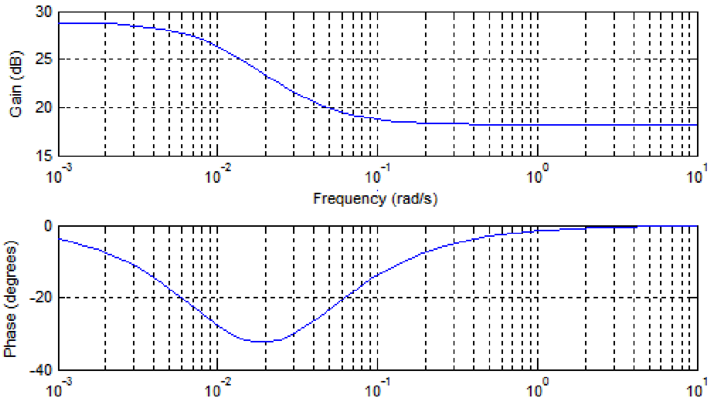

Fig. 6. Bode plot of FOPID controller designed for pH neutralization process

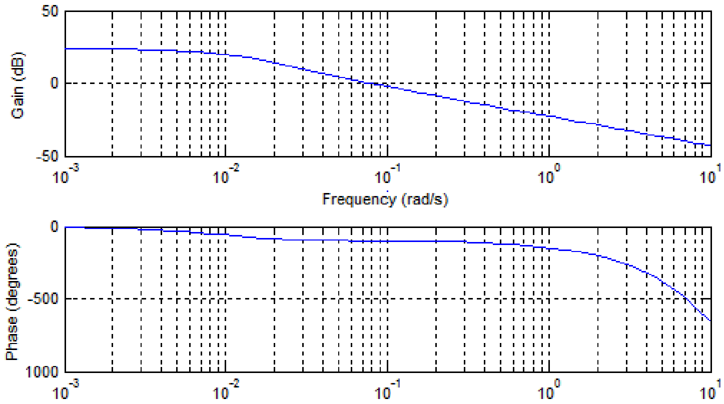

Fig. 7. Bode plot of pH neutralization process when the controller is applied

6. Ziegler-Nichols type tuning rules

In the previous section, a tuning method based on optimization techniques is proposed. The method is effective but allows local minima to be obtained. In practice, most solutions found with this optimization method are good enough, but they strongly depend on initial estimates of the parameters provided. Some may be discarded, because they are unfeasible or lead to unstable loops, but in many cases it is possible to find more than one acceptable FOPID. In others, only well-chosen initial estimates of the parameters allow finding a solution. Motivated from the fact that the optimization techniques depend on initial estimates, Valerio and Costa have introduced some Ziegler-Nichols-type tuning rules for FOPIDs. In this section, we will explain these tuning rules, and two illustrative examples will be presented. These tuning rules are applicable only for systems that have S-shaped step response. The simplest plant to have S-shaped step response can be described by

$$G(s) = \frac{K}{Ts+1}e^{-sL} \tag{24}$$

Valerio and Costa have employed the minimisation tuning method to plants given by (24) for several values of *L* and *T*, with *K* = 1. The parameters of FOPIDs thus obtained vary in a regular manner. Having translated the regularity into formulas, some tuning rules are obtained for particular desired responses.

6.1 First set of tuning rules

A first set of rules is given in Tables 2 and 3. These are to be read as

$$\mathrm{P} = -0.0048 + 0.2664\mathrm{L} + 0.4982\mathrm{T} + 0.0232\mathrm{L}^2 - 0.0720\mathrm{T}^2 - 0.0348\mathrm{TL} \tag{25}$$

and so on. They may be used if $0.1 \le \mathrm{T} \le 50, \mathrm{L} \le 2$ and were designed for the following specifications:

- $\omega_{cg} = 0.5\ \mathrm{rad/sec}$
- $\varphi_m = 2/3\ \mathrm{rad}$
- $\omega_t = 10\ \mathrm{rad/sec}$
- $\omega_s = 0.01\ \mathrm{rad/sec}$
- A = -10 *d*B
- B = -20 *d*B

	k_P	k_I	λ	k_D	μ
1	-0.0048	0.3254	1.5766	0.0662	0.8736
L	0.2664	0.2478	-0.2098	-0.2528	0.2746
T	0.4982	0.1429	-0.1313	0.1081	0.1489
L^2	0.0232	-0.1330	0.0713	0.0702	-0.1557
T^2	-0.0720	0.0258	0.0016	0.0328	-0.0250
LT	-0.0348	-0.0171	0.0114	0.2202	-0.0323

Table 2. Parameters for the first set of tuning rules when $0.1 \le \mathrm{T} \le 5$

	k_P	k_I	λ	k_D	μ
1	2.1187	-0.5201	1.0645	1.1421	1.2902
L	-3.5207	2.6643	-0.3268	-1.3707	-0.5371
T	-0.1563	0.3453	-0.0229	0.0357	-0.0381
L^2	1.5827	-1.0944	0.2018	0.5552	0.2208
T^2	0.0025	0.0002	0.0003	-0.0002	0.0007
LT	0.1824	-0.1054	0.0028	0.2630	-0.0014

Table 3. Parameters for the first set of tuning rules when $5 \le \mathrm{T} \le 50$

6.2 Second set of tuning rules

A second set of rules is given in Table 4. These may be applied for $0.1 \le T \le 50$ and $\mathrm{L} \le 0.5$. Only one set of parameters is needed in this case because the range of values of L these rules cope with is more reduced. They were designed for the following specifications:

- $\omega_{cg} = 0.5\,\text{rad/sec}$
- $\varphi_m = 1\ \text{rad}$
- $\omega_t = 10\,\text{rad/sec}$
- $\omega_s = 0.01\ \text{rad/sec}$
- $A = -20\ d\mathrm{B}$
- $B = -20\ d\mathrm{B}$

	k_P	k_I	λ	k_D	μ
1	-1.0574	0.6014	1.1851	0.8793	0.2778
L	24.5420	0.4025	-0.3464	-15.0846	-2.1522
T	0.3544	0.7921	-0.0492	-0.0771	0.0675
L²	-46.7325	-0.4508	1.7317	28.0388	2.4387
T²	-0.0021	0.0018	0.0006	-0.0000	-0.0013
LT	-0.3106	-1.2050	0.0380	1.6711	0.0021

Table 4. Parameters for the second set of tuning rules

6.3 Example: High-order process control

Consider the following high-order process

$$G(s) = \frac{1}{(s+1)^4} \tag{26}$$

The transfer function of the process is not on the form of FOPDT. In order to control the process via FOPID, let us approximate the process by a FOPDT model. The process can be approximated by the following model (see (Astrom & Hagglund, 1995))

$$G(s) = \frac{1}{2s+1}e^{-2s} \tag{27}$$

where K=1, L=2 and T=2. Fig.8 shows the step response of the process (26) and its approximated model. As we see, the model can approximate the process with satisfying accuracy. The step response of the process is of S-shaped type and we can use the Ziegler-Nichols type tuning rules for our FOPID controller.

Using the first set of tuning rules, one can obtain the following FOPID controller.

$$G_c(s) = 1.1900 + 0.6096\frac{1}{s^{1.2316}} + 1.0696s^{0.8686} \tag{28}$$

The closed step response of the system is depicted in Fig. 9.

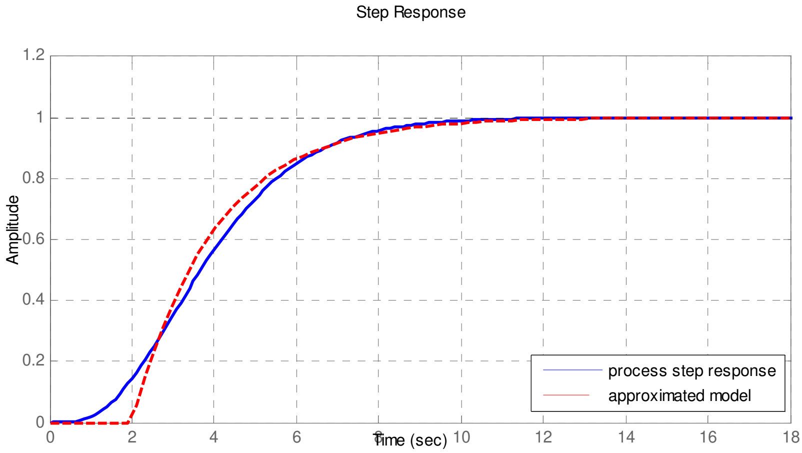

Fig. 8. Step response of the process and its approximated model

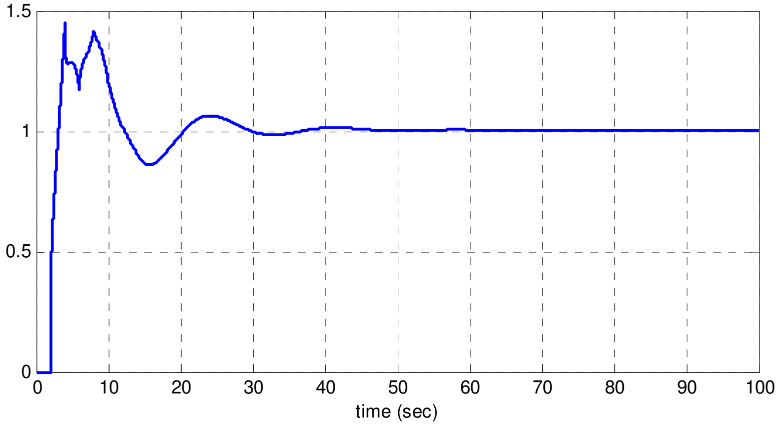

Fig. 9. Step response of high order process controlled by FOPID

6.4 Example: Non-minimum phase process control

When the transfer function of a process is not a FOPDT model, an approximated FOPDT model can be developed; this fact was shown in the previous example. Here, we consider a Non-Minimum phase process. We need to approximate a FOPDT model in order to use Ziegler-Nichols tuning rules. The following non-minimum phase process is considered

$$G(s) = \frac{1-s}{(s+0.5)(s+2)} \tag{29}$$

The process can be approximated by the following model

$$G(s) = \frac{1}{1.8s+1}e^{-1.7s} \tag{30}$$

The step response of the transfer function (30) is compared with the process (29) and depicted in Fig. 10. As we see, the FOPDT model of the process presents a good accuracy.

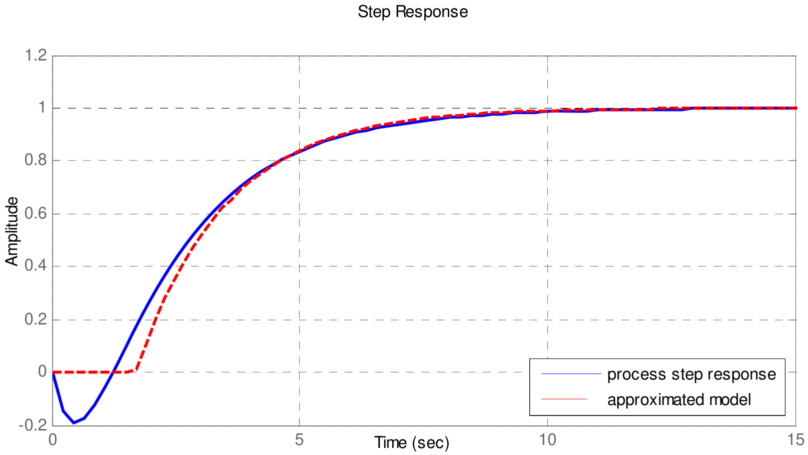

Fig. 10. Step response of the process and its approximated model

After having approximated the process with a FOPDT transfer function, application of the first set of tuning rules gives the following FOPID controller

$$G_c(s) = 1.0721 + 0.6508\frac{1}{s^{1.2297}} + 0.8140s^{0.9786} \tag{31}$$

while the step response of the closed loop control system for set point and is depicted in Fig. 11.

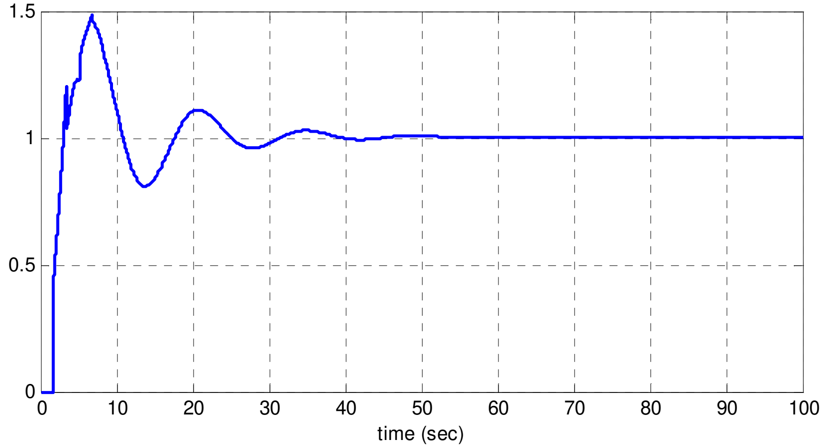

Fig. 11. Step response of non-minimum phase process controlled by FOPID

7. The Padula & Visioli method

In (Padula & Visioli, 2010a), a new set of tuning rules are presented for FOPID controllers. Based on FOPDT models, the tuning rules have been devised in order to minimise the integrated absolute error with a constraint on the maximum sensitivity. In this section, the tuning rules are presented and then the problem of heat exchanger temperature is given.

7.1 Tuning rules

Let us consider a process defined by FOPDT model as one given by Eq. (24). The process dynamics can be conveniently characterised by the normalised dead time and defined as

$$\tau = \frac{L}{L+T} \tag{32}$$

which represents a measure of difficulty in controlling the process. The proposed tuning rules are devised for values of the normalised dead time in the range $0.05 \le \tau \le 0.8$. In fact, for values of $\tau < 0.05$ the dead time can be virtually neglected and the design of a controller is rather trivial, while for values of $\tau > 0.8$ the process is significantly dominated by the dead time and therefore a dead time compensator should be employed. By the methodology developed in (Padula & Visioli, 2010a), the FOPID controller is modeled by the following transfer function

$$G_c(s) = K_p \frac{K_i s^{\lambda} + 1}{K_i s^{\lambda}} \frac{K_d s^{\mu} + 1}{\frac{K_d}{N} s^{\mu} + 1} \tag{33}$$

The major difference of FOPID defined by (33) with the standard form of FOPID defined by (15) is that an additional first-order filter has been employed in (33) in order to make the controller proper. The parameter N is chosen as $N = 10T^{(\mu-1)}$. The performance index is integrated absolute error which is defined as follows

$$\mathrm{IAE} = \int_0^{\infty} \left| e(t) \right| dt \tag{34}$$

Using Eq.(34) as performance index yields a low overshoot and a low settling time at the same time (Shinskey, 1994). The maximum sensitivity (Astrom and Hagglund, 1995) is defined as

$$M_s = \max\left\{ \frac{1}{1 + G_c(s)G(s)} \right\} \tag{35}$$

which represents the inverse of the maximum distance of the Nyquist plot from the critical point (-1,0). Obviously, the higher value of M_s yields the less robustness against uncertainties. Tuning rules are devised such that the typical values of M_s = 1.4 and M_s = 2.0 are achieved. If only the load disturbance rejection task is addressed, we have

$$K_p = \frac{1}{K}\left(a\tau^{b} + c \right) \tag{36}$$

$$K_i = T\left(a\left(\frac{L}{T}\right)^b + c\right) \tag{37}$$

$$K_d = T\left(a\left(\frac{L}{T}\right)^b + c\right) \tag{38}$$

where the values of the parameters are shown in Tables 5-8.

	a	b	c
k_P	0.2776	-1.097	-0.1426
k_D	0.6241	0.5573	0.0442
k_I	0.4793	0.7469	-0.0239

Table 5. Tuning rules for k_P, k_D and k_I when $M_s = 1.4$

λ	μ
1	1.0 if $\tau < 0.1$ 1.1 if $0.1 \le \tau < 0.4$ 1.2 if $0.4 \le \tau$

Table 6. Tuning rules for λ and μ when $M_s = 1.4$

	a	b	c
k_P	0.164	-1.449	-0.2108
k_D	0.6426	0.8069	0.0563
k_I	0.5970	0.5568	-0.0954

Table 7. Tuning rules for k_P, k_D and k_I when $M_s = 2.0$

λ	μ
1	1.0 if $\tau < 0.2$ 1.1 if $0.2 \le \tau < 0.6$ 1.2 if $0.6 \le \tau$

Table 8. Tuning rules for λ and μ when $M_s = 2.0$

7.2 Example: Heat exchanger temperature control

A chemical reactor called "stirring tank" is depicted in Fig. 12. The top inlet delivers liquid to be mixed in the tank. The tank liquid must be maintained at a constant temperature by varying the amount of steam supplied to the heat exchanger (bottom pipe) via its control valve. Variations in the temperature of the inlet flow are the main source of disturbances in this process.

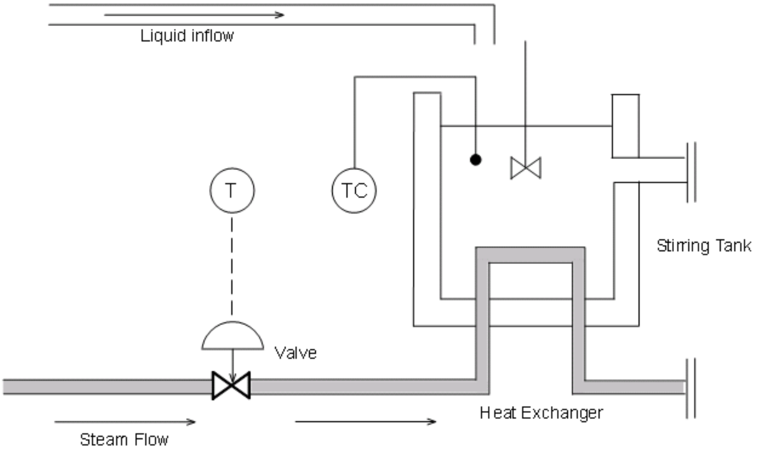

Fig. 12. Stirring Reactor with Heat Exchanger

The process can be modelled adequately by FOPDT models as shown in the Fig. 13.

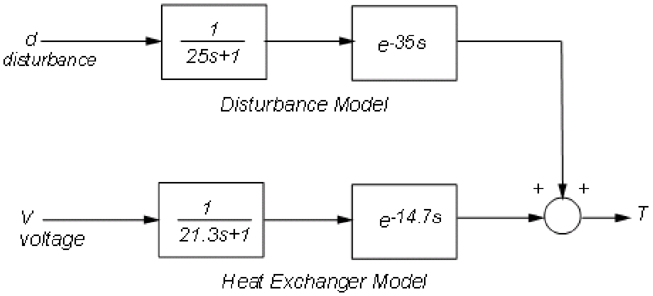

Fig. 13. Open loop process model

The transfer function

$$G(s) = \frac{e^{-14.7s}}{21.3s+1} \tag{38}$$

models how a change in the voltage *V* driving the steam valve opening effects the tank temperature *T*, while the transfer function

$$G_d(s) = \frac{e^{-35s}}{25s+1} \tag{39}$$

models how a change *d* in inflow temperature affects *T*.
The control problem is to regulate tank temperature *T* around a given setpoint. From Eq. (32), the normalized dead-time of the process (38) is obtained as 0.4083 which implies

that we can utilize the proposed tuning rules. From tuning table 5 and 6, the following FPOID can be obtained for the case of M_s = 1.4

$$G_{c1}(s) = 0.3511\frac{11.7527s+1}{11.7527s}\frac{7.2300s^{1.2}+1}{0.3923s^{1.2}+1} \tag{40}$$

And for the case of M_s = 2, from tables 7 and 8 we have

$$G_{c2}(s) = 0.1400\frac{11.3467s+1}{11.3467s}\frac{8.3116s^{1.1}+1}{0.4509s^{1.1}+1} \tag{41}$$

Simulation results are presented in Fig. 14. It is assumed that a load disturbance is applied at t=500 seconds, and the disturbance rejection of both controllers are verified. Simulations also show that the transient states of both controllers are approached.

9. Conclusion

In this chapter, some of the well-known tuning methods of FOPID controllers are presented and several illustrative examples, verifying the effectiveness of the methods are given.

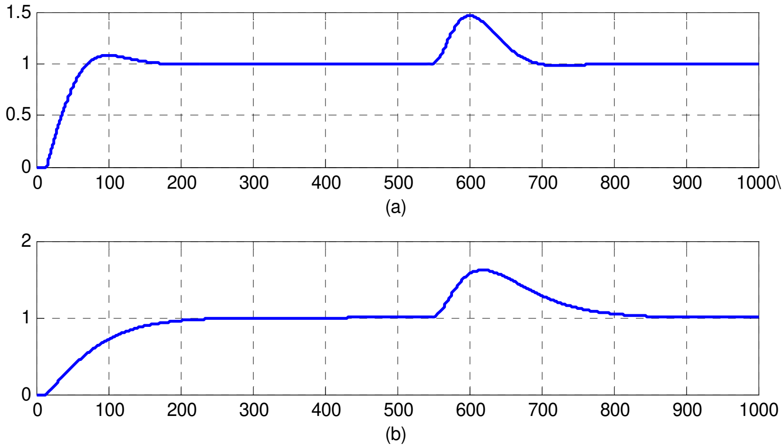

Fig. 13 Closed response of heat exchanger system and disturbance rejection of controllers (a) $G_{c1}(s)$ (b) $G_{c2}(s)$

Simulations have been carried out using *MATLAB/SIMULINK* software via Ninteger toolbox. After discussion on fractional calculus and its approximation methods, the Ninteger toolbox is introduced briefly. Then optimization methods, Ziegler-Nichols tuning rules and a new tuning method were introduced. We have considered control of pH neutralization process, high-order process, Non-Minimum phase process and temperature control of heat exchanger as case studies. In spite of extensive research, tuning the parameters of a FOPID controller remains an open problem. Other analytical methods and new tuning rules may be further studied.

10. References

Ahmad, W.M., Sprott, J.C., (2003). Chaos in fractional-order autonomous nonlinear systems, *Chaos, Solitons & Fractals* Vol. 16, 2003, pp.339–351.

Åström, K.J., Hägglund, T., (1995) PID Controllers: Theory, Design and Tuning, *ISA Press*, Research Triangle Park, 1995

Charef, A., Fergani, N. (2010). $PI^{\lambda}D^{\mu}$ Controller Tuning For Desired Closed-Loop Response Using Impulse Response, *Proceedings of Fractional Differentiation and its Applications*, Badajoz, Spain, October 2010

Delavari, H., Ghaderi, R., Ranjbar, A., Hosseinnia, S.H., Momani S., (2010). Adaptive Fractional PID Controller for Robot Manipulator, *Proceedings of Fractional Differentiation and its Applications*, Badajoz, Spain, October 2010

Domingues, J., Vale´rio, D., Costa, J.S. (2010). Rule-based fractional control of an irrigation canal, *ASME Journal of Computational and Nonlinear Dynamics*, 2010. Accepted.

Jesus, I.S., Machado, J.A.T. (2008). Fractional control of heat diffusion systems, *Nonlinear Dynamics,* Vol. 54, pp. 263-282

Karimi, M., Zamani, M., Sadati, N., Parniani, M. (2009). An Optimal Fractional Order Controller for an AVR System Using Particle Swarm Optimization Algorithm, *Control Engineering Practice*, Vol. 17, pp. 1380–1387

Maiti, D., Biswas, S., Konar, K. (2007). Design of a Fractional Order PID Controller Using Particle Swarm Optimization Technique, *Proceedings of 2nd National Conference on Recent Trends in Information Systems,* 2008

Monje, C.A., Vinagre, B.M. , Chen, Y.Q. , Feliu, V., Lanusse, P. , Sabatier, J. (2004). Proposals for fractional $PI^{\lambda}D^{\mu}$ tuning, *Proceedings of Fractional Differentiation and its Applications*, Bordeaux, 2004

Oustaloup, A. (1991). La commande CRONE: commande robuste d'ordre non entier. Hermès, Paris, 1991

Oustaloup, A., Levron, F. , Mathieu, B. , Nanot, F.M. ,(2000). Frequency-band complex noninteger differentiator: characterization and synthesis, *IEEE Transactions on Circuits and Systems I*, Vol. 47, 2000, pp. 25-39.

Padula, F., Visioli, A., (2010a). Tuning rules for optimal PID and fractional-order PID controllers, *Journal of Process Control*, doi:10.1016/j.jprocont.2010.10.006

Padula, F., Visioli, A., (2010b). Set-point Weighting for Fractional PID Controllers, *Proceedings of Fractional Differentiation and its Applications,* Badajoz, Spain, October 2010

Padula, F., Visioli, A., (2010c). Tuning of Fractional PID Controllers for Integral Processes, *Proceedings of Fractional Differentiation and its Applications,* Badajoz, Spain, October 2010

Podlubny, I. (1999a). Fractional-Order Systems and $PI^{\lambda}D^{\mu}$ Controllers, *IEEE Transactions on Automatic Control*, Vol. 44, No. 1, January 1999, pp. 208-214

Podlubny, I. (1999b). Fractional Differential Equations, *Academic Press*, USA

Podlubny, I., Dorcak, L., Kostial, I. (1997). On Fractional Derivatives, Fractional-Order Dynamic Systems and $PI^{\lambda}D^{\mu}$ controllers, *Proceedings of the 36th Conference on Decision & Control,* San Diego, California, USA, December 1997

Shinskey, F.G. (1994). Feedback Controllers for the Process Industries, *McGraw-Hill*, New York, USA

Tabatabaei, M., Haeri, M., (2010). Loop Shaping Design of Fractional PD and PID Controllers, *Proceedings of Fractional Differentiation and its Applications*, Badajoz, Spain, October 2010

Vale´rio, D., Costa, J.S. (2004). NINTEGER: A Non-Integer Control Toolbox for MATLAB, *Proceedings of Fractional Differentiation and its Applications*, Bordeaux, 2004

Vale´rio, D., Costa, J.S. (2006). Tuning of fractional PID controllers with Ziegler–Nichols-type rules, *Signal Processing*, Vol. 86, pp. 2771–2784

Vinagre, B. M., Podlubny, I., Dorcak, L., Feliu, V. (2000). On Fractional PID Controllers: A Frequency Domain Approach, *Proceedings of IFAC Workshop on Digital Control: Past, Present and Future of PID Control*. Terrasa, Spain, pp. 53—58.

Xue, D., Zhao, C., Chen, Y.Q., (2006). Fractional Order PID Control of A DC-Motor with Elastic Shaft: A Case Study, *Proceedings of the 2006 American Control Conference,* Minneapolis, Minnesota, USA, June, 2006

Yeroglu, C., Özyetkin, M. M., Tan, N. (2010). Design of Robust $PI^{\lambda}D^{\mu}$ Controller for FOPDT Systems, *Proceedings of Fractional Differentiation and its Applications,* Badajoz, Spain, October 2010

APPLICATIONS OF MATLAB IN SCIENCE AND ENGINEERING
Edited by Tadeusz Michałowski
INTECH

Applications of MATLAB in Science and Engineering
Edited by Prof. Tadeusz Michalowski

ISBN 978-953-307-708-6
Hard cover, 510 pages
Publisher InTech
Published online 09, September, 2011
Published in print edition September, 2011

The book consists of 24 chapters illustrating a wide range of areas where MATLAB tools are applied. These areas include mathematics, physics, chemistry and chemical engineering, mechanical engineering, biological (molecular biology) and medical sciences, communication and control systems, digital signal, image and video processing, system modeling and simulation. Many interesting problems have been included throughout the book, and its contents will be beneficial for students and professionals in wide areas of interest.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821